Data Mining

Q1. What is data mining? Write down some of the applications of data mining in various fields. ***q 1st

Answer:

It is the extraction of interesting (non-trivial, implicit, previously unknown, and potentially useful) knowledge (rules, regularities, constraints), information, or patterns from data in large databases.

Alternative names and their "inside stories": Data mining: a misnomer?

Knowledge discovery (mining) in databases (KDD), knowledge extraction, data/pattern analysis, data archeology, data dredging, information harvesting, business intelligence, etc.

Data Mining Applications:

- Database analysis and decision support
  - Market analysis and management
    - Target marketing, customer relation management, market basket analysis, cross selling, market segmentation.
  - Risk analysis and management
    - Forecasting, customer retention, quality control etc.
  - Fraud detection and management

- Other Applications
  - Text mining (news group, email, documents) and Web analysis.
  - Intelligent query answering
  - Sports
  - Astronomy
  - Internet Web Surf-Aid

Q2. Describe how data mining can be applied in market analysis and management. ***q 1st

Answer:

Market Analysis and Management:

- Credit card transactions, loyalty cards, discount coupons, customer complaint calls, plus (public) lifestyle studies are data originators.
- Target marketing
  - Find clusters of "model" customers who share the same characteristics: interest, income level, spending habits, etc.
- Determine customer purchasing patterns over time
  - Conversion of single to a joint bank account: marriage, etc.
- Cross-market analysis
  - Associations/co-relations between product sales
  - Prediction based on the association information

- Customer profiling
  - Data mining can tell you what types of customers buy what products (clustering or classification)

- Identifying customer requirements
  - Identifying the best products for different customers
  - use prediction to find what factors will attract new customers

Q3. Describe the steps of a KDD (Knowledge Discovery in Databases) process with diagram. ***q 1st 2nd

Answer:

Data Mining: A KDD Process.

- KDD refers knowledge discovery in databases.
- Data mining is the core of knowledge discovery process.

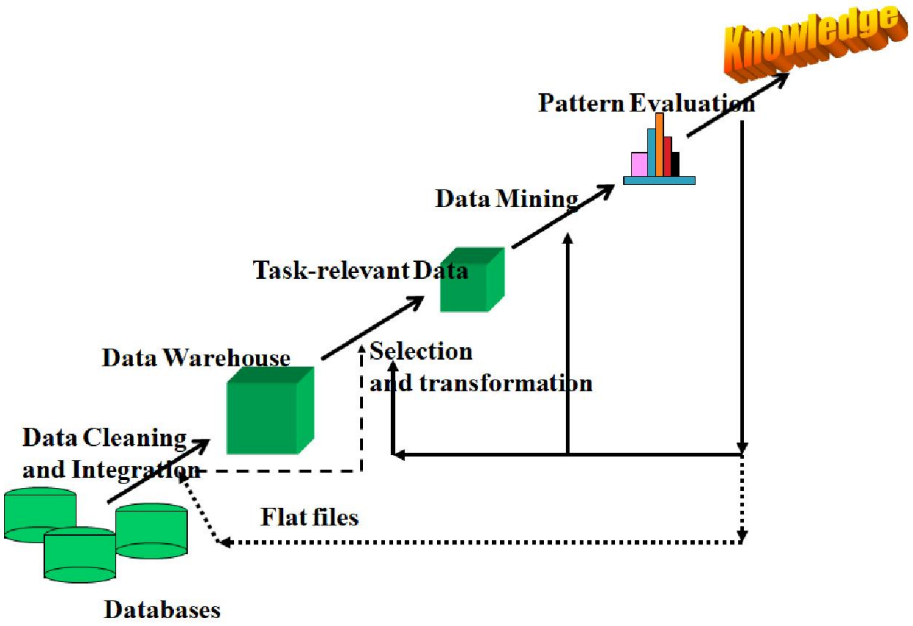


Fig: 1.1 Diagram of a KDD process.

Steps of a KDD process:

- **Learning the application domain**: Relevant prior knowledge and goals of application are learned
- **Data cleaning**: Removes noise and inconsistent data
- **Data integration**: Multiple data sources are combined
- **Data selection**: Data relevant to analysis task are retrieved
- **Data transformation**: Data are transformed or consolidated into forms appropriate for mining
- **Data mining**: An essential process where intelligent methods are applied in order to extract data patterns of interest
- **Pattern evaluation**: Identify the truly interesting patterns representing knowledge
- **Knowledge presentation**: Visualization and knowledge representation techniques are used to present the mined knowledge to the user.

Q4. What is the solution to the data explosion problem? Explain in brief. ***q 1st

Answer:

- **Data explosion problem**
  - Automated data collection tools and mature database technology lead to tremendous amounts of data stored in databases, data warehouses and other information repositories
- We are drowning in data, but starving for knowledge!
- Solution: Data warehousing and data mining
  - Data warehousing and on-line analytical processing
  - Extraction of interesting knowledge (rules, regularities, patterns, constraints) from data in large databases.

Q5. Differentiate between data mining and query processing.
Or What is data mining? How does it differ from query processing. ***q 1st

Answer:

- **Data mining** is a process of discovering patterns, correlations, relationships, and trends within large datasets using computational and statistical methods. The goal of data mining is to extract useful insights and knowledge from data that may not be readily apparent through simple analysis.

- **Data mining**
  - It is the extraction of interesting (non-trivial, implicit, previously unknown and potentially useful) knowledge (rules, regularities, constraints), information or patterns from data in large databases
  - Knowledge discovery (mining) in databases (KDD), knowledge extraction, data/pattern analysis, data archeology, data dredging, information harvesting, business intelligence, etc.
- **What is not data mining?**
  - (Deductive) query processing in not data mining.
  - Expert systems or small ML/statistical programs.
  - Data querying is the process of asking specific, structured questions of data in search of a specific answer, while data mining is the process of sifting through data to identify patterns and relationships using statistical algorithms.

Q6. What is outlier mining? ***q 1st

Answer:

Outlier analysis

- Outlier: a data object that does not comply with the general behavior or model of the data
- It can be considered as noise or exception but is quite useful in fraud detection
- The rare events can be more interesting than the more regularly occurring ones. The analysis of outlier data is referred to as outlier mining.

Q7. Describe how data mining can be applied in fraud detection and management. ***q 1st

Answer:

- Applications
  - widely used in health care, retail, credit card services, telecommunications (phone card fraud), etc.
- Approach
  - use historical data to build models of fraudulent behavior and use data mining to help identify similar instances
- Examples
  - auto insurance: detect a group of people who stage accidents to collect on insurance
  - money laundering: detect suspicious money transactions
  - medical insurance: detect professional patients and ring of doctors and ring of references
- Detecting inappropriate medical treatment
- Detecting telephone fraud

Q8. Describe the architecture of a typical data mining system with diagram. What is the function of data mining engine component. ***q 2nd

Answer:

Architecture of a Typical Data Mining System

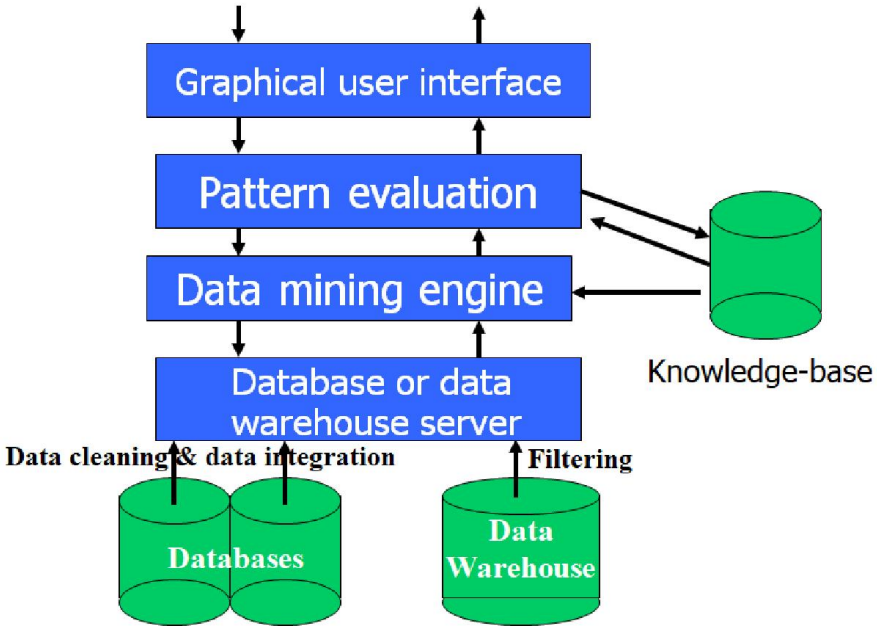


Fig: 7.1 Architecture of a Typical Data Mining System.

- **Database, data warehouse or other information repository**: These are set of databases, data warehouses, spreadsheets etc. Data cleaning and data integration techniques are applied to this data.
- **Database or data warehouse server:** Responsible for fetching the relevant data based on the user's data mining request.

- **Knowledge base:** This is the domain knowledge that is used to guide the search, or evaluate the interestingness of resulting patterns.
- **Data mining engine:** Consists of a set of functional modules for tasks such as characterization, association, classification, cluster analysis, and evolution and deviation analysis.
- **Pattern evaluation module:** This component typically employs interestingness measures and interact with the data mining modules so as to focus the search towards interesting patterns.
- **Graphical user interface:** This module communicates between users and the data mining system. It allows to specify a data mining query or task. Allows the user to browse database and data warehouse schemas or data structures, evaluate mined patterns and visualize the patterns in different forms.

Q9. What is a pattern? Are all of the discovered pattern interesting? Explain in brief. ***q 2nd

Answer:

Pattern mining concentrates on identifying rules that describe specific patterns within the data.

Are All the "Discovered" Patterns Interesting?

- A data mining system/query may generate thousands of patterns, not all of them are interesting. A pattern is interesting if
  - i) it is easily understood by humans
  - ii) valid on new or test data with some degree of certainty
  - iii) potentially useful
  - iv) novel, or validates some hypothesis that a user seeks to confirm.

Interestingness measures

- Objective interestingness measures: based on the structure of discovered patterns and the statistics underlying them. E.g., measure for association rules: support, confidence.

Q10. Can we find all and only interesting patterns in data mining? Explain the solutions in brief. Or Can we extract only the interesting patterns using data mining technique? Explain in brief. ***q 2nd

Can We Find All and Only Interesting Patterns?

- **Find all of the interesting patterns**
  - Can a data mining system generate all of the interesting patterns? It refers to the completeness of a data mining algorithm.
  - User provided constraints and interestingness measures should be used to focus the search, which is often sufficient to ensure completeness.

- **Search for only interesting patterns: Optimization**
  - Can a data mining system generate only interesting patterns? To generate only interesting patterns would be much more efficient for users and data mining systems- a challenging issue.
  - Approaches
    - First generate all of the patterns and then filter out the uninteresting ones.
    - Generate only the interesting patterns—mining query optimization

Q11. Define the terms: i) Classification, ii) Neural Network. iii) Prediction, iv) Cluster analysis. & v) Outlier Analysis.

Answer:

Classification: It is the process of finding a set of models (functions) that describe and distinguish data classes or concepts to predict the class of objects whose class label is unknown. The derived model is based on the analysis of a set of training data (i.e., data objects whose class label is known) E.g., classify countries based on climate, or classify cars based on gas mileage.

Neural Network: A neural network when used for classification, is typically a collection of neuron-like processing units with weighted connections between the units.

Prediction: Classification is the process of finding a set of models (functions) that describe and distinguish data classes or concepts to predict the class of objects whose class label is unknown. Prediction is the process to predict some unknown or missing numerical values.

Cluster Analysis: Clustering analyzes data objects without consulting a known class label. It groups data to form new classes, e.g., cluster houses to find distribution patterns.

The objects are clustered or grouped based on the principle of maximizing the intra-class similarity and minimizing the interclass similarity. Each cluster can be viewed as a class of objects from which new rules can be derived.

Outlier analysis: Outlier- a data object that does not comply with the general behavior or model of the data. It can be considered as noise or exception but is quite useful in fraud detection. The rare events can be more interesting than the more regularly occurring ones. The analysis of outlier data is referred to as outlier mining.

Q12. Define descriptive and predictive data mining.

Answer:

- Data mining tasks can be classified into two categories: descriptive and predictive.
  - **Descriptive**: characterize the general properties of the data in the database
  - **Predictive**: perform inference on the current data in order to make predictions

Q13. What are the different data/database source which can be use in data mining? / Q12. Write down the name of data mining data/database sources.

***q 2nd

Answer:

- Relational databases
- Data warehouses
- Transactional databases
- Advanced DB and information repositories
  - Object-oriented and object-relational databases
  - Spatial databases
  - Time-series data and temporal data
  - Text databases and multimedia databases
  - Heterogeneous and legacy databases
  - WWW

Q14. Define Classification. Describe decision tree induction.

Classification:

- It is the task of assigning objects to one of several predefined categories,

e.g., Detecting spam email messages based upon the message header and content

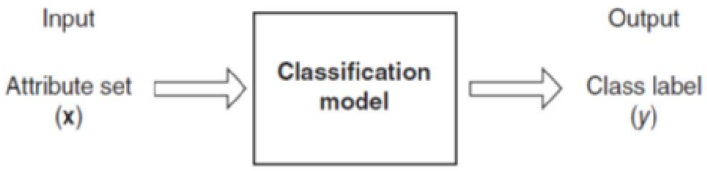


Figure 3.2. A schematic illustration of a classification task.

- It is the task of learning a target function *f* that maps each attribute set **x** to one of the predefined class labels, y.

Types

1. Descriptive modeling
2. Predictive modeling

Descriptive modeling

- A classification model can serve as a tool to distinguish between objects of different classes.

Predictive modeling

- A classification model can also be used to predict the class label of unknown records.

Example:

- Classify a person as a high, medium or low-income group.

Decision tree Induction:

- A decision tree is a hierarchical structure consisting of nodes and directed edges.

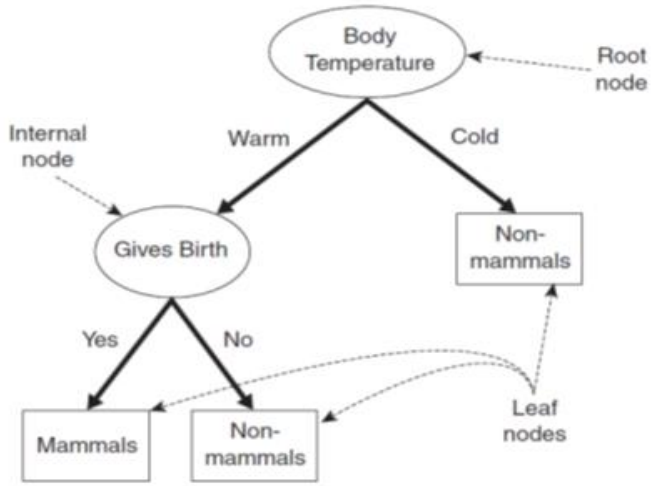


Figure 3.4. A decision tree for the mammal classification problem.

E.g., Fig. 4.4/3.4

- The tree has 3 types of nodes:
    1. Root
    2. Internal nodes
    3. Leaf or terminal nodes
- Each leaf node is assigned a class label. The non-terminal nodes (root, internal nodes) contain attribute test conditions to separate records that have different characteristics
- Classifying a test record is easy once a decision tree has been constructed
- Starting at the root (based on information gain), we apply the test condition to the record and follow the appropriate branch based on the outcome of the test.
- This will result another internal node (based on information gain), for which a new test condition is applied, or to a leaf node.

Q15. Define Classification. Draw a decision tree to represent mammal classification. ***q 2nd

- ***Classification:***
    - It is the task of assigning objects to one of several predefined categories,

e.g., Detecting spam email messages based upon the message header and content

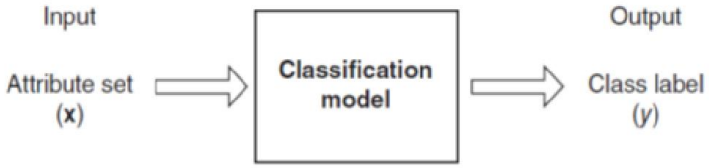


Figure 3.2. A schematic illustration of a classification task.

- It is the task of learning a target function *f* that maps each attribute set **x** to one of the predefined class labels, y.

Types

1. Descriptive modeling
2. Predictive modeling

Descriptive modeling

- A classification model can serve as a tool to distinguish between objects of different classes.

Predictive modeling

- A classification model can also be used to predict the class label of unknown records.

Example:

- Classify a person as a high, medium or low-income group.

Decision tree Induction:

- A decision tree is a hierarchical structure consisting of nodes and directed edges.

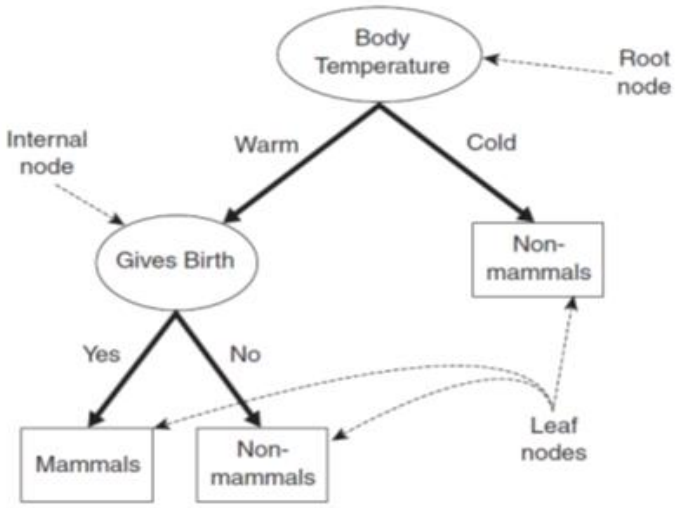


Figure 3.4. A decision tree for the mammal classification problem.

E.g., Fig. 4.4/3.4

- The tree has 3 types of nodes:
  1. Root
  2. Internal nodes
  3. Leaf or terminal nodes
- Each leaf node is assigned a class label. The non-terminal nodes (root, internal nodes) contain attribute test conditions to separate records that have different characteristics
- Classifying a test record is easy once a decision tree has been constructed
- Starting at the root (based on information gain), we apply the test condition to the record and follow the appropriate branch based on the outcome of the test.
- This will result another internal node (based on information gain), for which a new test condition is applied, or to a leaf node.

Q16. Explain decision tree induction. Describe the general approach for building a classification model. ***q 4th

- ***Decision tree Induction:***

- A decision tree is a hierarchical structure consisting of nodes and directed edges.

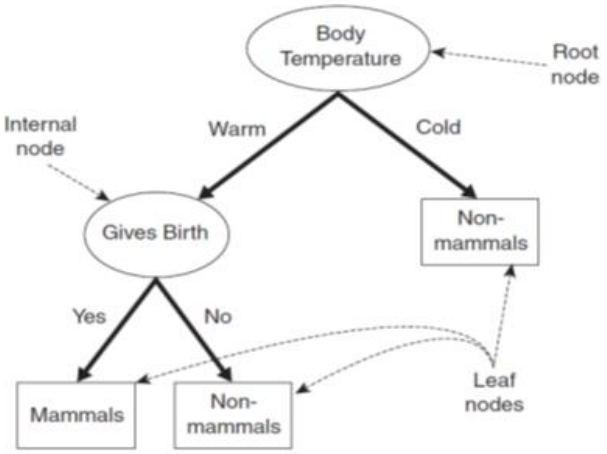


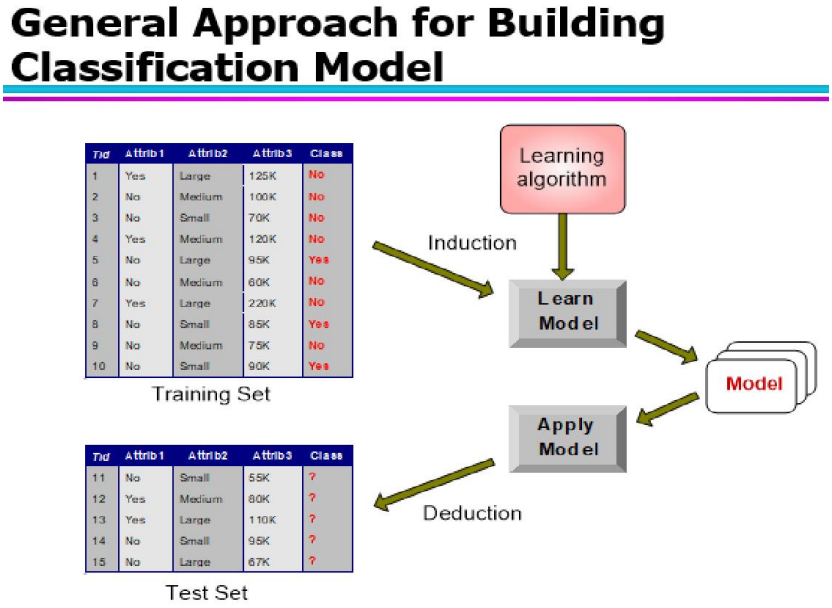
Figure 3.4. A decision tree for the mammal classification problem.

E.g., Fig. 4.4/3.4

- The tree has 3 types of nodes:
    1. Root
    2. Internal nodes
    3. Leaf or terminal nodes
- Each leaf node is assigned a class label. The non-terminal nodes (root, internal nodes) contain attribute test conditions to separate records that have different characteristics
- Classifying a test record is easy once a decision tree has been constructed
- Starting at the root (based on information gain), we apply the test condition to the record and follow the appropriate branch based on the outcome of the test.
- This will result another internal node (based on information gain), for which a new test condition is applied, or to a leaf node.

The General approach for building a classification model

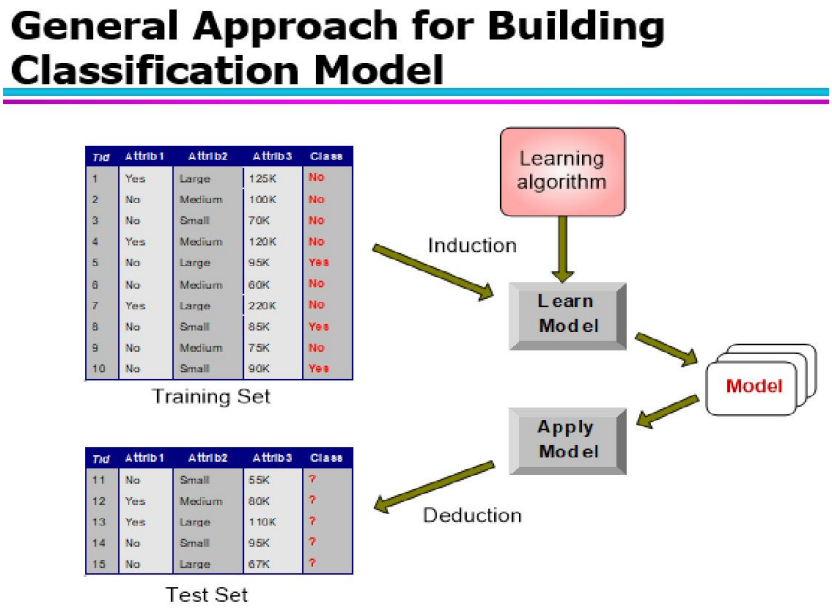
- A classification technique or a classifier is a systematic approach to building classification models from an input data set.
- E.g. decision tree classifiers, rule-based classifiers, neural networks etc.
- Each technique employs a learning algorithm to identify a model.



Q17. Describe the general approach for building a classification model. ***q 4th

The General approach for building a classification model

- A classification technique or a classifier is a systematic approach to building classification models from an input data set.
- E.g. decision tree classifiers, rule-based classifiers, neural networks etc.
- Each technique employs a learning algorithm to identify a model.



Q18. What is classification? What are the different techniques which are used to represent the data mining output of classification? ***q 1st

Classification is a type of supervised learning in data mining, where the goal is to build a model that can predict the class of a new data point based on a set of training data points that have already been classified.

There are many different techniques that can be used for classification, including:

- Decision trees
- Support vector machines
- Naïve Bayes classifiers
- K-nearest neighbors
- Random forests

The best technique to use will depend on the specific data set and the desired outcome of the classification task.

The different techniques which are used to represent the data mining output of classification are:

- **Decision trees:** Decision trees are a simple and intuitive way to represent a classification model. They are easy to understand and interpret, and they can be used to explain how the model makes its predictions.

- **Support vector machines:** Support vector machines are a more powerful technique than decision trees. They can be used to classify data that is not linearly separable, and they can achieve better accuracy than decision trees.
- **Naïve Bayes classifiers:** Naïve Bayes classifiers are a simple and fast technique for classification. They are based on the assumption that the probability of a data point belonging to a class is independent of the values of the other attributes.
- **K-nearest neighbors:** K-nearest neighbors is a simple and non-parametric technique for classification. It works by finding the K most similar data points to a new data point, and then predicting the class of the new data point based on the classes of the K nearest neighbors.
- **Random forests:** Random forests are a powerful technique for classification. They are an ensemble of decision trees, and they can achieve better accuracy than decision trees or support vector machines.

The choice of which technique to use will depend on the specific data set and the desired outcome of the classification task.

Here are some examples of how classification can be used in the real world:

- Banks use classification to predict which customers are likely to default on their loans.
- E-commerce websites use classification to recommend products to customers.
- Social media platforms use classification to identify and remove harmful content.
- Healthcare organizations use classification to identify patients who are at risk for certain diseases.
- Insurance companies use classification to determine the risk of insuring a particular customer.

Classification is a powerful tool that can be used to make predictions about a wide variety of things. By understanding the different techniques that can be used for classification, you can choose the best technique for your specific needs.

Q19. What is classification? In each semester, if you have to decide to register some prerequisite and require courses, which type of task you need to consider it is classification, or regression? ***q 1st

- Classification is a data mining task that involves predicting categorical or discrete class labels for a given set of input instances. It aims to assign instances to predefined classes based on their features or attributes. The goal is to learn a classification model from a labeled training dataset and use this model to classify new, unseen instances into the appropriate classes.

In the scenario you described, deciding which prerequisite and required courses to register for in each semester, the task can be framed as a classification problem. Here's how:

1. Input Instances: Each input instance represents a combination of courses or prerequisites that a student has already taken or plans to take.

2. Features or Attributes: The features or attributes of each instance can include the courses taken in previous semesters, the current semester, or any other relevant information about the student's academic progress.

3. Class Labels: The class labels represent the possible decisions or choices for the student, such as "Register" or "Do Not Register" for specific prerequisite or required courses.

4. Training Dataset: The training dataset consists of labeled instances where the class labels are known based on past student records or domain knowledge.

5. Classification Model: A classification model is trained using the labeled training dataset to learn the patterns and relationships between the input features and the corresponding class labels.

6. Classification Task: Once the model is trained, it can be used to classify new instances (combination of courses) into the appropriate class labels, i.e., determining whether the student should register for a particular prerequisite or required course.

Regression, on the other hand, is a data mining task used to predict continuous numerical values or quantities rather than discrete class labels. In the context of deciding which courses to register for, if the task involved predicting a numerical value, such as the expected GPA or credit hours for a particular semester, it would be a regression problem.

Therefore, in the scenario you described, the task of deciding prerequisite and required courses in each semester is a classification problem, as it involves predicting discrete class labels (e.g., "Register" or "Do Not Register") based on the student's past or current academic information.

Q20. For Table 1. Determine the attribute values in the blank space for the 3 classes Mammal, Reptile and Fish. Draw the Decision tree classifier for Table 1 using the attribute values you defined. .... ***q 4th

Name	Body temp	Skin cover	Gives birth	Aquatic creature	Aerial creature	Has legs	Hibernates	Class label
Human	Warm-blooded	---	---	---	---	---	---	Mammal
Python	Cold-blooded	---	---	---	---	---	---	Reptile
Salmon	Cold-blooded	---	---	---	---	---	---	Fish

- Answer:

Name	Body temp	Skin cover	Gives birth	Aquatic creature	Aerial creature	Has legs	Hibernates	Class label
Human	Warm-blooded	Hair	Yes	No	No	Yes	No	Mammal
Python	Cold-blooded	Scales	No	No	No	No	Yes	Reptile
Salmon	Cold-blooded	Scales	No	yes	No	No	No	Fish

Q21. What is a data warehouse? Write down the objective of a data warehouse. ***q 3rd

- A data warehouse is a large, centralized repository of integrated data from multiple sources that have been designed to support business intelligence and analytics applications. It is a separate database that is specifically optimized for querying and reporting, rather than for transaction processing like operational databases.

The data warehouse has become an increasingly important platform for data analysis, and online analytical processing and will provide an effective platform for data mining.

"A data warehouse is a subject-oriented, integrated, time-variant, and nonvolatile collection of data in support of management's decision-making process." —W. H. Inmon

It is a semantically consistent data store that serves as a physical implementation of a decision-support data model and stores the information which an enterprise needs to make strategic decisions.

Data warehousing is the process of constructing and using data warehouses.

Some of the key objectives of a data warehouse include:

- It provides architectures and tools for business executives to systematically organize, understand, and use their data to make strategic decisions.
- A data warehouse provides the decision-making body of the company with a platform that provides historical data for analysis.
- Supports structured queries, analytical reporting, and decision-making.
- It helps the decision-making body of the company to function efficiently.

Overall, the objective of a data warehouse is to provide a reliable, consistent, and comprehensive view of an organization's data, making it easier for decision-makers to access and analyze information for better decision-making.

Q22. Write down the usages of a data warehouse. ***q 3rd

- Three kinds of data warehouse applications
    - Information processing: supports querying, basic statistical analysis, and reporting, tables, charts and graphs
    - Analytical processing:
        - multidimensional analysis of data warehouse data
        - supports basic OLAP operations, slice-dice, drilling, pivoting
    - Data mining:
        - knowledge discovery from hidden patterns
        - supports associations, constructing analytical models, performing classification and prediction, and presenting the mining results using visualization tools.

Q23. Explain the features of data warehouse. ***q 3rd

- **Subject-Oriented**

- It is organized around major subjects, such as customer, product, sales.
- Focusing on the modeling and analysis of data for decision makers, not on daily operations or transaction processing.

Integrated

- It is constructed by integrating multiple, heterogeneous data sources such as relational databases, flat files, on-line transaction records

Time Variant

The time horizon for the data warehouse is significantly longer than that of operational systems. The operational database maintains current value data, but data warehouse data provide information from a historical perspective (e.g., past 5-10 years).

Non-Volatile

- It is always a physically separate store of data transformed from the application data found in the operational environment.
- Information from heterogeneous sources is integrated in advance and stored in warehouses for direct query and analysis.

Q24. What is OLAP? Define any two OLAP operations. ***q 3rd

- Data warehouse systems serve users or knowledge workers in the role of data analysis and decision making, can organize and present data in various format. Such systems are known as OLAP (on-line analytical processing).

- Two important OLAP (Online Analytical Processing) operations used in data warehousing are:

Roll-up: This operation involves aggregating data from a lower level of granularity to a higher level of granularity. For example, if a data warehouse contains daily sales data, roll-up can be used to aggregate the data to weekly, monthly, or yearly sales data. This allows for a more summarized view of the data that can be used for higher-level analysis and reporting.

Drill-down: This operation is the opposite of roll-up and involves breaking down aggregated data to a lower level of granularity. For example, if a data warehouse contains monthly sales data, drill-down can be used to break down the data to weekly or daily sales data. This allows for a more detailed view of the data that can be used for more granular analysis and reporting.

Other OLAP operations include slice and dice, which involve selecting a subset of data based on certain criteria (slice) and then rearranging the data in a different way (dice) for analysis, and pivot, which involves reorganizing the data to display it in a different way for analysis. All of these OLAP operations are designed to help users analyze and report on data in a flexible and efficient way.

Q25. Why should we need a separate data warehouse?

There are several reasons why organizations should consider investing in a separate data warehouse:

- **Centralization of data:** A data warehouse collects data from multiple sources and consolidates it into a single, centralized repository, making it easier to manage and maintain.
- **Integration of data:** A data warehouse integrates data from disparate sources, enabling cross-functional analysis and reporting.
- **Historical analysis:** A data warehouse stores historical data, allowing for the analysis of trends and patterns over time.
- **Business intelligence and analytics:** A data warehouse provides a foundation for business intelligence and analytics applications, enabling organizations to gain insights and make data-driven decisions.
- **Performance optimization:** A data warehouse is designed for fast, efficient querying and analysis, enabling users to access and analyze large volumes of data quickly and easily.
- Data quality: A data warehouse provides a platform for improving data quality through data cleaning, standardization, and validation.
- Cost-effective storage: A data warehouse can store large volumes of data cost-effectively, reducing the need for expensive hardware and software investments.

Overall, a separate data warehouse can help organizations to better manage, integrate, and analyze their data, leading to improved decision-making and business performance. It allows organizations to overcome the limitations of traditional transaction processing systems, which are designed primarily for day-to-day operations rather than analysis and reporting.

Q26. Differentiate between operational DBMS and Data Warehouse.

- **On-line operational database systems** perform the online transaction and query processing. These systems are called OLTP (online transaction processing).
    - Performs day-to-day operations: purchasing, inventory, banking, manufacturing, payroll, registration, accounting, etc.
- **Data warehouse systems** serve users or knowledge workers in the role of data analysis and decision-making and can organize and present data in various formats. Such systems are known as OLAP (online analytical processing).
- **Distinct features (OLTP vs. OLAP)**
    - **User and system orientation**: OLTP is customer-oriented, but OLAP is market-oriented
    - **Data contents**: OLTP manages current data in detail, but the OLAP system manages a large amount of historical and consolidated data
        - **Database design**: OLTP usually adopts ER data model and application-oriented database design, but OLAP system adopts star and subject-oriented database design

- **View**: OLTP system focuses on current data within an enterprise or department, but OLAP system often spans multiple versions of a database schema and integrates information from many data stores.
- **Access patterns**: OLTP access patterns consist of short, atomic transactions, and require update operation, but OLAP systems require only read-only operations and also may involve complex queries.

In summary, an operational DBMS is optimized for transaction processing and real-time data updates, while a data warehouse is optimized for analytical processing and historical data analysis. They serve different purposes and have different characteristics, but both are important components of a modern data management ecosystem.

Q27. Define the terms: i) fact table, ii) measures. ***q 3rd

- In the context of data warehousing and OLAP (Online Analytical Processing), the terms fact table and measures are defined as follows:

- **Fact table:** A fact table is a central table in a data warehouse schema that contains quantitative and numerical data (facts) about a particular business process or activity, such as sales, revenue, inventory, or customer transactions. The fact table typically contains foreign keys to link to one or more-dimension tables, which provide additional context and descriptive information about the facts, such as the date, product, location, or customer attributes. The fact table is usually designed in a star schema or snowflake schema, which simplifies queries and analysis by reducing the number of tables joins.

- **Measures:** Measures are the numerical values or metrics that are stored in the fact table and used to perform data analysis and reporting. Measures represent the quantitative aspects of the business process or activity being analyzed, such as the total sales revenue, the average order size, the number of units sold, or the profit margin. Measures can be aggregated or summarized in various ways to provide different levels of granularity and insights into the data, such as by time period, product category, sales region, or customer segment. Measures are typically organized into hierarchies or levels, which enable drill-down and roll-up operations in OLAP cubes and reports.

Q28. Differentiate between OLTP and OLAP. What are the different OLAP operations? ***q 3rd

- **Distinct features (OLTP vs. OLAP)**
    - **User and system orientation**: OLTP is customer-oriented, but OLAP is market-oriented
    - **Data contents**: OLTP manages current data in detail, but the OLAP system manages a large amount of historical and consolidated data
        - **Database design**: OLTP usually adopts ER data model and application-oriented database design, but OLAP system adopts star and subject-oriented database design

- **View**: OLTP system focuses on current data within an enterprise or department, but OLAP system often spans multiple versions of a database schema and integrates information from many data stores.
- **Access patterns**: OLTP access patterns consist of short, atomic transactions, and require update operation, but OLAP systems require only read-only operations and also may involve complex queries.

Overall, while both OLTP and OLAP systems are important for managing and processing data in organizations, they serve different purposes and are optimized for different types of processing and analysis.

OLAP (Online Analytical Processing) operations are used to perform multidimensional analysis of data in data warehouses. OLAP operations allow users to analyze data in a flexible and interactive manner, enabling them to gain insights into the data quickly and easily. Here are the most commonly used OLAP operations:

1) Slice, 2) Dice, 3) Roll-Up, 4) Drill-Down, 5) Pivot, 6) Slice-and-Dice

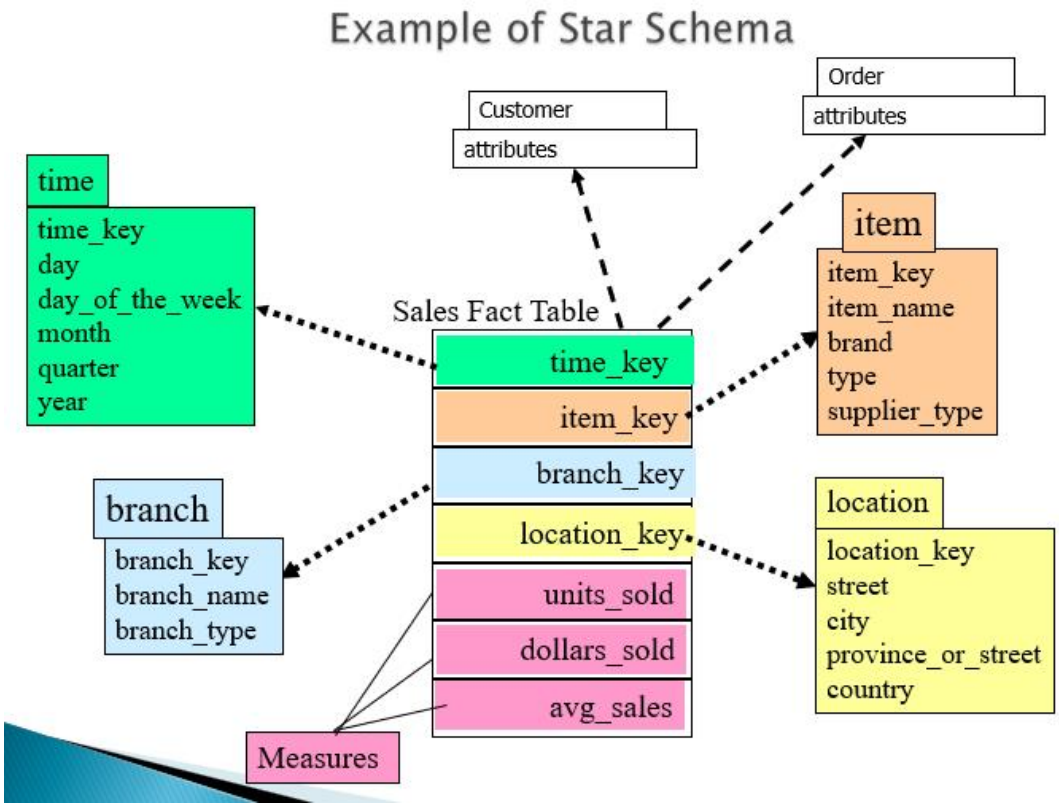
These OLAP operations enable users to analyze data in various ways, from high-level summaries to detailed views, and from different perspectives, such as by product, region, or time period. By using these operations, users can gain insights into the data quickly and easily, enabling them to make informed decisions based on the data.

Q29. Design a warehouse schema for Airlines ticket sales system using Star model showing its dimensions and fact table with required attributes and measures. The measures are aggregated values for attributes contained in the fact table. These measures are usually used by the decision making body of the organization, e.g., total sales in BDT, total flight, total passengers per flight etc. ***q 3rd

Q29. Design a data warehouse schema for the sales system using the star model showing its dimension and fact table. The measures are contained in the fact table which is aggregated attribute usually used by the decision-making body of an organization.

Star Schema:

- It is the most common modelling paradigm.
- In this model, the data warehouse contains
  1. A large central table (fact table) containing the bulk of the data, with no redundancy, and
  2. A set of smaller attendant tables (dimension tables), one for each dimension
- In the figure, Sales are considered along six dimensions, namely time, item, branch, customer, order, and location.
- The schema contains a central fact table for **sales** that contain keys to each of the six dimensions, along with three measures dollars_sold, units_sold, and avg_sales.
- Each dimension table contains a set of attributes

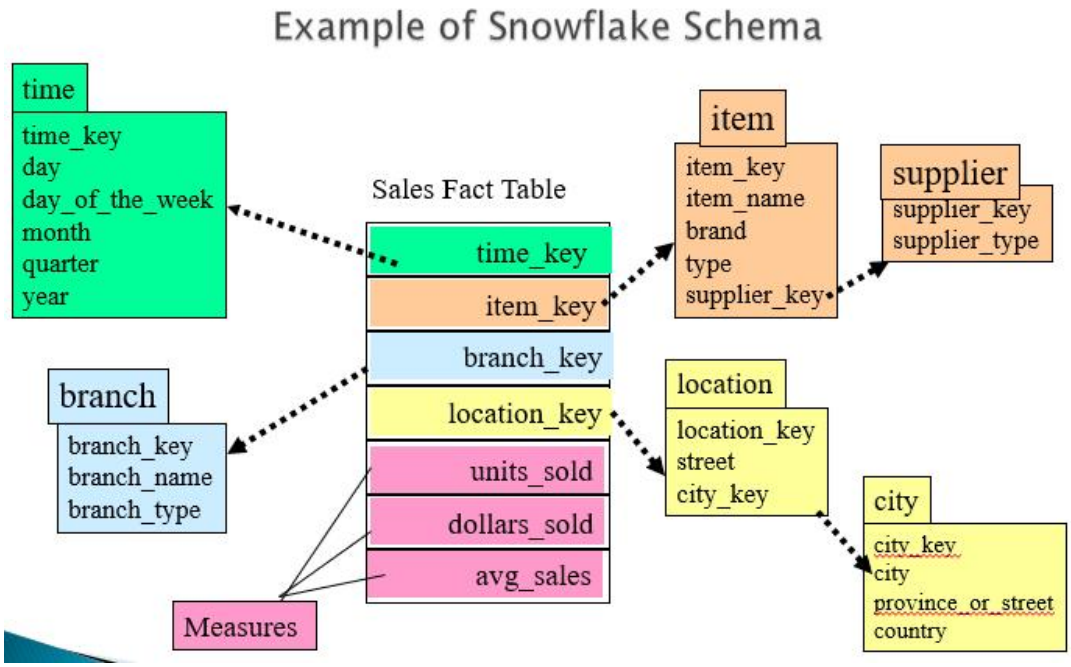


Q30. Design a data warehouse schema for Agriculture Crop production system using Snowflake model showing its dimensions and fact table with required attributes and measures. The information about area, total crops and the year of crop production will be stored in the dimension and fact table. The measures are usually used by the decision making body of the organization, e.g., total crops produced in metric ton in an area etc.
***q 3rd

Q. Design a data warehouse schema for the sales system using the snowflake model showing its dimensions and fast table. The model should contain multiple dimensions with the attributes and measures of the fact table as required by the sales system.

- ***Snowflake Schema:***

- It is a variant of the star schema model. Some dimension tables are normalised, thereby further splitting the data into additional tables.
- The resulting schema graph forms a shape similar to a snowflake.
- The major difference between the star and the snowflake schema models is that the dimension table of the snowflake model may be kept in normalised form to reduce redundancies. Such a table is easy to maintain and saves space.
- In the example, the dimension table **item** is normalised, resulting into new item and **supplier** tables. The item dimension table now contains the attributes item_key, item_name, brand, type and supplier_key where supplier_key is linked to the supplier dimension table.



Q31. Define the terms: i) Classification, ii) Decision tree.

- **Classification:** Classification is a supervised learning technique in machine learning that involves categorizing data into predefined classes or categories based on their features or attributes. The goal of classification is to build a predictive model that can accurately classify new, unseen instances based on their features. Classification is used in various applications such as image recognition, spam filtering, fraud detection, and medical diagnosis.

- **Decision tree:** A decision tree is a tree-like model that represents a sequence of decisions and their possible consequences or outcomes. In machine learning, decision trees are used for classification and regression tasks. A decision tree consists of nodes that represent decisions, branches that represent the possible outcomes or options, and leaves that represent the final classification or prediction. The root node is the topmost node that represents the initial decision or question, and each internal node represents a decision based on one of the features or attributes of the data. The branches represent the possible values of the feature, and the leaves represent the final classification or prediction. Decision trees can be visualized as a tree-like structure, which is easy to interpret and understand, making them a popular choice for many machine learning applications.

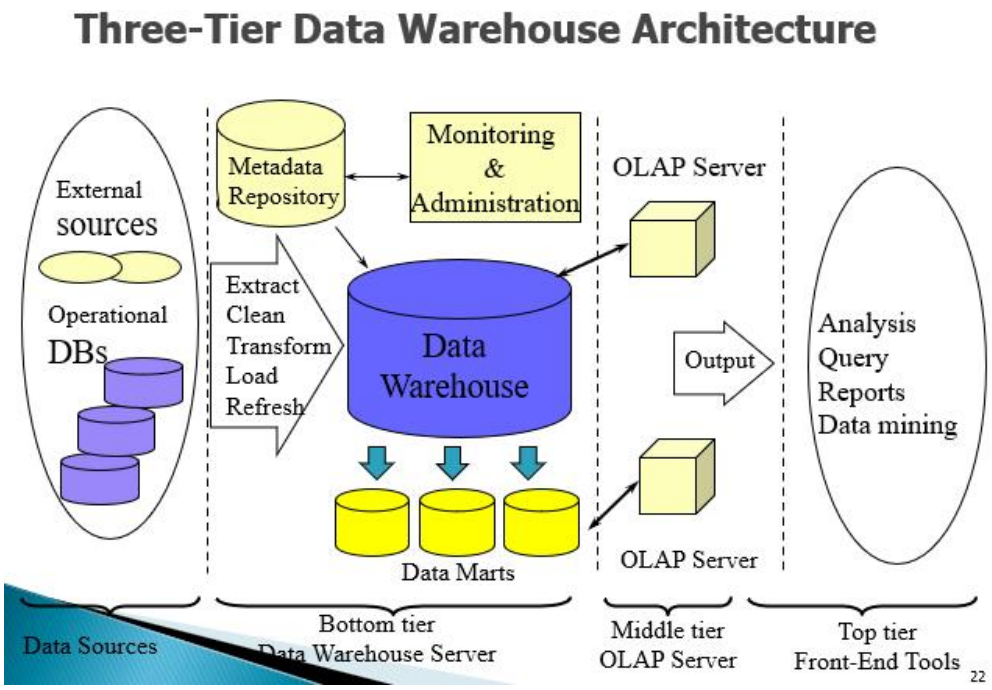
Q32. Describe the three-tier data warehouse architecture.

- The three-tier data warehouse architecture is a commonly used architecture for designing and implementing data warehouses. This architecture consists of three layers, each with a specific function:

- **Bottom Tier (Data Storage Layer):** The bottom tier is also known as the data storage layer, and it is responsible for storing the raw data that is extracted from various sources. This layer is also known as the data warehouse layer because it is where the data warehouse is actually located. The data in this layer is typically stored in a relational database management system (RDBMS), and it is optimized for fast write operations and data integrity.

- **Middle Tier (Data Integration Layer):** The middle tier is also known as the data integration layer, and it is responsible for integrating and transforming the raw data from the data storage layer into a format that is suitable for analysis. This layer is where data cleansing, data transformation, and data aggregation take place. The data in this layer is often stored in a separate database or file system, and it is optimized for fast read and write operations.

- **Top Tier (Data Presentation Layer):** The top tier is also known as the data presentation layer, and it is responsible for presenting the transformed data to the end-user in a meaningful way. This layer is where data visualization, reporting, and analysis take place. The data in this layer is often stored in a separate database or file system, and it is optimized for fast-read operations and query performance.



The three-tier data warehouse architecture provides a clear separation of concerns between the different layers, which makes it easier to maintain and modify the system over time. It also enables scalability and flexibility because each layer can be upgraded or replaced independently without affecting the other layers.

Q33. Describe Hunt's algorithm for classification. ***q 4th

Hunt's Algorithm

- A decision tree is grown in a recursive fashion
- Partitions the training records into successively purer subsets
- The following variables are used:
    - D_t = set of training records for node t
    - Y = {y_1, y_2, …, y_c} are the class labels

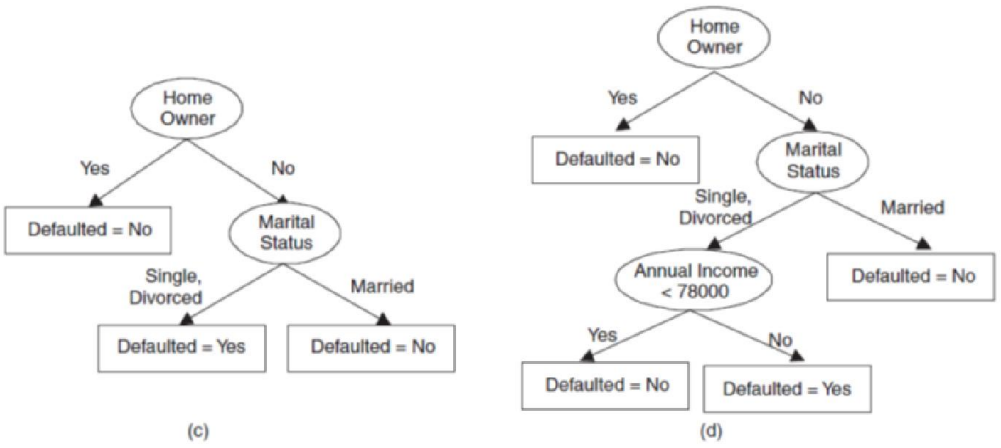


Figure 3.6. Hunt's algorithm for building decision trees.

See Fig. 4.6 from Book

Hunt's Algorithm

Step 1: If all the records in D_t belong to the same class y_t then t is a leaf node labeled as y_t

Step 2: If D_t contains records that belong to more than one class, an attribute test condition is selected to partition the records into smaller subsets. A child node is created for each outcome of the test condition. The records in D_t are distributed to the children based on the outcomes. The algorithm is then recursively applied to each child node.

Q34. Describe Design Issues for Decision Tree Induction.

Design Issues for Decision Tree Induction

- How should the training records be split?
    - Each recursive step of the tree-growing process must select an attribute (based on information gain) test condition to divide the records into smaller subsets.
- How should the splitting procedure stop?
    - A possible strategy is to continue expanding a node until either all the records belong to the same class or all the records have identical attribute values.

- **Methods of Expressing Attribute Test Condition**

  i) Binary attributes ii) Nominal attributes iii) Ordinal attributes iv) Continuous attributes

Q35. What is DMQL? Give the full specification of a DMQL query for mining characteristics about customer purchasing in an electronics shop.

- DMQL stands for Data Mining Query Language, a specialized query language that extracts valuable information from large datasets.

Here's an example of a DMQL query that could be used to mine characteristics of customer purchasing in an electronics shop:

```
SELECT
    customer.age,
    customer.gender,
    product.category,
    AVG(sale.amount) as avg_purchase_amount,
    COUNT(sale.id) as purchase_count
FROM
    customer
    JOIN sale ON customer.id = sale.customer_id
    JOIN product ON sale.product_id = product.id
WHERE
    product.category LIKE 'electronics%'
GROUP BY
    customer.age,
    customer.gender,
    product.category
```

This query selects the age and gender of each customer, as well as the category of the electronics product they purchased. It also calculates the average purchase amount and counts for each customer and product category.

The query then filters the results only to include products in the "electronics" category and groups the results by customer age, gender, and product category.

Overall, this query can be used to gain insights into which demographics purchase electronics products most frequently and how much they spend on average.

Q36. What is DMQL? Write down the primitives for defining a data mining task. ***q 5th

DMQL stands for Data Mining Query Language. It is a specialized language used for querying and manipulating data in the context of data mining tasks. DMQL provides a set of primitives or commands that allow users to define and execute various data mining operations. These primitives help in specifying the desired data mining task and provide instructions to the data mining system on how to perform the task.

- The primitives for defining a data mining task in the form of a data mining query are the following:

- The set of task relevant data to be mined
- The kind of knowledge to be mined
- The background knowledge to be used in the discovery process
- The interestingness measures and thresholds for pattern evaluation
- The expected representation for visualizing the discovered patterns

- Based on these primitives, a query language for data mining called data mining query language (DMQL) is designed.
- DMQL allows the ad hoc mining of several kinds of knowledge from relational databases and data warehouses at multiple levels of abstraction.

By using these primitives in DMQL, data analysts can define a structured and repeatable process for performing data mining tasks and extracting valuable insights from large datasets.

Or

- DMQL, or Data Mining Query Language, is a specialized language used to extract useful information from large datasets through data mining techniques.

The primitives for defining a data mining task in DMQL typically include the following:

1. ***Data Selection:*** This involves selecting a specific subset of data from a larger dataset based on certain criteria or filters, such as time range, specific attributes, or certain values.
2. ***Data Transformation:*** This involves transforming the selected data into a form that is suitable for analysis. This may include aggregating, grouping, or cleaning the data.
3. ***Data Mining:*** This is the core of the data mining task and involves applying specific data mining algorithms or techniques to the transformed data to discover patterns, trends, or relationships.
4. ***Pattern Evaluation:*** This involves evaluating the patterns or relationships discovered through data mining to determine their relevance or usefulness for the intended purpose.
5. ***Knowledge Representation:*** This involves representing the patterns or relationships discovered through data mining in a meaningful way, such as through visualizations, summary statistics, or descriptive language.
6. ***Deployment:*** This involves deploying the results of the data mining task in a practical context, such as using them to make predictions, inform decision-making, or improve business processes.

By using these primitives in DMQL, data analysts can define a structured and repeatable process for performing data mining tasks and extracting valuable insights from large datasets.

Q37. What is DMQL? Write down the DMQL query for mining the general characteristics of customers who frequently shop in ABC Electronics. The tables are an item, customers, purchases, and items sold to store data values. Define the DMQL query to characterize the buying habits of customers who purchase items priced at no less than BDT 5500 with respect to the customer's age, the type of item purchased, and the place in which the item was made. For each characteristic discovered, you would like to know the percentage of customers having that characteristic. In particular, you are only interested in purchases in Dhaka and paid for with an "SC Bank" credit card. You would like to view the resulting descriptions in the form of a table. ***q 5th

Give the data mining query for the above problem in DMQL form.

- DMQL, or Data Mining Query Language, is a specialized language used to extract useful information from large datasets through data mining techniques.

Here's an example of a DMQL query that could be used to mine the general characteristics of customers who frequently shop in ABC Electronics, specifically focusing on purchases of items priced at no less than BDT 5500 in Dhaka, paid for with an "SC Bank" credit card:

```
SELECT
    customer.age,
    item.type,
    purchases.place,
    COUNT(DISTINCT purchases.customer_id) as customer_count,
    ROUND((COUNT(DISTINCT purchases.customer_id) * 100) / (SELECT COUNT(*) FROM
customers), 2) as percentage
FROM
    purchases
    JOIN items_sold ON purchases.id = items_sold.purchase_id
    JOIN item ON items_sold.item_id = item.id
    JOIN customer ON purchases.customer_id = customer.id
WHERE
    purchases.place = 'Dhaka'
    AND purchases.payment_method = 'SC Bank'
    AND item.price >= 5500
GROUP BY
    customer.age,
    item.type,
    purchases.place
```

This query selects the age of each customer, the type of item purchased, and the place in which the item was made, filtering only purchases in Dhaka that were paid for with an "SC Bank" credit card and with items priced at no less than BDT 5500.

The query then groups the results by customer age, item type, and place, and calculates the number of distinct customers and their percentage out of the total number of customers in the dataset.

Overall, this query can be used to gain insights into the general characteristics of customers who frequently shop in ABC Electronics, explicitly focusing on high-value purchases made with an "SC Bank" credit card in Dhaka. The resulting descriptions are presented in the form of a table.

Q38. Write Down the decision tree induction algorithm. ***q 5th 4th

- The decision tree induction algorithm is a popular machine learning algorithm for classification and regression problems. It creates a decision tree by recursively partitioning the training data into subsets based on the values of the input features. Here is the general algorithm:

Description of the algorithm

- A skeleton decision tree induction algorithm called TreeGrowth is given.
- The algorithm consists of the training records E, and the attribute set F

Decision Tree Induction Algorithm

Algorithm 1 A skeleton decision tree induction algorithm TreeGrowth (E,F)

```
1:  if stopping_cond(E,F)=true then
2:      leaf =createNode()
3:      leaf.label=Classify(E)
4:      return leaf
5:  else
6:      root= createNode()
7:      root.test_cond=find_best_split(E,F)
8:      let V={v|v is a possible outcome of root.test_cond}
9:      for each v ∈ V do
10:         Ev = {e| root.test_cond (e)=v and e ∈ E}
11:         child = TreeGrowth(Ev, F)
12:     add child as descendant of root and label the edge as (root->child) as v
13:     endfor
14:  end if
15: return root
```

Q39. Explain the basic idea of the decision tree induction algorithm defining the functions used in the algorithm. ***q 5th 4th

- Functions Used in Decision Tree Induction Algorithm

Four functions are used in this algorithm:

1. CreateNode(): It extends the decision tree by creating a new node. A node has either a test condition *node.test_cond*, or a class label *node.label*.
2. Find_best_split(): It determines which attribute should be used as a test condition for splitting the training records.
3. Classify(): It determines the class label to be assigned to a leaf node.
4. Stopping_cond(): It is used to terminate the tree growing process. It tests whether all the records have either the same class label or the same attribute values.

Tree Pruning

- It is used to reduce the size of the tree

Overfitting

- A model that fits the training data too well can have a poorer generalization error than a model with a higher training error, which is known as model overfitting.

In summary, the decision tree induction algorithm creates a tree-like model of decisions and their possible consequences by recursively partitioning the dataset based on the values of the input features. The model predicts the class or value of new instances based on their feature values, using decision rules at each node in the tree. The algorithm uses different functions, such as splitting, stopping, pruning, and prediction functions, to create and optimize the decision tree model.

Q40. Define the terms: i) Tree Pruning ii) Overfitting. ***q 5th

- **Tree Pruning**

  - It is used to reduce the size of the tree

  i) Tree Pruning: Tree pruning is a technique used in decision tree learning to reduce the size of a decision tree by removing sections of the tree that do not contribute significantly to its accuracy. Pruning helps to avoid overfitting, improve the generalization of the model, and reduce the complexity of the tree. There are several methods of pruning, including reduced error pruning, cost-complexity pruning, and rule post-pruning.

Overfitting

- A model that fits the training data too well can have a poorer generalization error than a model with a higher training error, which is known as model overfitting.

ii) Overfitting: Overfitting is a phenomenon that occurs when a machine learning model is too complex or flexible, such that it fits the training data too closely and captures the noise in the data rather than the underlying patterns or relationships. Overfitting results in poor generalization of the model, meaning that it performs well on the training data but poorly on new, unseen data. Overfitting can be addressed through various techniques, including regularization, cross-validation, and early stopping.

Q41. Define any two methods of expressing attribute test conditions for creating new branches of a decision tree. ***q 4th ans: 5.chap3_basic_classification, need to check

Methods of Expressing Attribute Test Condition

i) Binary attributes
ii) Nominal attributes
iii) Ordinal attributes
iv) Continuous attributes

The definitions of four types of attributes in data mining:

1. Binary attributes

Binary attributes are attributes that can have only two values. For example, the "gender" attribute can have the values "male" and "female".

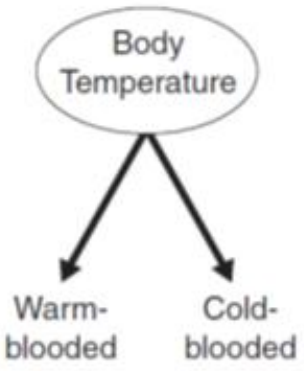
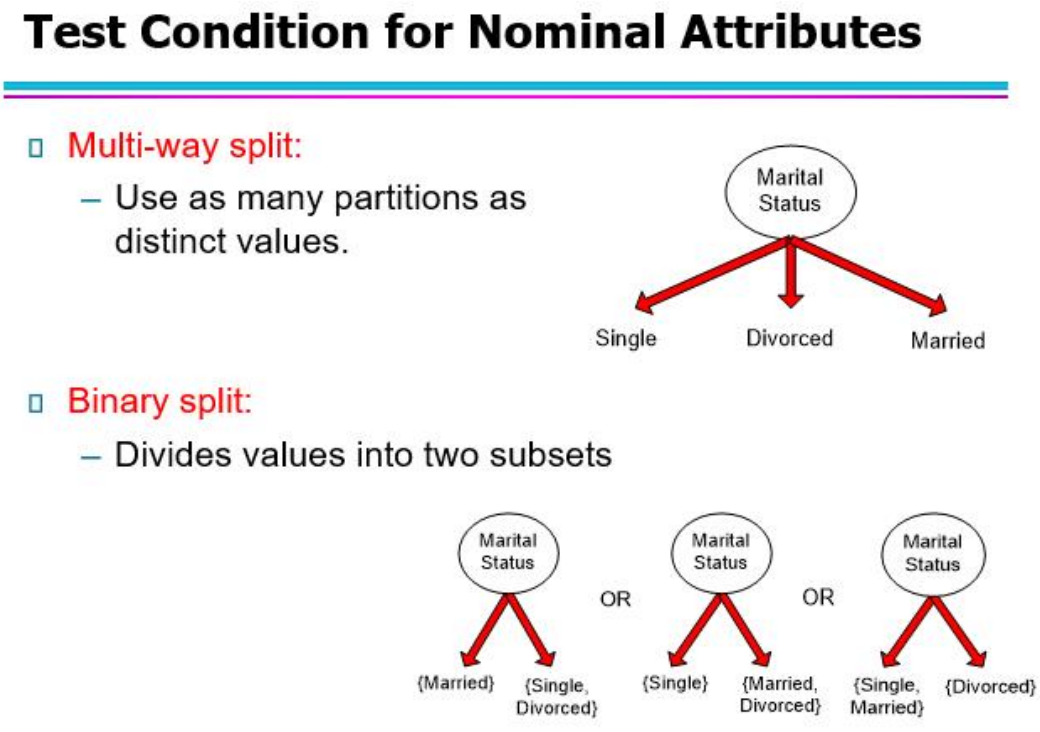


Figure 3.7. Attribute test condition for a binary attribute.

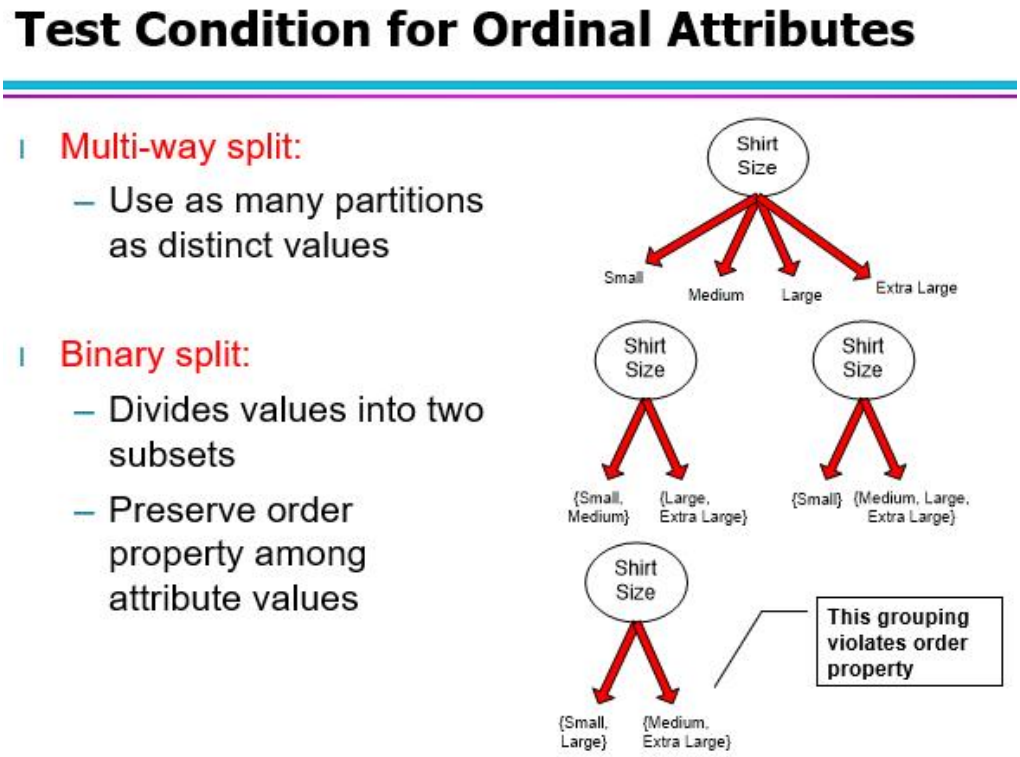
2. Nominal attributes

Nominal attributes are attributes that can have any number of values, but the values have no inherent order. For example, the "color" attribute can have the values "red", "green", "blue", etc.



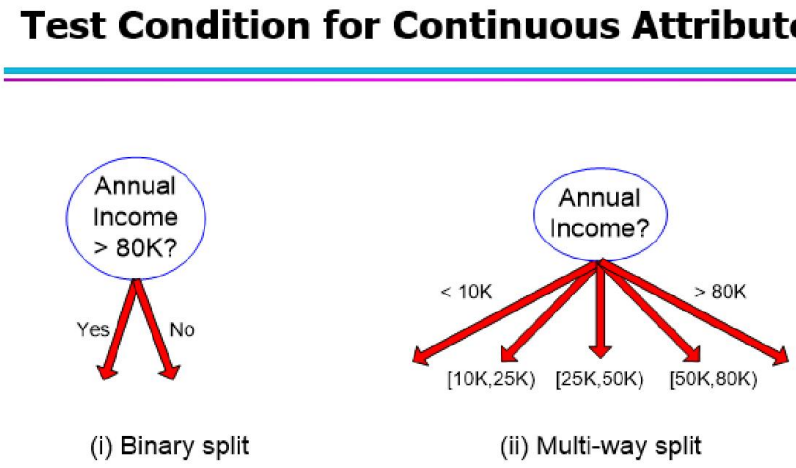
3. Ordinal attributes

Ordinal attributes are attributes that can have any number of values, and the values have a natural order. For example, the "size" attribute can have the values "small", "medium", "large", etc.



4. Continuous attributes

Continuous attributes are attributes that can have any number of values, and the values can be ordered and spaced evenly. For example, the "height" attribute can have any value between 0 and infinity.



In general, it is best to use the simplest type of attribute that can represent the desired data. However, if the simplest type of attribute is not expressive enough, then a more complex type of attribute may be necessary.

See Fig. 4.8

Q42. Explain the decision tree construction process using decision tree induction.

Decision Tree Construction

- Exponentially many decision trees can be constructed from a given set of attributes
- Finding the optimal tree is computationally infeasible
- Efficient algorithms have been developed to induce suboptimal decision trees
- Hunt's algorithm is the basis of many decision tree induction algorithms like ID3, C4.5, and CART

Fig. 4.5

Q43. Define the functions used in the decision tree induction algorithm for classification. ***q 4th

- The decision tree induction algorithm for classification involves the following functions the Four functions are used in this algorithm:

1. **CreateNode**(): It extends the decision tree by creating a new node. A node has either a test condition *node.test_cond*, or a class label *node.label*.
2. **Find_best_split**(): It determines which attribute should be used as a test condition for splitting the training records.
3. **Classify**(): It determines the class label to be assigned to a leaf node.
4. **Stopping_cond**(): It is used to terminate the tree growing process. It tests whether all the records have either the same class label or the same attribute values.

These functions work together to construct a decision tree that accurately represents the underlying data and can be used to make predictions on new instances. The quality of the decision tree depends on the choice of the splitting criterion, the stopping criterion, the leaf node labeling function, and the pruning method.

Q44. Define rule-based classifier with example. ***q 5th 7th

- It is a technique for classifying records using a collection of "if…then…" rules.
- Rules for the model are represented in a disjunctive normal form

  $R = (r_1 \vee r_2 \vee \ldots r_k)$

  where R is the rule set and r_i 's are the classification rules or disjuncts.

  Example of a rule set for the vertebrate classification problem

  r1: (Gives Birth = no) ∧ (Aerial Creature = yes) → Birds
  r2: (Gives Birth = no) ∧ (Aquatic Creature = yes) → Fishes

Table 5.1. Example of a rule set for the vertebrate classification problem.

r_1:	(Gives Birth = no) ∧ (Aerial Creature = yes) ⟶ Birds
r_2:	(Gives Birth = no) ∧ (Aquatic Creature = yes) ⟶ Fishes
r_3:	(Gives Birth = yes) ∧ (Body Temperature = warm-blooded) ⟶ Mammals
r_4:	(Gives Birth = no) ∧ (Aerial Creature = no) ⟶ Reptiles
r_5:	(Aquatic Creature = semi) ⟶ Amphibians

- Left hand side is called antecedent or precondition
- Right hand side is called rule consequent

Q45. Describe Bayesian Classifier. ***q 5th

- In many applications, the relationship between the attribute set and the class variable is non-deterministic
- The class label of a test record cannot be predicted with certainty even though its attribute set is identical to some of the training records because of the presence of noisy data
- The approach for modeling probabilistic relationships between the attribute set and the class variable is represented with the help of Bayes theorem.

Q46. Describe Bayes theorem. **

- It is a statistical principle for combining prior knowledge of the classes with new evidence gathered from data
- Two implementations of Bayesian classifiers:
    1. Naïve Bayes
    2. Bayesian belief networks

- **Bayes theorem:**

$$P(Y|X) = \frac{P(X|Y)P(Y)}{P(X)}.$$

The Bayes theorem can be used to solve the prediction problem stated at the beginning of this section. For notational convenience, let X be the random variable that represents the team hosting the match and Y be the random variable that represents the winner of the match. Both X and Y can take on values from the set {0,1}. We can summarize the information given in the problem as follows:
Probability Team 0 wins is P(Y=0) =0.65.
Probability Team 1 wins is P(Y=1) =1−P(Y=0) = 0.35.
Probability Team 1 hosted the match it won is P(X=1|Y=1) =0.75.
Probability Team 1 hosted the match won by Team 0 is P(X=1|Y=0) =0.3.

Q47. Describe Bayesian Theorem for Classification. **

- Let X denote the attribute set and Y denote the class variable
- If X and Y are random variables, we can capture their relationship probabilistically using P(Y|X)- conditional probability, or posterior probability for Y where prior probability is P(Y).

  e.g., Task of predicting whether a loan borrower will default on their payments
- If P(Yes|X) > P(No|X), then the record is classified as Yes, otherwise it is classified as No.

Q48. Describe the Bayesian classifier for data mining classification. ***q

- The Bayesian classifier is a data mining classification algorithm that uses Bayes' theorem to compute the probabilities of each class label given the attribute values of an instance. The Bayesian classifier is a probabilistic model that estimates the probability distribution of each class label based on the training data and uses this distribution to make predictions for new instances.

- In many applications, the relationship between the attribute set and the class variable is non-deterministic
- The class label of a test record cannot be predicted with certainty even though its attribute set is identical to some of the training records because of the presence of noisy data
- The approach for modeling probabilistic relationships between the attribute set and the class variable is represented with the help of Bayes theorem.

The Bayesian classifier can handle missing attribute values and can also handle continuous attribute values by modeling their probability distributions. However, the Bayesian classifier assumes that the attributes are conditionally independent given the class label, which may not hold in all domains. In such cases, other classification algorithms, such as decision trees and neural networks, may be more appropriate.

Q49. What is Association Analysis? Explain with an example of association rule using customer purchase records. ***q 6th

It is useful for discovering interesting relationships hidden in large data sets. The uncovered relationships can be represented in the form of association rules or sets of frequent items. For example, the following rule can be extracted from the data set shown in Table 6.1:

Table 6.1. An example of market basket transactions.

TID	Items
1	{Bread, Milk}
2	{Bread, Diapers, Beer, Eggs}
3	{Milk, Diapers, Beer, Cola}
4	{Bread, Milk, Diapers, Beer}
5	{Bread, Milk, Diapers, Cola}

E.g., The rule {Diapers} $\rightarrow$ {Beer} can be extracted from Table: 6.1.
Another example of an association rule: {Bread, Sweet} $\rightarrow$ {Milk}

An example of an association rule using a customer purchase record could be:

Association rule:

- It is an implication expression of the form X $\rightarrow$ Y where X and Y are disjoint item sets, X $\cap$ Y = $\varnothing$.
- The strength of an association rule is measured using support and confidence

"If a customer buys milk and eggs, they are likely to buy bread as well"

In this rule, milk and eggs are called the antecedent and bread is called the consequent. The support of this rule measures how frequently milk, eggs, and bread are purchased together, while the confidence of the rule measures the conditional probability of buying bread given that a customer has already bought milk and eggs.

By identifying strong association rules like this, businesses can improve their product placement and promotions to increase sales. For example, a grocery store might place bread next to milk and eggs to encourage customers to purchase all three items together.

Q50. What is Association rule? Write down some application. ***q 6th

- **Association rule:**

- It is an implication expression of the form $X \rightarrow Y$ where X and Y are disjoint item sets, $X \cap Y = \emptyset$.
- The strength of an association rule is measured using support and confidence

It is useful for discovering interesting relationships hidden in large data sets. The uncovered relationships can be represented in the form of association rules or sets of frequent items. For example, the following rule can be extracted from the data set shown in Table 6.1:

Table 6.1. An example of market basket transactions.

TID	Items
1	{Bread, Milk}
2	{Bread, Diapers, Beer, Eggs}
3	{Milk, Diapers, Beer, Cola}
4	{Bread, Milk, Diapers, Beer}
5	{Bread, Milk, Diapers, Cola}

E.g., The rule {Diapers} $\rightarrow$ {Beer} can be extracted from Table: 6.1.
Another example of an association rule: {Bread, Sweet} $\rightarrow$ {Milk}

Applications:

- Market basket analysis, Bioinformatics, Medical diagnosis, Web mining, and scientific data analysis.

Q51. Define support and confidence of an association rule. ***q 6th

Association rule:

- It is an implication expression of the form $X \rightarrow Y$ where X and Y are disjoint item sets, $X \cap Y = \emptyset$.
- The strength of an association rule is measured using support and confidence

Support:

- It determines how often a rule is applicable to a given data set

  Support, $s(X \rightarrow Y) = \sigma(X \cup Y) / N$ where N is the total number of records

Confidence:

- It determines how frequently items in Y appear in transactions that contain X

  Confidence $c(X \rightarrow Y) = \sigma(X \cup Y) / \sigma(X)$

Association rule discovery:

- Given a set of transactions T, find all the rules having support ≥ minsup and confidence ≥ minconf, where minsup and minconf are the corresponding support and confidence thresholds.

Q52. Describe Association rule mining algorithm. **

- Decomposes the problem into two major subtasks:

  - Frequent item set generation: Find all the item sets that satisfy the minsup threshold called *frequent item sets*
  - Rule generation: Extract all the high confidence rules from the frequent item sets found in the previous step called strong rules
  - A data set that contains k items can potentially generate up to 2^k -1 frequent item sets excluding the null set.
  - To find frequent item sets, determine the support count for every candidate item set.
  - Compare each candidate against every transaction, If the candidate is contained in a transaction, the support count will be incremented.

Q53. What is frequent item? Describe the frequently item set generation technique using the Apriori algorithm. ***q 6th

- A frequent item is an item that appears in a dataset with a frequency that meets or exceeds a predefined threshold, known as the minimum support threshold. It is an important concept in association rule mining, as it helps to identify patterns and relationships between different items in the dataset.

The Apriori Principe

- If an item set is frequent, then all of its subsets must also be frequent.

Frequent Item set Generation using The Apriori algorithm

- It uses support-based pruning to systematically control the exponential growth of candidate item sets
- Initially, every item is considered as a candidate 1-item set. The items which have minimum support are discarded
- In the next iteration, candidate-2 item sets are generated using only the frequent 1-item sets
- Candidate 3-itemsets are formed in the similar way. This process is continued until the frequent item set is null.
- The following variables are used in the Apriori algorithm:

  $C_k \rightarrow$ denotes the set of candidate k-item sets

  $F_k \rightarrow$ denotes the set of frequent k-item sets

Q54. Write Algorithm: Frequent itemset generation of the Apriori algorithm. **

Algorithm 6.1: Frequent itemset generation of the Apriori algorithm
1: k=1
2: F_k={ i|i ∈ I ∧ σ({i}) >= N x minsup }
3: repeat
4: k=k+1
5: C_k = apriori_gen(F_{k-1})
6: for each transaction t ∈ T do
7: C_t = subset(C_k, t)
8: for each candidate itemset c ∈ C_t do
9: σ(c) = σ(c) + 1
10: end for
11: end for
12: F_k={ c|c ∈ C_k ∧ σ(c) >= N x minsup }
13: until F_k = ϕ
14: Result = ∪ F_k

Result F = F_1∪ F_2∪ F_3∪ F_4∪ … ∪ F_k

Q55. Explain candidate generation and pruning. **

- Candidate generation and pruning are two important steps in frequent itemset generation algorithms, such as the Apriori algorithm, used in association rule mining.

Candidate Generation and Pruning

- The apriori-gen() function generates candidate itemsets by performing the following two functions:
    1. **Candidate generation:** it generates new candidate k-itemsets based upon the frequent (k-1) itemsets found in the previous iteration
    2. **Candidate pruning:** This operation eliminates some of the candidate k-itemsets using the support-based pruning strategy. The infrequent item is immediately prunned.

Both candidate generation and pruning are important steps in frequent itemset generation algorithms, and their effectiveness determines the accuracy and efficiency of the association rule mining algorithm.

Q56. Define the Items: Sequence & Sub-Sequence. ***q 7th

- **Sequence**

- It is an ordered list of elements
- It can be denoted as $s = <e_1\ e_2\ e_3 \ldots\ e_n>$ where each element e_j is a collection of one or more events $e_j = <i_1, i_2, \ldots, i_k>$

Example

- Sequence of web pages viewed by a web site visitor: <{Homepage} {Electronics} {Cameras and Camcorders} {Digital Cameras} {Shopping cart} {Order confirmation} {Return to shopping}>
- A sequence can be characterized by its length i.e., number of elements, and the number of occurring events. Hence, a *k*-sequence is a sequence that contains *k* events.

Subsequence

- A sequence *t* is a subsequence of another sequence *s* if each ordered element in *t* is a subset of an ordered element in *s*.
- The sequence $t = <t_1\ t_2 \dots\ t_m>$ is a subsequence of $s = <s_1\ s_2 \dots\ s_n>$ if there exists integers $1 \leq j_1 < j_2 < j_m \leq n$ such that $t_1 \subseteq s_{j1}$, $t_2 \subseteq s_{j2}$, …, $t_m \subseteq s_{jm}$
- If *t* is a subsequence of *s*, then *t* is contained in *s*

Sequence, s	Sequence, *t*	Is *t* a subsequence of *s?*
<{2,4} {3,5,6} {8}>	<{2} {3,6} {8}>	yes
<{1,2} {3,4} >	<{1} {2} >	no

Q57. Answer the following and explain the reason in favor of your answer. *** 7th (value different thakbe just)

Sequence, s	Sequence, *t*	Is *t* a subsequence of *s?*
<{2,4} {3,5,6} {8}>	<{2} {3,6} {8}>	
<{1,2} {3,4} >	<{1} {2} >	

Subsequence

- A sequence *t* is a subsequence of another sequence *s* if each ordered element in *t* is a subset of an ordered element in *s*.
- The sequence $t = <t_1\ t_2 \dots\ t_m>$ is a subsequence of $s = <s_1\ s_2 \dots\ s_n>$ if there exists integers $1 \leq j_1 < j_2 < j_m \leq n$ such that $t_1 \subseteq s_{j1}$, $t_2 \subseteq s_{j2}$, …, $t_m \subseteq s_{jm}$
- If *t* is a subsequence of *s*, then *t* is contained in *s*

Sequence, s	Sequence, *t*	Is *t* a subsequence of *s?*
<{2,4} {3,5,6} {8}>	<{2} {3,6} {8}>	yes
<{1,2} {3,4} >	<{1} {2} >	no

Q58. Define Sequential pattern discovery. What is frequent sequence? ***q 7th

- Given a sequence data set D and a user specified minimum support threshold *minsup*, the task of sequential pattern discovery is to find all sequences with support ≥ *minsup*.
- The input to sequential pattern discovery is **a sequence data set**

Object	Timestamp	Events
A	10	2, 3, 5
A	20	6, 1
A	23	1
B	11	4, 5, 6
B	17	2
B	21	7, 8, 1, 2
B	28	1, 6
C	14	1, 8, 7

- Each row records the occurrences of events associated with a particular object at a given time

Sequential pattern or frequent sequence:

- If the support for a sequence *s* is greater than or equal to a user specified threshold *minsup*, then *s* is a *sequential pattern* or *frequent sequence*.

Sequential pattern discovery definition

- Given a sequence data set D and a user specified minimum support threshold *minsup*, the task of sequential pattern discovery is to find all sequences with support ≥ *minsup*.

Q59. Describe Counting support for a sequence. **

Counting support for a sequence

- A data set that contains 5 data sequences for data objects A, B, C, D and E
- Support for the sequence <{1}{2}>: s=80% as it occurs in the 4 of the 5 data sequences every object except for D
- Assume minimum support threshold is 50%, any sequence that appears in at least three data sequences is a sequential pattern.
- See Figure below: Sequential patterns derived from a data set that contains five data sequences.

Object	Timestamp	Events
A	1	1,2, 4
A	2	2,3
A	3	5
B	1	1,2
B	2	2,3,4
C	1	1,2
C	2	2,3,4
C	3	2,4,5
D	1	2
D	2	3,4
D	3	4,5
E	1	1,3
E	2	2,4,5

Q60. Describe Apriori principle with example. **

It holds for sequential data as any data sequence that contains a particular k-sequence must also contain all of its (k-1) subsequences.

Example

- Consider minimum support = 50%
- Sequential pattern: <{1, 2}>, s = 60%
- Five data sequences
- Support for the sequence <{1} {2}> = 80%, appear in 4 of the 5 data sequences (every object except D)

Q61. What is Big-Data? Explain Big-Data analysis. ***q 7th

- **Definition**

As in Source: Gartner IT Glossary

- Big-data is high-volume, high-velocity, and high-variety information assets that demand cost effective, innovative forms of information processing, for enhanced insight and decision making.
- Big data is data that is big in volume, velocity, and variety
- Extremely large, very fast, highly diverse, and complex data that cannot be managed with traditional data management tools. See Figure.

Big Data Analytics

It is technology enabled analytics-

- A quite few data analytics and visualization tools are available e.g., from SAS, R Analytics to help process and analyze big data.
- About gaining a meaningful, deeper and richer insight into the business, better leveraging the services of the vendors and suppliers.
- Enables to obtain findings that allow quicker and better decision making in a competitive edge over the competitors
- A handshake between 3 communities- IT, business users and data scientists
- Working with datasets exceeding the storage and processing capability and infrastructure of the organization
- About moving code to data

Big Data processing tools

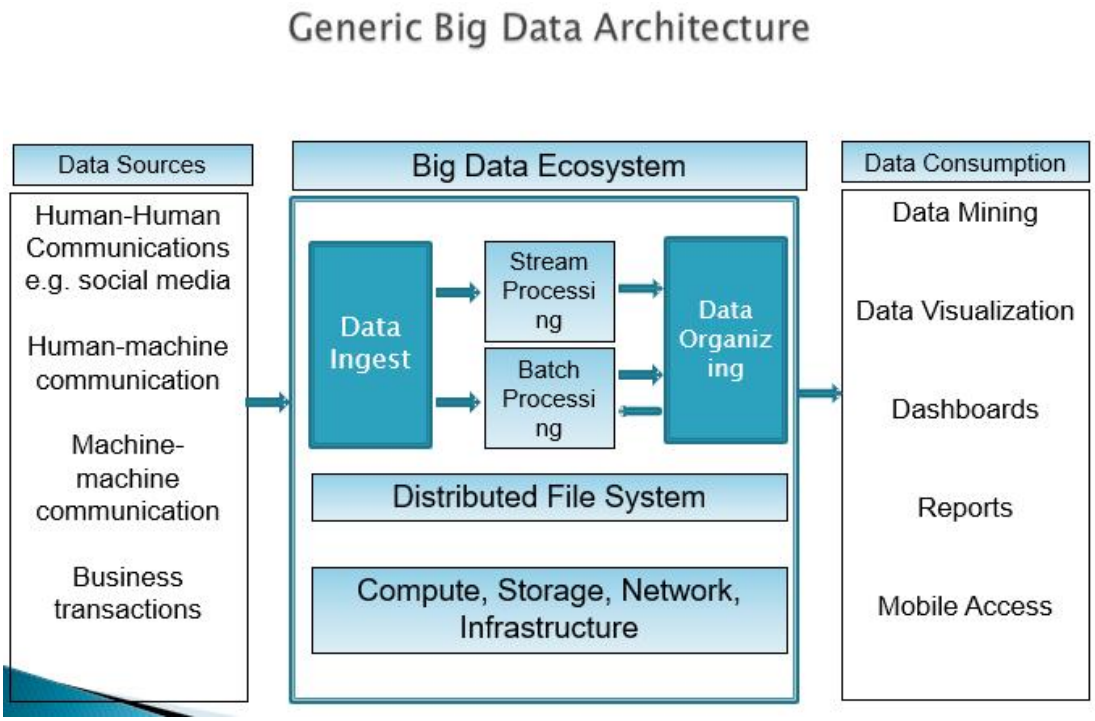
- Hadoop
- NoSQL
- Mongo DB
- Lotus Domino

Q62. Write Down some application of Big-Data. ***q 7th

The Application of Big-Data:

- Monitoring and Tracking Applications
    - Public health monitoring
    - Consumer sentiment monitoring
    - Electricity consumption tracking
- Analysis and Insight Applications
    - Predictive policing
    - Winning political elections
    - Personal health
- New product development
    - Flexible auto insurance
    - Location-based retail promotion

Q63. Draw the diagram of Generic Big-Data architecture. ***q 7th



Big Data Source layer

- Can be internal or external to the system
- Access could be limited
- Choice of sources depend on the application to analyze data

Data Ingest layer

- Responsible for acquiring data from various data sources

Batch processing layer

- Data is processed using parallel programming techniques, e.g. MapReduce to produce the desired results

Stream processing layer

Data is processed using parallel programming techniques, e.g. MapReduce to process it in real time and understand super-light algorithms

Data organizing layer

- Receives data from both the batch and stream processing layers to organize the data for easy access. NoSQL databases can be used

Infrastructure layer

- Manages the raw resources of storage, computation and communication

Distributed file system layer

- Heart of the system
- Can store huge quantities of data to make it quickly and securely available and accessible to other layers. Hadoop Distributed File System (HDFS) is used in this layer.

Data consumption layer

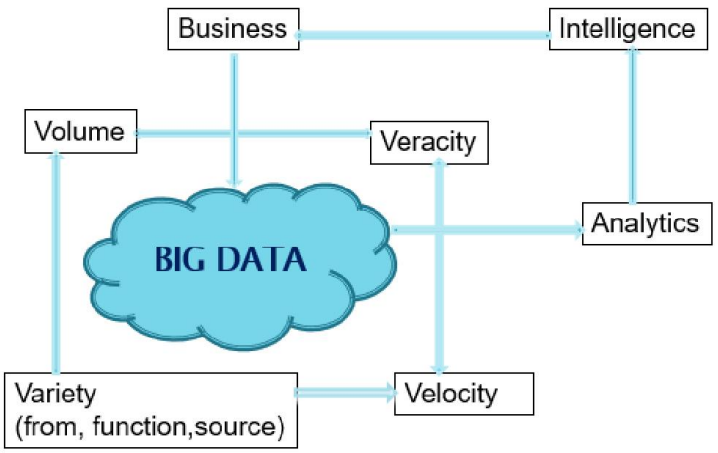
- Consumes the output provided by the analysis layers using reports, dashboards, data analytics etc.

Q64. Differentiate between traditional data and Big-Data. ***q 7th

Feature	Traditional data	Big data
Source	Business transactions, documents	Social media, web access
Volume	Gigabytes, Terabytes	Petabytes, Exabytes
Velocity	Ingest level is controlled	Realtime unpredictable ingest
Variety	Alphanumeric	Audio, Video, Text
Veracity	Clean, more trustworthy	Varies depending on source
Structure of data	Well structured	Semi or unstructured
Data manipulation	Conventional	Parallel

Q65. Draw a diagram on Big-Data Context. ***

- Big data is data that is big in volume, velocity, and variety
- Extremely large, very fast, highly diverse, and complex data that cannot be managed with traditional data management tools. See Figure.



Big Data Context

Q66. Differentiate between clustering and classification. ***q 8th

- **Clustering**: it can be regarded as a form of classification in that it creates a labeling of objects with class (cluster) labels; It derives these labels only from the data
- An entire collection of clusters is commonly referred to as clustering
- **Classification:** Unlabelled objects are assigned a class label using a model developed from objects with known class labels (training data objects) Classification is supervised classification. Clustering is unsupervised classification.

Q67. Explain hierarchical clustering with example. ***q 8th

Hierarchical

Hierarchical clustering is a clustering algorithm that recursively partitions the data into smaller subgroups based on their similarity or distance, to form a hierarchy of clusters. It can be visualized as a dendrogram and is commonly used in exploratory data analysis and pattern discovery.

- A set of nested clusters that are organized as a tree
- Figure 8.1 (a), (b), (c), (d) form a hierarchical clustering with 1, 2, 4, and 6 clusters on each level.

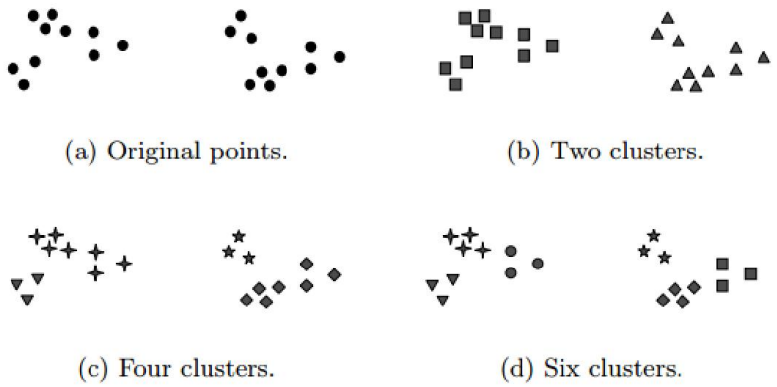


Figure 8.1. Different ways of clustering the same set of points.

Q68. What is the different clustering analysis techniques? ***q 8th

Cluster analysis techniques

- K-means
- Agglomerative Hierarchical Clustering
- DBSCAN

Q69. Define centroid. Describe the basic K-means algorithm for clustering a sample small data set following the steps. ***q 8th

Centroid: It defines a prototype in terms of a centroid- the mean of a group of points.
In K-means clustering, a centroid is a point that represents the center of a cluster. It is calculated as the mean of all the data points that belong to the cluster.

K-means

- It is a prototype-based, partitional clustering technique that attempts to find a user-specified number of clusters (K), which are represented by their centroids.

Basic K-means algorithm

- It defines a prototype in terms of a centroid- the mean of a group of points
- First, choose K initial centroids where k is a user specified parameter- number of clusters desired
- Each point is then assigned to the closest centroid to form a cluster
- The centroid if each cluster is then updated
- Repeat the assignment and update steps until no point changes clusters

Basic K-means algorithm

- 1: Select K points as initial centroids
- 2: repeat
- 3: Form K clusters by assigning each point to its closest centroid
- 4: Recompute the centroid of each cluster
- 5: Until centroids do not change

Q70. Define cluster analysis. What are the different types of clustering? ***q 8th

Cluster Analysis

- It groups data objects based only on information found in the data that describes the objects and their relationships
- The objects within a group be similar to one another and different from the objects in other groups
- The greater the similarity within a group and the greater the differences between groups, the better the clustering.
- It is the study of techniques for automatically finding classes
- It is the study of techniques for finding the most representative cluster prototypes: summarization, compression, efficiently finding nearest neighbors.

Different types of clustering

- Hierarchical versus partitional
- Exclusive versus overlapping versus fuzzy
- Complete versus partial

Q71. Differentiate between hierarchical and partitional clustering with diagram. ***q 8th

Hierarchical versus partitional

- Whether the set of clusters is nested or unnested

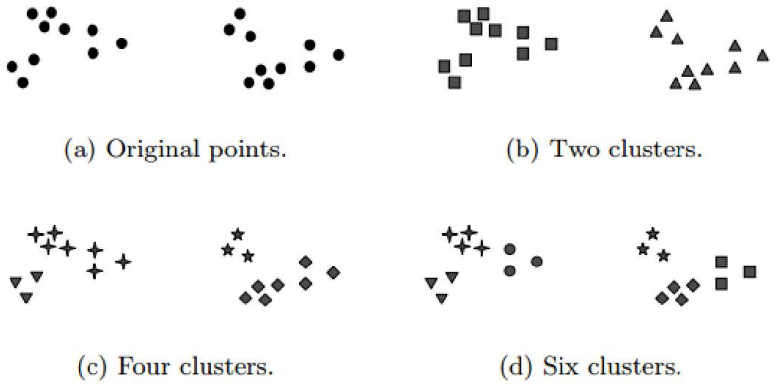


Figure 8.1. Different ways of clustering the same set of points.

Hierarchical

- A set of nested clusters that are organized as a tree
- Figure 8.1 (a), (b), (c), (d) form a hierarchical clustering with 1, 2, 4, and 6 clusters on each level.

Partitional

- It is a division of the set of data objects into non-overlapping subsets (clusters). Each data object is in exactly one subset.
- Individually, each collection of clusters in Figures 8.1 (b)-(d) is a partitional clustering.

Q. Answer the following and explain the reason in favor of your answer. *** q 7th

*** **Sequence / Subsequence:** *** q

1. Initialize a pointer i to the beginning of s.
2. Initialize a pointer j to the beginning of t.
3. While j is not at the end of t:
    - If the character at position i of s is equal to the character at position j of t:
        - Move both pointers forward by one.
    - Otherwise:
        - Move the pointer i forward by one.
4. If j is at the end of t, then t is a subsequence of s. Otherwise, t is not a subsequence of s.

Answer the following and explain the reason in favor of your answer.

Sequence, s	Sequence, *t*	Is *t* a subsequence of *s*?
<{2,4} {3,5,6} {18,25}>	<{2} {3,6} {18}>	Yes
<{1,2,14} {3,4} >	<{1,14} {2} >	No

Q. Answer the following and explain the reason in favor of your answer. *** q 7th

*** **Sequence / Subsequence:** *** q

- A sequence *t* is a subsequence of another sequence *s* if each ordered element in *t* is a subset of an ordered element in *s*.
- The sequence $t = \langle t_1\ t_2\ \ldots\ t_m \rangle$ is a subsequence of $s = \langle s_1\ s_2\ \ldots\ s_n \rangle$ if there exists integers $1 \le j_1 < j_2 < j_m \le n$ such that $t_1 \subseteq s_{j1},\ t_2 \subseteq s_{j2}, \ldots, t_m \subseteq s_{jm}$
- If *t* is a subsequence of *s*, then *t* is contained in *s*

Answer the following and explain the reason in favor of your answer.

Sequence, s	Sequence, *t*	Is *t* a subsequence of *s*?
<{12,14} {13,15,16} {18,25}>	<{12} {13,16} {18}>	Yes
<{11,12,14} {13,24} >	<{11,14} {12} >	No

Q. We have massive data but we are lack of knowledge. What is the solution to this problem *** q CT

When confronted with a situation where you possess massive amounts of data but lack knowledge, data mining can be a valuable solution. Data mining refers to the process of discovering patterns, relationships, and insights from large datasets. Here's how data mining can help address the problem:

1. **Exploratory data analysis:** Data mining techniques allow you to explore and analyze your data comprehensively. You can employ statistical methods, visualization tools, and exploratory techniques to uncover hidden patterns, trends, and correlations within the data. By gaining a deeper understanding of the data through exploration, you can start building knowledge.

2. **Pattern discovery:** Data mining enables you to identify meaningful patterns in your data. Through techniques such as association rule mining, you can discover relationships and dependencies between variables. This can provide insights into customer behavior, product associations, market trends, and more. By extracting patterns, you can derive valuable knowledge from your data.

3. **Classification and prediction:** Data mining algorithms, such as decision trees, support vector machines, and neural networks, can be used for classification and prediction tasks. These algorithms learn from the data to build models that can accurately classify new instances or make predictions about future outcomes. By leveraging these models, you can gain knowledge about the factors influencing certain outcomes or predict future trends based on historical data.

4. **Clustering and segmentation:** Data mining techniques like clustering help group similar data points together based on their characteristics. This can aid in identifying distinct segments within your data, such as customer segments or market clusters. By understanding these segments, you can tailor your strategies, products, or services to specific groups, enhancing your knowledge of customer preferences and needs.

5. **Outlier detection:** Outliers are data points that significantly deviate from the norm. Detecting outliers using data mining methods can provide valuable insights into unusual or unexpected phenomena within your data. These outliers may represent anomalies, errors, or unique patterns that can lead to new knowledge or actionable insights.

6. **Text mining and sentiment analysis:** If your data includes textual information, data mining techniques can be applied to extract knowledge from text. Text mining can help identify sentiment, extract key topics, and analyze textual patterns. By analyzing customer reviews, social media posts, or other textual data, you can gain insights into customer opinions, preferences, and trends.

7. **Knowledge discovery process:** Data mining is part of a broader knowledge discovery process. It involves iteratively applying data mining techniques, interpreting results, and refining your models and hypotheses. Through this iterative process, you continuously enhance your knowledge and refine your understanding of the data.

By employing data mining techniques, you can effectively extract knowledge from your massive datasets. These techniques allow you to explore, discover patterns, predict outcomes, segment data, detect outliers, analyze text, and uncover valuable insights. Data mining serves as a powerful tool to bridge the gap between your data and the knowledge you seek.

Q. Define Clustering. Write down some application of clustering? *** q CT

Clustering:

- Classes, or conceptually meaningful groups of objects that share common characteristics, play an important role in how people analyze and describe the world.
- It is a technique which involves dividing objects into groups

Applications:

- Biology: hierarchical classification of all living beings
- Information retrieval: Clustering can be used to group web search results into a small number of clusters, e.g., movie, trailers, stars etc. categories can be divided into subcategories (sub-clusters).
- Climate
- Psychology and Medicine
- Business

Q. Write down the DMQL expression for specifying task relevant data. *** q CT

- DMQL adopts an SQL-like syntax. In this syntax, [] represents 0 or one occurrences, {} represents 0 or more occurrences, and words in sans serif font represent keywords.

Defining a data mining task

- Specification of the task relevant data, i.e., the data on which the mining is to be performed. This involves:
    - Specifying the database and tables or data warehouse containing the relevant data
    - Conditions for selecting the relevant data
    - The relevant attributes or dimensions for exploration and
    - Instructions for the ordering or grouping of the data retrieved

Example: this example shows how to use DMQL to specify the task-relevant data described in example, for the mining association between items frequently purchased at AllElectronics by Canadian customers, with respect to customer income and age, In addition, the user specifies that she would like the data to be grouped by data. The data are retrieved from a relational database.

Query:

```
Use database AllElectronics_db
In relevance to I.name, I.price, C.income, C.age
From customer C,item I, purchases P, items_sold S
Where I.item_ID = S.item_ID and S.trans_ID = P.trans_ID and P.cust_ID = C.cust_ID
and C.address = 'Canada'
group by P.date
```