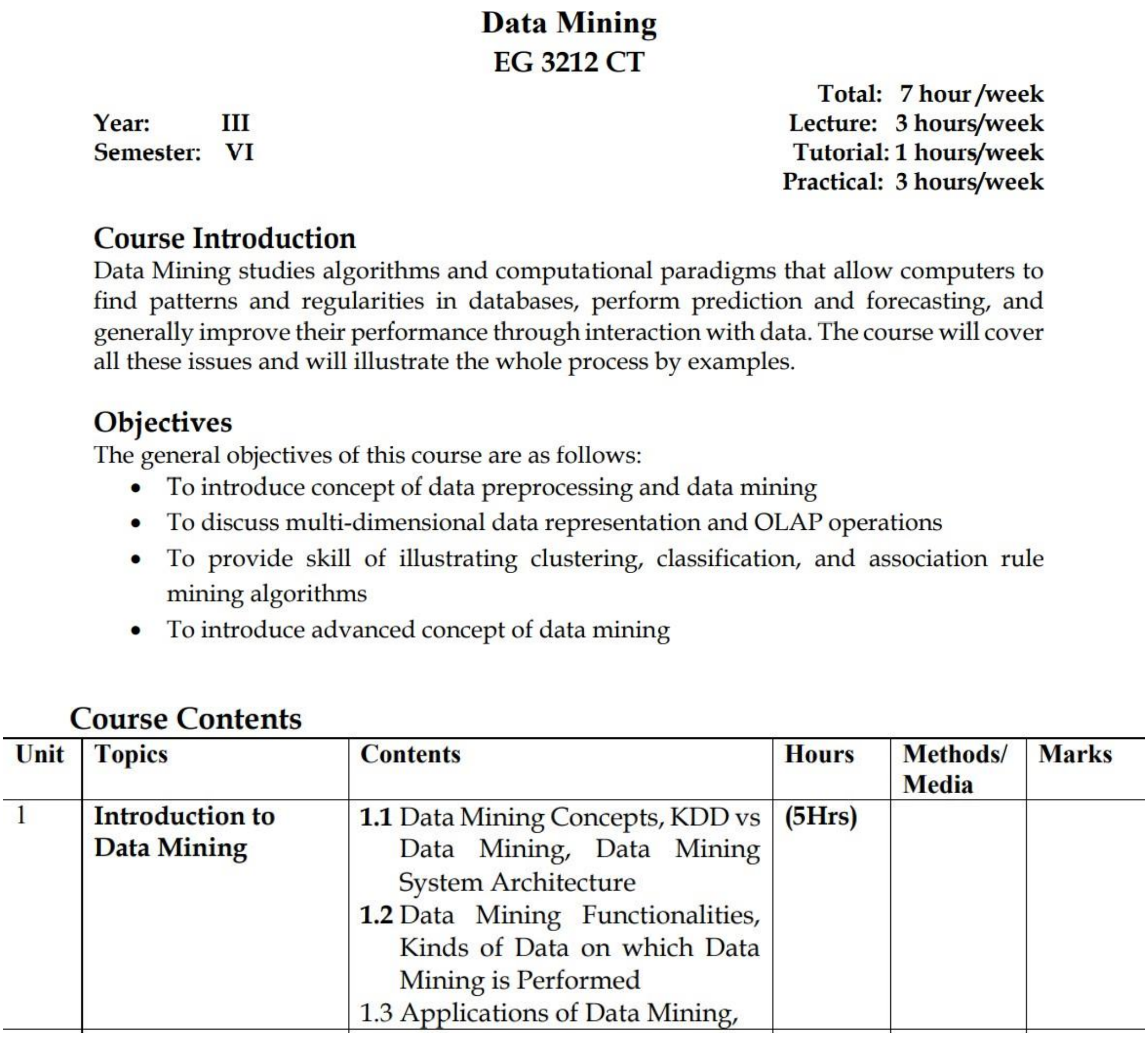


Data Mining
EG 3212 CT

		Total:	7 hour /week
Year:	III	Lecture:	3 hours/week
Semester:	VI	Tutorial:	1 hours/week
		Practical:	3 hours/week

Course Introduction

Data Mining studies algorithms and computational paradigms that allow computers to find patterns and regularities in databases, perform prediction and forecasting, and generally improve their performance through interaction with data. The course will cover all these issues and will illustrate the whole process by examples.

Objectives

The general objectives of this course are as follows:

- To introduce concept of data preprocessing and data mining
- To discuss multi-dimensional data representation and OLAP operations
- To provide skill of illustrating clustering, classification, and association rule mining algorithms
- To introduce advanced concept of data mining

Course Contents

Unit	Topics	Contents	Hours	Methods/ Media	Marks
1	**Introduction to Data Mining**	**1.1** Data Mining Concepts, KDD vs Data Mining, Data Mining System Architecture **1.2** Data Mining Functionalities, Kinds of Data on which Data Mining is Performed 1.3 Applications of Data Mining,	**(5Hrs)**		

Introduction

We live in a world where vast amounts of data are collected daily. Analyzing such data is an important need.

“We are living in the information age” is a popular saying; however, we are actually living in the data age. Terabytes or petabytes1 of data pour into our computer networks, the World Wide Web (WWW), and various data storage devices every day from business, society, science and engineering, medicine, and almost every other aspect of daily life. This explosive growth of available data volume is a result of the computerization of our society and the fast development of powerful data collection and storage tools. Businesses worldwide generate gigantic data sets, including sales transactions, stock trading records, product descriptions, sales promotions, company profiles and performance, and customer feedback. For example, large stores, such as Wal-Mart, handle hundreds of millions of transactions per week at thousands of branches around the world. Scientific and engineering practices generate high orders of petabytes of data in a continuous manner, from remote sensing, process measuring, scientific experiments, system performance, engineering observations, and environment surveillance.

Global backbone telecommunication networks carry tens of petabytes of data traffic every day. The medical and health industry generates tremendous amounts of data from medical records, patient monitoring, and medical imaging. Billions of Web searches supported by search engines process tens of petabytes of data daily. Communities and social media have become increasingly important data sources, producing digital pictures and videos, blogs, Web communities, and various kinds of social networks. The list of sources that generate huge amounts of data is endless. This explosively growing, widely available, and gigantic body of data makes our time truly the data age. Powerful and versatile tools are badly needed to automatically uncover valuable information from the tremendous amounts of data and to transform such data into organized knowledge. This necessity has led to the birth of data mining. The field is young, dynamic, and promising. Data mining has and will continue to make great strides in our journey from the data age toward the coming information age.

Data mining turns a large collection of data into knowledge. A search engine (e.g., Google) receives hundreds of millions of queries every day. Each query can be viewed as a transaction where the user describes her or his information need. What novel and useful knowledge can a search engine learn from such a huge collection of queries collected from users over time? Interestingly, some patterns found in user search queries can disclose invaluable knowledge that cannot be obtained by reading individual data items alone. For example, Google's Flu Trends uses specific search terms as indicators of flu activity. It found a close relationship between the number of people who search for flu-related information and the number of people who actually have flu symptoms. A pattern emerges when all of the search queries related to flu are aggregated. Using aggregated Google search data, Flu Trends can estimate flu activity up to two weeks faster than traditional systems can.2 This example shows how data mining can turn a large collection of data into knowledge that can help meet a current global challenge.

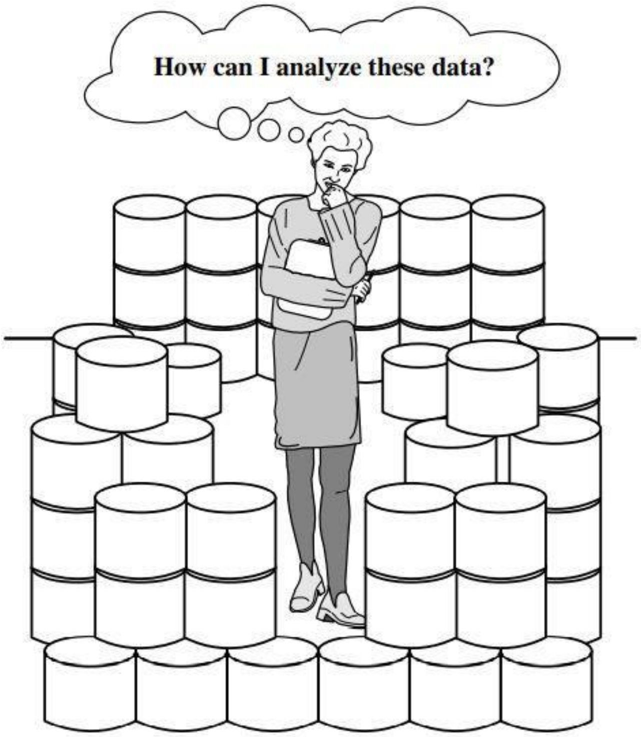


Figure 1.2 The world is data rich but information poor.

What Is Data Mining?

It is no surprise that data mining, as a truly interdisciplinary subject, can be defined in many different ways.
Nevertheless, mining is a vivid term characterizing the process that finds a small set of precious nuggets from a great deal of raw material.

Figure 1.3 Data mining—searching for knowledge (interesting patterns) in data

- *"The process of extracting information to identify patterns, trends, and useful data that would allow the business to take the data-driven decision from huge sets of data is called Data Mining."*
- *"Data mining is the process of analyzing massive volumes of data to discover business intelligence that helps companies solve problems, mitigate risks, and seize new opportunities."*
- *"Data mining is the process of finding anomalies, patterns and correlations within large data sets to predict outcomes. Using a broad range of techniques, you can use this information to increase revenues, cut costs, improve customer relationships, reduce risks and more."*

Data Mining is a process used by organizations to extract specific data from huge databases to solve business problems. It primarily turns raw data into useful information. Data Mining is similar to Data Science carried out by a person, in a specific situation, on a particular data set, with an objective. This process includes various types of services such as text mining, web mining, audio and video mining, pictorial data mining, and social media mining.

KEY TAKEAWAYS

- ✓ Data mining is the process of analyzing a large batch of information to discern trends and patterns.
- ✓ Data mining can be used by corporations for everything from learning about what customers are interested in or want to buy to fraud detection and spam filtering.
- ✓ Data mining programs break down patterns and connections in data based on what information users request or provide.
- ✓ Social media companies use data mining techniques to commodify their users in order to generate profit.
- ✓ This use of data mining has come under criticism lately s users are often unaware of the data mining happening with their personal information, especially when it is used to influence preferences.

What kinds of patterns can be mined?

There are a number of **data mining functionalities**. **These include characterization and discrimination** the mining of frequent patterns, associations, and correlations classification and regression; clustering analysis; and outlier analysis.

Data mining functionalities are used to specify the kinds of patterns to be found in data mining tasks. In general, such tasks can be classified into two categories: **descriptive and predictive.**

Descriptive mining tasks characterize properties of the data in a target data set.

Predictive mining tasks perform induction on the current data in order to make predictions

The output of **data characterization** can be presented in various forms. Examples include pie charts, bar charts, curves, multidimensional data cubes, and multidimensional tables, including crosstabs. The resulting descriptions can also be presented as generalized relations or in rule form (called characteristic rules).

Data discrimination is a comparison of the general features of the target class data objects against the general features of objects from one or multiple contrasting classes. The target and contrasting classes can be specified by a user, and the corresponding data objects can be retrieved through database queries. The methods used for data discrimination are similar to those used for data characterization.

KDD vs. Data Mining

KDD is a field of computer science, which deals with extraction of previously unknown and interesting information from raw data. KDD is the whole process of trying to make sense of data by developing appropriate methods or techniques. This process deals with the mapping of low-level data into other forms those are more compact, abstract and useful. This is achieved by creating short reports, modeling the process of generating data and developing predictive models that can predict future cases. Due to the exponential growth of data, especially in areas such as business, KDD has become a very important process to convert this large wealth of data in to business intelligence, as manual extraction of patterns has become seemingly impossible in the past few decades.

Data Mining is only a step within the overall KDD process. There are two major Data Mining goals as defined by the goal of the application, and they are namely **verification or discovery**. Verification is verifying the user's hypothesis about data, while discovery is automatically finding interesting patterns.

There are four major data mining tasks: **clustering, classification, regression, and association** (summarization).

Clustering is identifying similar groups from unstructured data.

Classification is learning rules that can be applied to new data.

Regression is finding functions with minimal error to model data.

Association is looking for relationships between variables. Then, the specific data mining algorithm needs to be selected. Depending on the goal, different algorithms like linear regression, logistic regression, decision trees and Naïve Bayes can be selected.

Many people treat data mining as a synonym for another popularly used term, knowledge discovery from data, or KDD, while others view data mining as merely an essential step in the process of knowledge discovery.

The knowledge discovery process is shown in Figure 1.4 as an iterative sequence of the following steps:

1. **Data cleaning** (to remove noise and inconsistent data)
2. **Data integration** (where multiple data sources may be combined)
3. **Data selection** (where data relevant to the analysis task are retrieved from the database)
4. **Data transformation** (where data are transformed and consolidated into forms appropriate for mining by performing summary or aggregation operations)
5. **Data mining** (an essential process where intelligent methods are applied to extract data patterns)
6. **Pattern evaluation** (to identify the truly interesting patterns representing knowledge based on interestingness measures)
7. **Knowledge presentation** (where visualization and knowledge representation techniques are used to present mined knowledge to users)

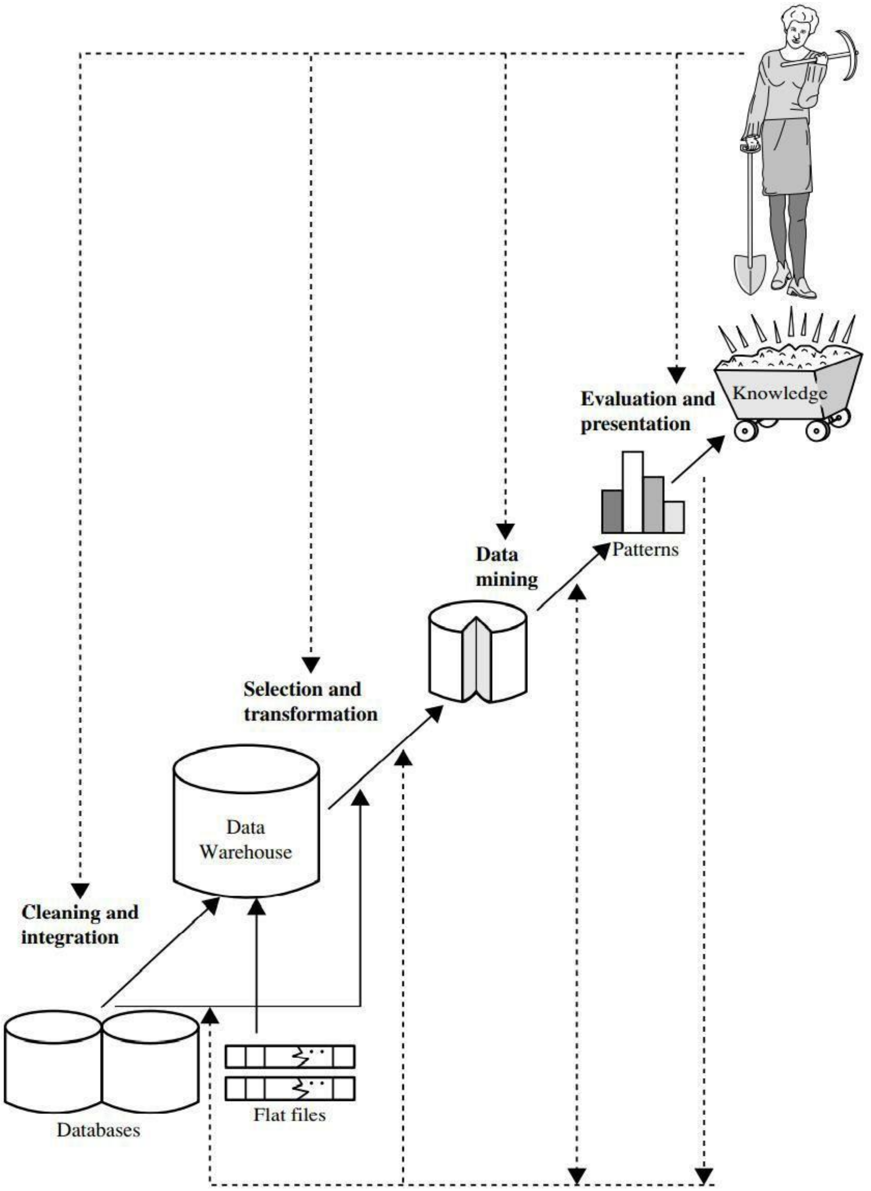


Figure 1.4 Data mining as a step in the process of knowledge discovery.

1. ***Data Cleaning***: Data cleaning is defined as removal of noisy and irrelevant data from collection.
    - Cleaning in case of ***Missing values***.
    - Cleaning ***noisy*** data, where noise is a random or variance error.
    - Cleaning with ***Data discrepancy detection*** and ***Data transformation tools***.
2. ***Data Integration***: Data integration is defined as heterogeneous data from multiple sources combined in a common source (Datawarehouse).
    - Data integration using ***Data Migration tools***.
    - Data integration using ***Data Synchronization tools***.
    - Data integration using ***ETL***(Extract-Load-Transformation) process.
3. ***Data Selection***: Data selection is defined as the process where data relevant to the analysis is decided and retrieved from the data collection.
    - Data selection using ***Neural network***.
    - Data selection using ***Decision Trees***.
    - Data selection using ***Naive bayes***.
    - Data selection using ***Clustering***, ***Regression***, etc.
4. ***Data Transformation***: Data Transformation is defined as the process of transforming data into appropriate form required by mining procedure.
   Data Transformation is a twostep process:

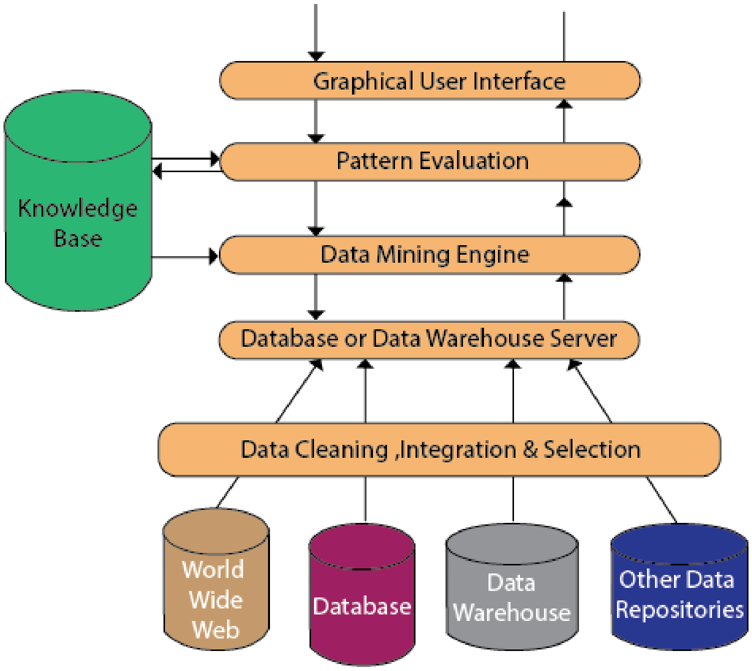
    - ***Data Mapping***: Assigning elements from source base to destination to capture transformations.
    - ***Code generation***: Creation of the actual transformation program.
5. ***Data Mining***: Data mining is defined as clever techniques that are applied to extract patterns potentially useful.
    - Transforms task relevant data into ***patterns***.
    - Decides purpose of model using ***classification*** or ***characterization***.
6. ***Pattern Evaluation***: Pattern Evaluation is defined as identifying strictly increasing patterns representing knowledge based on given measures.
    - Find ***interestingness score*** of each pattern.
    - Uses ***summarization*** and ***Visualization*** to make data understandable by user.
7. ***Knowledge representation***: Knowledge representation is defined as technique which utilizes visualization tools to represent data mining results.
    - Generate ***reports***.
    - Generate ***tables***.
    - Generate ***discriminant rules***, ***classification rules***, ***characterization rules***, etc.

Data Mining System Architecture

The significant components of data mining systems are a data source, data mining engine, data warehouse server, the pattern evaluation module, graphical user interface, and knowledge base.

Data Source

The actual source of data is the Database, data warehouse, World Wide Web (WWW), text files, and other documents. You need a huge amount of historical data for data mining to be successful. Organizations typically store data in databases or data warehouses.



Database or Data Warehouse Server
The database or data warehouse server consists of the original data that is ready to be processed. Hence, the server is cause for retrieving the relevant data that is based on data mining as per user request.

Data Mining Engine
The data mining engine is a major component of any data mining system. It contains several modules for operating data mining tasks, including association, characterization, classification, clustering, prediction, time-series analysis, etc.
In other words, we can say data mining is the root of our data mining architecture. It comprises instruments and software used to obtain insights and knowledge from data collected from various data sources and stored within the data warehouse.

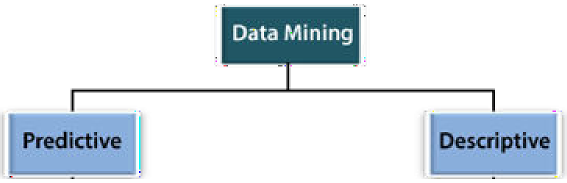
Pattern Evaluation Module
The Pattern evaluation module is primarily responsible for the measure of investigation of the pattern by using a threshold value. It collaborates with the data mining engine to focus the search on exciting patterns.

Graphical User Interface
The graphical user interface (GUI) module communicates between the data mining system and the user. This module helps the user to easily and efficiently use the system without knowing the complexity of the process. This module cooperates with the data mining system when the user specifies a query or a task and displays the results.

Knowledge Base
The knowledge base is helpful in the entire process of data mining. It might be helpful to guide the search or evaluate the stake of the result patterns. The knowledge base may even contain user views and data from user experiences that might be helpful in the data mining process.

Data Mining Functionalities

Data Mining functions are used to define the trends or correlations contained in data mining activities. In comparison, data mining activities can be divided into 2 categories:



1. Descriptive Data Mining

It includes certain knowledge to understand what is happening within the data without a previous idea. The common data features are highlighted in the data set.
For examples: count, average etc.

Given below are functions listed in this kind of Data Mining:

1. Class or Concept Description
2. Mining of Frequent Patterns
3. Mining of Associations
4. Mining of Correlations
5. Mining of Clusters

Class or Concept Description

Class or Concept refers to the data that is linked or correlated with some classes or some concepts. For instance, let's say there is a company and in that company, the classes of things for sales include mugs and glasses, and concepts of customers include big spenders and budget spenders. The types of descriptions of a class or a concept are known as class or concept descriptions. These descriptions can be acquired with the help of the ways listed below which are:

Data Characterization: Data Characterization is a method which sums up a dataset of class under study. This class which is under study is known as Target Class.
Data Discrimination: It means to classify a class with the help of some predefined group or class.

Mining of Frequent Patterns

The second function in Descriptive Data Mining is the "Frequent patterns". They can be defined as the patterns that takes place very often in transactional data, which are:

Frequent Item Set: As the name suggest, the meaning of Frequent Item set is a set of items that are often appeared together. For instance, shoes and socks.
Frequent Subsequence: Much similar to the above point, a sequence of patterns that takes place very often like putting on socks and then shoes is called Frequent Subsequence.
Frequent Sub Structure: Substructure is called different structural forms, for instances graphs, charts, etc. combined with subsequences.

Mining of Association

Mining of Associations are mainly used in retail sales in order to identify patterns that are very often purchased together. The process of Mining of Association can be defined as the process of revealing the relationship among the set of data and finding out association rules.

Mining of Correlations

Mining of Correlations refers to a type of Descriptive Data Mining's Functions that are usually executed in order to reveal or expose some statistical correlations between associated attribute value pairs or between two item sets. This is helpful to analyze that whether they are having positive, negative or no effect on each other.

Mining of Clusters

The literal meaning of the word "Cluster" is a group of things which are similar to one another in some way or another. Now coming to the term "Cluster analysis", it means to form group of things that are almost alike each other but at the same time, they are very different from the things that are in other clusters.

2. **Predictive Data Mining**
   It helps developers to provide unlabeled definitions of attributes. Based on previous tests, the software estimates the characteristics that are absent.
   For example: Judging from the findings of a patient's medical examinations that is he suffering from any particular disease.

The model derived in the process are represented in various formations, the formations are listed below:

1. **Decision Tree**
2. **Neutral Network**

Decision Tree

A decisiontree is a like a flow chart with a tree structure, in which every junction/node is used to represent a test on an attribute value, moreover, each and every branch is responsible for representing the concluding outcome of the test, and tree leaves are used to represent the classes or the distribution of classes.

Neural Network

A neural network is mainly used for classification can be defined as a collection of processing units with connections between the units. In other and simpler words, Neural networks searches for patterns or trends in large quantity of different sets of data, which allows organizations to understand more and better about their clients or users need which is directly responsible for rendering their marketing strategies, increase sales and lowers costs.

Kinds of Data on which Data mining is performed

Data mining can be performed on the following types of data:

1. **Relational Database**
   A relational database is a collection of multiple data sets formally organized by tables, records, and columns from which data can be accessed in various ways without having to recognize the

database tables. Tables convey and share information, which facilitates data searchability, reporting, and organization.

2. **Data Warehouse**
A Data Warehouse is the technology that collects the data from various sources within the organization to provide meaningful business insights. The huge amount of data comes from multiple places such as Marketing and Finance. The extracted data is utilized for analytical purposes and helps in decision- making for a business organization. The data warehouse is designed for the analysis of data rather than transaction processing.

3. **Data Repositories**
The Data Repository generally refers to a destination for data storage. However, many IT professionals utilize the term more clearly to refer to a specific kind of setup within an IT structure. For example, a group of databases, where an organization has kept various kinds of information.

4. **Object-Relational Database**
A combination of an object-oriented database model and relational database model is called an object-relational model. It supports Classes, Objects, Inheritance, etc. One of the primary objectives of the Object-relational data model is to close the gap between the Relational database and the object-oriented model practices frequently utilized in many programming languages, for example, C++, Java, C#, and so on.

5. **Transactional Database**
A transactional database refers to a database management system (DBMS) that has the potential to undo a database transaction if it is not performed appropriately. Even though this was a unique capability a very long while back, today, most of the relational database systems support transactional database activities.

Advantages of Data Mining

- ✓ The Data Mining technique enables organizations to obtain knowledge-based data.
- ✓ Data mining enables organizations to make lucrative modifications in operation and production.
- ✓ Compared with other statistical data applications, data mining is a cost-efficient.
- ✓ Data Mining helps the decision-making process of an organization.
- ✓ It Facilitates the automated discovery of hidden patterns as well as the prediction of trends and behaviors.
- ✓ It can be induced in the new system as well as the existing platforms.
- ✓ It is a quick process that makes it easy for new users to analyze enormous amounts of data in a short time.

Disadvantages of Data Mining

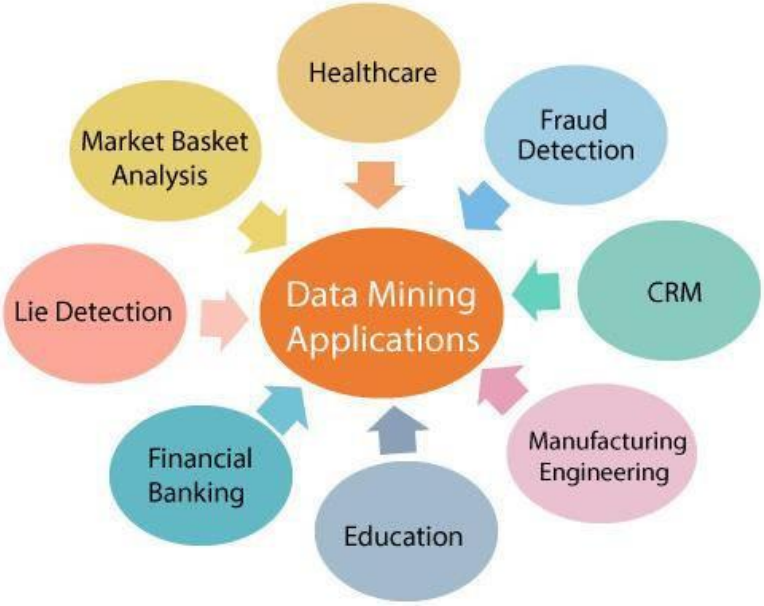
- ✓ There is a probability that the organizations may sell useful data of customers to other organizations for money. As per the report, American Express has sold credit card purchases of their customers to other organizations.
- ✓ Many data mining analytics software is difficult to operate and needs advance training to work on.

- ✓ Different data mining instruments operate in distinct ways due to the different algorithms used in their design. Therefore, the selection of the right data mining tools is a very challenging task.
- ✓ The data mining techniques are not precise, so that it may lead to severe consequences in certain conditions.

Applications of Data Mining

Data Mining is primarily used by organizations with intense consumer demands- Retail, Communication, Financial, marketing company, determine price, consumer preferences, product positioning, and impact on sales, customer satisfaction, and corporate profits. Data mining enables a retailer to use point-of-sale records of customer purchases to develop products and promotions that help the organization to attract the customer.

These are the following areas where data mining is widely used:



- ➢ **Data Mining in Healthcare:**

Data mining in healthcare has excellent potential to improve the health system. It uses data and analytics for better insights and to identify best practices that will enhance health care services and reduce costs. Analysts use data mining approaches such as Machine learning, multi-dimensional database, Data visualization, soft computing, and statistics. Data Mining can be used to forecast patients in each category. The procedures ensure that the patients get intensive care at the right place and at the right time. Data mining also enables healthcare insurers to recognize fraud and abuse.

- ➢ **Data Mining in Market Basket Analysis:**

Market basket analysis is a modeling method based on a hypothesis. If you buy a specific group of products, then you are more likely to buy another group of products. This technique may enable the retailer to understand the purchase behavior of a buyer. This data may assist the retailer in understanding the requirements of the buyer and altering the store's layout accordingly. Using a different analytical comparison of results between various stores, between customers in different demographic groups can be done.

- **Data mining in Education**

Education data mining is a newly emerging field, concerned with developing techniques that explore knowledge from the data generated from educational Environments. EDM objectives are recognized as affirming student's future learning behavior, studying the impact of educational support, and promoting learning science. An organization can use data mining to make precise decisions and also to predict the results of the student. With the results, the institution can concentrate on what to teach and how to teach.

- **Data Mining in Manufacturing Engineering:**

Knowledge is the best asset possessed by a manufacturing company. Data mining tools can be beneficial to find patterns in a complex manufacturing process. Data mining can be used in system-level designing to obtain the relationships between product architecture, product portfolio, and data needs of the customers. It can also be used to forecast the product development period, cost, and expectations among the other tasks.

- **Data Mining in CRM (Customer Relationship Management):**

Customer Relationship Management (CRM) is all about obtaining and holding Customers, also enhancing customer loyalty and implementing customer-oriented strategies. To get a decent relationship with the customer, a business organization needs to collect data and analyze the data. With data mining technologies, the collected data can be used for analytics.

- **Data Mining in Fraud detection:**

Billions of dollars are lost to the action of frauds. Traditional methods of fraud detection are a little bit time consuming and sophisticated. Data mining provides meaningful patterns and turning data into information. An ideal fraud detection system should protect the data of all the users. Supervised methods consist of a collection of sample records, and these records are classified as fraudulent or non-fraudulent. A model is constructed using this data, and the technique is made to identify whether the document is fraudulent or not.

- **Data Mining in Lie Detection:**

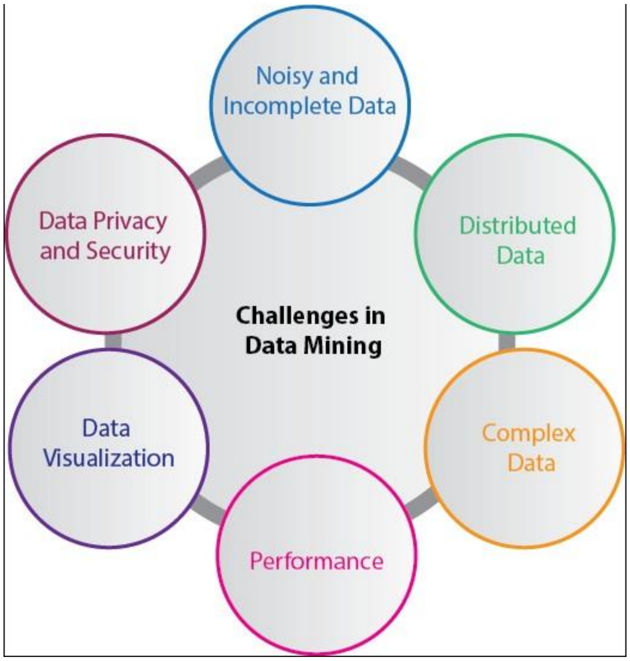
Apprehending a criminal is not a big deal but bringing out the truth from him is a very challenging task. Law enforcement may use data mining techniques to investigate offenses, monitor suspected terrorist communications, etc. This technique includes text mining also, and it seeks meaningful patterns in data, which is usually unstructured text. The information collected from the previous investigations is compared, and a model for lie detection is constructed.

- **Data Mining Financial Banking:**

The Digitalization of the banking system is supposed to generate an enormous amount of data with every new transaction. The data mining technique can help bankers by solving business-related problems in banking and finance by identifying trends, casualties, and correlations in business information and market costs that are not instantly evident to managers or executives because the data volume is too large or are produced too rapidly on the screen by experts. The manager may find these data for better targeting, acquiring, retaining, segmenting, and maintain a profitable customer.

Challenges of Implementation in Data mining

Although data mining is very powerful, it faces many challenges during its execution. Various challenges could be related to performance, data, methods, and techniques, etc. The process of data mining becomes effective when the challenges or problems are correctly recognized and adequately resolved.


Noisy and Incomplete Data
Distributed Data
Complex Data
Performance
Data Visualization
Data Privacy and Security
Challenges in Data Mining

Incomplete and noisy data

The data in the real-world is heterogeneous, incomplete, and noisy. Data in huge quantities will usually be inaccurate or unreliable. These problems may occur due to data measuring instrument or because of human errors.

Data Distribution

For example, various regional offices may have their servers to store their data. It is not feasible to store, all the data from all the offices on a central server. Therefore, data mining requires the development of tools and algorithms that allow the mining of distributed data.

Complex Data

Real-world data is heterogeneous, and it could be multimedia data, including audio and video, images, complex data, spatial data, time series, and so on. Managing these various types of data and extracting useful information is a tough task. Most of the time, new technologies, new tools, and methodologies would have to be refined to obtain specific information.

Performance

The data mining system's performance relies primarily on the efficiency of algorithms and techniques used. If the designed algorithm and techniques are not up to the mark, then the efficiency of the data mining process will be affected adversely.

Data Privacy and Security

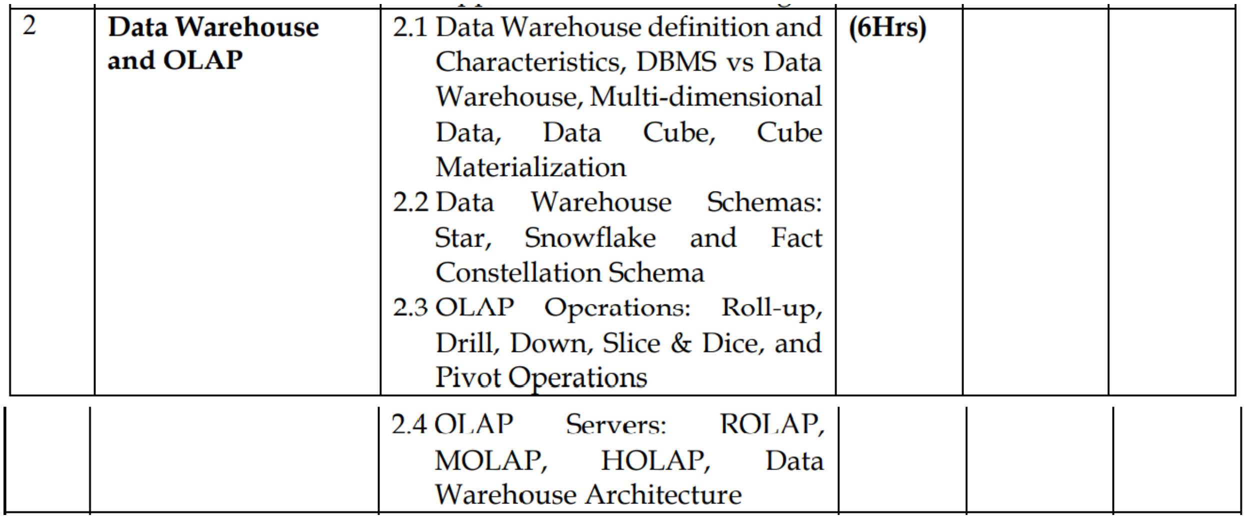
Data mining usually leads to serious issues in terms of data security, governance, and privacy. For example, if a retailer analyzes the details of the purchased items, then it reveals data about buying habits and preferences of the customers without their permission.

Data Visualization

The extracted data should convey the exact meaning of what it intends to express. But many times, representing the information to the end-user in a precise and easy way is difficult. The input data and the output information being complicated, very efficient, and successful data visualization processes need to be implemented to make it successful.

There are many more challenges in data mining in addition to the problems above-mentioned. More problems are disclosed as the actual data mining process begins, and the success of data mining relies on getting rid of all these difficulties.

Unit 2



2	**Data Warehouse and OLAP**	2.1 Data Warehouse definition and Characteristics, DBMS vs Data Warehouse, Multi-dimensional Data, Data Cube, Cube Materialization 2.2 Data Warehouse Schemas: Star, Snowflake and Fact Constellation Schema 2.3 OLAP Operations: Roll-up, Drill, Down, Slice & Dice, and Pivot Operations	**(6Hrs)**		
		2.4 OLAP Servers: ROLAP, MOLAP, HOLAP, Data Warehouse Architecture			

Data Warehouse

A data warehouse is a repository of information collected from multiple sources, stored under a unified schema, and usually residing at a single site. Data warehouses are constructed via a process of data cleaning, data integration, data transformation, data loading, and periodic data refreshing.

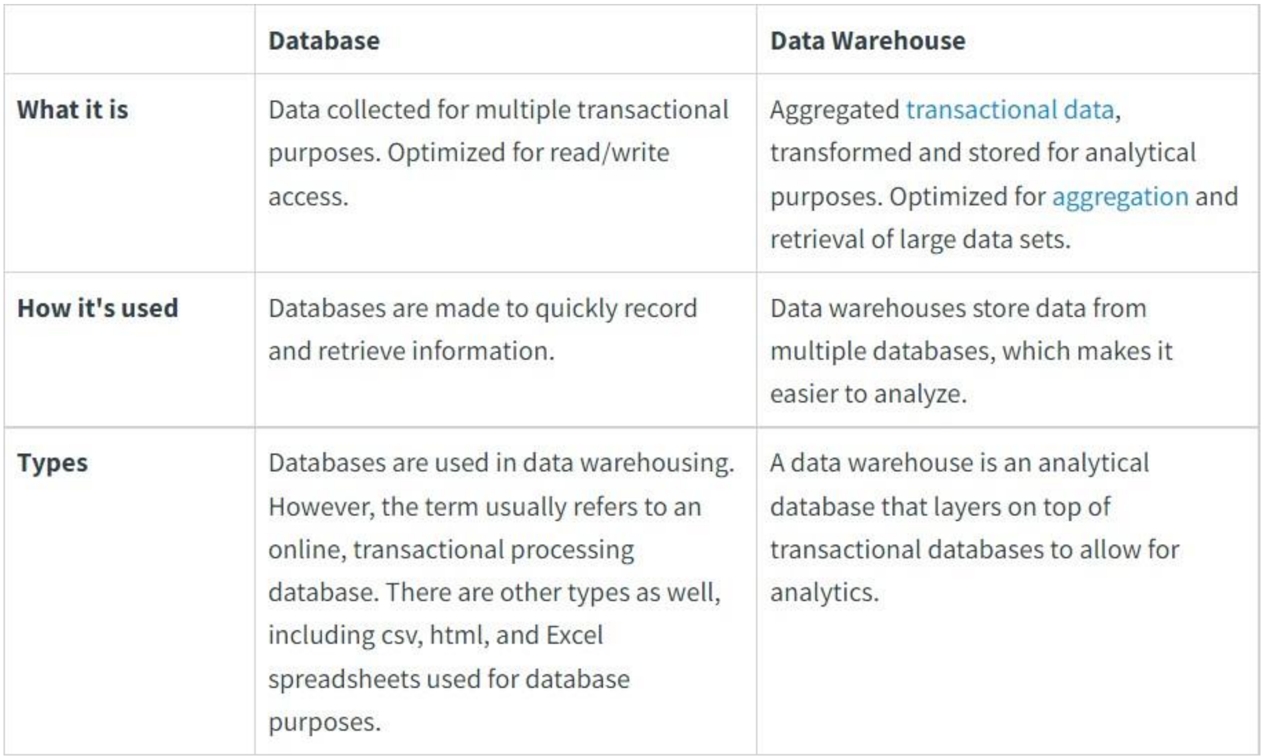
A Data Warehouse works as a central repository where information arrives from one or more data sources. Data flows into a data warehouse from the transactional system and other relational databases. Data may be:

- Structured
- Semi-structured
- Unstructured data

Today, businesses can invest in cloud-based data warehouse software services from companies including Microsoft, Google, Amazon, and Oracle, among others.

1. It is not database
2. It is not data lake
3. It is not data mart

A data warehouse is not the same as a database: For example, a database might only have the most recent address of a customer, while a data warehouse might have all the addresses for the customer for the past 10 years.



	Database	Data Warehouse
What it is	Data collected for multiple transactional purposes. Optimized for read/write access.	Aggregated transactional data, transformed and stored for analytical purposes. Optimized for aggregation and retrieval of large data sets.
How it's used	Databases are made to quickly record and retrieve information.	Data warehouses store data from multiple databases, which makes it easier to analyze.
Types	Databases are used in data warehousing. However, the term usually refers to an online, transactional processing database. There are other types as well, including csv, html, and Excel spreadsheets used for database purposes.	A data warehouse is an analytical database that layers on top of transactional databases to allow for analytics.

Types of Data Warehouse

1. Enterprise Data Warehouse (EDW)
2. Operational data store (ODS)
3. Data mart

Multidimensional Data Model

When data is grouped or combined in multidimensional matrices called Data Cubes. The data cube method has a few alternative names or a few variants, such as "Multidimensional databases," "materialized views," and "OLAP (On-Line Analytical Processing)

A data cube is a multidimensional data model that store the optimized, summarized or aggregated data which eases the OLAP tools for the quick and easy analysis. Data cube stores the precomputed data and eases online analytical processing. A data cube provides a multidimensional view of data and allows the precomputation and fast access of summarized data.

A data cube allows data to be viewed in multiple dimensions. Dimensions are entities with respect to which an organization wants to keep records. For example, in store sales record, dimensions allow the store to keep track of things like monthly sales of items and the branches and locations.

A multidimensional database helps to provide data-related answers to complex business queries quickly and accurately.

Data warehouses and Online Analytical Processing (OLAP) tools are based on a multidimensional data model. LAP in data warehousing enables users to view data from different angles and dimensions.
By providing multidimensional data views and the precomputation of summarized data, data warehouse systems can provide inherent support for OLAP. Online analytical processing operations make use of background knowledge regarding the domain of the data being studied to allow the presentation of data at different levels of abstraction.

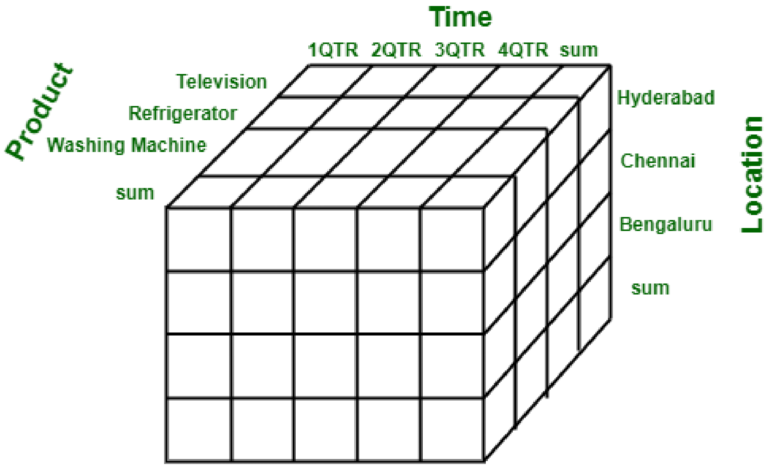


Fig: Multidimensional Data Representation

Advantages of Multi-Dimensional Data Model

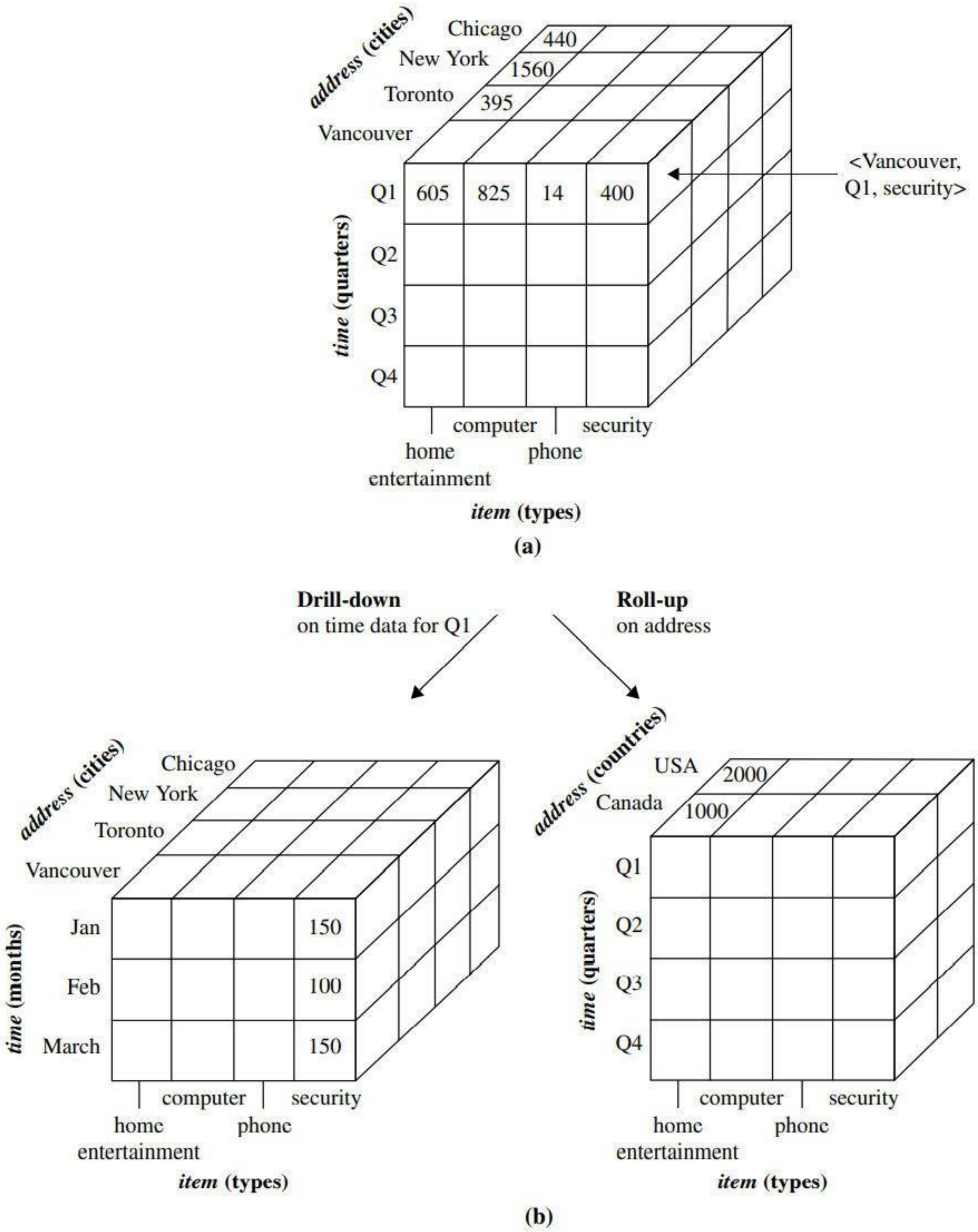
The following are the advantages of a multi-dimensional data model:

- ✓ A multi-dimensional data model is easy to handle.
- ✓ It is easy to maintain.
- ✓ Its performance is better than that of normal databases (e.g., relational databases).
- ✓ The representation of data is better than traditional databases. That is because the multi-dimensional databases are multi-viewed and carry different types of factors.
- ✓ It is workable on complex systems and applications, contrary to the simple one-dimensional database systems.
- ✓ The compatibility in this type of database is an upliftment for projects having lower bandwidth for maintenance staff.

Disadvantages of Multi-Dimensional Data Model

The following are the disadvantages of a Multi-Dimensional Data Model:

- ✓ The multi-dimensional Data Model is slightly complicated in nature, and it requires professionals to recognize and examine the data in the database.
- ✓ During the work of a Multi-Dimensional Data Model, when the system caches, there is a great effect on the working of the system.
- ✓ It is complicated in nature due to which the databases are generally dynamic in design.
- ✓ The path to achieving the product is complicated most of the time.
- ✓ As the Multi-Dimensional Data Model has complicated systems, databases have a large number of databases due to which the system is very insecure when there is a security break.

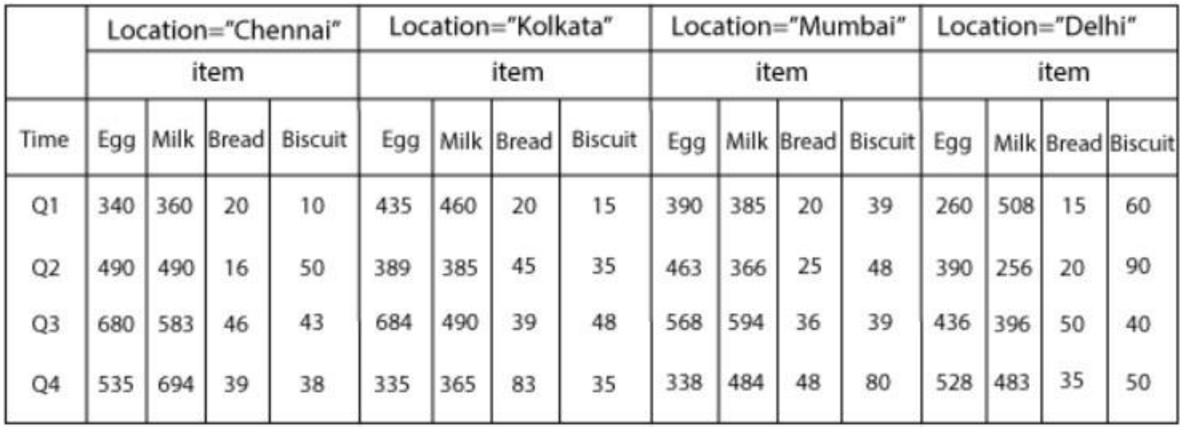


A multidimensional data cube, commonly used for data warehousing,

(a) showing summarized data

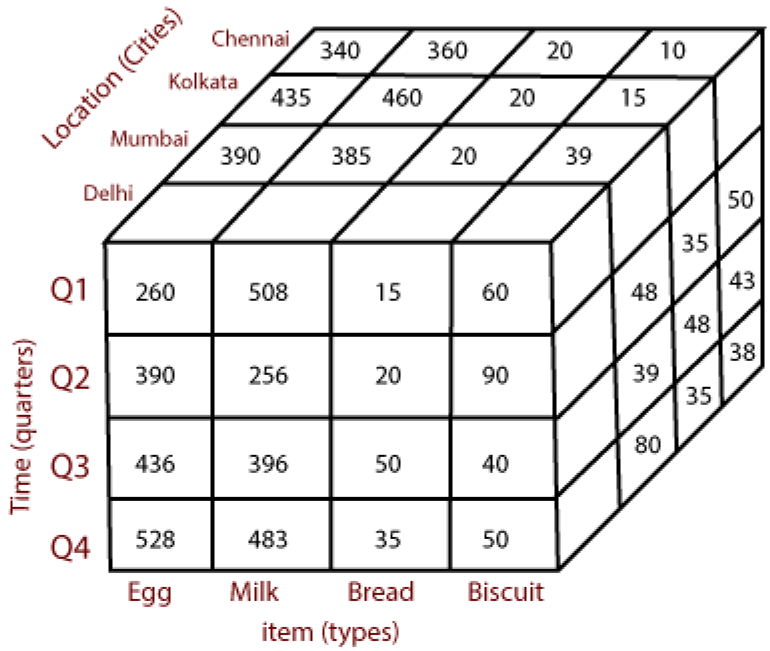
(b) showing summarized data resulting from drill-down and roll-up operations on the cube in (a).

For example, suppose the data according to time and item, as well as the location is considered for the cities Chennai, Kolkata, Mumbai, and Delhi. These 3D data are shown in the table. The 3D data of the table are represented as a series of 2D tables.



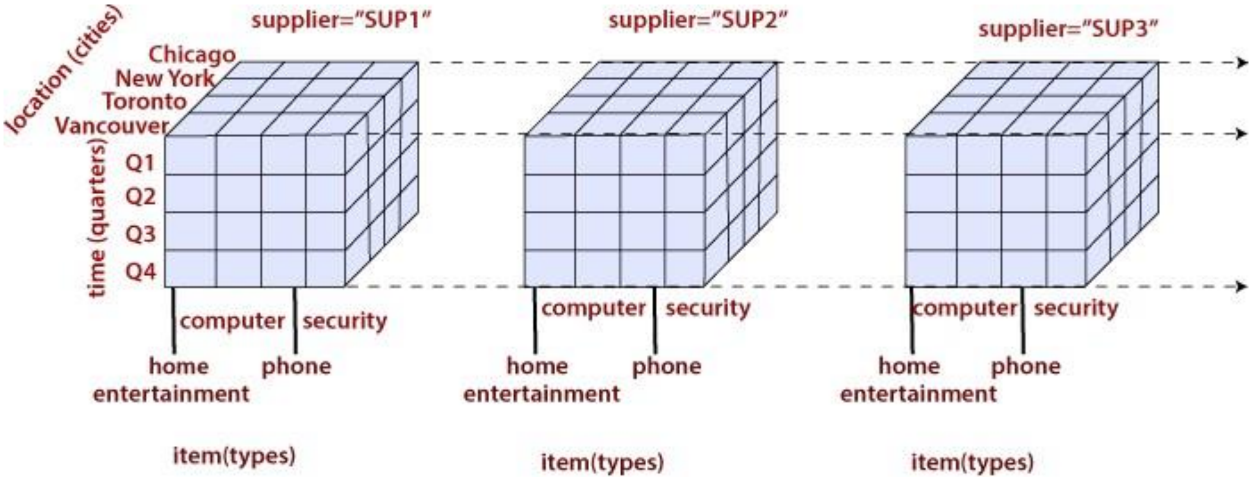
	Location="Chennai"				Location="Kolkata"				Location="Mumbai"				Location="Delhi"			
	item				item				item				item			
Time	Egg	Milk	Bread	Biscuit	Egg	Milk	Bread	Biscuit	Egg	Milk	Bread	Biscuit	Egg	Milk	Bread	Biscuit
Q1	340	360	20	10	435	460	20	15	390	385	20	39	260	508	15	60
Q2	490	490	16	50	389	385	45	35	463	366	25	48	390	256	20	90
Q3	680	583	46	43	684	490	39	48	568	594	36	39	436	396	50	40
Q4	535	694	39	38	335	365	83	35	338	484	48	80	528	483	35	50

Conceptually, it may also be represented by the same data in the form of a 3D data cube, as shown in fig:



Let us suppose that we would like to view our sales data with an additional fourth dimension, such as a supplier.

- ✓ In data warehousing, the data cubes are **n dimensional.** The **cuboid** which holds the lowest level of summarization is called a base cuboid.
- ✓ For example, the 4-D cuboid in the figure is the base cuboid for the given time, item, location, and supplier dimensions.
- ✓ *The topmost 0-D cuboid, which holds the highest level of summarization, is known as the **apex cuboid.***
- ✓ The lattice of cuboid forms a data cube.

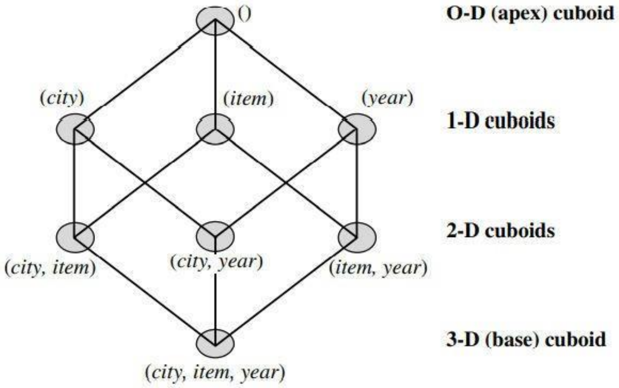


Cube Materialization

A data cube is a lattice of cuboids. Taking the three attributes, city, item, and year, as the dimensions for the data cube, and sales in dollars as the measure, the total number of cuboids, or group by's, that can be computed for this data cube is $2^3 = 8$. The possible group-by's are the following: {(city, item, year), (city, item), (city, year), (item, year), (city), (item), (year), ()}, where () means that the group-by is empty (i.e., the dimensions are not grouped). These group-by's form a lattice of cuboids for the data cube, as shown in Figure.

Another aspect of data warehouses that is worth mentioning briefly is *materialized aggregates.* As discussed earlier, data warehouse queries often involve an aggregate function, such as COUNT, SUM, AVG, MIN, or MAX in SQL. If the same aggregates are used by many different queries, it can be wasteful to crunch through the raw data every time. Why not cache some of the counts or sums that queries use most often?

One way of creating such a cache is a ***materialized view***.



There are three choices for data cube materialization given a base cuboid:

a. **No materialization:** Do not precompute any of the "nonbase" cuboids.
b. **Full materialization:** Precompute all of the cuboids. The resulting lattice of computed cuboids is referred to as the full cube. This choice typically requires huge amounts of memory space in order to store all the precomputed cuboids.
c. **Partial materialization:** Selectively compute a proper subset of the whole set of possible cuboids.

Data Warehouse Schemas

A schema is defined as a logical description of database where fact and dimension tables are joined in a logical manner. Data Warehouse is maintained in the form of Star, Snowflakes, and Fact Constellation schema.
In the data warehouse there includes the name and description of records. It has all data items and also different aggregates associated with the data. Like a database has a schema, it is required to maintain a schema for a data warehouse as well.

1. Star Schema

This is the simplest and most effective schema in a data warehouse. A fact table in the center surrounded by multiple dimension tables resembles a star in the Star Schema model.
An example of a Star Schema is given below.

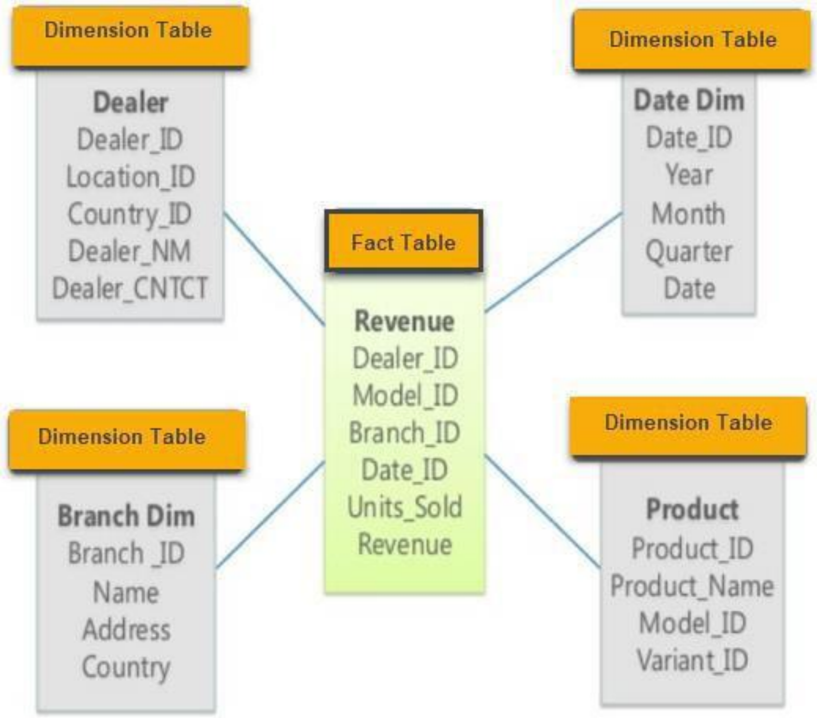


Fig: Example of Star Schema Diagram

Characteristics

- In a Star schema, there is only one fact table and multiple dimension tables.
- In a Star schema, each dimension is represented by one-dimension table.
- Dimension tables are not normalized in a Star schema.
- Each Dimension table is joined to a key in a fact table.

Disadvantages Of Star Schema

- If there are many changes in the requirements, the existing star schema is not recommended to modify and reuse in the long run.
- Data redundancy is more as tables are not hierarchically divided.

2. Snowflakes

Snowflake Schema in data warehouse is a logical arrangement of tables in a multidimensional database such that the ER diagram resembles a snowflake shape. A Snowflake Schema is an extension of a Star Schema, and it adds additional dimensions. The dimension tables are normalized which splits data into additional tables.

In the following Snowflake Schema example, Country is further normalized into an individual table.

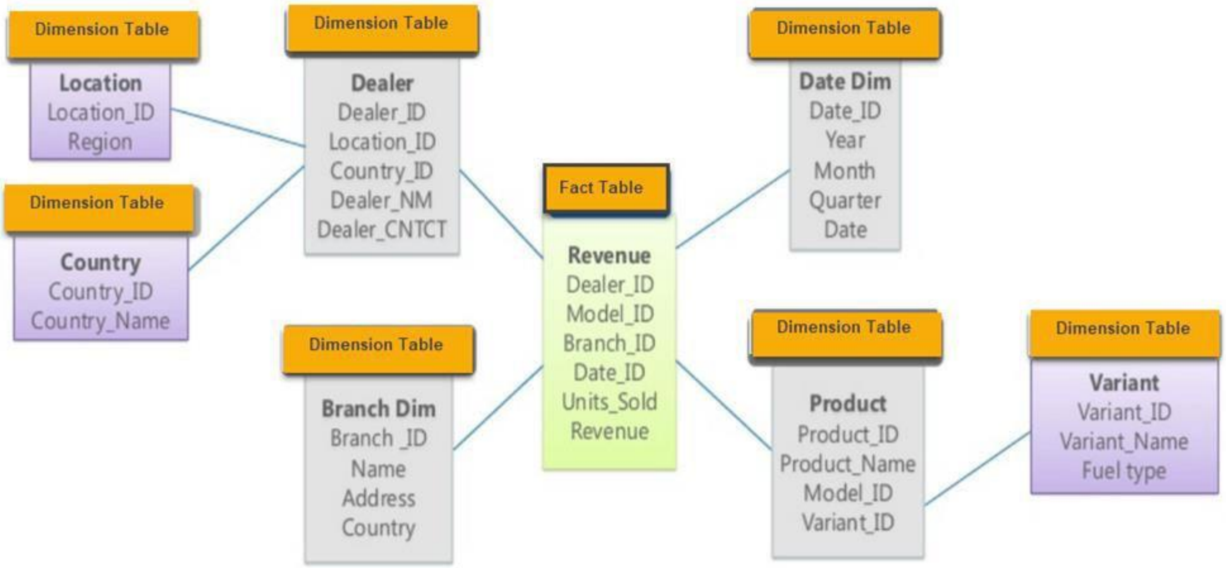


Fig: Example of Snowflake Schema

Characteristics of Snowflake Schema:

- ✓ The main benefit of the snowflake schema it uses smaller disk space.
- ✓ Easier to implement a dimension is added to the Schema
- ✓ Due to multiple tables query performance is reduced
- ✓ The primary challenge that you will face while using the snowflake Schema is that you need to perform more maintenance efforts because of the more lookup tables.

Star Schema Vs Snowflake Schema: Key Differences

Star Schema	Snowflake Schema
Hierarchies for the dimensions are stored in the dimensional table.	Hierarchies are divided into separate tables.
It contains a fact table surrounded by dimension tables.	One fact table surrounded by dimension table which are in turn surrounded by dimension table
In a star schema, only single join creates the relationship between the fact table and any dimension tables.	A snowflake schema requires many joins to fetch the data.
Simple DB Design.	Very Complex DB Design.
Denormalized Data structure and query also run faster.	Normalized Data Structure.

High level of Data redundancy	Very low-level data redundancy
Single Dimension table contains aggregated data.	Data Split into different Dimension Tables.
Cube processing is faster.	Cube processing might be slow because of the complex join.
Offers higher performing queries using Star Join Query Optimization. Tables may be connected with multiple dimensions.	The Snowflake schema is represented by centralized fact table which unlikely connected with multiple dimensions.

3. Fact Constellation Schema (Galaxy Schema)

A Galaxy Schema contains two fact table that share dimension tables between them. It is also called Fact Constellation Schema. The schema is viewed as a collection of stars hence the name Galaxy Schema.

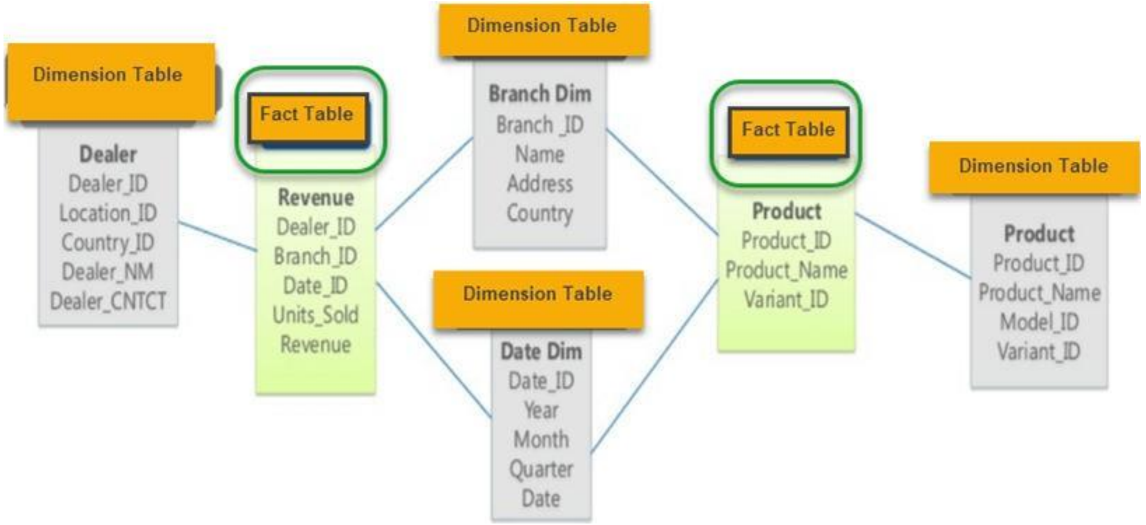


Fig: Example of Galaxy Schema

Characteristics of Galaxy Schema

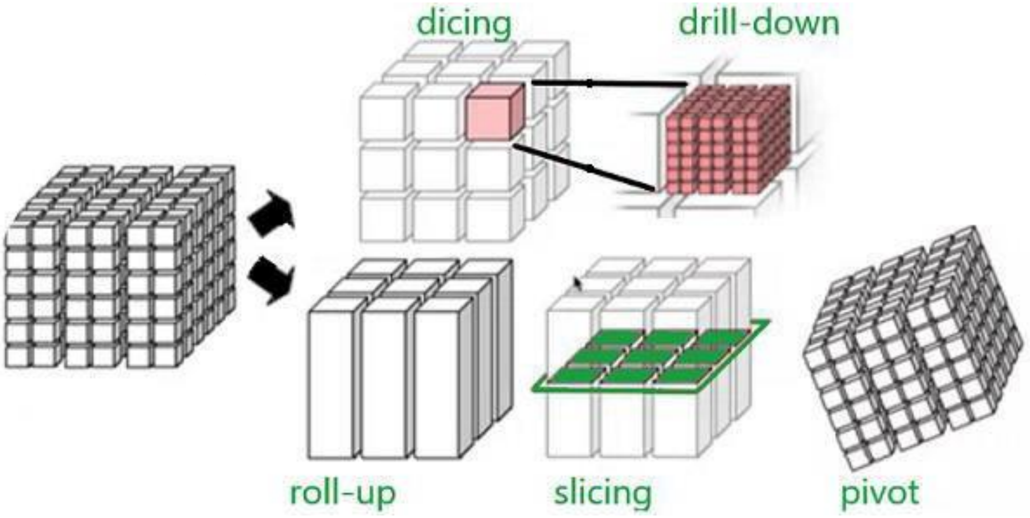
- ✓ The dimensions in this schema are separated into separate dimensions based on the various levels of hierarchy.
- ✓ For example, if geography has four levels of hierarchy like region, country, state, and city then Galaxy schema should have four dimensions.
- ✓ Moreover, it is possible to build this type of schema by splitting the one-star schema into more Star schemes.
- ✓ The dimensions are large in this schema which is needed to build based on the levels of hierarchy.
- ✓ This schema is helpful for aggregating fact tables for better understanding.

Snowflake schema contains fully expanded hierarchies. However, this can add complexity to the Schema and requires extra joins. On the other hand, star schema contains fully collapsed hierarchies, which may lead to redundancy. So, the best solution may be a balance between these two schemas which is **Star Cluster Schema design.**

Data cube operations:

Data cube operations are used to manipulate data to meet the needs of users. These operations help to select particular data for the analysis purpose. There are mainly 5 operations listed below-

1. **Roll-up**: operation and aggregate certain similar data attributes having the same dimension together.
2. **Drill-down**: this operation is the reverse of the roll-up operation. It allows us to take particular information and then subdivide it further for coarser granularity analysis.
3. **Slicing**: this operation filters the unnecessary portions.
4. **Dicing**: this operation does a multidimensional cutting, that not only cuts only one dimension but also can go to another dimension and cut a certain range of it.
5. **Pivot**: this operation is very important from a viewing point of view. It basically transforms the data cube in terms of view. It doesn't change the data present in the data cube.

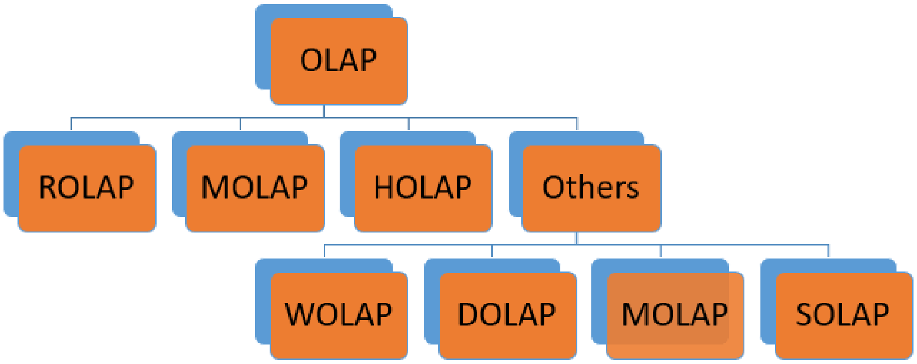


OLAP Servers

Online Analytical Processing Server (OLAP) is based on the multidimensional data model. It allows managers, and analysts to get an insight of the information through fast, consistent, and interactive access to information.
Online Analytical Processing (OLAP) is a category of software that allows users to analyze information from multiple database systems at the same time. It is a technology that enables analysts to extract and view business data from different points of view.

There are three main types of OLAP servers are as following:

1. Relational OLAP (ROLAP)
2. Multidimensional OLAP (MOLAP)
3. Hybrid OLAP (HOLAP)

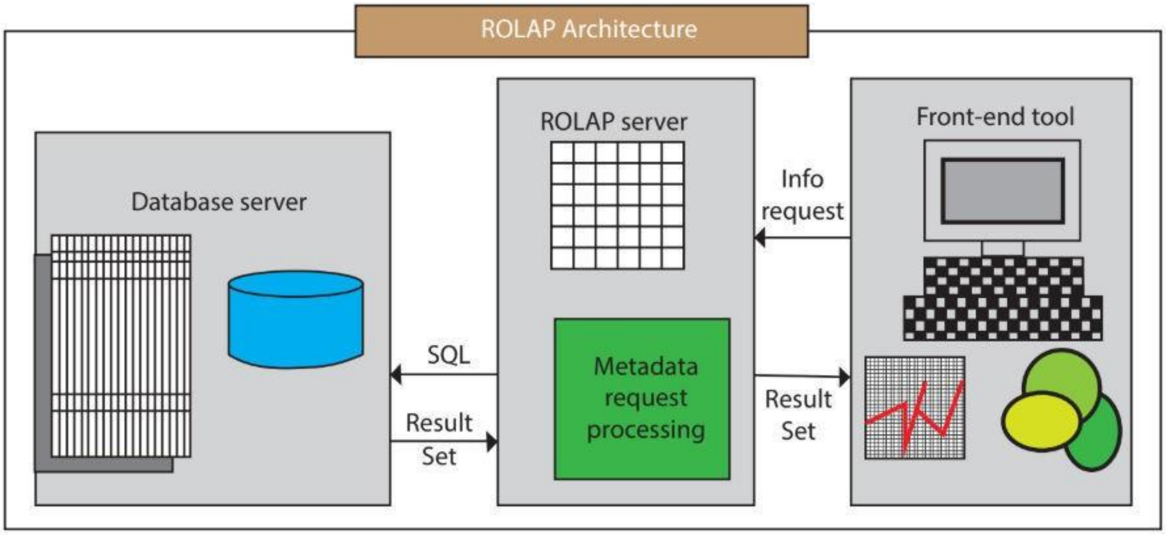


Relational OLAP (ROLAP) Server

- ✓ These are intermediate servers which stand in between a relational back-end server and user frontend tools.
- ✓ ROLAP servers contain optimization for each DBMS back end, implementation of aggregation navigation logic, and additional tools and services.
- ✓ ROLAP technology tends to have higher scalability than MOLAP technology.
- ✓ ROLAP systems work primarily from the data that resides in a relational database, where the base data and dimension tables are stored as relational tables. This model permits the multidimensional analysis of data.

ROLAP Architecture includes the following components

- ➢ Database server.
- ➢ ROLAP server.
- ➢ Front-end tool.



Relational OLAP (ROLAP) is the latest and fastest-growing OLAP technology segment in the market. This method allows multiple multidimensional views of two-dimensional relational tables to be created, avoiding structuring record around the desired view.

Advantages of ROLAP

1. ROLAP can handle large amounts of data.
2. Can be used with data warehouse and OLTP systems.

Disadvantages of ROLAP

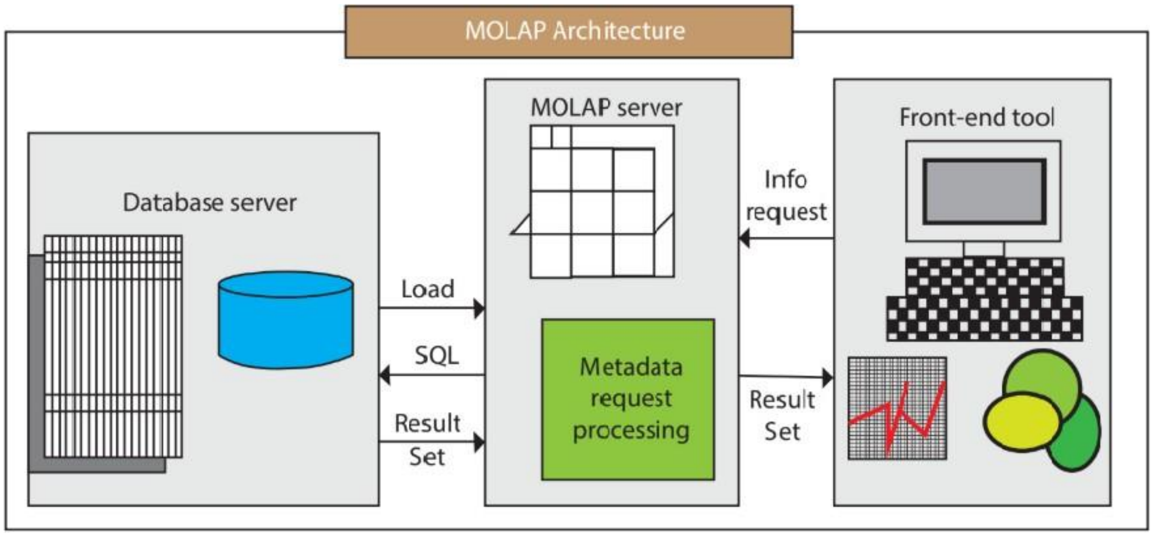
1. Limited by SQL functionalities.
2. Hard to maintain aggregate tables.

Multidimensional OLAP (MOLAP) Server

Multidimensional On-Line Analytical Processing (MOLAP) support multidimensional views of data through array-based multidimensional storage engines. With multidimensional data stores, the storage utilization may be low if the data set is sparse.
One of the significant distinctions of MOLAP against a ROLAP is that data are summarized and are stored in an optimized format in a multidimensional cube, instead of in a relational database. In MOLAP model, data are structured into proprietary formats by client's reporting requirements with the calculations pre-generated on the cubes.

MOLAP Architecture includes the following components

- ➢ Database server.
- ➢ MOLAP server.
- ➢ Front-end tool.



Advantages of MOLAP

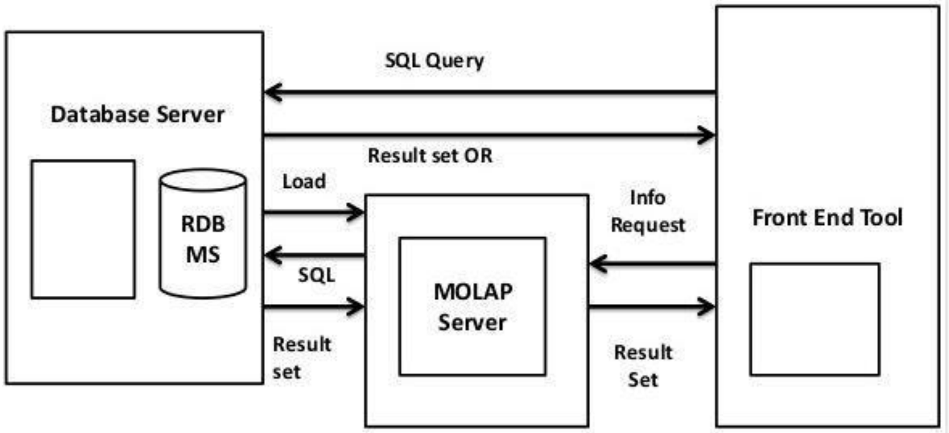
- ✓ Optimal for slice and dice operations.
- ✓ Performs better than ROLAP when data is dense.
- ✓ Can perform complex calculations.

Disadvantages of MOLAP

- ✓ Difficult to change dimension without re-aggregation.
- ✓ MOLAP can handle limited amount of data.

Hybrid OLAP (HOLAP) Server

HOLAP incorporates the best features of MOLAP and ROLAP into a single architecture. HOLAP systems save more substantial quantities of detailed data in the relational tables while the aggregations are stored in the pre-calculated cubes. HOLAP also can drill through from the cube down to the relational tables for delineated data. The Microsoft SQL Server 2000 provides a hybrid OLAP server.
Hybrid On-Line Analytical Processing (HOLAP) is a combination of ROLAP and MOLAP. HOLAP provide greater scalability of ROLAP and the faster computation of MOLAP.



Advantages of HOLAP

- ✓ HOLAP provide advantages of both MOLAP and ROLAP.
- ✓ Provide fast access at all levels of aggregation.

Disadvantages of HOLAP

- ✓ HOLAP architecture is very complex because it supports both MOLAP and ROLAP servers.

Advantages of OLAP

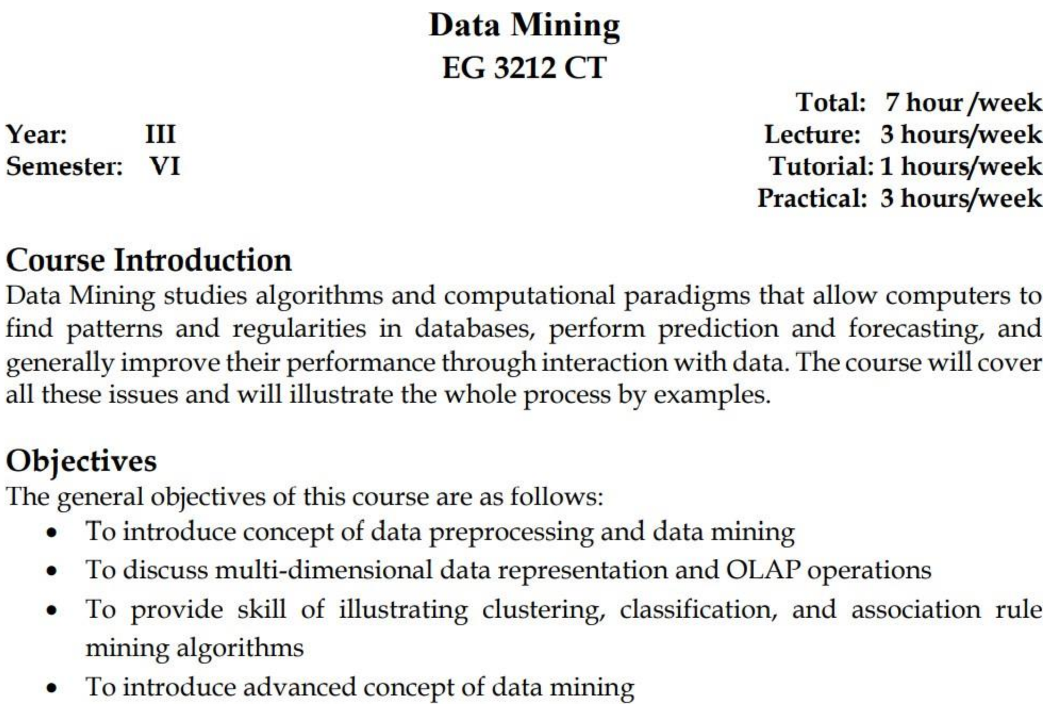
- ✓ OLAP is a platform for all type of business includes planning, budgeting, reporting, and analysis.
- ✓ Information and calculations are consistent in an OLAP cube. This is a crucial benefit.
- ✓ Quickly create and analyze “What if” scenarios
- ✓ Easily search OLAP database for broad or specific terms.
- ✓ OLAP provides the building blocks for business modeling tools, Data mining tools, performance reporting tools.
- ✓ Allows users to do slice and dice cube data all by various dimensions, measures, and filters.
- ✓ It is good for analyzing time series.
- ✓ Finding some clusters and outliers is easy with OLAP.
- ✓ It is a powerful visualization online analytical process system which provides faster response times

Disadvantages of OLAP

- ✓ OLAP requires organizing data into a star or snowflake schema. These schemas are complicated to implement and administer
- ✓ You cannot have large number of dimensions in a single OLAP cube
- ✓ Transactional data cannot be accessed with OLAP system.
- ✓ Any modification in an OLAP cube needs a full update of the cube. This is a time-consuming process

OLAP Vs OLTP

S.N.	Data Warehouse (OLAP)	Operational Database (OLTP)
1	Involves historical processing of information.	Involves day-to-day processing.
2	OLAP systems are used by knowledge workers such as executives, managers and analysts.	OLTP systems are used by clerks, DBAs, or database professionals.
3	Useful in analyzing the business.	Useful in running the business.
4	It focuses on Information out.	It focuses on Data in.
5	Based on Star Schema, Snowflake, Schema and Fact Constellation Schema.	Based on Entity Relationship Model.
6	Contains historical data.	Contains current data.
7	Provides summarized and consolidated data.	Provides primitive and highly detailed data.
8	Provides summarized and multidimensional view of data.	Provides detailed and flat relational view of data.
9	Number or users is in hundreds.	Number of users is in thousands.
10	Number of records accessed is in millions.	Number of records accessed is in tens.
11	Database size is from 100 GB to 1 TB	Database size is from 100 MB to 1 GB.
12	Highly flexible.	Provides high performance.



Data Mining
EG 3212 CT

Year: III	Total: 7 hour /week
Semester: VI	Lecture: 3 hours/week
	Tutorial: 1 hours/week
	Practical: 3 hours/week

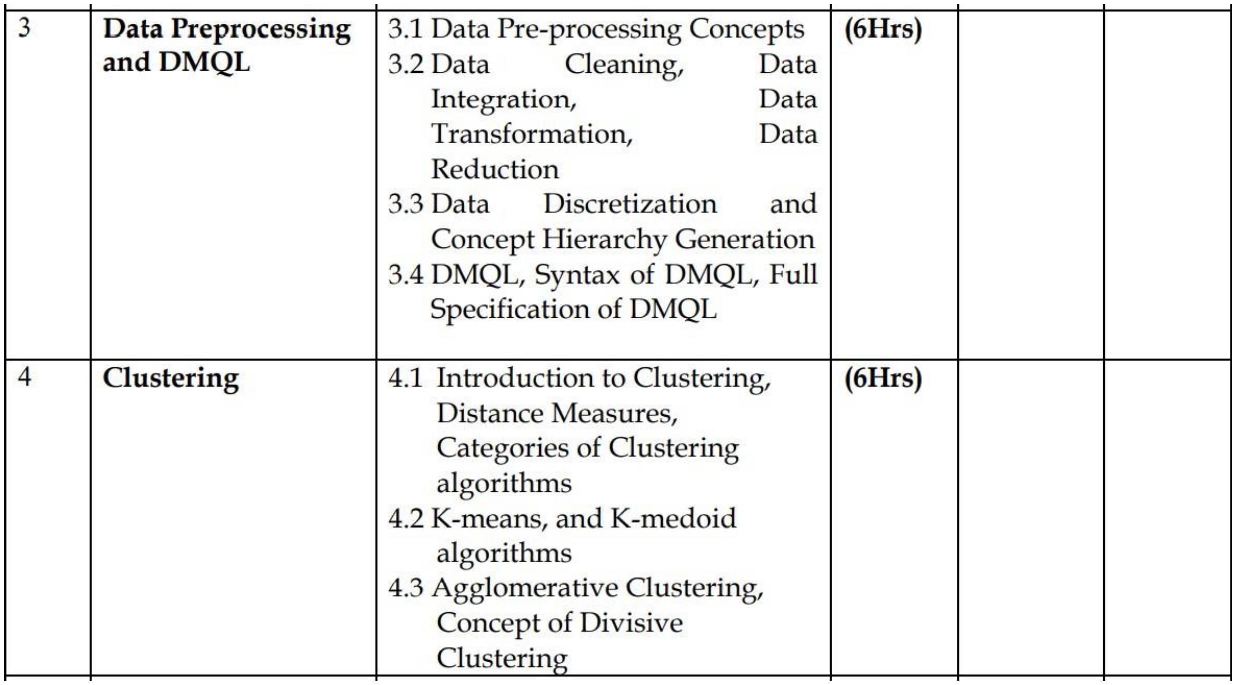
Course Introduction

Data Mining studies algorithms and computational paradigms that allow computers to find patterns and regularities in databases, perform prediction and forecasting, and generally improve their performance through interaction with data. The course will cover all these issues and will illustrate the whole process by examples.

Objectives

The general objectives of this course are as follows:

- To introduce concept of data preprocessing and data mining
- To discuss multi-dimensional data representation and OLAP operations
- To provide skill of illustrating clustering, classification, and association rule mining algorithms
- To introduce advanced concept of data mining



3	**Data Preprocessing and DMQL**	3.1 Data Pre-processing Concepts 3.2 Data Cleaning, Data Integration, Data Transformation, Data Reduction 3.3 Data Discretization and Concept Hierarchy Generation 3.4 DMQL, Syntax of DMQL, Full Specification of DMQL	**(6Hrs)**		
4	**Clustering**	4.1 Introduction to Clustering, Distance Measures, Categories of Clustering algorithms 4.2 K-means, and K-medoid algorithms 4.3 Agglomerative Clustering, Concept of Divisive Clustering	**(6Hrs)**		

Data Preprocessing Concepts

Today’s real-world databases are highly susceptible to noisy, missing, and inconsistent data due to their typically huge size (often several gigabytes or more) and their likely origin from multiple, heterogenous sources. Low-quality data will lead to low-quality mining results.

- "How can the data be preprocessed in order to help improve the quality of the data and, consequently, of the mining results?
- How can the data be preprocessed so as to improve the efficiency and ease of the mining process?"

There are several data preprocessing techniques. **Data cleaning** can be applied to remove noise and correct inconsistencies in data. **Data integration** merges data from multiple sources into a coherent data store such as a data warehouse. **Data reduction** can reduce data size by, for instance, aggregating, eliminating redundant features, or clustering. **Data transformations** (e.g., normalization) may be applied, where data are scaled to fall within a smaller range like 0.0 to 1.0. This can improve the accuracy and efficiency of mining algorithms involving distance measurements.

These techniques are not mutually exclusive; they may work together. For example, data cleaning can involve transformations to correct wrong data, such as by transforming all entries for a date field to a common format.

Data Quality: Why Preprocess the Data?

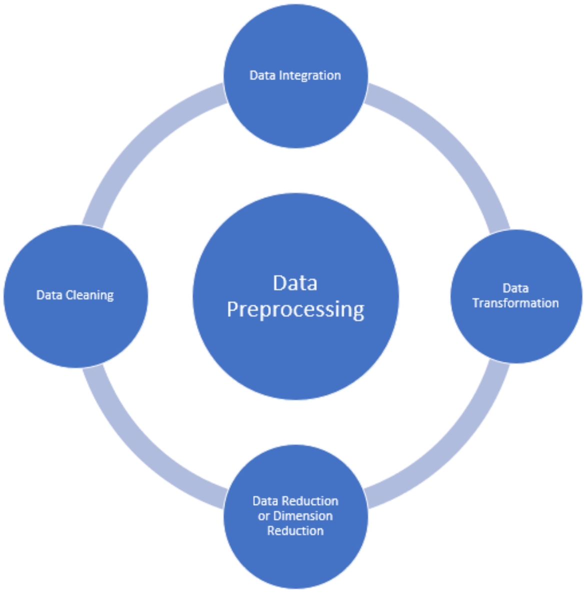
Data have quality if they satisfy the requirements of the intended use. There are many factors comprising data quality, including accuracy, completeness, consistency, timeliness, believability, and interpretability.

Preprocessing of data is mainly to check the data quality. The quality can be checked by the following

- ✓ **Accuracy**: To check whether the data entered is correct or not.
- ✓ **Completeness**: To check whether the data is available or not recorded.
- ✓ **Consistency:** To check whether the same data is kept in all the places that do or do not match.
- ✓ **Timeliness**: The data should be updated correctly.
- ✓ **Believability**: The data should be trustable.
- ✓ **Interpretability**: The understandability of the data.

Major Tasks in Data Preprocessing

The major steps involved in data preprocessing, namely, data cleaning, data integration, data reduction, and data transformation.



1. Data Cleaning

Data cleaning routines work to "clean" the data by filling in missing values, smoothing noisy data, identifying or removing outliers, and resolving inconsistencies. If users believe the data are dirty, they are unlikely to trust the results of any data mining that has been applied. Furthermore, dirty data can cause confusion for the mining procedure, resulting in unreliable output. Although most mining routines have some procedures for dealing with incomplete or noisy data, they are not always robust. Instead, they may concentrate on avoiding overfitting the data to the function being modeled. Therefore, a useful preprocessing step is to run your data through some data cleaning routines.

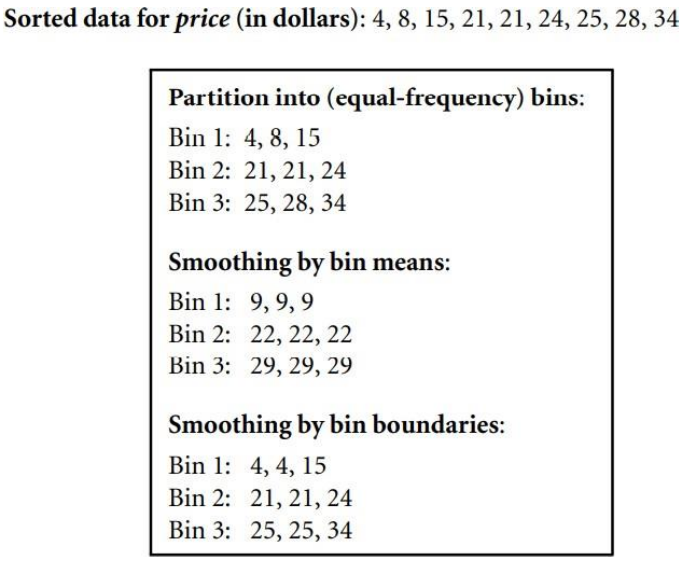
> *Data cleaning is the process to remove incorrect data, incomplete data and inaccurate data from the datasets, and it also replaces the missing values.*

When some data is missing in the data. It can be handled in various ways. Some of them are:

- ✓ Ignore the tuple
- ✓ Fill the missing value manually
- ✓ Use global constant to fill the missing values
- ✓ Use a measure of central tendency for the attribute (e.g., the mean or median) to fill in the missing value.
- ✓ Use the most probable value to fill in the missing value.

Noisy data can be handled in following ways:

- ✓ Binning method
- ✓ Regression method
- ✓ Clustering



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

Partition into (equal-frequency) bins:

Bin 1: 4, 8, 15
Bin 2: 21, 21, 24
Bin 3: 25, 28, 34

Smoothing by bin means:

Bin 1: 9, 9, 9
Bin 2: 22, 22, 22
Bin 3: 29, 29, 29

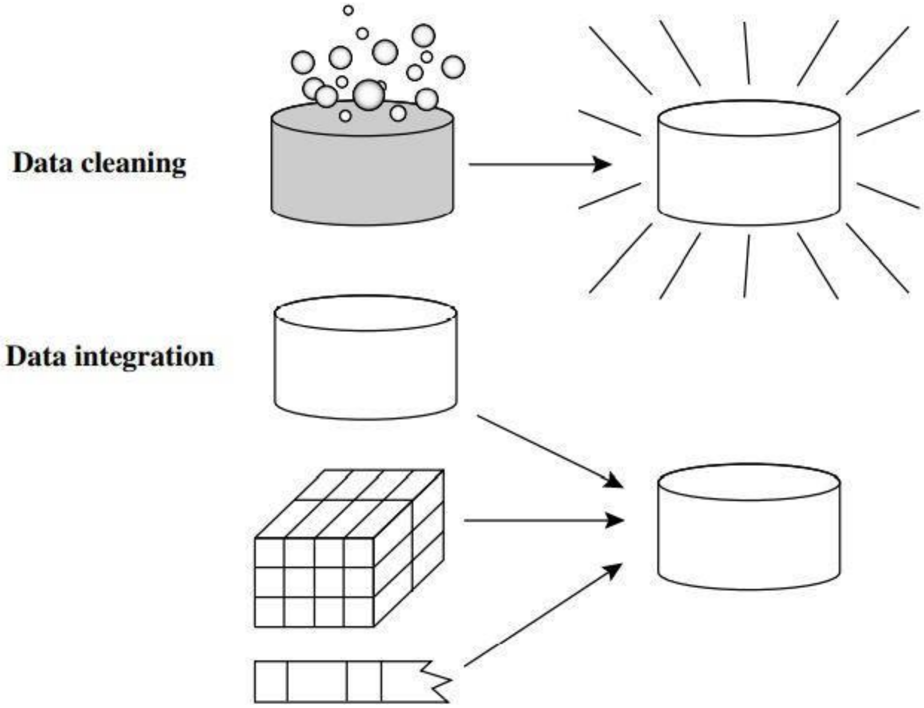
Smoothing by bin boundaries:

Bin 1: 4, 4, 15
Bin 2: 21, 21, 24
Bin 3: 25, 25, 34

Figure: Binning methods for data smoothing.

2. Data Integration

The process of combining multiple sources into a single dataset. The Data integration process is one of the main components in data management. Data integration is the method to assists when the information is collected from diversified data origin and information is merging to form continuous information.



3. Data Transformation

The change made in the format or the structure of the data is called data transformation. This step can be simple or complex based on the requirements. There are some methods in data transformation.
This involves following ways:

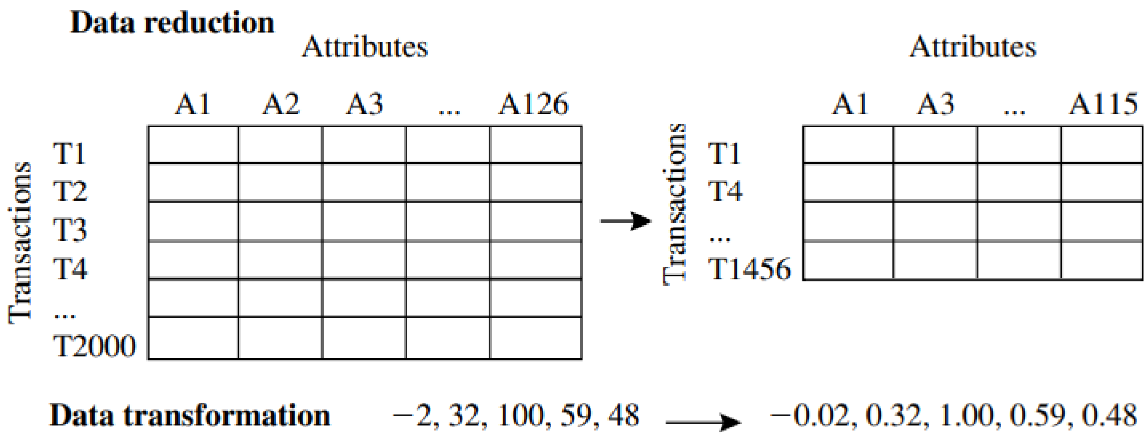
- ✓ Smoothing
- ✓ Aggregation
- ✓
- ✓ Normalization
- ✓ Attribute Selection
- ✓ Discretization
- ✓ Concept hierarchy generation

4. Data Reduction

Since data mining is a technique that is used to handle huge amount of data. While working with huge volume of data, analysis became harder in such cases. In order to get rid of this, we use data reduction technique. It aims to increase the storage efficiency and reduce data storage and analysis costs.
When the volume of data is huge, databases can become slower, costly to access, and challenging to properly store. Data reduction aims to present a reduced representation of the data in a data warehouse.
The various steps to data reduction are:

- ✓ Data Cube aggregation
- ✓ Attribute Subset Selection

- ✓ Numerosity Reduction
- ✓ Dimensionality Reduction



Although numerous methods of data preprocessing have been developed, data preprocessing remains an active area of research, due to the huge amount of inconsistent or dirty data and the complexity of the problem.

Data Discretization and Concept of Hierarchy Generation

Data discretization refers to a method of converting a huge number of data values into smaller ones so that the evaluation and management of data become easy. In other words, data discretization is a method of converting attributes values of continuous data into a finite set of intervals with minimum data loss.

Now, we can understand this concept with the help of an example
Suppose we have an attribute of Age with the given values

Age	1,5,9,4,7,11,14,17,13,18, 19,31,33,36,42,44,46,70,74,78,77

Table before Discretization

Attribute	Age	Age	Age	Age
	1,5,4,9,7	11,14,17,13,18,19	31,33,36,42,44,46	70,74,77,78
After Discretization	Child	Young	Mature	Old

A concept hierarchy for a given numeric attribute attribute defines a discretization of the attribute. Concept hierarchies can be used to reduce the data y collecting and replacing low-level concepts (such as numeric value for the attribute age) by higher level concepts (such as young, middle-aged, or senior). Although detail is lost by such generalization, it becomes meaningful and it is easier to interpret.

Some Famous techniques of data discretization

- **Histogram analysis**

  Histogram refers to a plot used to represent the underlying frequency distribution of a continuous data set. Histogram assists the data inspection for data distribution. For example, Outliers, skewness representation, normal distribution representation, etc.
- **Binning**

  Binning refers to a data smoothing technique that helps to group a huge number of continuous values into smaller values. For data discretization and the development of idea hierarchy, this technique can also be used.
- **Cluster Analysis**

  Cluster analysis is a form of data discretization. A clustering algorithm is executed by dividing the values of x numbers into clusters to isolate a computational feature of x.
- **Decision tree analysis**

  Data discretization refers to a decision tree analysis in which a top-down slicing technique is used. It is done through a supervised procedure.
- **Correlation analysis**

  Discretizing data by linear regression technique, you can get the best neighboring interval, and then the large intervals are combined to develop a larger overlap to form the final 20 overlapping intervals. It is a supervised procedure.

Concept Hierarchy Generation

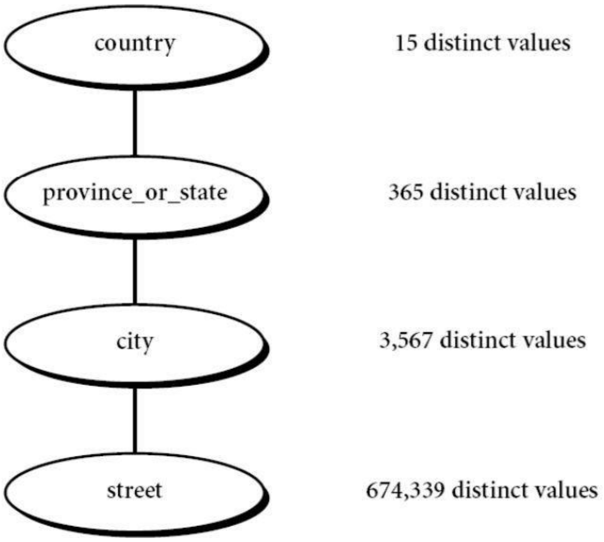
The term hierarchy represents an organizational structure or mapping in which items are ranked according to their levels of importance. For example, in computer science, there are different types of hierarchical systems. A document is placed in a folder in windows at a specific place in the tree structure is the best example of a computer hierarchical tree model. There are two types of hierarchy: top-down mapping and the second one is bottom-up mapping.

Top-down mapping

Top-down mapping generally starts with the top with some general information and ends with the bottom to the specialized information.

Bottom-up mapping

Bottom-up mapping generally starts with the bottom with some specialized information and ends with the top to the generalized information.



DMQL (Data Mining Query Language)

DMQL is designed based on Structured Query Language (SQL) which in turn is a relational query language. To support the whole knowledge discovery process, we need for integrated systems which can deal either with patterns and data. The inductive database approach has emerged as a unifying framework for such systems. Following this database perspective, knowledge discovery processes become querying processes for which query languages have to be designed.
The Data Mining Query Language (DMQL) was proposed by Han, Fu, Wang, et al. for the DBMiner data mining system. The Data Mining Query Language is actually based on the Structured Query Language (SQL). Data Mining Query Languages can be designed to support ad hoc and interactive data mining. This DMQL provides commands for specifying primitives. The DMQL can work with databases and data warehouses as well.

Data Mining Query Language (DMQL) adopts SQL-like syntax.
Hence, it can be easily integrated with relational query languages.

Syntax of DMQL

DMQL is designed for Data Mining Task Primitives.
Its syntax goes for all of the primitives.
Syntax for the specification of

- ✓ Task-relevant data
- ✓ The kind of knowledge to be mined
- ✓ Concept hierarchy specification
- ✓ Interestingness measure
- ✓ Pattern presentation and visualization

Putting it all together — a DMQL query is formed.

1. DMQL-Syntax for task-relevant data specification

Names of the relevant database or data warehouse, conditions, and relevant attributes or dimensions must be specified:

use database ‹database_name› or **use data warehouse** ‹data_warehouse_name›
from ‹relation(s)/cube(s)› [**where** condition] (data cubes and tables)
in relevance to ‹attribute_or_dimension_list› (attributes or dimension for exploration)
order by ‹order_list› (sorting order)
group by ‹grouping_list› (specifies criteria to group)
having ‹condition› (it represent which group of data are considered relevant)

2. Syntax for specifying the kind of knowledge.

Syntax for Characterization, Discrimination, Association, Classification, and Prediction.

a. Data mining characterization

The syntax for characterization is −

mine characteristics [as pattern_name]
analyze {measure(s)}

b. Data mining discrimination

The syntax for Discrimination is –

```
mine comparison [as {pattern_name]}
For {target_class } where {t arget_condition }
{versus {contrast_class_i }
where {contrast_condition_i}}
analyze {measure(s) }
```

c. Data mining association

The syntax for Association is–

```
mine associations [ as {pattern_name} ]
{matching {metapattern} }
```

d. Data mining classification

The syntax for Classification is –

```
mine classification [as pattern_name]
analyze classifying_attribute_or_dimension
```

e. Data mining prediction

The syntax for prediction is –

```
mine prediction [as pattern_name]
analyze  prediction_attribute_or_dimension
{set  {attribute_or_dimension_i=  value_i}}
```

Example 1: where 'numeric_field' is equal to 2
DMQL query: (numeric_field=2)
Equivalent SQL server query: WHERE numeric_field = 2

Example 2: where 'numeric_field' is greater than or equal to 3
DMQL query: (numeric_field=3+)
Equivalent SQL server query: WHERE numeric_field >= 3

Example 3: where 'numeric_field' is less than or equal to 8
DMQL query:(numeric_field=8-)
Equivalent SQL server query: WHERE numeric_field

Example 4: where 'numeric_field' is between 5 and 9 (including 5 and 9)
DMQL query:(numeric_field=5-9)
Equivalent SQL server query: WHERE numeric_field between 5 and 9

3. Syntax for Concept Hierarchy Specification

To specify concept hierarchies, use the following syntax –
use hierarchy <hierarchy> for <attribute_or_dimension>

For Example –

```
with support threshold = 0.05
with confidence threshold = 0.7
```

4. **Syntax for Pattern Presentation and Visualization Specification**
   We have a syntax, which allows users to specify the display of discovered patterns in one or more forms.

```
display as <result_form>
For Example –
display as table
```

Full Specification of DMQL

As a market manager of a company, you would like to characterize the buying habits of customers who can purchase items priced at no less than $100; with respect to the customer's age, type of item purchased, and the place where the item was purchased. You would like to know the percentage of customers having that characteristic. In particular, you are only interested in purchases made in Canada, and paid with an American Express credit card. You would like to view the resulting descriptions in the form of a table.

Putting it all together: A DMQL query

Here is an example DMQL for AllElectronics Database

```
use database AllElectronics_db
use hierarchy location_hierarchy for B.address
mine characteristics as customerPurchasing
analyze count%
in relevance to C.age, I.type, I.place_made
from customer C,  item I, purchases P, items_sold S, works_at W, branch
where I.item_ID = S.item_ID  and S.trans_ID = P.trans_ID
   and P.cust_ID = C.cust_ID and P.method_paid = ""AmEx""
   and P.empl_ID = W.empl_ID and W.branch_ID = B.branch_ID and B.address =
""Canada""  and I.price >= 100
with noise threshold = 0.05
display as table
```

This above query represents the whole Data Mining Query for a Database or Data Warehouse.

Purposes of – Data Mining Query

Data Mining Queries are useful for many purposes are:

- ✓ Apply the model to new data, to make single or multiple predictions. You can provide input values as parameters, or in a batch.
- ✓ Get a statistical summary of the data used for training.
- ✓ Extract patterns and rule of the typical case representing a pattern in the model.
- ✓ Extract regression formulas and other calculations that explain patterns.
- ✓ Get the cases that fit a particular pattern.
- ✓ Retrieve details about individual cases used in the model. Also, it includes data not used in an analysis.
- ✓ Retrain a model by adding new data, or perform cross-prediction.

Introduction to clustering

In clustering, a group of different data objects is classified as similar objects. One group means a cluster of data. Data sets are divided into different groups in the cluster analysis, which is based on the similarity of the data. After the classification of data into various groups, a label is assigned to the group. It helps in adapting to the changes by doing the classification.

Cluster Analysis in Data Mining means that to find out the group of objects which are similar to each other in the group but are different from the object in other groups.

Applications of cluster analysis:

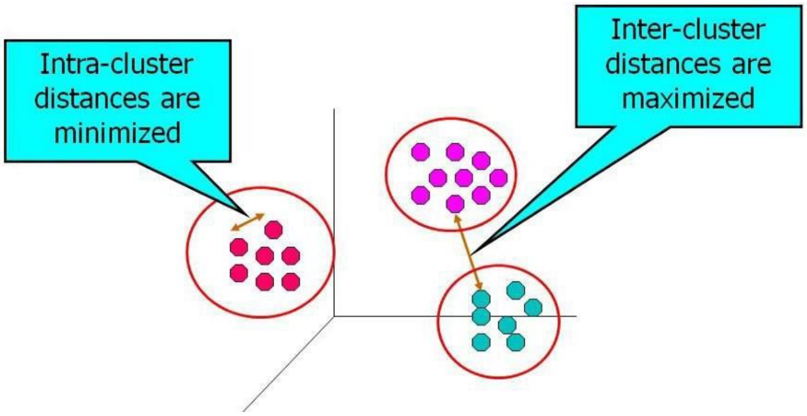
- ✓ It is widely used in many applications such as image processing, data analysis, and pattern recognition.
- ✓ It helps marketers to find the distinct groups in their customer base and they can characterize their customer groups by using purchasing patterns.
- ✓ It can be used in the field of biology, by deriving animal and plant taxonomies, identifying genes with the same capabilities.
- ✓ It also helps in information discovery by classifying documents on the web.

In this sense, clustering is sometimes called automatic classification. Again, a critical difference here is that clustering can automatically find the groupings. This is a distinct advantage of cluster analysis. Clustering is also called data segmentation in some applications because clustering partitions large data sets into groups according to their similarity.

High quality clusters can be created by reducing the distance between the objects in the same cluster known as intra-cluster minimization and increasing the distance with the objects in the other cluster known as inter-cluster maximization.

Intra-cluster minimization: The closer the objects in a cluster, the more likely they belong to the same cluster.

Inter-cluster Maximization: This makes the separation between two clusters. The main goal is to maximize the distance between 2 clusters.


Intra-cluster distances are minimized
Inter-cluster distances are maximized

As a branch of statistics, cluster analysis has been extensively studied, with the main focus on distance-based cluster analysis. Cluster analysis tools based on k-means, k-medoids, and several other methods also have been built into many statistical analysis software packages or systems, such as S-Plus, SPSS, and SAS.

The following are typical requirements of clustering in data mining.

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

Distance Measures

In the clustering setting, a distance (or equivalently a similarity) measure is a function that quantifies the similarity between two objects. Distance measures play an important role in machine learning.
A distance measure is an objective score that summarizes the relative difference between two objects in a problem domain.

Most commonly, the two objects are rows of data that describe a subject (such as a person, car, or house), or an event (such as a purchase, a claim, or a diagnosis).
Perhaps the most likely way you will encounter distance measures is when you are using a specific machine learning algorithm that uses distance measures at its core. The most famous algorithm of this type is the k-nearest neighbors' algorithm, or KNN for short.

Some of the more popular machine learning algorithms that use distance measures at their core is as follows:

- K-Nearest Neighbors
- Learning Vector Quantization (LVQ)
- Self-Organizing Map (SOM)
- K-Means Clustering

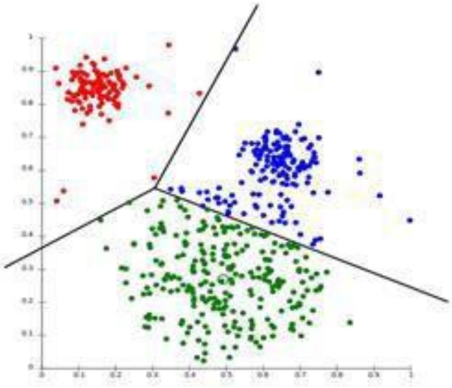
Perhaps four of the most commonly used distance measures in machine learning are as follows:

- Hamming Distance
- Euclidean Distance
- Manhattan Distance
- Minkowski Distance

Clustering Methods

Clustering methods can be classified into the following categories –

- ➢ **Partitioning Method**
  Given a set of n objects, a partitioning method constructs k partitions of the data, where each partition represents a cluster and k ≤ n. That is, it divides the data into k groups such that each group must contain at least one object. In other words, partitioning methods conduct one-level partitioning on data sets.
  Most partitioning methods are distance-based. Given k, the number of partitions to construct, a partitioning method creates an initial partitioning.



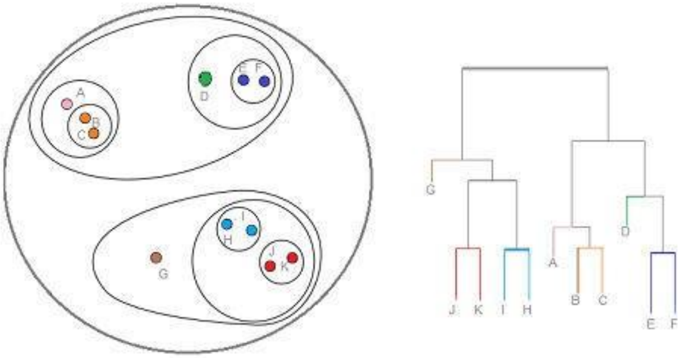
- **Hierarchical Method**

  A hierarchical method creates a hierarchical decomposition of the given set of data objects. A hierarchical method can be classified as being either agglomerative or divisive, based on how the hierarchical decomposition is formed.

  The agglomerative approach, also called the bottom-up approach, starts with each object forming a separate group. It successively merges the objects or groups close to one another, until all the groups are merged into one (the topmost level of the hierarchy), or a termination condition holds. The divisive approach, also called the top-down approach, starts with all the objects in the same cluster. In each successive iteration, a cluster is split into smaller clusters, until eventually each object is in one cluster, or a termination condition holds. Hierarchical clustering methods can be distance-based or density- and continuity based. Various extensions of hierarchical methods consider clustering in subspaces as well. Hierarchical methods suffer from the fact that once a step (merge or split) is done, it can never be undone.

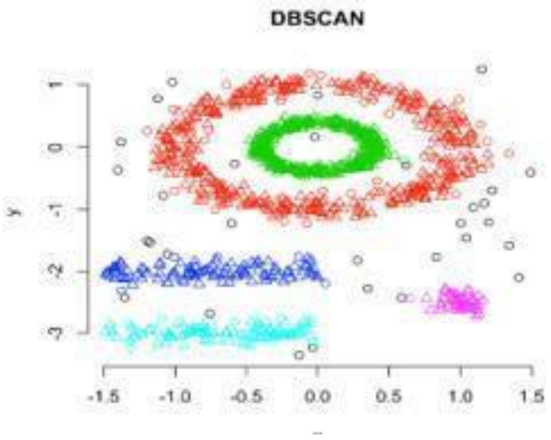
  There are two types of approaches for the creation of hierarchical decomposition, which are: –

  a. Divisive Approach
  b. Agglomerative Approach



- **Density-based Method**

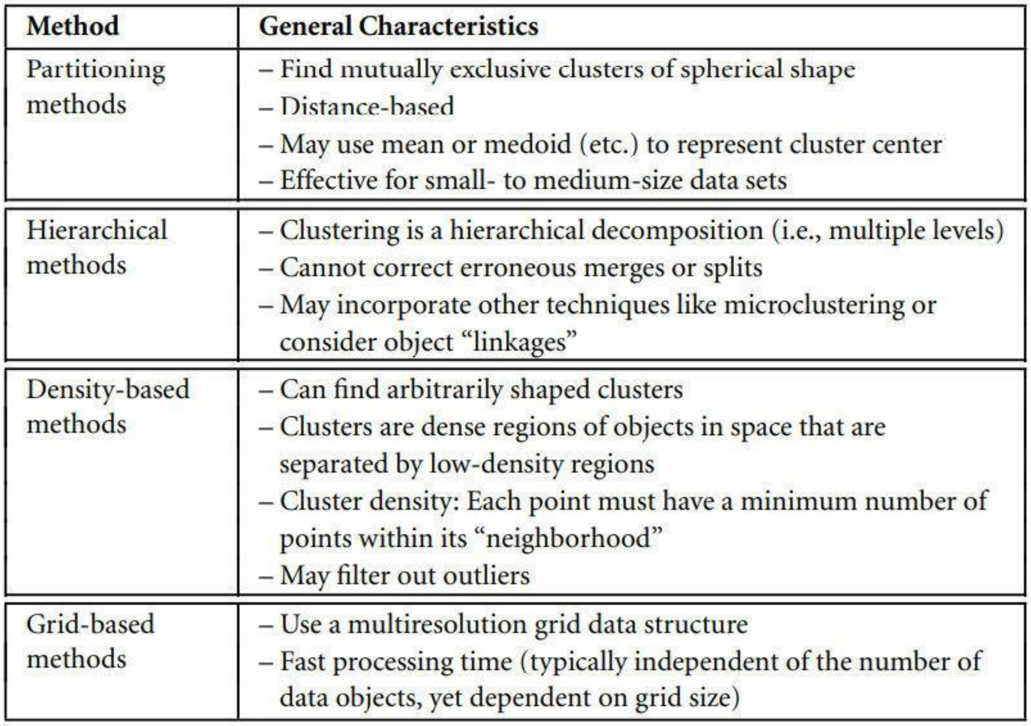
  Most partitioning methods cluster objects based on the distance between objects. Such methods can find only spherical-shaped clusters and encounter difficulty in discovering clusters of arbitrary shapes. Other clustering methods have been developed based on the notion of density. Their general idea is to continue growing a given cluster as long as the density (number of objects or data points) in the "neighborhood" exceeds some threshold. For example, for each data point within a given cluster, the neighborhood of a given radius has to contain at least a minimum number of points. Such a method can be used to filter out noise or outliers and discover clusters of arbitrary shape.



- **Grid-Based Method**

  Grid-based methods quantize the object space into a finite number of cells that form a grid structure. All the clustering operations are performed on the grid structure (i.e., on the quantized space). The main advantage of this approach is its fast-processing time, which is typically independent of the number of data objects and dependent only on the number of cells in each dimension in the quantized space.

  Using grids is often an efficient approach to many spatial data mining problems, including clustering. Therefore, grid-based methods can be integrated with other clustering methods such as density-based methods and hierarchical methods.



Method	General Characteristics
Partitioning methods	– Find mutually exclusive clusters of spherical shape – Distance-based – May use mean or medoid (etc.) to represent cluster center – Effective for small- to medium-size data sets
Hierarchical methods	– Clustering is a hierarchical decomposition (i.e., multiple levels) – Cannot correct erroneous merges or splits – May incorporate other techniques like microclustering or consider object "linkages"
Density-based methods	– Can find arbitrarily shaped clusters – Clusters are dense regions of objects in space that are separated by low-density regions – Cluster density: Each point must have a minimum number of points within its "neighborhood" – May filter out outliers
Grid-based methods	– Use a multiresolution grid data structure – Fast processing time (typically independent of the number of data objects, yet dependent on grid size)

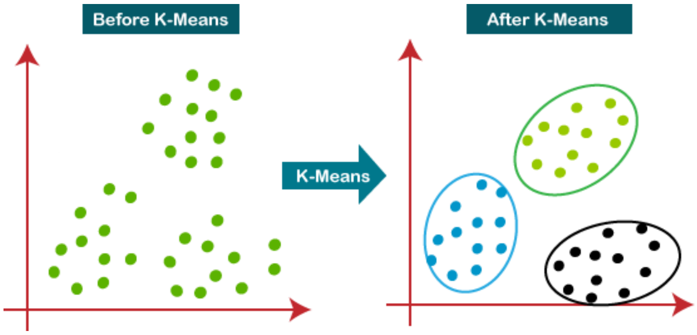
Figure: Overview of clustering methods

K-Means Algorithms

> It is an iterative algorithm that divides the unlabeled dataset into k different clusters in such a way that each dataset belongs only one group that has similar properties.

K-means clustering is one of the simplest and popular unsupervised machine learning algorithms. To process the learning data, the K-means algorithm in data mining starts with a first group of randomly selected centroids, which are used as the beginning points for every cluster, and then performs iterative (repetitive) calculations to optimize the positions of the centroids. It halts creating and optimizing clusters when either:

- ✓ The centroids have stabilized — there is no change in their values because the clustering has been successful.
- ✓ The defined number of iterations has been achieved.

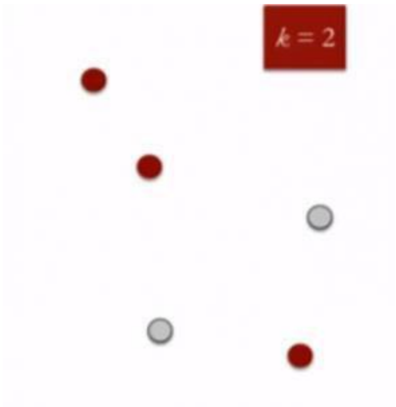


The working of the K-Means algorithm is explained in the below steps:

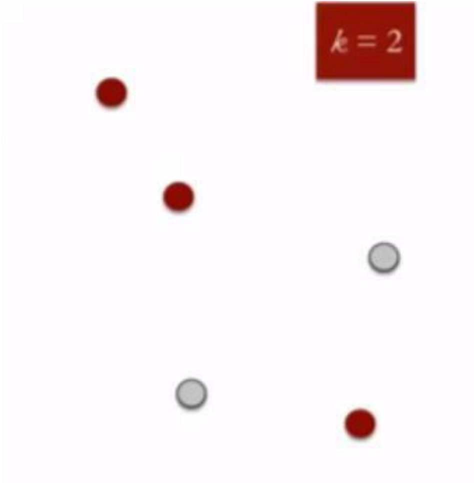
Step-1: Select the number K to decide the number of clusters.
Step-2: Select random K points or centroids. (It can be other from the input dataset).
Step-3: Assign each data point to their closest centroid, which will form the predefined K clusters.
Step-4: Calculate the variance and place a new centroid of each cluster.
Step-5: Repeat the third steps, which means reassign each datapoint to the new closest centroid of each cluster.
Step-6: If any reassignment occurs, then go to step-4 else go to FINISH.
Step-7: The model is ready.

Example:

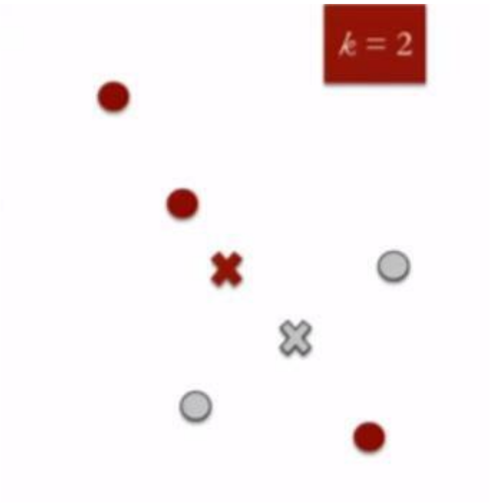
1. Specify the desired number of clusters K: Let us choose k=2 for these 5 data points in 2-D space.



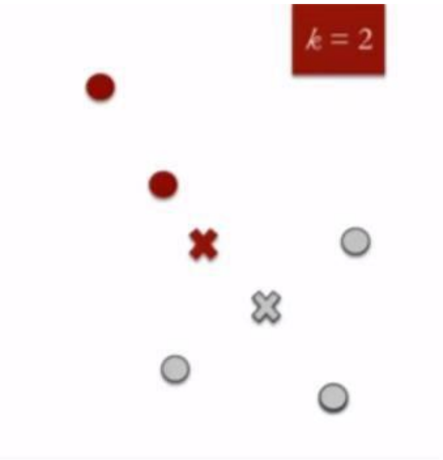
2. Randomly assign each data point to a cluster: Let's assign three points in cluster 1 shown using red color and two points in cluster 2 shown using grey color.



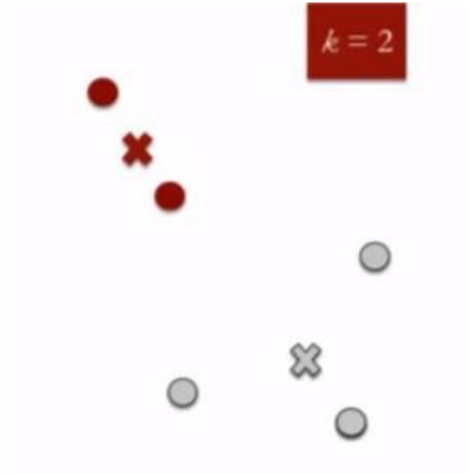
3. Compute cluster centroids: The centroid of data points in the red cluster is shown using red cross and those in grey cluster using grey cross.



4. Re-assign each point to the closest cluster centroid: Note that only the data point at the bottom is assigned to the red cluster even though its closer to the centroid of grey cluster. Thus, we assign that data point into grey cluster.



5. Re-compute cluster centroids: Now, re-computing the centroids for both the clusters.



6. Repeat steps 4 and 5 until no improvements are possible: Similarly, we'll repeat the 4th and 5th steps until we'll reach global optima. When there will be no further switching of data points between two clusters for two successive repeats. It will mark the termination of the algorithm if not explicitly mentioned.

K-medoid Algorithm

K-Medoids is a clustering algorithm resembling the K-Means clustering technique**.** K-Medoids (also called as Partitioning Around Medoid) algorithm was proposed in 1987 by Kaufman and Rousseeuw. A medoid can be defined as the point in the cluster, whose dissimilarities with all the other points in the cluster is minimum.

It majorly differs from the K-Means algorithm in terms of the way it selects the clusters' centers. The former selects the average of a cluster's points as its center (which may or may not be one of the data points) while the latter always picks the actual data points from the clusters as their centers (also known as '**exemplars' or** 'medoids').

The most common realization of *k*-medoid clustering is the **Partitioning Around Medoids(PAM)** algorithm and is as follows:

Algorithm:

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

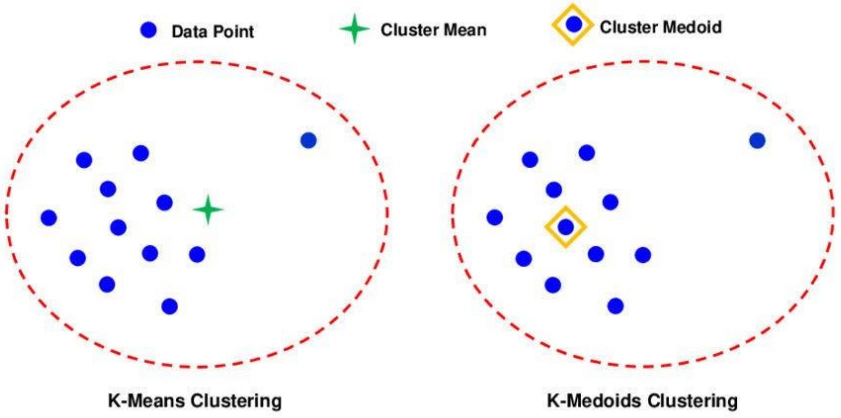
Advantages:

- ✓ It is simple to understand and easy to implement.
- ✓ K-Medoid Algorithm is fast and converges in a fixed number of steps.
- ✓ PAM is less sensitive to outliers than other partitioning algorithms.

Suppose, we have 5 diamonds with prices 2, 100, 102, 110, 115.
For K-Medoids, we take each diamond and compute its distance with the other diamonds. Then we select the diamond with the minimum distance.

Distance between diamond with price 2 and the other diamonds = 419
Distance between diamond with price 100 and the other diamonds = 125
Distance between diamond with price 102 and the other diamonds = 123
Distance between diamond with price 110 and the other diamonds = 131
Distance between diamond with price 115 and the other diamonds = 146

This time, we chose 102 as the center. We call it a medoid. It is a better option in our case. A medoid as a median is not sensitive to outliers. **But a medoid is not a median**.



Hierarchical Clustering (Agglomerative and Divisive Clustering)

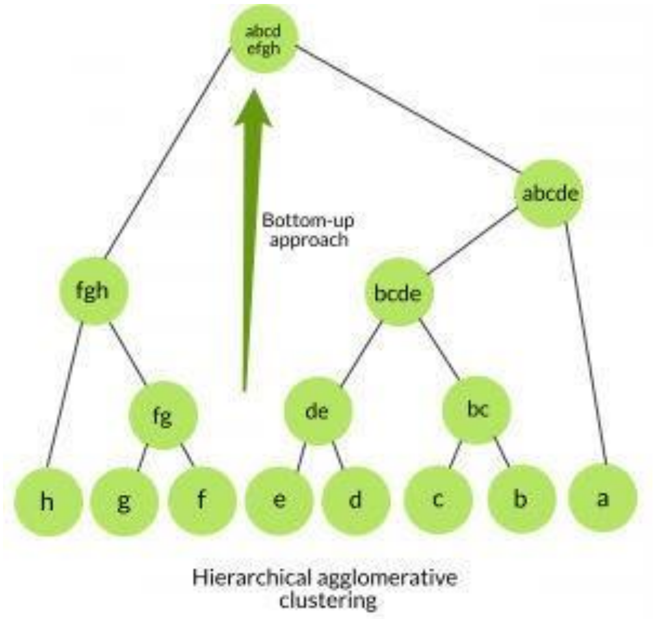
Hierarchical Clustering creates clusters in a hierarchical tree-like structure (also called a Dendrogram). Meaning, a subset of similar data is created in a tree-like structure in which the root node corresponds to entire data, and branches are created from the root node to form several clusters.

Hierarchical clustering is of two types:

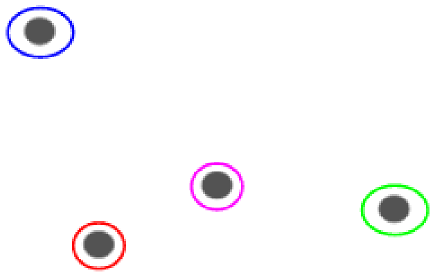
1. Agglomerative Clustering
2. Divisive clustering

1. Agglomerative Clustering

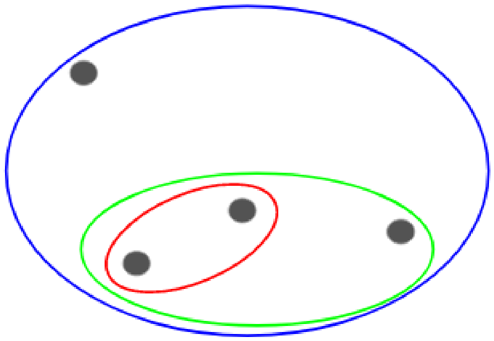
Agglomerative Hierarchical Clustering is popularly known as a bottom-up approach, wherein each data or observation is treated as its cluster. A pair of clusters are combined until all clusters are merged into one big cluster that contains all the data.



We assign each point to an individual cluster in this technique. Suppose there are 4 data points. We will assign each of these points to a cluster and hence will have 4 clusters in the beginning:



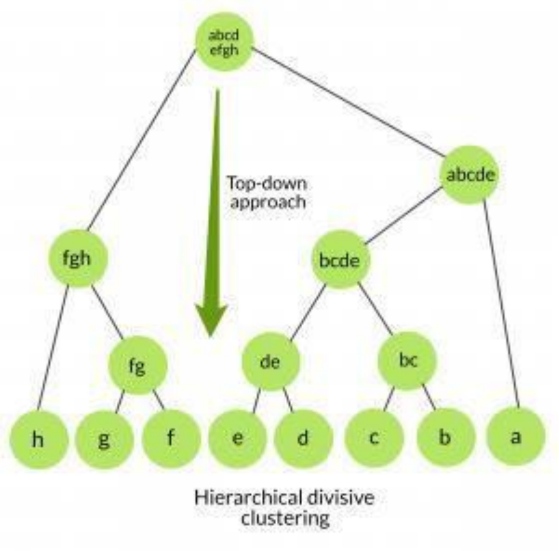
Then, at each iteration, we merge the closest pair of clusters and repeat this step until only a single cluster is left:



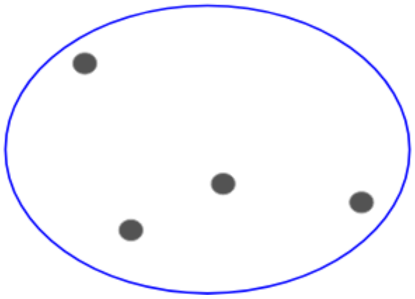
We are merging (or adding) the clusters at each step, right? Hence, this type of clustering is also known as additive hierarchical clustering.

2. Divisive clustering

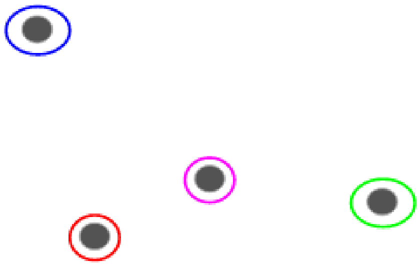
Divisive Hierarchical Clustering is also termed as a top-down clustering approach. In this technique, entire data or observation is assigned to a single cluster. The cluster is further split until there is one cluster for each data or observation. Divisive hierarchical clustering works by starting with 1 cluster containing the entire data set.



Instead of starting with n clusters (in case of n observations), we start with a single cluster and assign all the points to that cluster.
So, it doesn't matter if we have 10 or 1000 data points. All these points will belong to the same cluster at the beginning:



Now, at each iteration, we split the farthest point in the cluster and repeat this process until each cluster only contains a single point:



We are splitting (or dividing) the clusters at each step, hence the name divisive hierarchical clustering. Agglomerative Clustering is widely used in the industry and that will be the focus in this article. Divisive hierarchical clustering will be a piece of cake once we have a handle on the agglomerative type.

Both algorithms are exactly the opposite of each other.

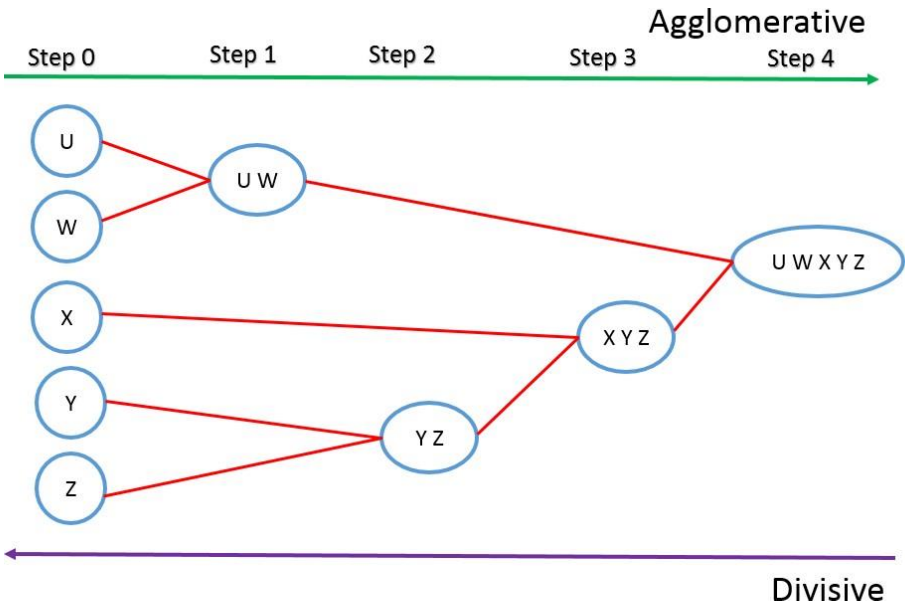
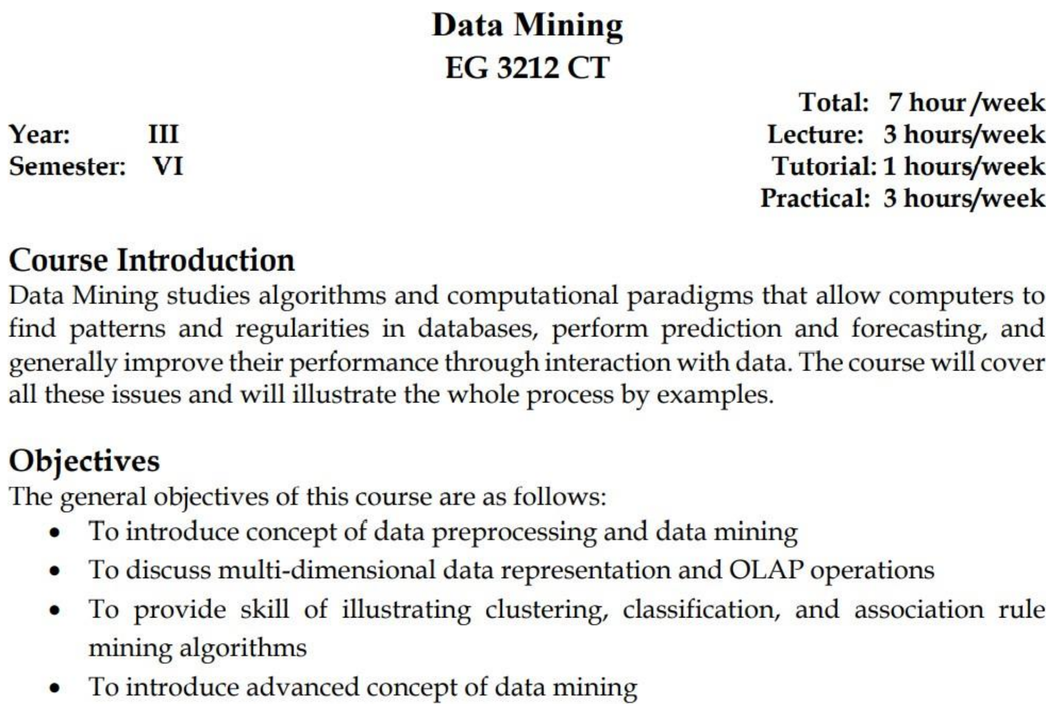


Figure: Comparative schematic diagram of agglomerative clustering and divisive clustering

THE END



Data Mining
EG 3212 CT

Total: 7 hour /week

Year: III	Lecture: 3 hours/week
Semester: VI	Tutorial: 1 hours/week
	Practical: 3 hours/week

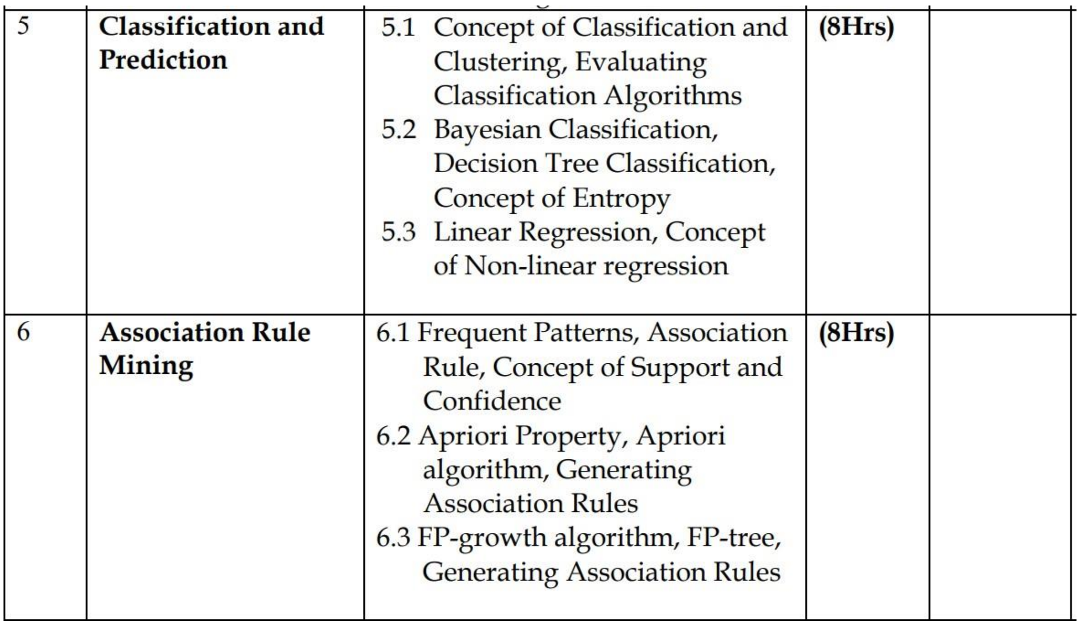
Course Introduction

Data Mining studies algorithms and computational paradigms that allow computers to find patterns and regularities in databases, perform prediction and forecasting, and generally improve their performance through interaction with data. The course will cover all these issues and will illustrate the whole process by examples.

Objectives

The general objectives of this course are as follows:

- To introduce concept of data preprocessing and data mining
- To discuss multi-dimensional data representation and OLAP operations
- To provide skill of illustrating clustering, classification, and association rule mining algorithms
- To introduce advanced concept of data mining



5	**Classification and Prediction**	5.1 Concept of Classification and Clustering, Evaluating Classification Algorithms 5.2 Bayesian Classification, Decision Tree Classification, Concept of Entropy 5.3 Linear Regression, Concept of Non-linear regression	**(8Hrs)**	
6	**Association Rule Mining**	6.1 Frequent Patterns, Association Rule, Concept of Support and Confidence 6.2 Apriori Property, Apriori algorithm, Generating Association Rules 6.3 FP-growth algorithm, FP-tree, Generating Association Rules	**(8Hrs)**	

Classification and Prediction

Concept of Classification and Clustering

Data classification in data mining is a common technique that helps in organizing data sets that are both complicated and large. This technique often involves the use of algorithms that can be easily adapted to improve the quality of data.

Classification in data mining is a common technique that separates data points into different classes. It allows you to organize data sets of all sorts, including complex and large datasets as well as small and simple ones.
There are multiple types of classification algorithms, each with its unique functionality and application. All of those algorithms are used to extract data from a dataset. Which application you use for a particular task depends on the goal of the task and the kind of data you need to extract.

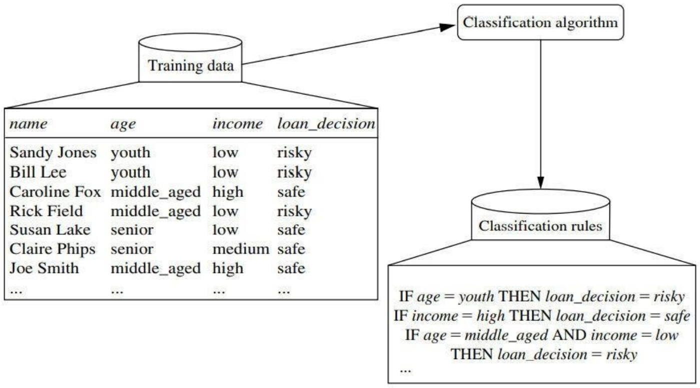
Following are the examples of cases where the data analysis task is Classification –

- A bank loan officer wants to analyze the data in order to know which customer (loan applicant) are risky or which are safe.
- A marketing manager at a company needs to analyze a customer with a given profile, who will buy a new computer.

In both of the above examples, a model or classifier is constructed to predict the categorical labels. These labels are risky or safe for loan application data and yes or no for marketing data.

It is a two-step process such as:

1. **Learning Step (Training Phase)**: Construction of Classification Model
   Different Algorithms are used to build a classifier by making the model learn using the training set available. The model has to be trained for the prediction of accurate results.

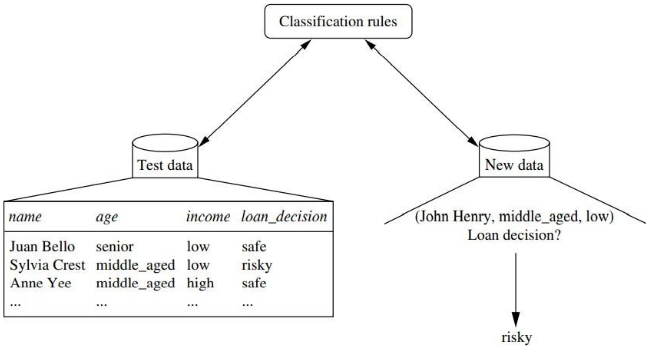


name	*age*	*income*	*loan_decision*
Sandy Jones	youth	low	risky
Bill Lee	youth	low	risky
Caroline Fox	middle_aged	high	safe
Rick Field	middle_aged	low	risky
Susan Lake	senior	low	safe
Claire Phips	senior	medium	safe
Joe Smith	middle_aged	high	safe
...	...	...	...

IF *age* = *youth* THEN *loan_decision* = *risky*
IF *income* = *high* THEN *loan_decision* = *safe*
IF *age* = *middle_aged* AND *income* = *low*
THEN *loan_decision* = *risky*
...

Learning: Training data are analyzed by a classification algorithm. Here, the class label attribute is loan decision, and the learned model or classifier is represented in the form of classification rules.

2. **Classification Step**: Model used to predict class labels and testing the constructed model on test data and hence estimate the accuracy of the classification rules.



name	*age*	*income*	*loan_decision*
Juan Bello	senior	low	safe
Sylvia Crest	middle_aged	low	risky
Anne Yee	middle_aged	high	safe
...	...	...	...

Classification: Test data are used to estimate the accuracy of the classification rules. If the accuracy is considered acceptable, the rules can be applied to the classification of new data tuples.

Difference between Classification and Clustering

Classification	**Clustering**
Classification is a supervised learning approach where a specific label is provided to the machine to classify new observations. Here the machine needs proper testing and training for the label verification.	Clustering is an unsupervised learning approach where grouping is done on similarities basis.
Supervised learning approach.	Unsupervised learning approach.
It uses a training dataset.	It does not use a training dataset.
It uses algorithms to categorize the new data as per the observations of the training set.	It uses statistical concepts in which the data set is divided into subsets with the same features.
In classification, there are labels for training data.	In clustering, there are no labels for training data.
Its objective is to find which class a new object belongs to form the set of predefined classes.	Its objective is to group a set of objects to find whether there is any relationship between them.
It is more complex as compared to clustering.	It is less complex as compared to clustering.

Bayesian Classification

Bayesian classification is based on Bayes' Theorem. Bayesian classifiers are the statistical classifiers. Bayesian classifiers can predict class membership probabilities such as the probability that a given tuple belongs to a particular class.

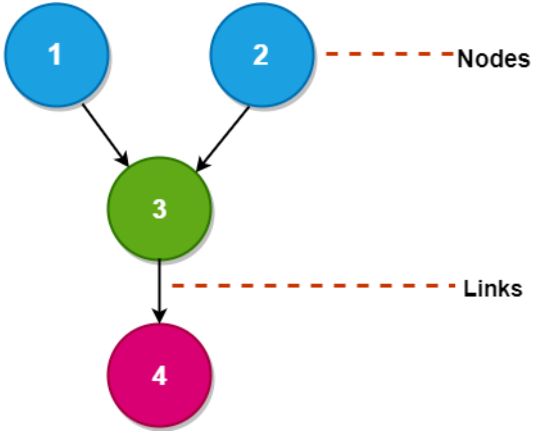
There are two components that define a Bayesian Belief Network –

- Directed acyclic graph
- A set of conditional probability table

Bayesian Network

A Bayesian Network falls under the classification of Probabilistic Graphical Modelling (PGM) procedure that is utilized to compute uncertainties by utilizing the probability concept. Generally known as **Belief Networks, Bayesian Networks** are used to show uncertainties using **Directed Acyclic Graphs** (DAG)

A **Directed Acyclic Graph** is used to show a Bayesian Network, and like some other statistical graph, a DAG consists of a set of nodes and links, where the links signify the connection between the nodes.



A DAG models the uncertainty of an event taking place based on the Conditional Probability Distribution (CDP) of each random variable. A Conditional Probability Table (CPT) is used to represent the CPD of each variable in a network.

Bayes theorem came into existence after Thomas Bayes, who first utilized conditional probability to provide an algorithm that uses evidence to calculate limits on an unknown parameter.

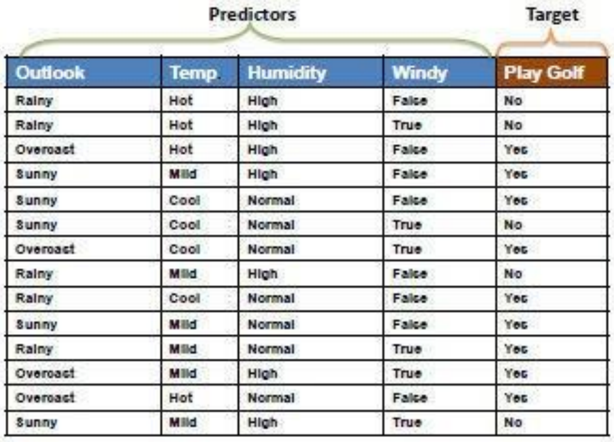
$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

Bayes' Theorem finds the probability of an event occurring given the probability of another event that has already occurred. Bayes' theorem is stated mathematically as the following equation:
where A and B are events and P(B) ≠ 0.

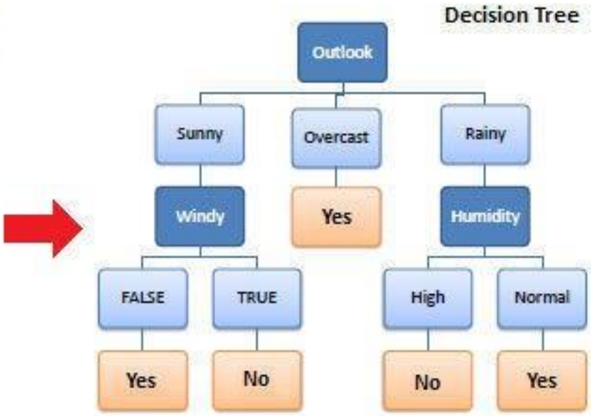
- Basically, we are trying to find probability of event A, given the event B is true. Event B is also termed as evidence.
- P(A) is the priori of A (the prior probability, i.e., Probability of event before evidence is seen). The evidence is an attribute value of an unknown instance(here, it is event B).
- P(A|B) is a posteriori probability of B, i.e., probability of event after evidence is seen.

Decision Tree Classification

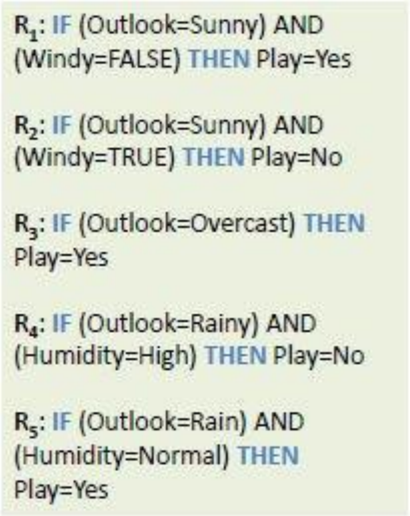
Decision tree builds classification or regression models in the form of a tree structure. It breaks down a dataset into smaller and smaller subsets while at the same time an associated decision tree is incrementally developed. The final result is a tree with decision nodes and leaf nodes. A decision node (e.g., Outlook) has two or more branches (e.g., Sunny, Overcast and Rainy). Leaf node (e.g., Play) represents a classification or decision. The topmost decision node in a tree which corresponds to the best predictor called root node. Decision trees can handle both categorical and numerical data.
Decision Tree consists of:
Nodes: Test for the value of a certain attribute.
Edges/ Branch: Correspond to the outcome of a test and connect to the next node or leaf.
Leaf nodes: Terminal nodes that predict the outcome (represent class labels or class distribution).



Predictors				Target
Outlook	Temp	Humidity	Windy	Play Golf
Rainy	Hot	High	False	No
Rainy	Hot	High	True	No
Overcast	Hot	High	False	Yes
Sunny	Mild	High	False	Yes
Sunny	Cool	Normal	False	Yes
Sunny	Cool	Normal	True	No
Overcast	Cool	Normal	True	Yes
Rainy	Mild	High	False	No
Rainy	Cool	Normal	False	Yes
Sunny	Mild	Normal	False	Yes
Rainy	Mild	Normal	True	Yes
Overcast	Mild	High	True	Yes
Overcast	Hot	Normal	False	Yes
Sunny	Mild	High	True	No



A decision tree can easily be transformed to a set of rules by mapping from the root node to the leaf nodes one by one.



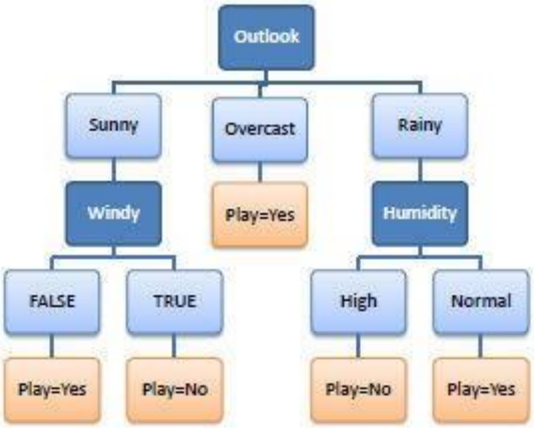
R_1: IF (Outlook=Sunny) AND (Windy=FALSE) THEN Play=Yes

R_2: IF (Outlook=Sunny) AND (Windy=TRUE) THEN Play=No

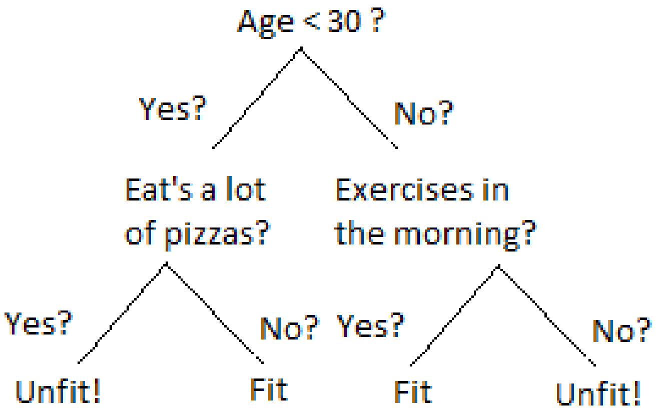
R_3: IF (Outlook=Overcast) THEN Play=Yes

R_4: IF (Outlook=Rainy) AND (Humidity=High) THEN Play=No

R_5: IF (Outlook=Rain) AND (Humidity=Normal) THEN Play=Yes



Is a Person Fit?



Advantages of Classification with Decision Trees:

1. Inexpensive to construct.
2. Extremely fast at classifying unknown records.
3. Easy to interpret for small-sized trees
4. Accuracy comparable to other classification techniques for many simple data sets.
5. Excludes unimportant features.

Disadvantages of Classification with Decision Trees:

1. Easy to overfit.
2. Decision Boundary restricted to being parallel to attribute axes.
3. Decision tree models are often biased toward splits on features having a large number of levels.
4. Small changes in the training data can result in large changes to decision logic.
5. Large trees can be difficult to interpret and the decisions they make may seem counter intuitive.

Applications of Decision trees in real life:

1. Biomedical Engineering (decision trees for identifying features to be used in implantable devices).
2. Financial analysis (Customer Satisfaction with a product or service).
3. Astronomy (classify galaxies).
4. System Control.
5. Manufacturing and Production (Quality control, Semiconductor manufacturing, etc).
6. Medicines (diagnosis, cardiology, psychiatry).
7. Physics (Particle detection).

Concept of Entropy

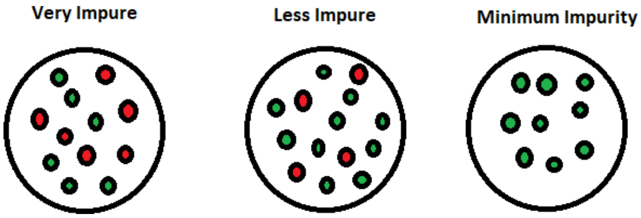
Definition: Entropy is the measures of impurity, disorder or uncertainty in a bunch of examples.

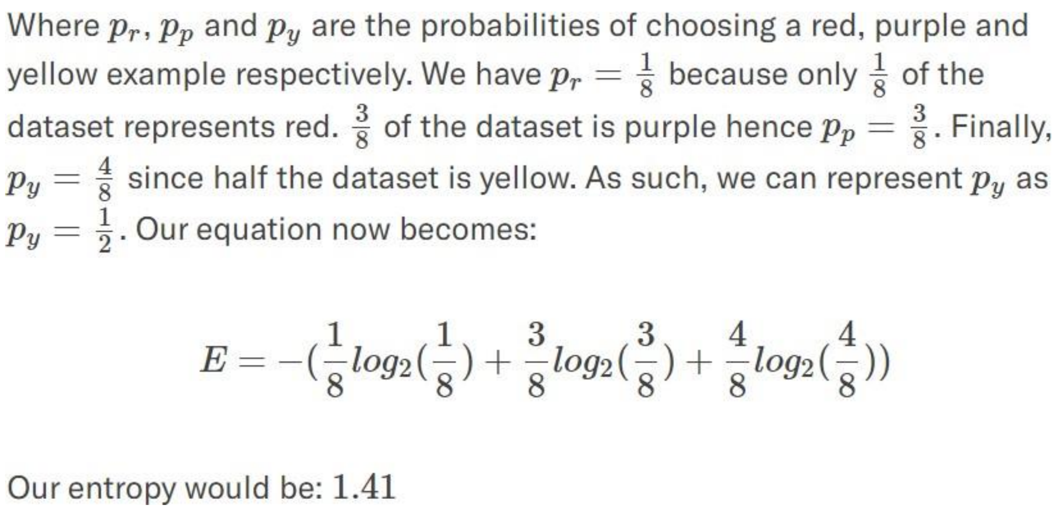
Entropy controls how a Decision Tree decides to split the data. It actually effects how a Decision Tree draws its boundaries.

$$Entropy = -\sum p(X)\log p(X)$$

here p(x) is a fraction of examples in a given class

Entropy is an information theory metric that measures the impurity or uncertainty in a group of observations. It determines how a decision tree chooses to split data. The image below gives a better description of the purity of a set.





Where p_r, p_p and p_y are the probabilities of choosing a red, purple and yellow example respectively. We have $p_r = \frac{1}{8}$ because only $\frac{1}{8}$ of the dataset represents red. $\frac{3}{8}$ of the dataset is purple hence $p_p = \frac{3}{8}$. Finally, $p_y = \frac{4}{8}$ since half the dataset is yellow. As such, we can represent p_y as $p_y = \frac{1}{2}$. Our equation now becomes:

$$E = -(\frac{1}{8}log_2(\frac{1}{8}) + \frac{3}{8}log_2(\frac{3}{8}) + \frac{4}{8}log_2(\frac{4}{8}))$$

Our entropy would be: 1.41

Evaluating Classification Algorithms

It is important to understand both what a classification metric expresses and what it hides.
There are so many performances evaluation measures when it comes to selecting a classification model.
What are the Performance Evaluation Measures for Classification Models?

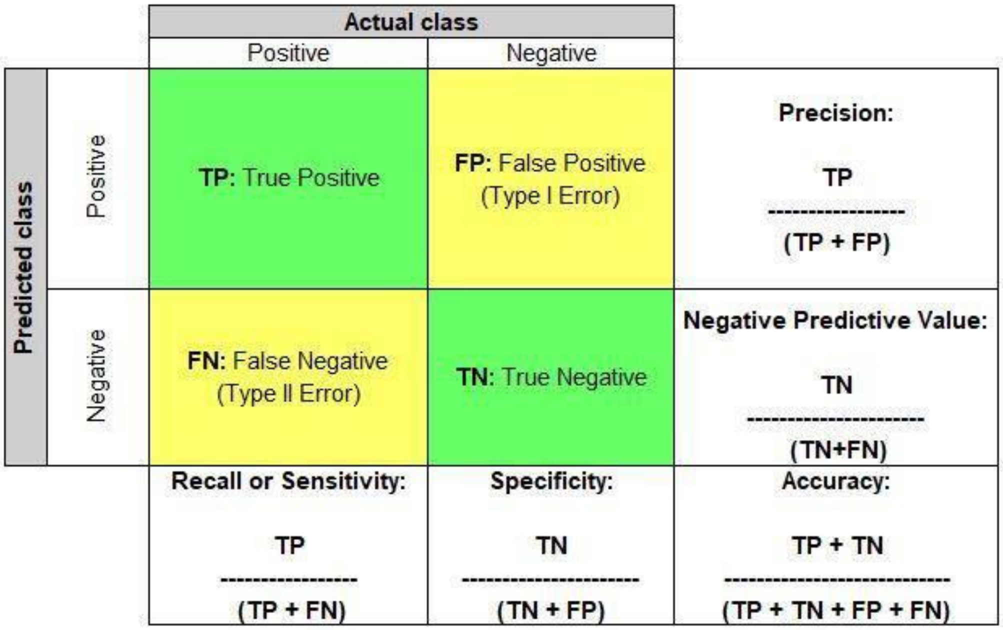
- Confusion Matrix
  Confusion Matrix usually causes a lot of confusion even in those who are using them regularly. Terms used in defining a confusion matrix are TP, TN, FP, and FN.

- Precision
  Out of all that were marked as positive, how many are actually truly positive.

- Recall/ Sensitivity
  Out of all the actual real positive cases, how many were identified as positive.

- Specificity
  Out of all the real negative cases, how many were identified as negative.

- F1-Score
  F1 score is a weighted average of Precision and Recall, which means there is equal importance given to FP and FN.



		Actual class		
		Positive	Negative	
Predicted class	Positive	**TP:** True Positive	**FP:** False Positive (Type I Error)	**Precision:** **TP** **-----------------** **(TP + FP)**
	Negative	**FN:** False Negative (Type II Error)	**TN:** True Negative	**Negative Predictive Value:** **TN** **----------------------** **(TN+FN)**
		Recall or Sensitivity: **TP** **-----------------** **(TP + FN)**	**Specificity:** **TN** **----------------------** **(TN + FP)**	**Accuracy:** **TP + TN** **-----------------------------** **(TP + TN + FP + FN)**

- Accuracy: This term tells us how many right classifications were made out of all the classifications.

- AUC and ROC curve: Area Under Curve and Receiver and Operating Characteristics

Linear Regression

Linear regression attempts to model the relationship between two variables by fitting a linear equation (= a straight line) to the observed data.

What linear regression does is simply tell us the value of the dependent variable for an arbitrary independent/explanatory variable. e.g., Twitter revenues based on number of Twitter users.

From a machine learning context, it is the simplest model one can try out on your data. If you have a hunch that the data follows a straight-line trend, linear regression can give you quick and reasonably accurate results.

Linear regression models use a straight line, while logistic and nonlinear regression models use a curved line.

Simple linear regression is used to estimate the relationship between two quantitative variables. You can use simple linear regression when you want to know:

1. How strong the relationship is between two variables (e.g., the relationship between rainfall and soil erosion).
2. The value of the dependent variable at a certain value of the independent variable (e.g., the amount of soil erosion at a certain level of rainfall).

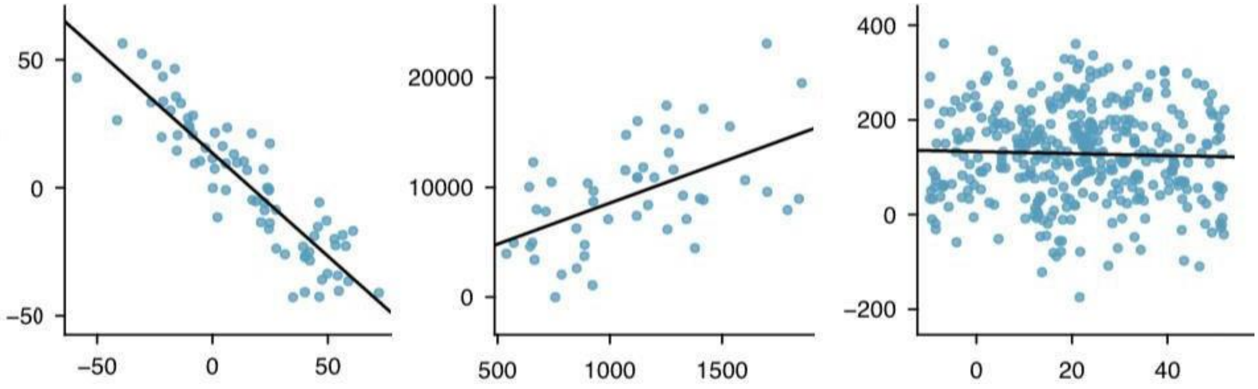


Figure: Three data sets where a linear model may be useful even though the data do not all fall exactly on the line.

Table: Example data.

X	Y
1.00	1.00
2.00	2.00
3.00	1.30
4.00	3.75
5.00	2.25

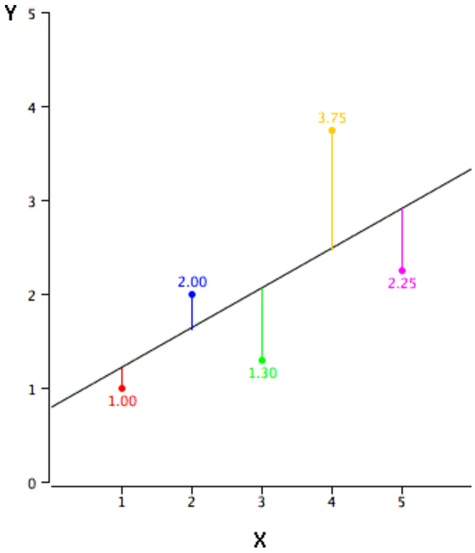


Figure 2. A scatter plot of the example data.

Linear regression at its core is a method to find values for parameters that represent a line.

The equation Y=mX+C

Linear regression consists of finding the best-fitting straight line through the points. The best-fitting line is called a regression line. The black diagonal line in Figure 2 is the regression line and consists of the predicted score on Y for each possible value of X. The vertical lines from the points to the regression line represent the errors of prediction. As you can see, the red point is very near the regression line; its error of prediction is small. By contrast, the yellow point is much higher than the regression line and therefore its error of prediction is large.
In terms of coordinate geometry if dependent variable is called Y and independent variable is called X then a straight line can be represented as Y = m*X+c. Where m and c are two numbers that linear regression tries to figure out to estimate that white line.

Regression is a procedure that lets us predict a continuous target variable with the help of one or more explanatory variables.

Concept of Non-Linear Regression

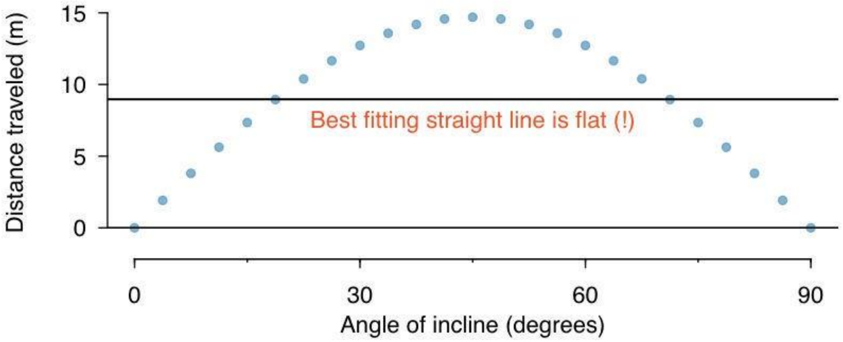
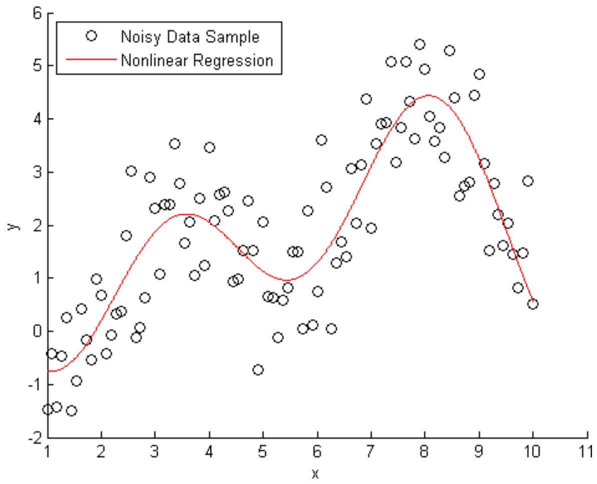
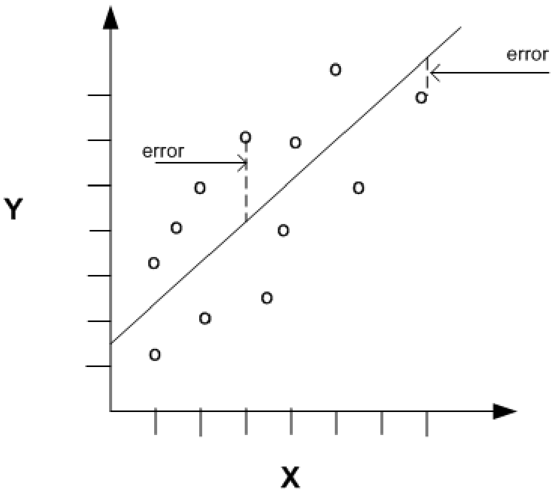


Figure: A linear model is not useful in this nonlinear case. These data are from an introductory physics experiment.

- Both linear and nonlinear regression predict Y responses from an X variable (or variables).
- Nonlinear regression is a curved function of an X variable (or variables) that is used to predict a Y variable
- Nonlinear regression can show a prediction of population growth over time.

➢ Now, we'll focus on the "non" in nonlinear! If a regression equation doesn't follow the rules for a linear model, then it must be a nonlinear model. It's that simple! A nonlinear model is literally not linear.





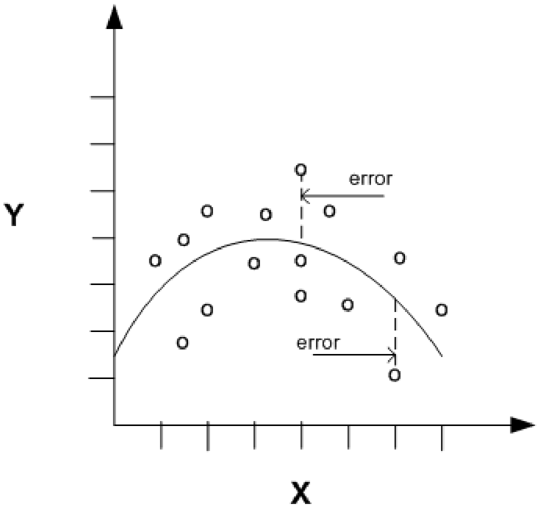


Figure: Linear Regression with a Single Predictor   Figure: Nonlinear Regression with a Single Predictor

Nonlinear regression uses nonlinear regression equations, which take the form:

Y = f (X, β) + ε

Where:

X = a vector of p predictors,

β = a vector of k parameters,

f () = a known regression function,

ε = an error term.

Association Rule Mining

Association rule mining is a procedure which is meant to find frequent patterns, correlations, associations, or causal structures from data sets found in various kinds of databases such as relational databases, transactional databases, and other forms of data repositories.

Given a set of transactions, association rule mining aims to find the rules which enable us to predict the occurrence of a specific item based on the occurrences of the other items in the transaction.

> iurpriiingly, diaperi and beer are bought together becauie, ai it turni out, that dadi are often taiked to do the ihopping while the momi are left with the baby.

> "If a customer buys bread, he's 70% likely of buying milk."

In the above association rule, bread is the antecedent and milk is the consequent.

- Association rules are created by thoroughly analyzing data and looking for frequent if/then patterns.

The association rule learning is one of the very important concepts of machine learning, and it is employed in **Market Basket analysis, Web usage mining, continuous production, etc.** Here market basket analysis is a technique used by the various big retailer to discover the associations between items. We can understand it by taking an example of a supermarket, as in a supermarket, all products that are purchased together are put together.
For example, if a customer buys bread, he most likely can also buy butter, eggs, or milk, so these products are stored within a shelf or mostly nearby.

Frequent Patterns

Frequent Pattern Mining is also known as the Association Rule Mining. The identification of frequent patterns is an important task in data mining.
In Data Mining, Frequent Pattern Mining is a major concern because it is playing a major role in Associations and Correlations.
Frequent Pattern is a pattern which appears frequently in a data set. By identifying frequent patterns, we can observe strongly correlated items together and easily identify similar characteristics, associations among them. By doing frequent pattern mining, it leads to further analysis like clustering, classification and other data mining tasks.

- Frequent patterns are item sets, subsequences, or substructures that appear in a data set with frequency no less than a user-specified threshold.
- For example, a set of items, such as milk and bread, that appear frequently together in a transaction data set, is a frequent itemset.
- A subsequence, such as buying first a PC, then a digital camera, and then a memory card, if it occurs frequently in a shopping history database, is a (frequent) sequential pattern.
- When a series of transactions are given, pattern mining's main motive is to find the rules that enable us to speculate a certain item based on the happening of other items in the transaction.

Association Rule

Association rule mining finds interesting associations and relationships among large sets of data items. This rule shows how frequently an itemset occurs in a transaction. A typical example is Market Based Analysis.
Given a set of transactions, we can find rules that will predict the occurrence of an item based on the occurrences of other items in the transaction.

An implication expression of the form X -> Y, where X and Y are any 2 item sets.

TID	Items
1	Bread, Milk
2	Bread, Diaper, Beer, Eggs
3	Milk, Diaper, Beer, Coke
4	Bread, Milk, Diaper, Beer
5	Bread, Milk, Diaper, Coke

Example: {Milk, Diaper}->{Beer}

Support and Confidence

Depending on the following two parameters, the important relationships are observed:

Support: Support indicates how frequently the if/then relationship appears in the database.
Support is the frequency of A or how frequently an item appears in the dataset. It is defined as the fraction of the transaction T that contains the itemset X. If there are X datasets, then for transactions T, it can be written as:

$$Support(\{X\} \rightarrow \{Y\}) = \frac{Transactions\ containing\ both\ X\ and\ Y}{Total\ number\ of\ transactions}$$

Confidence: Confidence talks about the number of times these relationships have been found to be true.
Confidence indicates how often the rule has been found to be true. Or how often the items X and Y occur together in the dataset when the occurrence of X is already given. It is the ratio of the transaction that contains X and Y to the number of records that contain X.

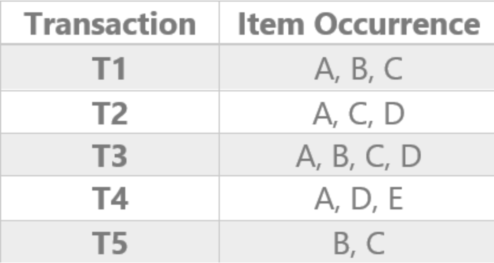
$$Confidence(\{X\} \rightarrow \{Y\}) = \frac{Transactions\ containing\ both\ X\ and\ Y}{Transactions\ containing\ X}$$

Lift

It is the strength of any rule, which can be defined as below formula:

$$Lift(\{X\} \rightarrow \{Y\}) = \frac{(Transactions\ containing\ both\ X\ and\ Y)/(Transactions\ containing\ X)}{Fraction\ of\ transactions\ containing\ Y}$$

- ✓ If **Lift= 1**: The probability of occurrence of antecedent and consequent is independent of each other.
- ✓ **Lift>1**: It determines the degree to which the two item sets are dependent to each other.
- ✓ **Lift<1**: It tells us that one item is a substitute for other items, which means one item has a negative effect on another.



Transaction	Item Occurrence
T1	A, B, C
T2	A, C, D
T3	A, B, C, D
T4	A, D, E
T5	B, C

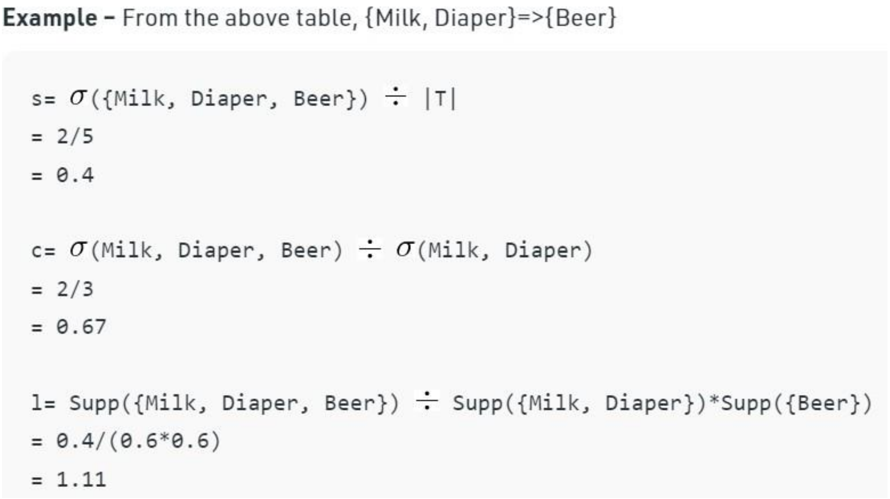
Support = Frequency (A, B) / N

Confidence = Frequency (A, B) / Frequency (A)

- ✓ Example: One of possible Association Rule is A → D
- ✓ Total no of Transactions(N) = 5
- ✓ Frequency (A, D) = > Total no of instances together A with D is 3
- ✓ Frequency(A) => Total no of occurrence in A
- ✓ Support = 3 / 5
- ✓ Confidence = 3 / 4

Example 2

TID	Items
1	Bread, Milk
2	Bread, Diaper, Beer, Eggs
3	Milk, Diaper, Beer, Coke
4	Bread, Milk, Diaper, Beer
5	Bread, Milk, Diaper, Coke



Example – From the above table, {Milk, Diaper}=>{Beer}

```
s= σ({Milk, Diaper, Beer}) ÷ |T|
= 2/5
= 0.4

c= σ(Milk, Diaper, Beer) ÷ σ(Milk, Diaper)
= 2/3
= 0.67

l= Supp({Milk, Diaper, Beer}) ÷ Supp({Milk, Diaper})*Supp({Beer})
= 0.4/(0.6*0.6)
= 1.11
```

Support and Confidence can be represented by the following example:

Bread=> butter [support=2%, confidence-60%]

The above statement is an example of an association rule. This means that there is a 2% transaction that bought bread and butter together and there are 60% of customers who bought bread as well as butter.

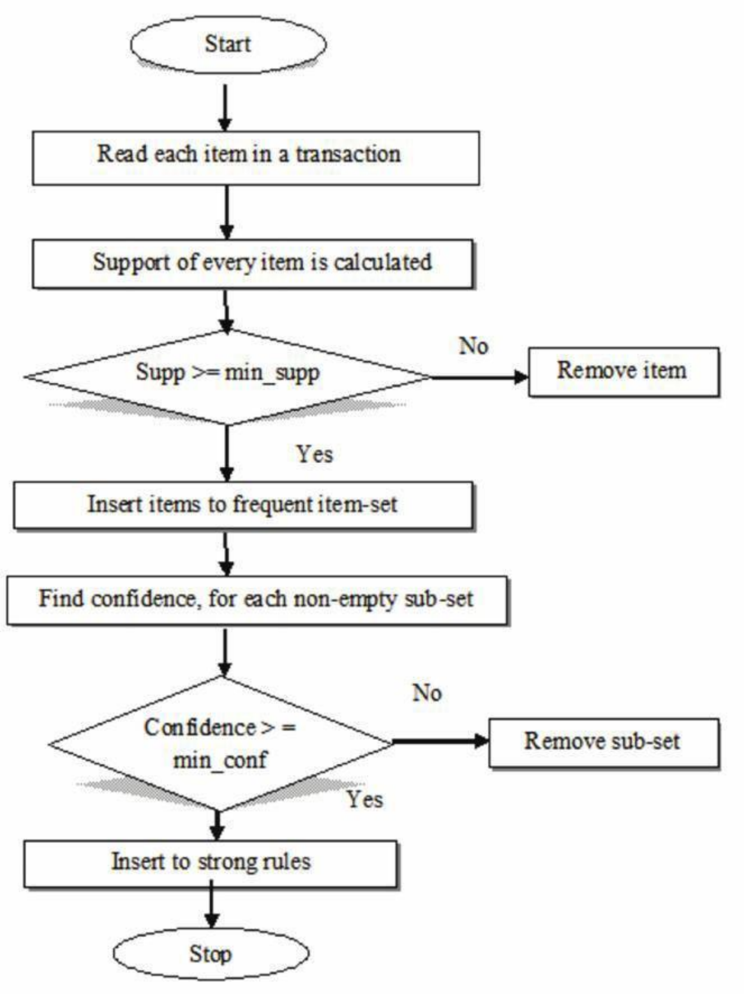
The main applications of association rule mining:

- **Basket data analysis-** is to analyze the association of purchased items in a single basket or single purchase as per the examples given above.
- **Cross marketing-** is to work with other businesses that complement your own, not competitors. For example, vehicle dealerships and manufacturers have cross marketing campaigns with oil and gas companies for obvious reasons.
- **Catalog design-** the selection of items in a business' catalog are often designed to complement each other so that buying one item will lead to buying of another. So, these items are often complements or very related.

Apriori Algorithm

Apriori algorithm was the first algorithm that was proposed for frequent itemset mining. This algorithm uses frequent datasets to generate association rules. It is designed to work on the databases that contain transactions. This algorithm uses a breadth-first search and Hash Tree to calculate the itemset efficiently. It is an iterative approach to discover the most frequent item sets.

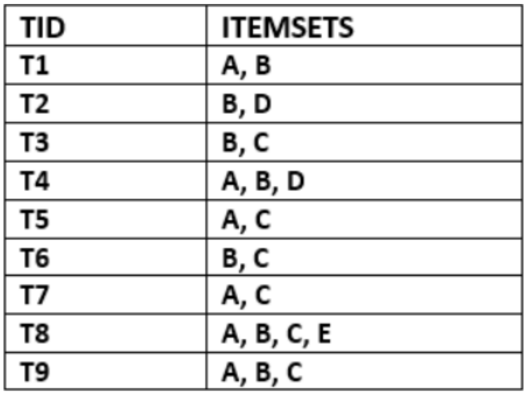
Steps for Apriori Algorithm


Start
Read each item in a transaction
Support of every item is calculated
Supp >= min_supp
No
Remove item
Yes
Insert items to frequent item-set
Find confidence, for each non-empty sub-set
Confidence > =
min_conf
No
Remove sub-set
Yes
Insert to strong rules
Stop

Below are the steps for the apriori algorithm:

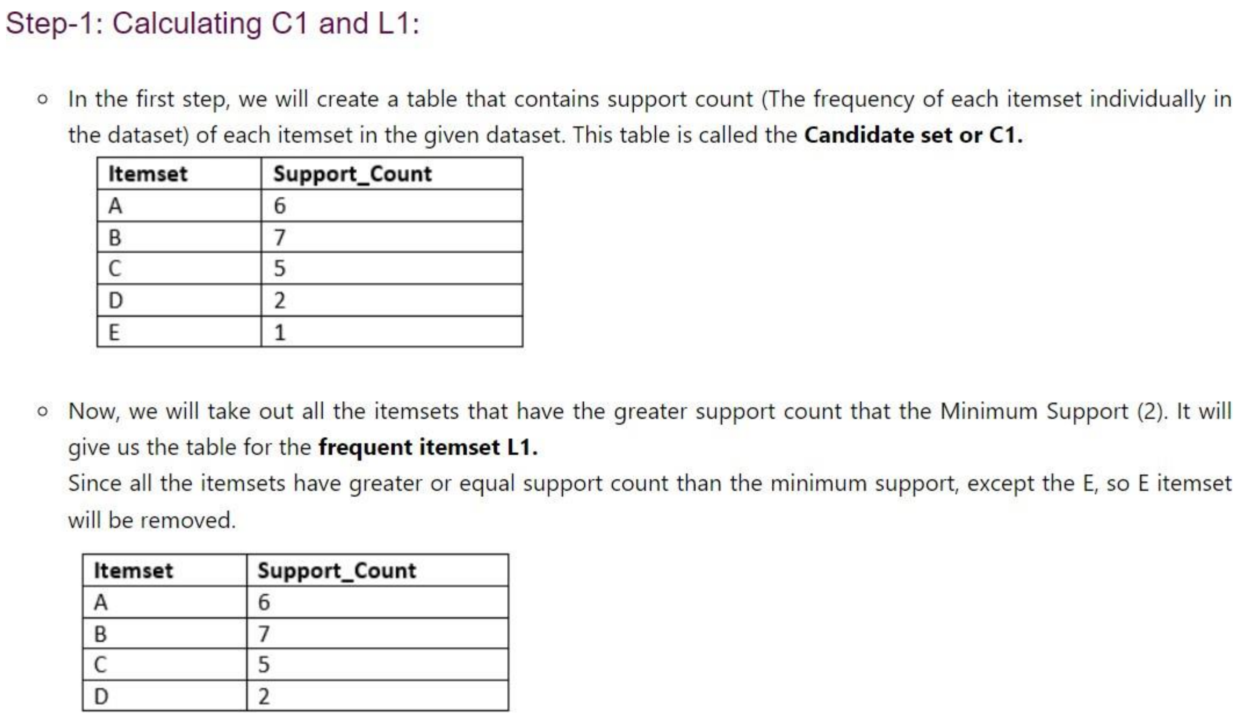
Step-1: Determine the support of item sets in the transactional database, and select the minimum support and confidence.
Step-2: Take all supports in the transaction with higher support value than the minimum or selected support value.
Step-3: Find all the rules of these subsets that have higher confidence value than the threshold or minimum confidence.
Step-4: Sort the rules as the decreasing order of lift.

Suppose we have the following dataset that has various transactions, and from this dataset, we need to find the frequent item sets and generate the association rules using the Apriori algorithm:



TID	ITEMSETS
T1	A, B
T2	B, D
T3	B, C
T4	A, B, D
T5	A, C
T6	B, C
T7	A, C
T8	A, B, C, E
T9	A, B, C

Given: Minimum support =2; minimum confidence = 50%



Step-1: Calculating C1 and L1:

- In the first step, we will create a table that contains support count (The frequency of each itemset individually in the dataset) of each itemset in the given dataset. This table is called the **Candidate set or C1.**

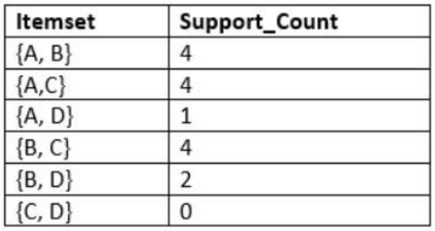
Itemset	Support_Count
A	6
B	7
C	5
D	2
E	1

- Now, we will take out all the itemsets that have the greater support count that the Minimum Support (2). It will give us the table for the **frequent itemset L1.**
  Since all the itemsets have greater or equal support count than the minimum support, except the E, so E itemset will be removed.

Itemset	Support_Count
A	6
B	7
C	5
D	2

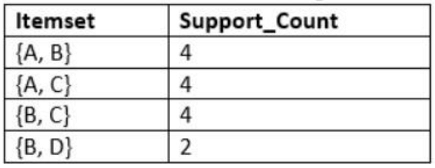
Step-2: Candidate Generation C2, and L2:

- In this step, we will generate C2 with the help of L1. In C2, we will create the pair of the itemsets of L1 in the form of subsets.
- After creating the subsets, we will again find the support count from the main transaction table of datasets, i.e., how many times these pairs have occurred together in the given dataset. So, we will get the below table for C2:



Itemset	Support_Count
{A, B}	4
{A,C}	4
{A, D}	1
{B, C}	4
{B, D}	2
{C, D}	0

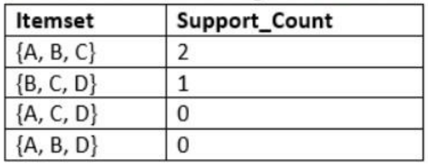
- Again, we need to compare the C2 Support count with the minimum support count, and after comparing, the itemset with less support count will be eliminated from the table C2. It will give us the below table for L2



Itemset	Support_Count
{A, B}	4
{A, C}	4
{B, C}	4
{B, D}	2

Step-3: Candidate generation C3, and L3:

- For C3, we will repeat the same two processes, but now we will form the C3 table with subsets of three itemsets together, and will calculate the support count from the dataset. It will give the below table:



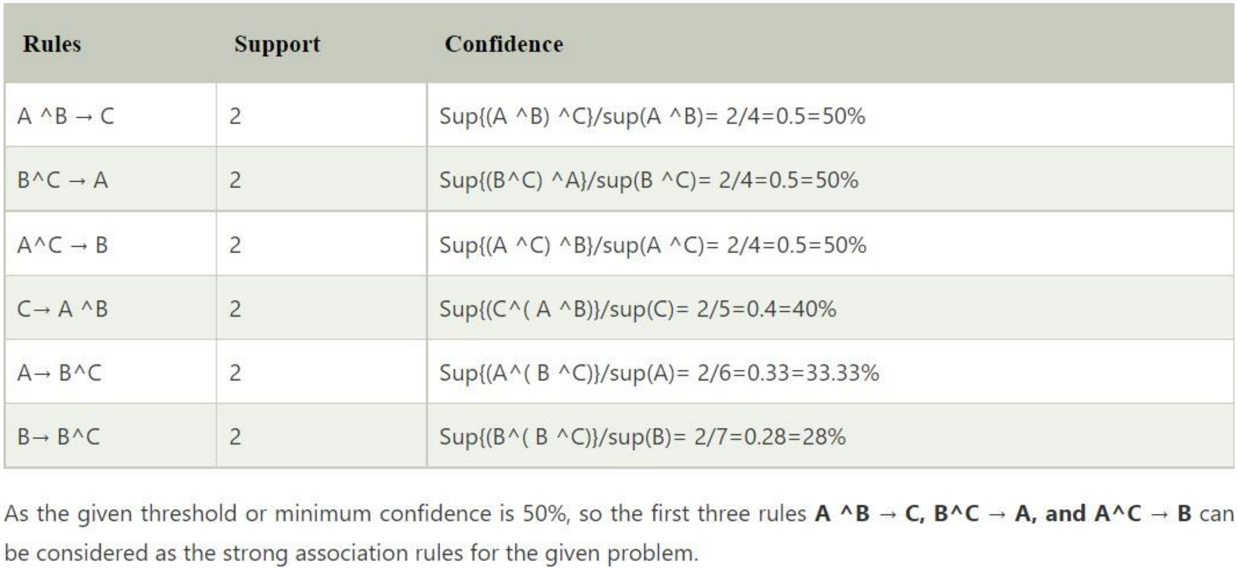
Itemset	Support_Count
{A, B, C}	2
{B, C, D}	1
{A, C, D}	0
{A, B, D}	0

- Now we will create the L3 table. As we can see from the above C3 table, there is only one combination of itemset that has support count equal to the minimum support count. So, the L3 will have only one combination, i.e., **{A, B, C}.**

Step-4: Finding the association rules for the subsets:

To generate the association rules, first, we will create a new table with the possible rules from the occurred combination {A, B.C}. For all the rules, we will calculate the Confidence using formula **sup(A ^B)/A.** After calculating the confidence value for all rules, we will exclude the rules that have less confidence than the minimum threshold(50%).

Consider the below table:



Rules	Support	Confidence
A ^B → C	2	Sup{(A ^B) ^C}/sup(A ^B)= 2/4=0.5=50%
B^C → A	2	Sup{(B^C) ^A}/sup(B ^C)= 2/4=0.5=50%
A^C → B	2	Sup{(A ^C) ^B}/sup(A ^C)= 2/4=0.5=50%
C→ A ^B	2	Sup{(C^(A ^B)}/sup(C)= 2/5=0.4=40%
A→ B^C	2	Sup{(A^(B ^C)}/sup(A)= 2/6=0.33=33.33%
B→ B^C	2	Sup{(B^(B ^C)}/sup(B)= 2/7=0.28=28%

As the given threshold or minimum confidence is 50%, so the first three rules **A ^B → C, B^C → A, and A^C → B** can be considered as the strong association rules for the given problem.

Advantages of Apriori Algorithm

✓ This is easy to understand algorithm

✓ The join and prune steps of the algorithm can be easily implemented on large datasets.

Disadvantages of Apriori Algorithm

✓ The apriori algorithm works slow compared to other algorithms.

✓ The overall performance can be reduced as it scans the database for multiple times.

✓ The time complexity and space complexity of the apriori algorithm is O(2D), which is very high. Here D represents the horizontal width present in the database.

Shortcomings Of Apriori Algorithm

- Using Apriori needs a generation of candidate itemset. These item sets may be large in number if the itemset in the database is huge.
- Apriori needs multiple scans of the database to check the support of each itemset generated and this leads to high costs.
- These shortcomings can be overcome using the FP growth algorithm.

FP-Growth Algorithm

This algorithm is an improvement to the Apriori method. A frequent pattern is generated without the need for candidate generation. FP growth algorithm represents the database in the form of a tree called a frequent pattern tree or FP tree.

This tree structure will maintain the association between the item sets. The database is fragmented using one frequent item. This fragmented part is called "pattern fragment". The item sets of these fragmented patterns are analyzed. Thus, with this method, the search for frequent item sets is reduced comparatively.

FP Tree

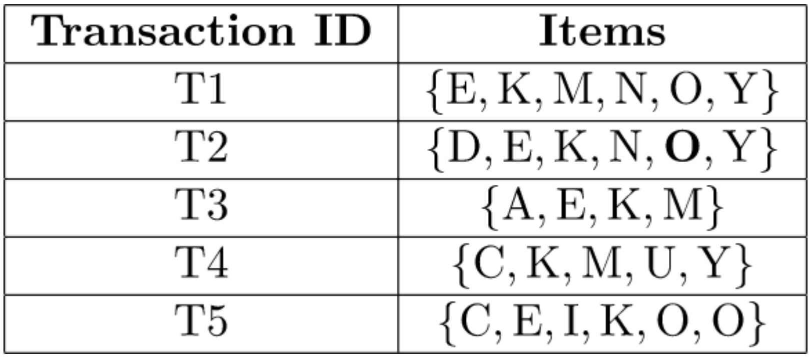
- ✓ Frequent Pattern Tree is a tree-like structure that is made with the initial item sets of the database. The purpose of the FP tree is to mine the most frequent pattern. Each node of the FP tree represents an item of the itemset.
- ✓ The root node represents null while the lower nodes represent the item sets. The association of the nodes with the lower nodes that is the item sets with the other item sets are maintained while forming the tree.

FP Growth vs Apriori

FP Growth	Apriori
Pattern Generation	
FP growth generates pattern by constructing a FP tree	Apriori generates pattern by pairing the items into singletons, pairs and triplets.
Candidate Generation	
There is no candidate generation	Apriori uses candidate generation
Process	
The process is faster as compared to Apriori. The runtime of process increases linearly with increase in number of item sets.	The process is comparatively slower than FP Growth, the runtime increases exponentially with increase in number of item sets
Memory Usage	
A compact version of database is saved	The candidates' combinations are saved in memory

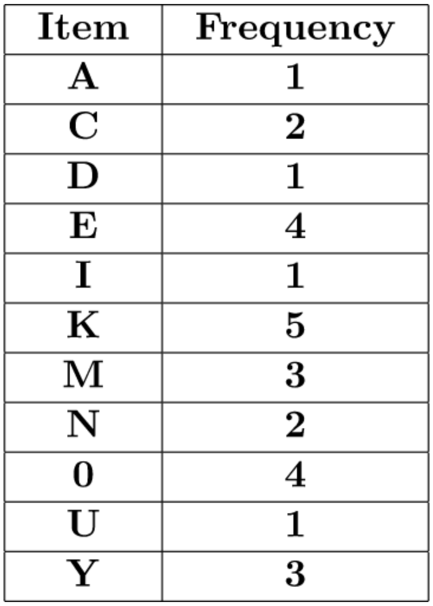
FP-tree Pseudocode and Explanation

Step 1: Deduce the ordered frequent items. For items with the same frequency, the order is given by the alphabetical order.
Step 2: Construct the FP-tree from the above data
Step 3: From the FP-tree above, construct the FP-conditional tree for each item (or itemset).
Step 4: Determine the frequent patterns.



Transaction ID	Items
T1	$\{E, K, M, N, O, Y\}$
T2	$\{D, E, K, N, \mathbf{O}, Y\}$
T3	$\{A, E, K, M\}$
T4	$\{C, K, M, U, Y\}$
T5	$\{C, E, I, K, O, O\}$

The above-given data is a hypothetical dataset of transactions with each letter representing an item. The frequency of each individual item is computed: -



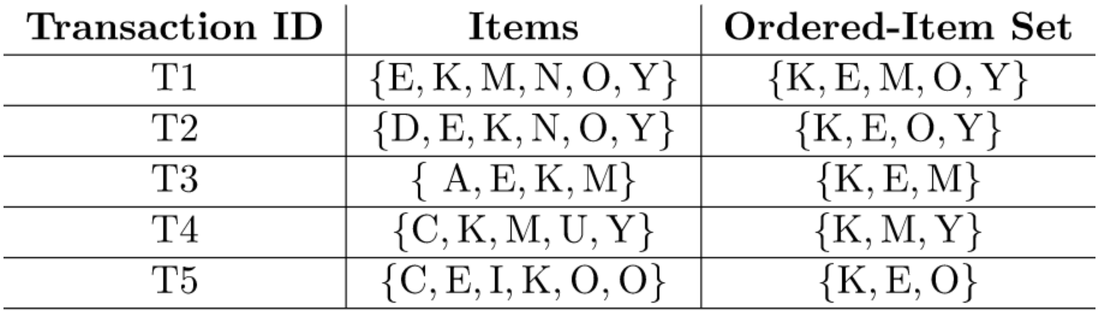
Item	Frequency
A	1
C	2
D	1
E	4
I	1
K	5
M	3
N	2
0	4
U	1
Y	3

Let the minimum support be **3.**

A **Frequent Pattern set** is built which will contain all the elements whose frequency is greater than or equal to the minimum support. These elements are stored in descending order of their respective frequencies. After insertion of the relevant items, the set L looks like this: -

L = {K: 5, E: 4, M: 3, O: 4, Y: 3}

Now, for each transaction, the respective Ordered-Item set is built.

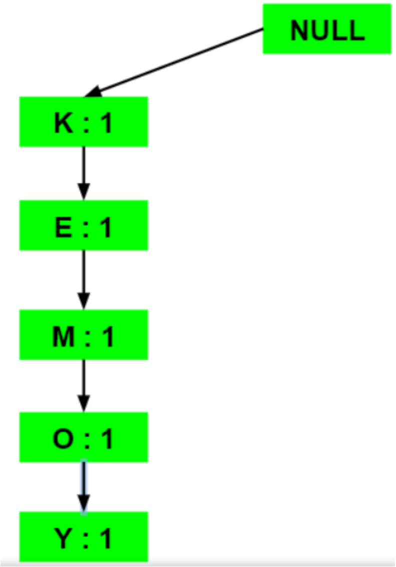


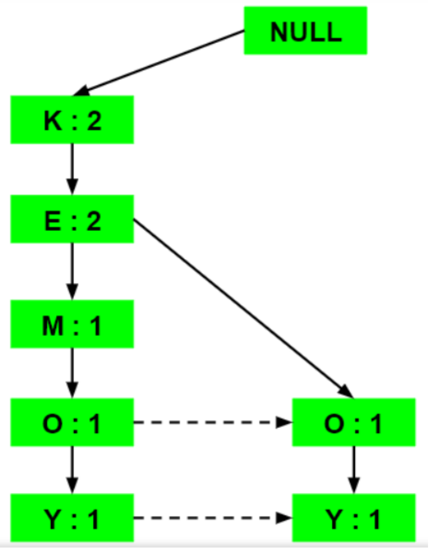
Transaction ID	Items	Ordered-Item Set
T1	{E, K, M, N, O, Y}	{K, E, M, O, Y}
T2	{D, E, K, N, O, Y}	{K, E, O, Y}
T3	{ A, E, K, M}	{K, E, M}
T4	{C, K, M, U, Y}	{K, M, Y}
T5	{C, E, I, K, O, O}	{K, E, O}

Now, all the Ordered-Item sets are inserted into a Tree Data Structure.

Inserting the set {K, E, M, O, Y}:

Here, all the items are simply linked one after the other in the order of occurrence in the set and initialize the support count for each item as 1.

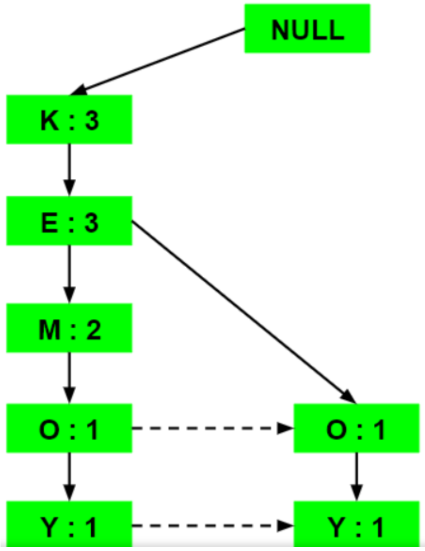


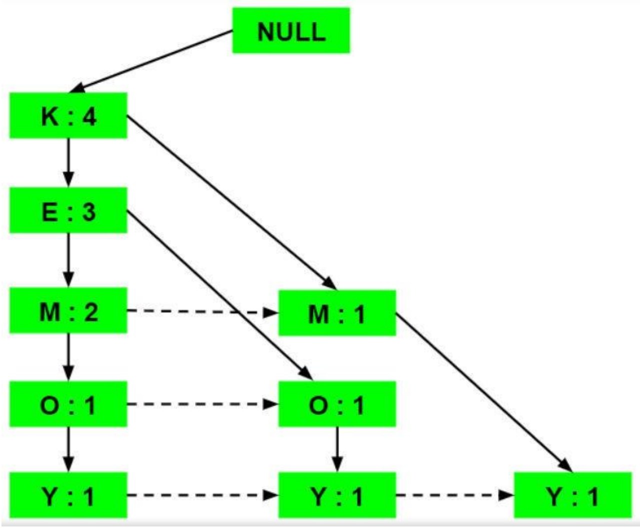


b) Inserting the set {K, E, O, Y}:

c) Inserting the set {K, E, M}:
Here simply the support count of each element is increased by 1.

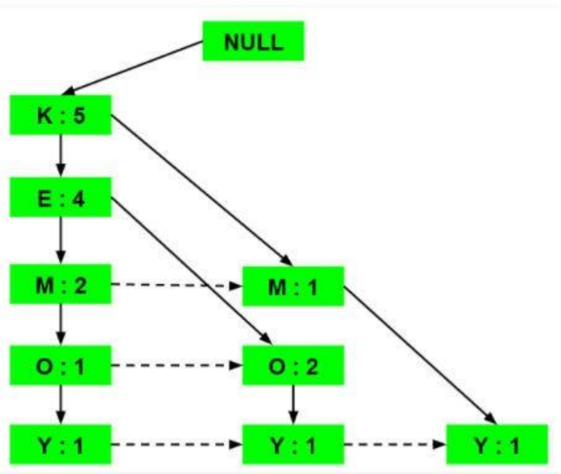
d) Inserting the set {K, M, Y}:
Similar to step b), first the support count of K is increased, then new nodes for M and Y are initialized and linked accordingly.



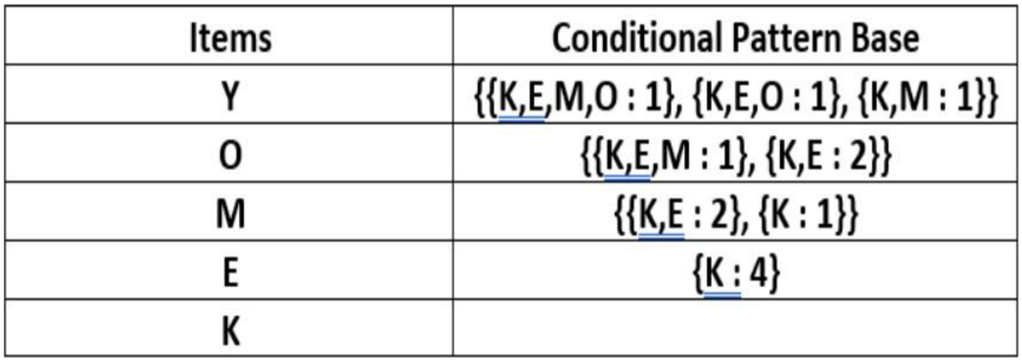


e) Inserting the set {K, E, O}:

Here simply the support counts of the respective elements are increased. Note that the support count of the new node of item O is increased.

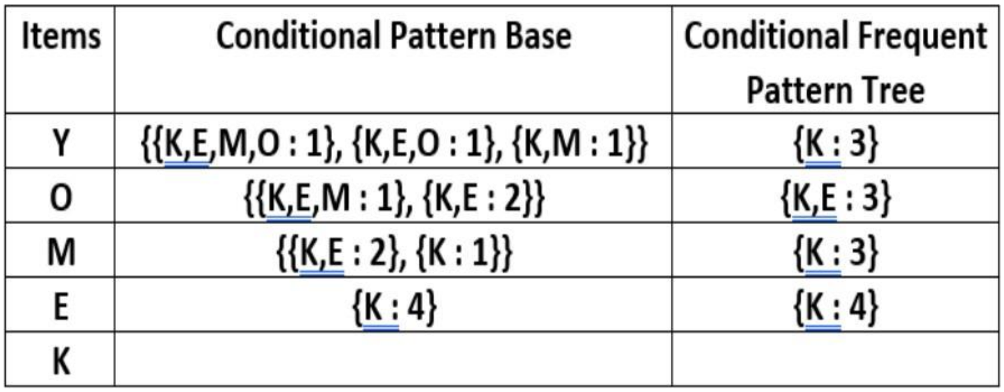


Now, for each item, the Conditional Pattern Base is computed which is path labels of all the paths which lead to any node of the given item in the frequent-pattern tree. Note that the items in the below table are arranged in the ascending order of their frequencies.



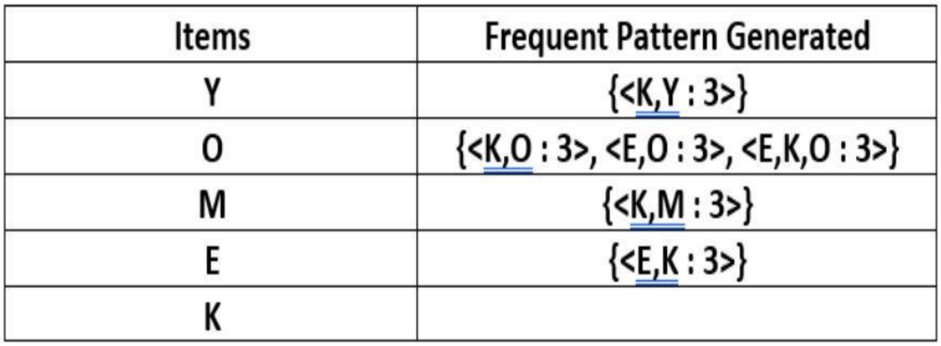
Items	Conditional Pattern Base
Y	{{K,E,M,O : 1}, {K,E,O : 1}, {K,M : 1}}
O	{{K,E,M : 1}, {K,E : 2}}
M	{{K,E : 2}, {K : 1}}
E	{K : 4}
K	

Now for each item, the Conditional Frequent Pattern Tree is built.



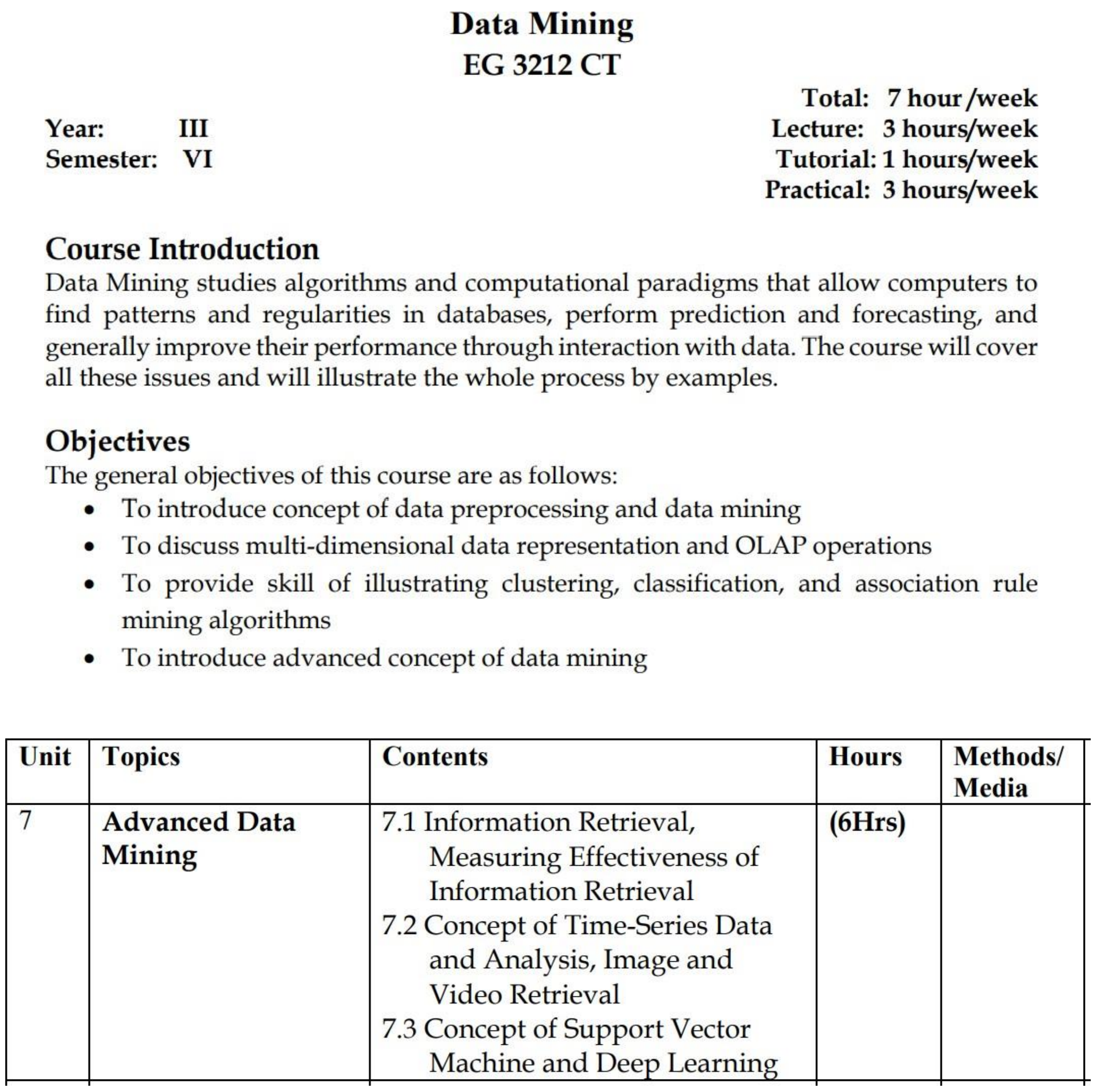
Items	Conditional Pattern Base	Conditional Frequent Pattern Tree
Y	{{K,E,M,O : 1}, {K,E,O : 1}, {K,M : 1}}	{K : 3}
O	{{K,E,M : 1}, {K,E : 2}}	{K,E : 3}
M	{{K,E : 2}, {K : 1}}	{K : 3}
E	{K : 4}	{K : 4}
K		

From the Conditional Frequent Pattern tree, the Frequent Pattern rules are generated by pairing the items of the Conditional Frequent Pattern Tree set to the corresponding to the item as given in the below table.



Items	Frequent Pattern Generated
Y	{<K,Y : 3>}
O	{<K,O : 3>, <E,O : 3>, <E,K,O : 3>}
M	{<K,M : 3>}
E	{<E,K : 3>}
K	

For each row, two types of association rules can be inferred for example for the first row which contains the element, the rules K -> Y and Y -> K can be inferred. To determine the valid rule, the confidence of both the rules is calculated and the one with confidence greater than or equal to the minimum confidence value is retained.



Data Mining

EG 3212 CT

		Total:	**7 hour /week**
Year:	**III**	**Lecture:**	**3 hours/week**
Semester:	**VI**	**Tutorial:**	**1 hours/week**
		Practical:	**3 hours/week**

Course Introduction

Data Mining studies algorithms and computational paradigms that allow computers to find patterns and regularities in databases, perform prediction and forecasting, and generally improve their performance through interaction with data. The course will cover all these issues and will illustrate the whole process by examples.

Objectives

The general objectives of this course are as follows:

- To introduce concept of data preprocessing and data mining
- To discuss multi-dimensional data representation and OLAP operations
- To provide skill of illustrating clustering, classification, and association rule mining algorithms
- To introduce advanced concept of data mining

Unit	Topics	Contents	Hours	Methods/ Media
7	**Advanced Data Mining**	7.1 Information Retrieval, Measuring Effectiveness of Information Retrieval 7.2 Concept of Time-Series Data and Analysis, Image and Video Retrieval 7.3 Concept of Support Vector Machine and Deep Learning	**(6Hrs)**	

Information Retrieval

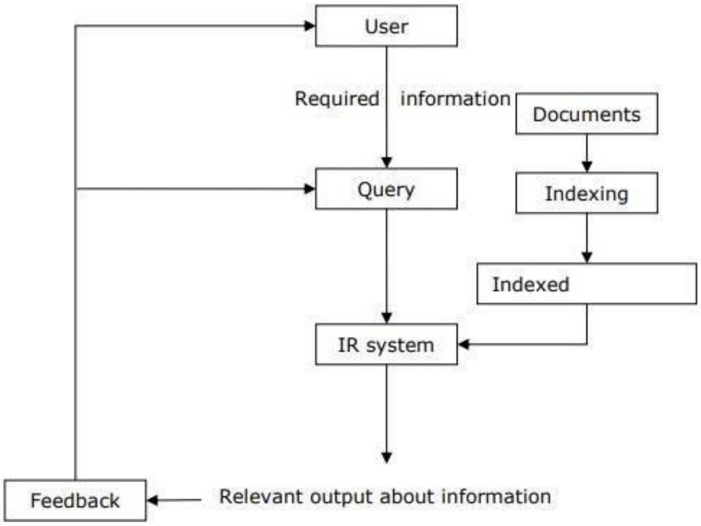
Information Retrieval is understood as a fully automatic process that responds to a user query by examining a collection of documents and returning a sorted document list that should be relevant to the user requirements as expressed in the query.

Information retrieval deals with the retrieval of information from a large number of text-based documents. Some of the database systems are not usually present in information retrieval systems because both handle different kinds of data. Examples of information retrieval system include –

- Online Library catalogue system
- Online Document Management Systems
- Web Search Systems etc.

The main problem in an information retrieval system is to locate relevant documents in a document collection based on a user's query. This kind of user's query consists of some keywords describing an information need.

In such search problems, the user takes an initiative to pull relevant information out from a collection. This is appropriate when the user has ad-hoc information need, i.e., a short-term need. But if the user has a long-term information need, then the retrieval system can also take an initiative to push any newly arrived information item to the user.
This kind of access to information is called Information Filtering. And the corresponding systems are known as Filtering Systems or Recommender Systems.
Information retrieval (IR) may be defined as a software program that deals with the organization, storage, retrieval and evaluation of information from document repositories particularly textual information. The system assists users in finding the information they require but it does not explicitly return the answers of the questions. It informs the existence and location of documents that might consist of the required information. The documents that satisfy user’s requirement are called relevant documents. A perfect IR system will retrieve only relevant documents.



It is clear from the above diagram that a user who needs information will have to formulate a request in the form of query in natural language. Then the IR system will respond by retrieving the relevant output, in the form of documents, about the required information.

Information retrieval (IR) is the science of searching for **information** in documents for text, sound, images or data.

Effectiveness of Information Retrieval (IR) Systems

In evaluating information storage and retrieval systems, those that deal with the retrieval of references to documents, much of the effort has gone into measuring variables based on the relevance of documents to the question put to the system. This aspect of evaluation is clearly only one part of the overall evaluation of any retrieval system. These relevance-based variables are chosen to reflect in some way what has now become known as the retrieval effectiveness: the ability of the system to retrieve relevant documents while at the same time suppressing the retrieval of non-relevant documents.

The most well-known pair of variables jointly measuring retrieval effectiveness are precision and recall, precision being the proportion of the retrieved documents that are relevant, and recall being the proportion of the relevant documents that have been retrieved. Singly, each variable (or parameter as it is sometimes called) measures some aspect of retrieval effectiveness; jointly they measure retrieval effectiveness completely.

Precision and recall are the measures used in the information retrieval domain to measure how well an information retrieval system retrieves the relevant documents requested by a user. The measures are defined as follows:

Precision= Total number of documents retrieved that are relevant/Total number of documents that are retrieved.
Recall = Total number of documents retrieved that are relevant/Total number of relevant documents in the database.

For IR, precision and recall are best explained by an example. Suppose you have 100 documents and a search query of “ford”. Of the 100 documents, suppose 20 are related/relevant to the term “ford”, and the other 80 are not relevant to “ford”.
Now suppose your search algorithm returns 25 result documents where 15 docs are in fact relevant (but meaning you incorrectly missed 5 relevant docs) and 10 result docs are not relevant (meaning you correctly omitted 70 of the irrelevant docs).

Precision and recall are calculated as:

```
precision = 15 retrieved relevant / 25 total retrieved

          = 0.60

recall = 15 retrieved relevant / 20 total relevant

       = 0.75
```

Concept of Time-Series Data and Analysis

A time series is a sequence of data points that occur in successive order over some period of time. This can be contrasted with cross-sectional data, which captures a point-in-time.

A time series can be taken on any variable that changes over time. In investing, it is common to use a time series to track the price of a security over time. This can be tracked over the short term, such as the price of a security on the hour over the course of a business day, or the long term, such as the price of a security at close on the last day of every month over the course of five years.

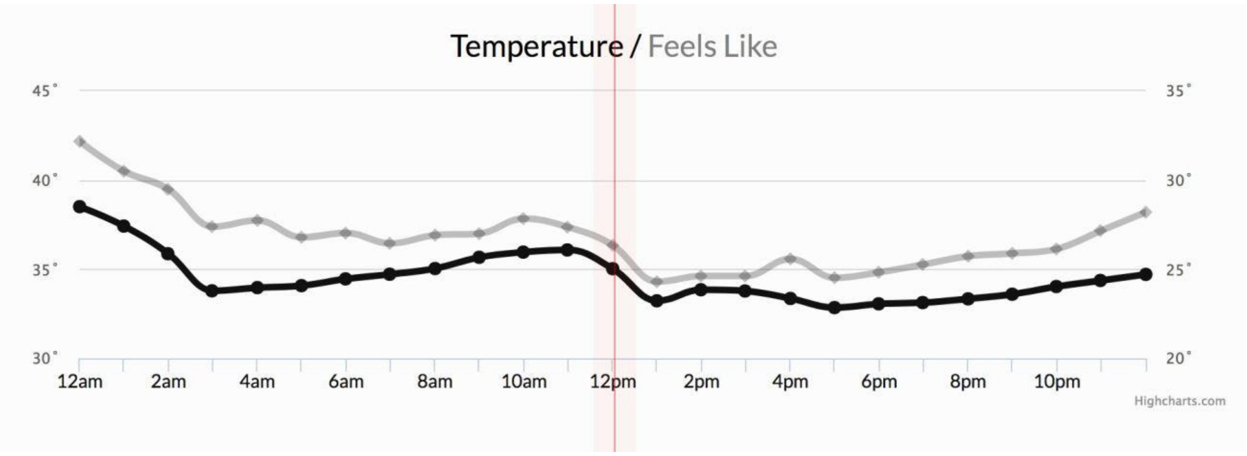
Time series analysis can be useful to see how a given asset, security, or economic variable changes over time. It can also be used to examine how the changes associated with the chosen data point compare to shifts in other variables over the same time period.

Time series forecasting uses information regarding historical values and associated patterns to predict future activity. Most often, this relates to trend analysis, cyclical fluctuation analysis, and issues of seasonality.

A time series can be constructed by any data that is measured over time at evenly-spaced intervals. Historical stock prices, earnings, GDP, or other sequences of financial or economic data can be analyzed as a time series.

Examples of time series analysis in action include:

- Weather data
- Rainfall measurements
- Temperature readings
- Heart rate monitoring (EKG)
- Brain monitoring (EEG)
- Quarterly sales
- Stock prices
- Automated stock trading
- Industry forecasts
- Interest rates



Another familiar example of time series data is patient health monitoring, such as in an electrocardiogram (ECG), which monitors the heart’s activity to show whether it is working normally.

Time series Analysis

Because time series analysis includes many categories or variations of data, analysts sometimes must make complex models.

Models of time series analysis include:

- ✓ **Classification:** Identifies and assigns categories to the data.
- ✓ **Curve fitting:** Plots the data along a curve to study the relationships of variables within the data.
- ✓ **Descriptive analysis:** Identifies patterns in time series data, like trends, cycles, or seasonal variation.
- ✓ **Explanative analysis:** Attempts to understand the data and the relationships within it, as well as cause and effect.
- ✓ **Exploratory analysis:** Highlights the main characteristics of the time series data, usually in a visual format.
- ✓ **Forecasting:** Predicts future data. This type is based on historical trends. It uses the historical data as a model for future data, predicting scenarios that could happen along future plot points.
- ✓ **Intervention analysis:** Studies how an event can change the data.
- ✓ **Segmentation:** Splits the data into segments to show the underlying properties of the source information.

Image and Video Retrieval

An **image retrieval** system is a computer system used for browsing, searching and retrieving images from a large database of digital images. The first microcomputer-based image database retrieval system was developed at MIT, in the 1990s.

Information retrieval (IR) is the science of searching for documents or information in documents. Documents can be text or multimedia, and may reside on the Web.

Image and video retrieval continues to be one of the most exciting and fastest-growing research areas in the field of multimedia technology.

One could store the digital information on tapes, CDROMs, DVDs or any such device but the level of access would be less than the well-known shoe boxes filled with tapes, old photographs, and letters. What is needed is that the techniques for organizing images and video stay in tune with the amounts of information. Therefore, there is an urgent need for a semantic understanding of image and video.

Luckily, it can be argued that the access to video is somehow a simpler problem than access to still images. Video comes as a sequence, so what moves together most likely forms an entity in real life, so segmentation of video is intrinsically simpler than a still image, at the expense of only more data to handle. So the potential to make progress on video in a semantic understanding is there. Moving from images to video adds several orders of complexity to the retrieval problem due to indexing, analysis, and browsing over the inherently temporal aspect of video.

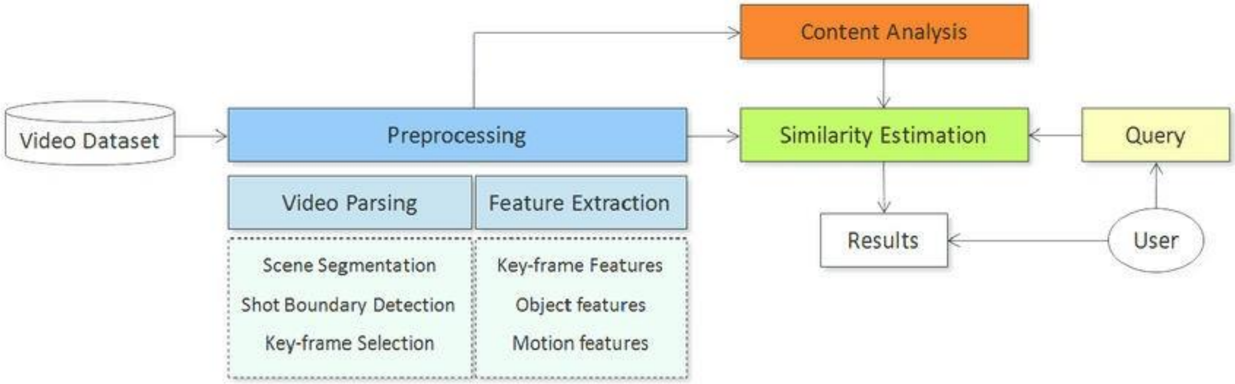


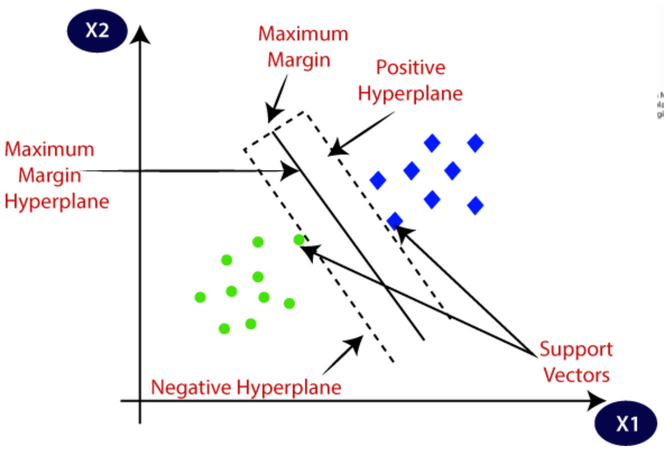
Fig: Overview of video retrieval system

The concept of Support Vector Machine and Deep Learning

Support Vector machine (SVM)

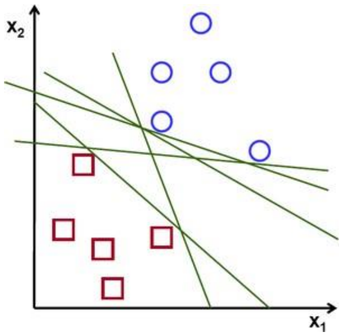
Support Vector Machine or SVM is one of the most popular Supervised Learning algorithms, which is used for Classification as well as Regression problems. However, primarily, it is used for Classification problems in Machine Learning.
The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future. This best decision boundary is called a hyperplane.

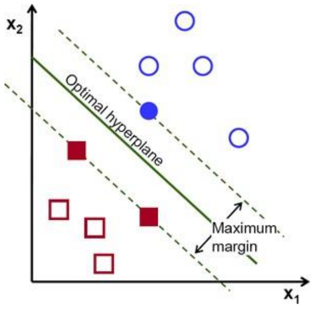
SVM chooses the extreme points/vectors that help in creating the hyperplane. These extreme cases are called as support vectors, and hence algorithm is termed as Support Vector Machine. Consider the below diagram in which there are two different categories that are classified using a decision boundary or hyperplane:



An SVM algorithm should not only place objects into categories, but have the margins between them on a graph as wide as possible.
Some applications of SVM include:

- Text and hypertext classification
- Image classification
- Recognizing handwritten characters
- Biological sciences, including protein classification





SVM algorithm can be used for **Face detection, image classification, text categorization,** etc.

Types of SVM

SVM can be of two types:

- ✓ **Linear SVM:** Linear SVM is used for linearly separable data, which means if a dataset can be classified into two classes by using a single straight line, then such data is termed as linearly separable data, and classifier is used called as Linear SVM classifier.
- ✓ **Non-linear SVM**: Non-Linear SVM is used for non-linearly separated data, which means if a dataset cannot be classified by using a straight line, then such data is termed as non-linear data and classifier used is called as Non-linear SVM classifier.

Hyperplane: There can be multiple lines/decision boundaries to segregate the classes in n-dimensional space, but we need to find out the best decision boundary that helps to classify the data points. This best boundary is known as the hyperplane of SVM.

The dimensions of the hyperplane depend on the features present in the dataset, which means if there are 2 features (as shown in image), then hyperplane will be a straight line. And if there are 3 features, then hyperplane will be a 2-dimension plane.

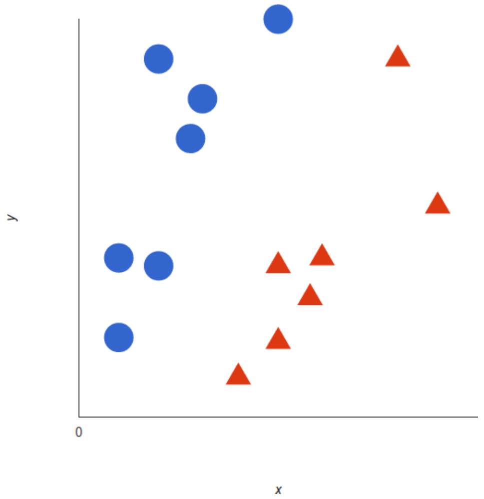
We always create a hyperplane that has a maximum margin, which means the maximum distance between the data points.

A hyperplane is an $n\text{-}1$ dimensional subspace in an n-dimensional space.

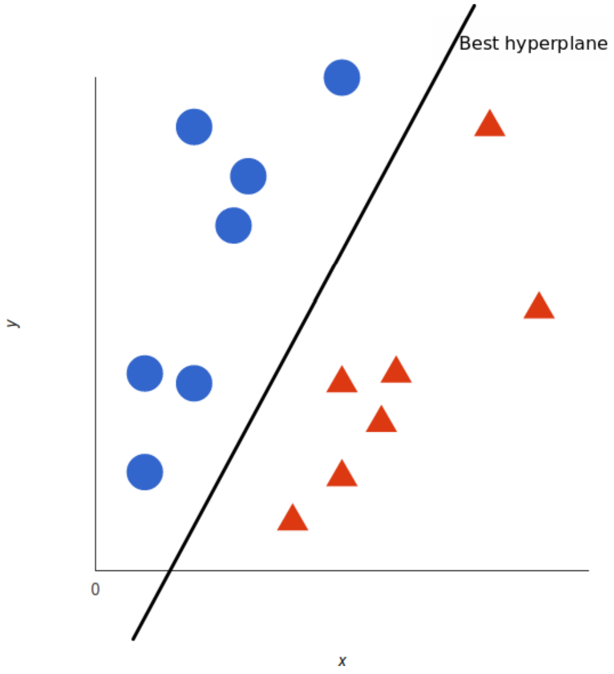
- ✓ SVC is a single point in a one-dimensional space
- ✓ SVC is a one-dimensional line in a two-dimensional space.
- ✓ SVC is a two-dimensional plane in a three-dimensional space.
- ✓ When the data-points are in more than three dimensions, SVC is a hyperplane.

How SVM work?

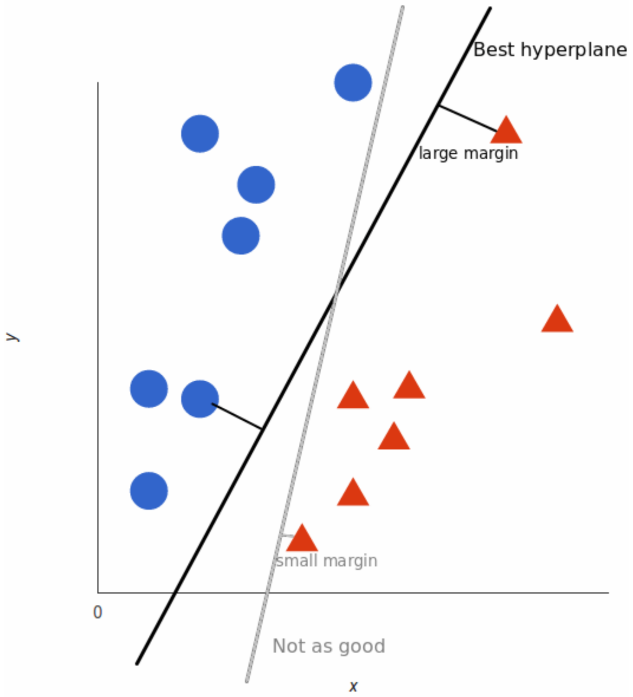
The basics of Support Vector Machines and how it works are best understood with a simple example. Let's imagine we have two tags: red and blue, and our data has two features: x and y. We want a classifier that, given a pair of (x, y) coordinates, outputs if it's either red or blue. We plot our already labeled training data on a plane:



A support vector machine takes these data points and outputs the hyperplane (which in two dimensions it's simply a line) that best separates the tags. This line is the decision boundary: anything that falls to one side of it we will classify as blue, and anything that falls to the other as red.



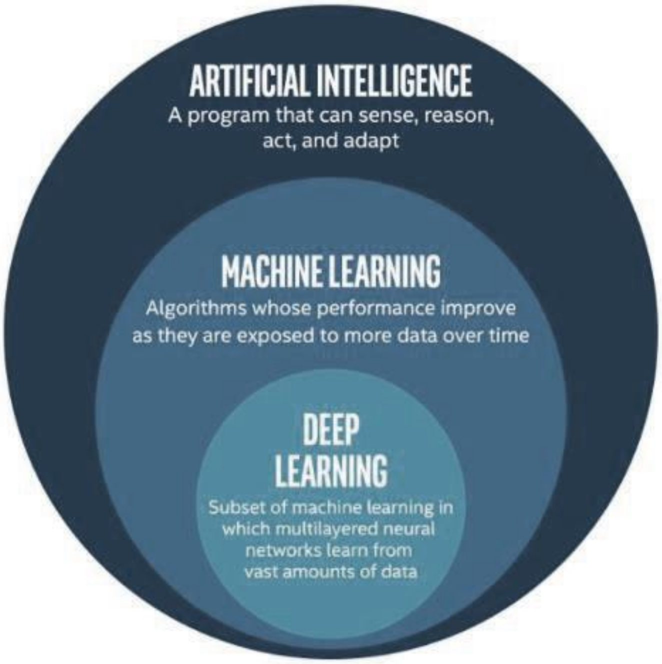
But what exactly is the best hyperplane? For SVM, it's the one that maximizes the margins from both tags. In other words: the hyperplane (remember it's a line in this case) whose distance to the nearest element of each tag is the largest.



A support vector machine (SVM) is a type of deep learning algorithm that performs supervised learning for classification or regression of data groups.

Deep Learning

Deep Learning is a subfield of machine learning concerned with algorithms inspired by the structure and function of the brain called artificial neural networks.


ARTIFICIAL INTELLIGENCE
A program that can sense, reason, act, and adapt
MACHINE LEARNING
Algorithms whose performance improve as they are exposed to more data over time
DEEP LEARNING
Subset of machine learning in which multilayered neural networks learn from vast amounts of data

Deep learning is a subset of machine learning, which is essentially a neural network with three or more layers. These neural networks attempt to simulate the behavior of the human brain—albeit far from matching its ability—allowing it to “learn” from large amounts of data. While a neural network with a single layer can still make approximate predictions, additional hidden layers can help to optimize and refine for accuracy.

Deep learning drives many artificial intelligence (AI) applications and services that improve automation, performing analytical and physical tasks without human intervention. Deep learning technology lies behind everyday products and services (such as digital assistants, voice-enabled TV remotes, and credit card fraud detection) as well as emerging technologies (such as self-driving cars).

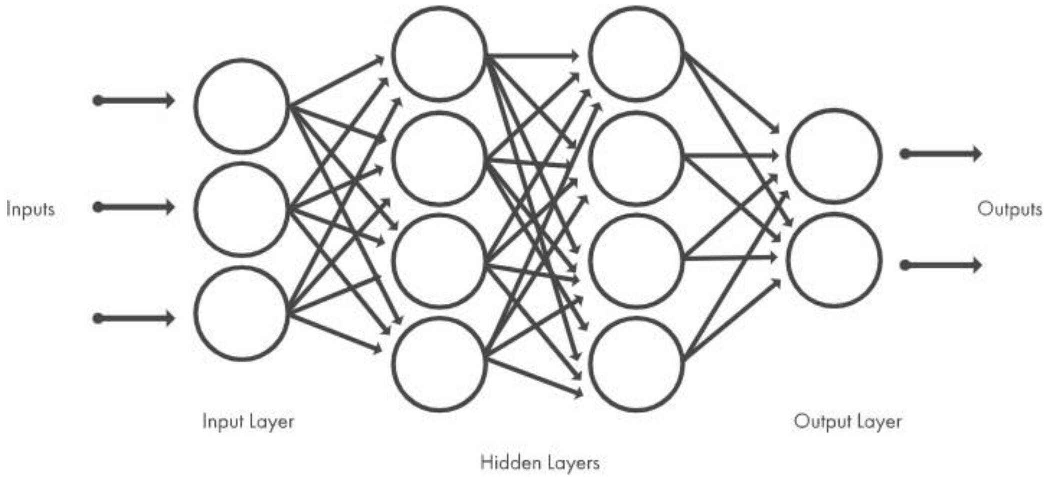
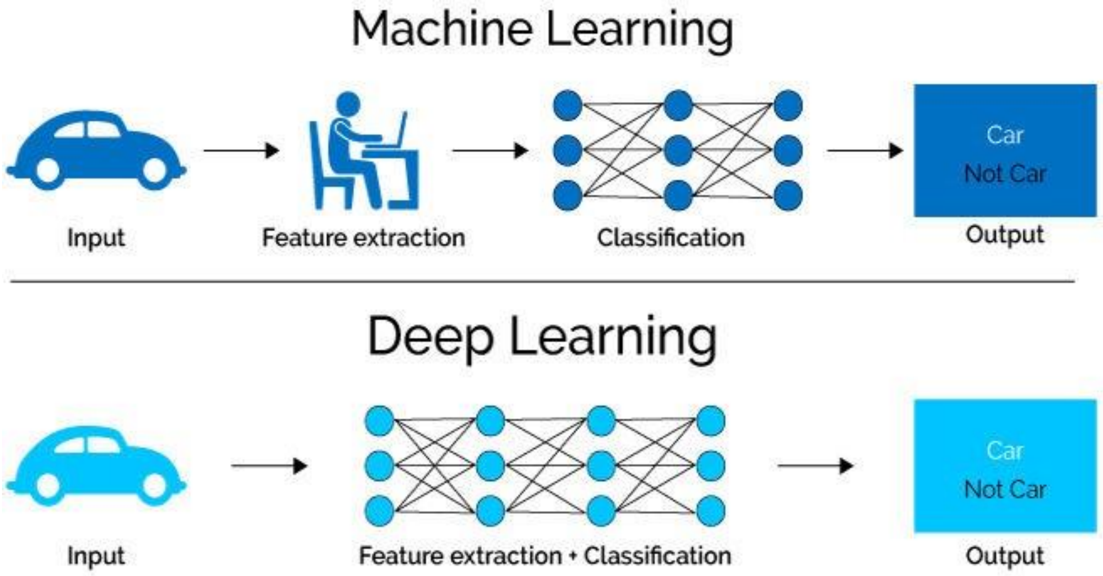


Fig: Neural networks, which are organized in layers consisting of a set of interconnected nodes. Networks can have tens or hundreds of hidden layers.

One of the most popular types of deep neural networks is known as convolutional neural networks (CNN or ConvNet). A CNN convolves learned features with input data, and uses 2D convolutional layers, making this architecture well suited to processing 2D data, such as images.

- ✓ Long before deep learning was used, traditional machine learning methods were mainly used. Such as Decision Trees, SVM, Naïve Bayes Classifier and Logistic Regression.



- ➢ Feature Extraction is only required for ML Algorithms.

Deep learning is currently the most sophisticated AI architecture in use today. Popular deep learning algorithms include:

- ✓ **Convolutional neural network -** the algorithm can assign weights and biases to different objects in an image and differentiate one object in the image from another. Used for object detection and image classification.
- ✓ **Recurrent neural networks -** the algorithm is able to remember sequential data. Used for speech recognition, voice recognition, time series prediction and natural language processing.
- ✓ **Long short-term memory networks -** the algorithm can learn order dependence in sequence prediction problems. Used in machine translation and language modeling.
- ✓ **Generative adversarial networks** - two algorithms compete against each other and use each other's mistakes as new training data. Used in digital photo restoration and deepfake video.
- ✓ **Deep belief networks -** an unsupervised deep learning algorithm in which each layer has two purposes: it functions as a hidden layer for what came before and a visible layer for what comes next. Used in healthcare sectors for cancer and other disease detection.

THE END