Course title: Data Mining and Data Warehousing

Course Code: CSC 425

Credit Unit : 2C

INTRODUCTION

What is Data Mining?

Data Mining is the computational process of discovering patterns in large data sets involving methods using artificial intelligence, machine learning, statistical analysis, and database systems to extract information from a data set and transform it into an understandable structure for further use. One of the most frequently cited definitions of data mining defines the technique as “the nontrivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data”.

Simply storing information in a data warehouse does not provide the benefits that an organization is seeking. To realise the value of a data warehouse, it is necessary to extract the knowledge hidden within the warehouse. However, as the amount and complexity of the data in a data warehouse grows, it becomes increasingly difficult, if not impossible, for business analysts to identify trends and relationships in the data using simple query and reporting tools. Data mining is one of the best ways to extract meaningful trends and patterns from huge amounts of data. Data mining discovers information within data warehouses that mere queries and reports cannot effectively reveal.

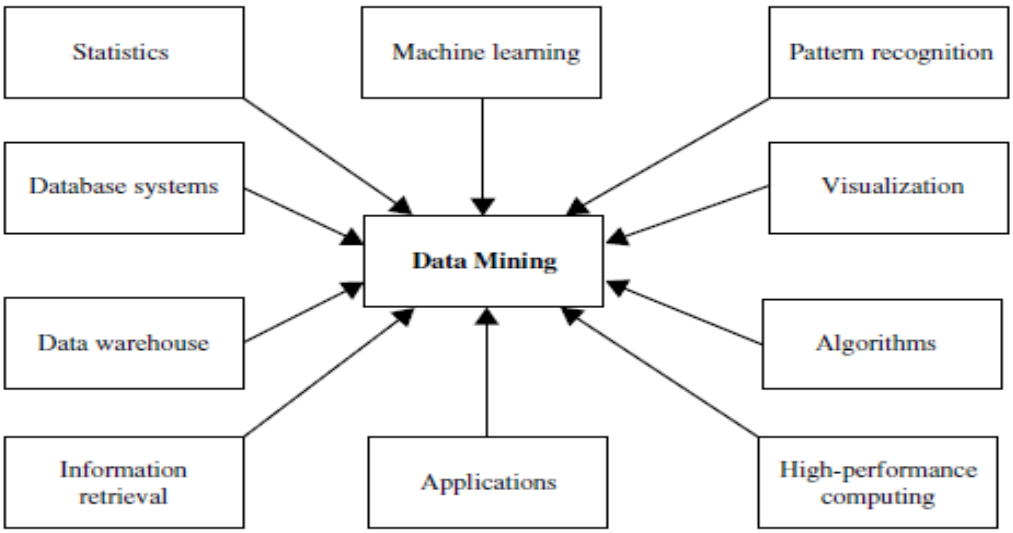
Many people treat data mining as a synonym for another popularly used term, *knowledge discovery from data, or KDD*, while others view data mining as merely an essential step in the process of knowledge discovery. There is a need to clarify between Data mining and querying a Database. While the former can reveal the hidden knowledge in data stored in database and other relevant information not stored in database; the latter is limited to retrieving information about the data stored in database. It is on record that statisticians were the first to use the term “data mining” and recently, computer scientists have looked at data mining as an algorithmic problem.

In the commercial field, the questions to be asked are not only ‘how many customers have bought this product in this period?’ but also ‘what is their profile?’, ‘what other products are they interested in?’ and ‘when will they be interested’? The profiles to be discovered are generally complex, more complicated combinations, in which the discriminant variables are not necessarily what we might have imagined at first, and could not be found by chance, especially in the case of rare behaviours. Data mining methods are certainly more complex than those of elementary descriptive statistics. They are based on artificial intelligence tools (neural networks), information theory (decision trees), machine learning theory, and above all, inferential statistics and ‘conventional’ data analysis including factor analysis, clustering and discriminant analysis, etc.

In today business market, the level of engagement between customers and companies, services or even product has changed. The companies have made their presence online prominent by becoming easily accessible through social platforms such as Facebook, Twitter, and WhatsApp. These platforms provide valuable data which is unstructured. That is a reason why most companies require Data Mining tools.

Data mining software help to explore the unknown patterns that are significant to the success of the business. The actual data mining task is an automatic analysis of large quantities of data to extract previously unknown, interesting patterns such as cluster analysis, unusual records (anomaly detection), and dependencies (association rule mining, sequential pattern mining).

Data mining has its origin in various disciplines; the two most important are statistics and machine learning. The figure below summarizes some disciplines through which data mining originates.



Data mining adopts techniques from many domains

Generally, knowledge discovery from data follows some basic steps:

1. **Data cleaning:** To remove noise and inconsistent data.
2. **Data integration:** Combination of multiple data sources.
3. **Data selection:** Process of selecting data that are relevant to the analysis task from the database.
4. **Data transformation:** Transformation and consolidation of data into forms appropriate for mining by performing summary or aggregation operations.
5. **Data mining:** This is an essential process where intelligent methods are applied to extract the patterns in the data.
6. **Pattern evaluation:** This is the identification of the truly interesting patterns representing knowledge based on *interestingness measures*.
7. **Knowledge presentation:** This is the visualization and knowledge representation techniques that are used to present mined knowledge to users.

Data Pre-processing

Data Quality: Why is it necessary to pre-process data?

Data have quality if they satisfy the requirements of the intended use. Data quality comprised of : *accuracy, completeness, consistency* and *interpretability*. Let us look at a scenario. Imagine that you are a manager at XYZ company and have been charged with analysing the company's data with respect to your branch's sales. You immediately set out to perform this task. You carefully inspect the company's database and data warehouse, identifying and selecting the attributes or dimensions such as *item*, *price*, and *units sold* to be included in your analysis. Then you notice

that several of the attributes for various tuples have no recorded value. For your analysis, you would like to include information as to whether each item purchased was advertised as on sale, yet you discover that this information has not been recorded. Furthermore, users of your database system have reported errors, unusual values, and inconsistencies in the data recorded for some transactions. In other words, the data you wish to analyze by data mining techniques are *incomplete i.e they lack* attribute values or certain attributes of interest, or containing only aggregate data; *inaccurate* or *noisy and inconsistent*.

This is a typical example of how real world data look like. This scenario illustrates three of the elements defining data quality: *accuracy, completeness,* and *consistency.* Inaccurate, incomplete, and inconsistent data are commonplace properties of large real-world databases and data warehouses. Today's real-world databases are highly susceptible to noisy, missing, and inconsistent data due to their typically huge size which sometimes may be of several gigabytes or more. Data inconsistency may also be as a result of where it originates from or due to its heterogeneous sources. Low-quality data will lead to low-quality mining results. However, pre-processing of data can:

i. *improve the quality of the data*
ii. *improve the mining results*
iii. *improve the efficiency and ease of the mining process*

There are several data pre-processing techniques. Although, not all may be required when a particular data is to be mined; the data to be explored determines the specific method that must be applied. Some of the data pre-processing techniques are:

i. Data cleaning*: Cleaning* removes noise and resolves inconsistencies in data. The process involves filling in the missing values, smoothing noisy data, and identifying or removing outliers, 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 confuse the mining procedure, resulting in unreliable output. The first step in data cleaning as a process is *discrepancy detection*. Discrepancies can be caused by several factors, including poorly designed data entry forms that have many optional fields, human error in data entry, and deliberate errors. For instance, if respondents are not willing to divulge their personal information, and data decay, e.g., out-dated addresses. Discrepancies may also arise from inconsistent data representations and inconsistent use of codes.

ii. Data integration: In the process of integrating data, several data are merged from multiple sources into a coherent data store such as a data warehouse. Getting back to your task at XYZ company, suppose that you would like to include data from multiple sources in your analysis. This would involve integrating multiple databases, or files.

Yet some attributes representing a given concept may have different names in different databases, causing inconsistencies and redundancies. For example, the attribute for customer identification may be referred to as *customer id* in one data store and *cust id* in another.

In a situation where the data selected for analysis is HUGE, the probability of having a slow mining process becomes very high. Such data set can be reduced without jeopardizing the data mining results. This is achievable through data reduction approach.

iii. Data reduction: This strategy obtains a reduced representation of the data set that is much smaller in volume and in the number of attributes, yet produces the same or almost the same analytical results. Data reduction strategies include:

a. *dimensionality reduction*
b. *numerosity reduction*.

In dimensionality reduction, data encoding schemes are applied so as to obtain a reduced or compressed representation of the original data. Examples include data compression techniques e.g., *wavelet transforms* and *principal components analysis*, *attribute subset selection* e.g., removing irrelevant attributes, and *attribute construction* e.g., where a small set of more useful attributes is derived from the original set.

In numerosity reduction, the data are replaced by alternative, smaller representations using parametric models e.g., *regression* or nonparametric models e.g., *clusters*, or *data aggregation*. In other words, the size of data set can be reduced by aggregation, elimination of redundant features, or clustering.

iv. Data transformations: *Again g*etting back to company's data, you have decided that you would like to use a distance based mining algorithm for your analysis, such as neural networks, nearest-neighbour classifiers, or clustering. Such methods provide better results if the data to be analysed have been *normalized*, that is, scaled to a smaller range such as [0.0, 1.0]. This technique usually improves the accuracy and efficiency of mining algorithms involving distance measurements.

v. Discretization: *Discretization* and *concept hierarchy generation* can also be useful, where raw data values for attributes are replaced by ranges or higher conceptual levels. For example, raw values for *age* may be replaced by higher-level concepts, such as *youth*, *adult*, or *senior*. Normalization, data discretization, and concept hierarchy generation are forms of data transformation. Concept hierarchies are a form of data discretization that can also be used for data smoothing. A concept hierarchy for *price*, for instance, may map real *price* values into *inexpensive,*

moderately priced, and *expensive*, thereby reducing the number of data values to be handled by the mining process.

Finally, in *data transformation*, the data are transformed or consolidated into forms appropriate for mining. Note that, transformation is one of the data pre-processing techniques, the strategies it uses can be summarized as follows:

1. Smoothing, this is a way of ensuring that data are free from noise. The techniques used here may include regression and clustering.

2. Attribute construction (or *feature construction*), where new attributes are constructed and added from the given set of attributes to help the mining process.

3. Aggregation, where summary or aggregation operations are applied to the data. For example, the daily sales data may be aggregated to compute monthly and annual total amounts. This step is typically used in constructing a data cube for data analysis at multiple abstraction levels.

4. Normalization, where the attribute data are scaled so as to fall within a smaller range, such as -1.0 to 1.0, or 0.0 to 1.0.

5. Discretization, where the raw values of a numeric attribute (e.g., *age*) are replaced by interval labels (e.g., 0–10, 11–20, etc.) or conceptual labels (e.g., *youth, adult*, *senior*).

What are Data Mining Software?

Most of the algorithms required to mine knowledge from data are already implemented in several software tools, few among them are:
Orange Data mining, Anaconda, R Software Environment, Scikit-learn, Weka Data Mining, Rapidminer, DataMelt, Natural Language Toolkit, Apache Mahout, GNU Octave, GraphLab Create, ELKI, KNIME Analytics Platform Community, TANAGRA, Rattle GUI, CMSR Data Miner and several others.

Tasks of Data Mining

Data mining involves six common classes of tasks:

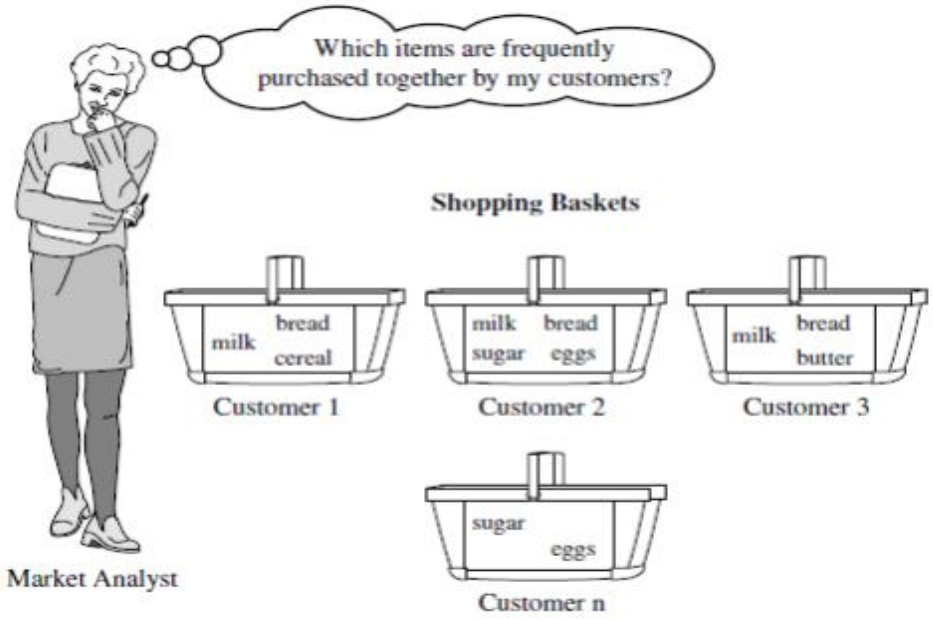
1. **Anomaly detection** (Outlier/change/deviation detection) – The identification of unusual data records, that might be interesting or data errors that require further investigation.

2. **Association rule mining** (Dependency modelling) – Searches for relationships between variables. For example, a supermarket might gather data on customer purchasing habits. Using association rule learning, the supermarket can

determine which products are frequently bought together and use this information for marketing purposes. This is sometimes referred to as market basket analysis.

Market basket analysis:

This process analyzes customer buying habits by finding associations between the different items that customers place in their shopping baskets. The discovery of such associations can help retailers develop marketing strategies by gaining insight into which items are frequently purchased together by customers. For instance, if customers are buying milk, how likely are they to also buy bread (and what kind of bread) on the same trip to the supermarket. Such information can lead to increased sales by helping retailers to selective marketing and plan their shelf space



Example:

If customers who purchase computers also tend to buy antivirus software at the same time, then placing the hardware display close to the software display may help increase the sales of both items. In an alternative strategy, placing hardware and software at opposite ends of the store may entice customers who purchase such items to pick up other items along the way. For instance, after deciding on an expensive computer, a customer may observe security systems for sale while heading toward the software display to purchase antivirus software and may decide to purchase a home security system as well. Market basket analysis can also help retailers plan which items to put on sale at reduced prices. If customers tend to purchase computers and

printers together, then having a sale on printers may encourage the sale of printers as well as computers.

3. Clustering

This is the task of discovering groups and structures in the data that are in some way or another "similar", without using known structures in the data.

Another approach of revealing patterns in the dataset is through description.
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**. A typical example of data description is clustering.

Clustering techniques apply when there is no class to be predicted but the instances are to be divided into natural groups. In other to achieve good cluster results, a correlation-based analysis method can be used to perform attribute *relevance analysis* and filter out statistically irrelevant or weakly relevant attributes from the descriptive mining process.

The following are typical requirements of clustering in data mining.

i. Scalability: Many clustering algorithms work well on small data sets containing little above hundred data objects; however, a large database may contain millions or even billions of objects, particularly in Web search scenarios. Clustering on only a sample of a given large data set may lead to biased results. Therefore, highly scalable clustering algorithms are needed.

ii. Ability to deal with different types of attributes: Many algorithms are designed to cluster numeric data. However, applications may require clustering other data types, such as binary, nominal (categorical), or mixtures of these data types. Recently, more and more applications need clustering techniques for complex data types such as graphs, sequences, images, and documents.

iii. Discovery of clusters with arbitrary shape: Many clustering algorithms determine clusters based on Euclidean or Manhattan distance measures. Algorithms based on such distance measures tend to find spherical clusters with similar size and density. However, a cluster could be of any shape. Consider sensors, for example, which are often deployed for environment surveillance. Cluster analysis on sensor readings can detect interesting phenomena. We may want to use clustering to find the frontier of a running forest fire, which is often not spherical. It is important to develop algorithms that can detect clusters of arbitrary shape.

iv. Requirements for domain knowledge to determine input parameters: Many clustering algorithms require users to provide domain knowledge in the form of input parameters such as the desired number of clusters. Consequently, the clustering results

may be sensitive to such parameters. Parameters are often hard to determine, especially for high-dimensionality data sets and where users have yet to grasp a deep understanding of their data. Requiring the specification of domain knowledge not only burdens users, but also makes the quality of clustering difficult to control.

v. Ability to deal with noisy data: Most real-world data sets contain outliers and/or missing, unknown, or erroneous data. Sensor readings, for example, are often noisy. Some readings may be inaccurate due to the sensing mechanisms, and some readings may be erroneous due to interferences from surrounding transient objects. Clustering algorithms can be sensitive to such noise and may produce poor-quality clusters. Therefore, we need clustering methods that are robust to noise.

vi. Capability of clustering high-dimensionality data: A data set can contain numerous dimensions or attributes. When clustering documents, for example, each keyword can be regarded as a dimension, and there are often thousands of keywords. Most clustering algorithms are good at handling low-dimensional data such as data sets involving only two or three dimensions. Finding clusters of data objects in a high dimensional space is challenging, especially considering that such data can be very sparse and highly skewed.

vii. Constraint-based clustering: Real-world applications may need to perform clustering under various kinds of constraints. Suppose that your job is to choose the locations for a given number of new automatic teller machines (ATMs) in a city. To decide upon this, you may cluster households while considering constraints such as the city's rivers and highway networks and the types and number of customers per cluster. A challenging task is to find data groups with good clustering behaviour that satisfy specified constraints.

viii. Interpretability and usability: Users want clustering results to be interpretable, comprehensible, and usable. That is, clustering may need to be tied in with specific semantic interpretations and applications. It is important to study how an application goal may influence the selection of clustering features and clustering methods.

Basic Clustering Methods

There are many clustering algorithms in the literature. It is difficult to provide a crisp categorization of clustering methods because these categories may overlap so that a method may have features from several categories. Nevertheless, it is useful to present a relatively organized picture of clustering methods. In general, the major fundamental clustering methods can be classified into the following categories:

1. Partitioning methods: 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. The basic partitioning methods typically adopt *exclusive cluster separation*. That is, each object must belong to exactly one group. Most partitioning methods are distance-based. Given *k*, the number of partitions to construct, a partitioning method creates an initial partitioning. It then uses an ***iterative relocation technique*** that attempts to improve the partitioning by moving objects from one group to another. The general criterion of a good partitioning is that objects in the same cluster are "close" or related to each other, whereas objects in different clusters are "far apart" or very different.

2. **Hierarchical methods:** 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.

    a. 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.
    b. The *divisive approach*, also called the *top-down* approach, starts with all the objects in the same cluster. In each of the successive iteration, a cluster is split into smaller clusters, until eventually each object is in one cluster, or a termination condition holds.

3. **Density-based methods:** Most partitioning methods cluster objects based on the distance between objects. Such methods can find only spherical-shaped clusters and encounter difficulty 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 "neighbourhood" exceeds some threshold. For example, for each data point within a given cluster, the neighbourhood 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.

4. **Grid-based methods:** 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. Some clustering algorithms integrate the ideas of several clustering methods, so that it is sometimes difficult to classify a given algorithm as uniquely belonging to only one clustering method category. Furthermore, some applications may have clustering criteria that require the integration of several clustering techniques.

Summary of Clustering methods and their characteristics

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)

4. Classification – is the task of generalizing known structure to apply to new data. For example, an e-mail program might attempt to classify an e-mail as "legitimate" or as "spam". Data classification involves tagging data to make it easily searchable and track-able. It also eliminates multiple duplications of data, which can reduce storage and backup costs while speeding up the search process.

5. Regression – attempts to find a function which models the data with the least error. Regression is a statistical method that tries to determine the strength and character of the relationship between one dependent variable and a series of others.

6. Summarization – providing a more compact representation of the data set, including visualization and report generation.

Limitations of Data Mining

While data mining can be very powerful tools, they are not self-sufficient applications. To be successful, data mining requires skilled technical and analytical specialists who can structure the analysis and interpret the output that is created. Consequently, the limitations of data mining are primarily data or personnel-related rather than technology-related.

Although data mining can help to reveal patterns and relationships, it does not tell the user the value or significance of these patterns. The user must make these types of determinations. Similarly, the validity of the patterns discovered is dependent on how they compare to "real world" circumstances.

Another limitation of data mining is that while it can identify connections between behaviours or variables, it does not necessarily identify causal relationships. For example, an application may identify that a pattern of behaviour such as the propensity to purchase an airline ticket just shortly before the flight is scheduled to depart, is related to characteristics such as income, level of education and internet use. However, that does not necessarily indicate that the ticket purchasing behaviour is caused by one or more of these variables. The individual's behaviour could be affected by some additional variable(s) such as occupation (the need to make the trip on short notice), family status (a sick relative needing care), or a hobby (taking advantage of last-minute discounts to visit new destinations).

TYPES OF LEARNING

In the domain of Data mining, or Machine learning, learning is basically of three forms: supervised, unsupervised and reinforcement. Each of this is discussed as follows:

A. Supervised Learning

In supervised learning, the training data is composed of input-output pairs. The learning algorithms use the training data that has been classified (has a target value for each training vector). The purpose of the supervised learning algorithm is to create a prediction function using the training data that will generalize for unseen training vectors to classify them correctly. The supervised *learning* approach has two phases: training and testing. Training builds a model using a large sample of historical data called a *training set*, and testing involves trying out the model on new, previously unseen data to determine its accuracy and physical performance characteristics.

The goal of the supervised learning approach is for the model to learn a mapping from inputs to outputs so it can predict the output for new, unseen data.

Characteristics:

- **Labelled Data**: The dataset contains both input data (features) and the corresponding correct output (labels).
- **Learning Task**: The model tries to learn the relationship between the input features and the output labels.

- **Goal**: To predict the output for new data based on the learned relationship.

B. Unsupervised Learning

In unsupervised learning, the agent learns patterns in the input even though no explicit feedback is supplied. The most common unsupervised learning task is **clustering:** detecting potentially useful clusters of input examples. For example, a taxi agent might gradually develop a concept of "good traffic days" and "bad traffic days" without ever being given labelled examples of each by a teacher. The goal is to find patterns, structures, or relationships in the data without predefined outputs. The model tries to infer the underlying structure or distribution of the data.

Characteristics

- **Unlabeled Data**: The dataset consists only of input data without any associated labels.
- **Learning Task**: The model attempts to identify patterns, clusters, or dimensionality reduction techniques from the data itself.
- **Goal**: To discover hidden patterns or intrinsic structures in the data

Key Differences:

Aspect	Supervised Learning	Unsupervised Learning
Data	Labeled data (inputs with corresponding outputs)	Unlabeled data (only inputs, no outputs)
Goal	Learn mapping from inputs to outputs (prediction)	Discover hidden patterns or structures in data
Example Tasks	Classification, Regression	Clustering, Dimensionality Reduction
Algorithms	Linear Regression, Decision Trees, SVM, etc.	K-Means, PCA, Hierarchical Clustering, etc.

C. Reinforcement learning

In **reinforcement learning** the agent learns from a series of reinforcements rewards or punishments. For example, the lack of a tip at the end of the journey for a special job done gives the taxi agent an indication that it did something wrong. The two points for a win at the end of a chess game tell the agent it did something right. It is up to the agent to decide which of the actions prior to the reinforcement were most responsible for it.

In practice, these distinctions are not always so crisp. In semi-supervised learning, we are given a few labelled examples and must make what we can of a large collection of un-labelled examples. Even the labels themselves may not be the oracular truths that we hope for. Imagine that you are trying to build a system to guess a person's age from a photo. You gather some labelled examples by snapping pictures of people and asking their age. That's supervised learning. But in reality, some of the people lied about their age. It's **not** just that there is random noise in the data; rather the inaccuracies are systematic, and to uncover them is an unsupervised learning problem involving images, self-reported ages, and true (un-known) ages. Thus, both noise and lack of labels create a continuum between supervised and unsupervised learning.

Classification and Prediction

Classification

Classification can be defined as the process of finding a model (or function) that describes and distinguishes data classes or concepts, for the purpose of being able to use the model to predict the class of objects whose class label is unknown. The derived model is based on the analysis of a set of training data (i.e., data objects whose class label is known).

Example: An airport security screening station is used to deter mine if passengers are potential terrorist or criminals. To do this, the face of each passenger is scanned and its basic pattern (distance between eyes, size, and shape of mouth, head etc) is identified. This pattern is compared to entries in a database to see if it matches any patterns that are associated with known offenders.

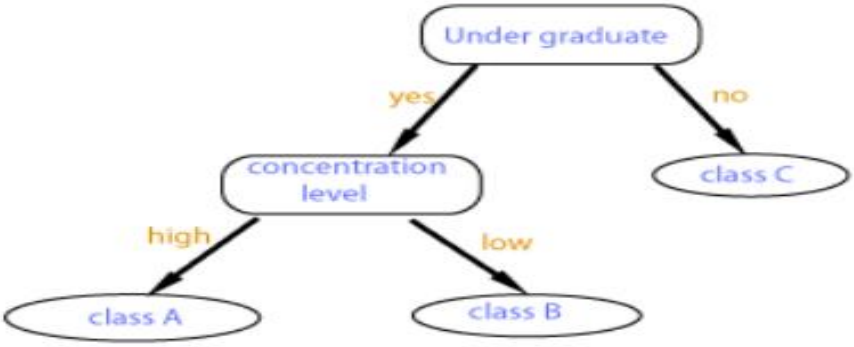
A classification model can be represented in various forms, such as:

1). IF-THEN rules, student (class , "undergraduate") AND concentration (level, "high") ==> class A

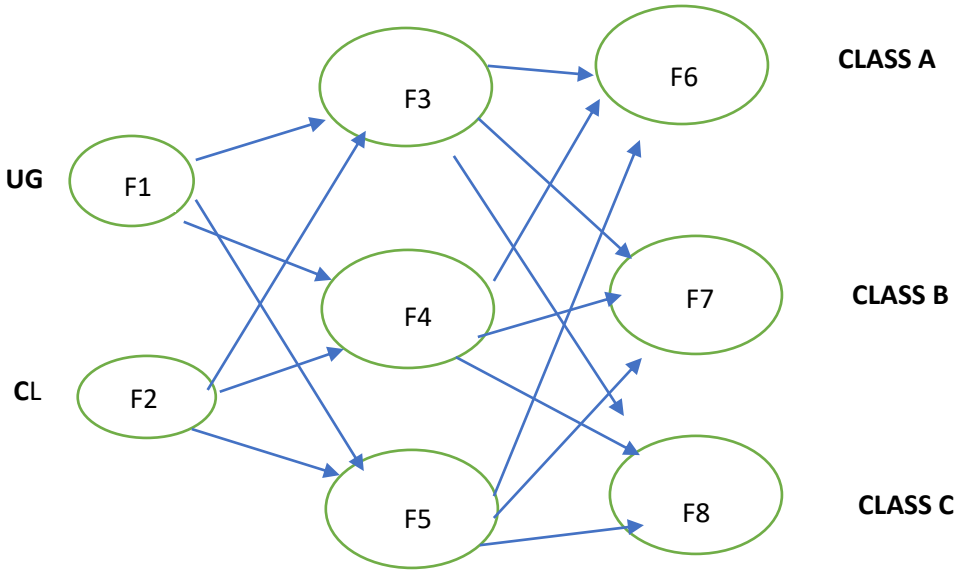
student (class ,"undergraduate") AND concentration (level,"low") ==> class B

student (class , "post graduate") ==> class C

2) Decision tree



3) Neural Network



UG- Undergraduate

CL – Concentration level

Prediction

This typically find some missing or unavailable data values rather than class labels. Although, prediction may refer to both data value prediction and class label prediction, it is usually confined to data value prediction and thus is distinct from classification. It is also referred to as regression most times. Prediction also encompasses the identification of distribution trends based on the available data.

Classification vs. Prediction

Classification differs from prediction in that the former i.e. classification is to construct a set of models (or functions) that describe and distinguish data class or concepts, whereas the latter is to predict some missing or unavailable, and often numerical, data values. Classification is used for predicting the class label of data objects and prediction is typically used for predicting missing numerical data values.

Regression can also be used to handle prediction problems. It is about using some independent variables to predict the dependent variable. For instance**, i**f the task is to predict house prices based on features like size, location, and age, the model learns from examples of houses with known prices to predict the price of new houses.

Classification vs. Clustering

- In classification you have a set of predefined classes and want to know which class a new object belongs to.

- Clustering tries to group a set of objects and find whether there is some relationship between the objects.

- In the context of machine learning, classification is supervised learning and clustering is unsupervised learning.

Artificial Neural Networks

An *artificial neural network (ANNs),* or simply *neural network,* is a type of artificial intelligence that attempts to mimic the way the human brain processes and stores information. ANNs are computational models inspired by the human brain, which can acquire and retain knowledge. It works by creating connections between mathematical processing elements, called *neurons.* Knowledge is encoded into the network through the strength of the connections between different neurons, called *weights,* and by creating *layers* of neurons that work in parallel. The system learns through a process of determining the number of neurons or *nodes* and adjusting the weights for the connections based upon training data.

In the 1940s foundational efforts at AI involved modelling the neurons in the brain, which resulted in the field of neural networks. An artificial neural network consists of a large collection of neural units (artificial neurons), whose behaviour is roughly based on how real neurons communicate with each other in the brain. Each neural unit is connected with many

other neural units, and links can enhance or inhibit the activation state of adjoining units.

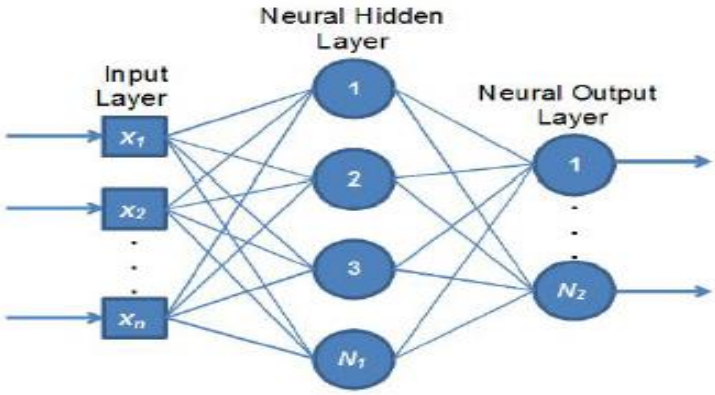
The network architecture consists of multiple layers of neural units. A signal initiates at the input layer, traverses through hidden layers, and finally culminates at the output layer. Once the logical approach to AI became dominant in the 1950s, neural networks fell from popularity. However, new algorithms for training neural networks and dramatically increased computer processing speed resulted in a re-emergence of the use of neural nets in the field called deep learning. Deep learning neural network
architectures differs from older neural networks in that they often have more hidden layers.

Furthermore, deep learning networks can be trained using both unsupervised and supervised learning. Deep learning has been used to solve tasks like computer vision and speech recognition, which were difficult with other approaches.

Learning takes two forms, supervised and unsupervised learning. In supervised learning, the training data is composed of input-output pairs. A neural network tries to find a function which, when given the inputs, produces the outputs. Through repeated application of training data, the network then approximates a function for that input domain. There are two main types of neural networks fixed (non-adaptive), and *dynamic* (adaptive).

Fixed neural networks, sometimes referred to as Pre-Trained Neural Networks (PTNN), are those that have undergone training and then become set. The internal structure of the network remains unchanged during operation. After training is complete, all weights, connections, and node configurations remain the same, and the network reduces to a repeatable function. A common use of a fixed neural network might be a classification system to identify malformed products on a manufacturing line where the definition of an undesirable characteristic would not change and the network would be expected to perform the same classification repeatedly.

A neural network without parameters will have no capability of associative memory. In particular, a neural network whenever its structure has been determined must possess the power for prediction in a supervised learning project. In order to make a neural network capable of prediction, it must have parameters which represent processed information. Among the various neural network architectures, there is the architecture of multiple layers, called MLP (Multilayer Perceptron). This type of architecture is usually used for pattern recognition, functional approximation, identification and control tasks. The structure of a neural network can be developed as shown in the figure below:



MLP neural network architecture

The focus here is not to discuss the detail about neural network but only to highlight how it works. However, explanation on neural network would be difficult to understand without mentioning transfer function, as it is a function that determines the output.

Transfer Function: The summation function computes the internal stimulation, or activation level of the neuron. The relationship between the internal activation level and the output can be linear or nonlinear. The relationship is expressed by one of several types of *transfer functions*. The transfer function combines (i.e., adds up) the inputs coming into a neuron from other neurons/sources and then produces an output based on the choice of the transfer function. Selection of the specific function affects the network's operation. The sigmoid activation function (or sigmoid *transfer function*) is an S-shaped transfer function in the range of 0 to 1, and it is a popular as well as useful nonlinear transfer function. Other activation functions that can be used are: step, sign, sigmoid and linear function. The *step* and *sign* activation functions, is sometimes referred to as hard limit functions. They are mostly used in decision-making neurons for classification and pattern recognition tasks.

Back-propagation Algorithm

Back-propagation is a neural network learning algorithm. During the learning phase, the network learns by adjusting the weights so as to be able to predict the correct class label of the input tuples. Neural network learning is also referred to as *connectionist learning* due to the connections between units. Neural networks involve long training times and are therefore more suitable for applications where this is feasible. They require several parameters that are typically best determined empirically such as the network topology or "structure." Neural networks have been criticized for their poor interpretability. For example, it is difficult for humans to interpret the symbolic meaning behind the learned weights and of "hidden units" in the network. These features initially made neural networks less desirable for data mining.

Advantages of neural networks, however, include their high tolerance of noisy data as well as their ability to classify patterns on which they have not been trained. They can be used when you may have little knowledge of the relationships between attributes and classes. They are well suited for continuous-valued inputs *and* outputs, unlike most decision tree algorithms. They have been successful on a wide array of real-world data, including handwritten character recognition, pathology and laboratory medicine, and training a computer to pronounce English text. Neural network algorithms are inherently parallel; parallelization techniques can be used to speed up the computation process.

Neural Network as a Classifier

Weakness

- ✓ Long training time
- ✓ Require a number of parameters typically best determined empirically, e.g., the network topology or ``structure."
- ✓ Poor interpretability: Difficult to interpret the symbolic meaning behind the learned weights and of ``hidden units" in the network

Strength

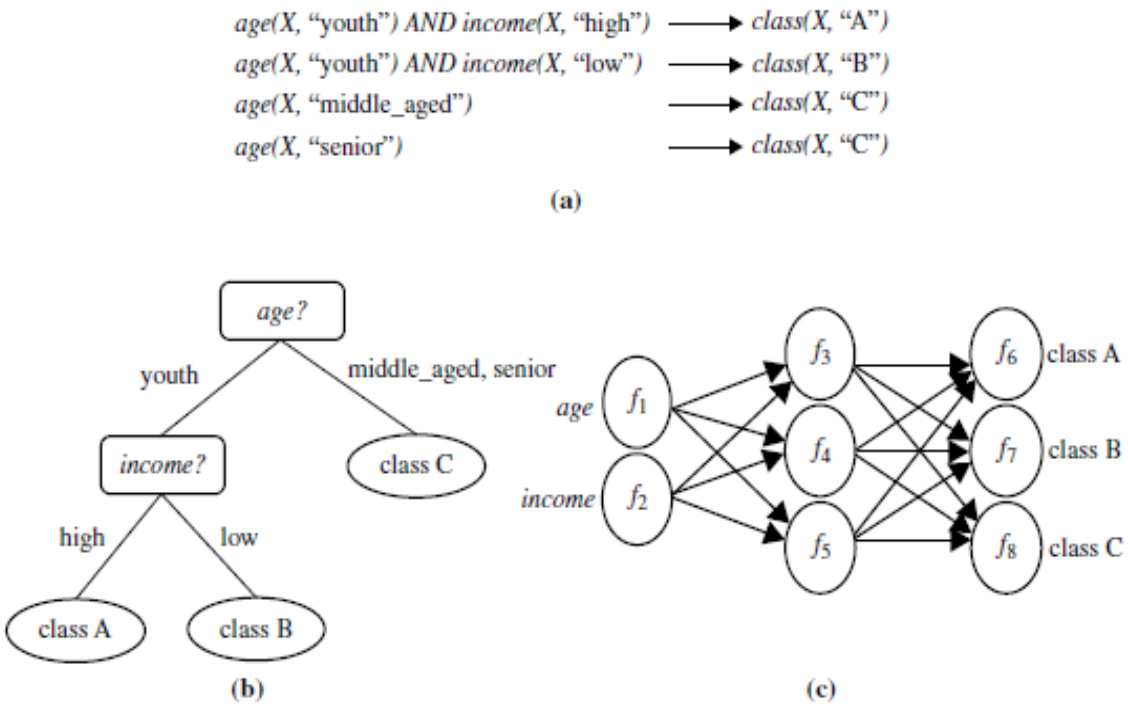
- ✓ High tolerance to noisy data
- ✓ Ability to classify untrained patterns
- ✓ Well-suited for continuous-valued inputs and outputs
- ✓ Successful on a wide array of real-world data
- ✓ Algorithms are inherently parallel
- ✓ Techniques have recently been developed for the extraction of rules from trained neural networks

DECISION TREES

Decision tree induction is one of the simplest and yet most successful forms of machine learning. A decision tree represents a function that takes as input a vector of attribute values and returns a "decision" in form of a single output value. The input and output values can be discrete or continuous. This is a tree-shaped structure that represents set of decisions. Nodes in a decision tree involve testing a particular attribute. Usually, the test compares an attribute value with a constant. Leaf nodes give a classification that applies to all instances that reach the leaf, or a set of classifications, or a probability distribution over all possible classifications.. Specific decision tree methods include Classification and Regression Trees (CART). More recently used decision tree algorithms include: ID3, C4.5 and C5.0.

To classify an unknown instance, it is routed down the tree according to the values of the attributes tested in successive nodes, and when a leaf is reached the instance is classified

according to the class assigned to the leaf. If the attribute is numeric, the test at a node usually determines whether its value is greater or less than a predetermined constant, giving a two-way split. Alternatively, a three-way split may be used, in which case there are several different possibilities. Decision tree can be illustrated as follows:



A classification model can be represented in various forms: (a) IF-THEN rules, (b) a decision tree, or (c) a neural network.

Data mining algorithms

We have discussed some techniques used in data mining especially for classification or Prediction purposes. This section list some of these techniques and some relevant algorithms required to be implemented to achieve the desired tasks

Pattern recognitions (Back propagation, Gradient Descent)

Association rule (Apriori algorithm, FP growth,)

Classification {SVM, Logical Regression, Linear Regression, DT etc}

Clustering {Db Scan, K-means, K-medoids, EM Algorithm etc}

Reinforcement (Q learning) – This algorithm learns the value of action-reward pairs which is usually referred to as Q-values.

Data Warehousing

Nowadays, the computing power of massively storing data has reached the point where virtually every data item generated by an enterprise can be saved. Also, enterprise databases have become extremely large and architecturally complex. An appropriate environment for applying various tools and techniques that can achieve a cleansed, stable, offline repository was needed and data warehouses were born.

A data warehouse is a set of databases with suitable properties for decision-making. The data are thematic, consolidated from different production information systems, user-oriented, non-volatile, documented and possibly aggregated. The purpose of a data warehouse is to support decision-making processes, business analysis, and reporting by consolidating data from different operational systems into one location for easy access and analysis. Data mining uses specialized data models; we cannot directly 'mine' the production data or the tables of a data centre or data warehouse.

As the data warehouses continue to grow, the need to create architecturally compatible functional subsets, or data marts, has been recognized. The immediate future is moving everything toward cloud computing. This will include the elimination of many local storage disks as data is pushed to a vast array of external servers accessible over the internet. Data mining in the cloud will continue to grow in importance as network connectivity and data accessibility become virtually infinite.

Data warehouses generalize and consolidate data in multidimensional space. The construction of data warehouses involves data cleaning, data integration, and data transformation, and can be viewed as an important pre-processing step for data mining. Moreover, data warehouses provide *online analytical processing (OLAP)* tools for the interactive analysis of multidimensional data of varied granularities, which facilitates effective data generalization and data mining. Many other data mining functions, such as association, classification, prediction, and clustering, can be integrated with OLAP operations to enhance interactive mining of knowledge at multiple levels of abstraction. Hence, the data warehouse has become an increasingly important platform for data analysis, OLAP and an effective platform for data mining. Therefore, data warehousing and OLAP form an essential step in the knowledge discovery process.

Integrating data from different sources usually presents many challenges—not deep issues of principle but nasty realities of practice. Different departments will use

different styles of recordkeeping, different conventions, different time periods, different degrees of data aggregation, and different primary keys, and will have different kinds of error. The data must be assembled, integrated, and cleaned up. The idea of companywide database integration is known as *data warehousing*.

Data warehouses provide a single consistent point of access to corporate or organizational data, transcending departmental divisions. They are the place where old data is published in a way that can be used to inform business decisions. The movement toward data warehousing is recognition of the fact that the fragmented information that an organization uses to support day-to-day operations at a departmental level can have immense strategic value when brought together. Clearly, the presence of a data warehouse is a very useful precursor to data mining, and if it is not available, many of the steps involved in data warehousing will have to be undertaken to prepare the data for mining.

A data warehouse is therefore, a subject-oriented, integrated, time-variant, and non-volatile collection of data in support of management's decision-making process. This short but comprehensive definition presents the major features of a data warehouse. The four keywords—*subject-oriented, integrated, time-variant*, and *non-volatile*—distinguish data warehouses from other data repository systems, such as relational database systems, transaction processing systems, and file systems.

Key features of data warehousing

The key features of data warehousing includes:

Subject-oriented: A data warehouse is organized around major subjects such as customer, supplier, product, and sales. Rather than concentrating on the day-to-day operations and transaction processing of an organization, a data warehouse focuses on the modelling and analysis of data for decision makers. Hence, data warehouses typically provide a simple and concise view of particular subject issues by excluding data that are not useful in the decision support process.

Data Integrated: A data warehouse is usually constructed by integrating multiple heterogeneous sources, such as e.g., transactional systems, log files, social media, etc.) are extracted, cleaned, and integrated into a single data warehouse. This often involves processes like ETL (Extract, Transform, Load). Data cleaning and data integration techniques are applied to ensure consistency in naming conventions, encoding structures, attribute measures, and so on.

Time-variant: Data are stored to provide information from an historic perspective (e.g., the past 5–10 years). Every key structure in the data warehouse contains, either implicitly or explicitly, a time element.

Non-volatile: A data warehouse is always a physically separate store of data transformed from the application data found in the operational environment. Due to this separation, a data warehouse does not require transaction processing, recovery, and concurrency control mechanisms. It usually requires only two operations in data accessing: *initial loading of data* and *access of data*.

Terminologies

In the context of data warehousing, there are several key terminologies that are important for understanding how data is structured, processed, and analysed. These terms help in organizing and managing large amounts of data that are used for decision-making and business analysis. Here's an overview of the most important terminologies:

.

1. ETL (Extract, Transform, Load):

Extract: The process of extracting data from different source systems.

Transform: The data is cleaned, validated, and transformed into a format suitable for analysis.

Load: The transformed data is loaded into the data warehouse for further processing and analysis.

2. Data Mart: A data mart is a subset of a data warehouse, often focused on a specific department or business function (like marketing, sales, finance). It is a smaller, more specialized version of a data warehouse.

3 OLAP (Online Analytical Processing): OLAP refers to the technology used to analyze and query multi-dimensional data. It allows users to view data from different perspectives (dimensions) and perform complex queries.

4. Fact Table: A fact table is the central table in a data warehouse schema. It contains quantitative data (measures) for analysis, such as sales figures, revenue, or quantities. It has the characteristics of been usually large and contains keys to link to dimension tables.

5. Dimension Table: A dimension table contains descriptive attributes (or dimensions) related to the facts in a fact table. These dimensions provide context to

the facts, such as time, product, location, or customer. For instance, in a sales data warehouse, dimensions could be Product, Time, and Store.

6. Star Schema: The star schema is a type of database schema used in data warehouses. It has a central fact table connected to multiple dimension tables, which resemble a star shape when visualized. Its benefits is the simple structure it produces and fast query performance.

7. Snowflake Schema: The snowflake schema is a more normalized version of the star schema, where dimension tables are further split into additional tables to reduce redundancy.

8. Normalization: Normalization is the process of organizing data to reduce redundancy and dependency by dividing large tables into smaller ones. It is used to ensure data integrity and efficiency in relational databases.

9. Metadata: A metadata is data about data. It provides information about the structure, source, and meaning of data in the warehouse. It can include details about data types, relationships between tables, and business rules applied during the ETL process.

Data cube

A data cube provides a multidimensional view of data and allows further computations and fast access of summarized data. The cube can display three dimensions, for example the customers' address or location, the item type and time with quarter values Q*1, Q2, Q3, and Q4*. By providing multidimensional data views and the pre-computation of summarized data; data warehouse systems can provide inherent support for Online Analytical Processing (OLAP). The operations of OLAP make use of background knowledge regarding the domain of the data being studied to allow the presentation of data at *different levels of abstraction*. Such operations accommodate different user viewpoints. Examples of OLAP operations include **drill-down** and **roll-up**, which allow the user to view the data at different degrees of summarization.

OLAP emerged as a response to the limitations of relational databases for analytical and multidimensional data processing. OLAP databases are optimized for complex queries, multidimensional analysis, and fast retrieval of processed data.

While data cubes can exist as a simple representation of data, without any extensive capabilities to analyze large volumes, OLAP data cubes are particularly valuable for complex data analysis, including business intelligence as they provide a comprehensive view of information across different dimensions, such as time, products, locations, or customer segments. For example, if you are looking at a sales data cube, different dimensions can show you data by year, product category, locations, customers, etc.

What are the data cube operations and What are the tools for creating a data cube?

Data cubes support various operations that allow users to examine and analyze data from different perspectives. Several tools can be used to create, manage, and visualize data cubes. Some popular software platforms include Microsoft SQL Server Analysis Services (SSAS), IBM Cognos Analytics, Oracle OLAP etc. All these tools can create and analyse data cubes for large datasets.

Here is an overview of some key data cube operations:

i. Roll-up: This operation adds up all the data from a category and presents it as a singular record. It is like zooming out of the cube and looking at the data from a broader perspective.

ii. Drill-down: While trying to access a transaction on the point-of-access, users need to descend into a dimension hierarchy. For instance, drilling down on the product dimension in the sales data cube would provide detailed sales figures for each product within each region.

iii. Slicing: When users want to focus on a specific set of facts from a particular dimension, they can filter the data to focus on that subset. Slicing a sales data cube to focus on "Electronics" would restrict the data to sales of electronic products only.

iv. Dicing: Breaking the data into multiple slices from a data cube can isolate a particular combination of factors for analysis. By selecting a subset of values from each dimension, the user can focus on the point where the two dimensions intersect each other. For example, dicing the product dimension to

"Electronics" and the region dimension to "Asia" would restrict the data to sales of electronic products in the Asian region.

v. Pivoting: Pivoting means rotating the cube to view the data from a unique perspective or reorienting analysis to focus on a different aspect. Pivoting the sales data cube to swap the product and region dimensions would shift the focus from sales by product to sales by region.

Example of a data cube

Banks collect and analyze data on customer interactions with their various products and services. This data-driven approach allows banks to offer personalized services and promotions, enhancing customer satisfaction and optimizing business performance. Here is an example of how banks collect and organize data:

Table 1: Banking Products

Product Type	Description
Checking Accounts	Everyday banking and payment transactions
Credit Cards	Credit card offerings for various needs
Personal Loans	Loans for personal expenses
Mortgage loans	Home loan products for buying properties
Business Accounts	Banking services for businesses

Table 2: Time Period

Time Period
January
February
March
April
May
June
July
August
September
October

Table 3: Customer Segments

Customer ID	Customer Name	Age	Employment Status	Income Level
001	Sarah Smith	28	Employed	Moderate
001	Sarah Smith	28	Employed	Moderate
002	John Johnson	42	Self-employed	High
003	Emily Davis	60	Retired	Low
004	David Brown	35	Employed	Moderate
005	Susan Lee	48	Employed	High
006	Mark Ross	30	Self-Employed	Moderate
007	Kevin Grey	52	Employed	High

Table 4: Customer Interactions

Time Period	Product Type	Customer ID	Number of Transactions
January	Checking Accounts	001	25
February	Credit Cards	002	12
April	Mortage Loans	003	3
August	Business Account	006	17
October	Personal Loans	004	8

An example of a data cube

The bank can gain insights into individual preferences and usage patterns by tracking customer interactions with these products. They can then use this data to create targeted marketing campaigns, offer personalized services, and adapt their strategies

to seasonal trends, enhancing customer satisfaction and improving business performance in the real world.

Advantages of a data cube

Programmed data cubes can accelerate data retrieval, support multi-dimensional analysis and enable easy processing to help businesses make informed decisions based on data-driven insights. Data cubes offer several advantages over traditional data analysis methods, including:

i. Fast: Data cubes are programmed before appending the semantic layer onto it, which means most of the required calculations reside in the cache memory. These calculations expedite query response times, which helps users retrieve and analyse large datasets quickly.

ii. Efficient: The multidimensional approach enables users to identify patterns, trends and relationships that might go unnoticed in a traditional two-dimensional table. By enabling users to slice, dice, format and pivot the data along every available axis in its multi-dimensional structure.

.

iii. Scalable: Data cubes can be scaled to accommodate billions of rows of data to adjust to evolving business requirements. At the introduction of any new data warehouses, data cubes are built to remain flexible and adapt to the new order.

iv. Convenient: By processing data in advance, these cubes ensure that operations remain smooth irrespective of data volume. As the data grows, some level of abstraction can creep in from the cracks, but the robust structure and pre-calculated relationships can still conveniently handle user queries.

The differences between a data warehouse and a data cube

Data warehouses are centralized repositories for storing data from various sources. No processing operations are done on the data which is presented to the user as soon as it is asked. However, while programming a data cube, the data is processed first and then shaped into a multi-dimensional structure. Users can then run queries on the data, helping them draw insights from the available facts. The following are some of the differences between the two:

Aspect	Data Warehouse	Data Cube
Purpose	Centralized data storage	Multidimensional data model
Data Structure	Relational database	Facts, dimensions, and measures for multiple viewpoints
Data Sources	Most suitable for data storage and retrieval speed	Most suitable for analytical operations and query speed
Query Complexity	Suitable for standard SQL queries and reporting	Supports complex OLAP queries and analytical operations
Schema Design	Star schema, snowflake schema etc.	Dimensions, hierarchies, facts and measures

Data warehousing framework

This refers to a structured approach or architecture used to design, implement, and manage a data warehouse. A data warehouse is a centralized repository that stores historical data from different sources, optimized for querying and reporting. The framework generally outlines best practices, tools, technologies, and processes for collecting, storing, transforming, and analysing data.

Several data warehousing frameworks are commonly used in the industry, each with its own unique components, but they all focus on ensuring data integration, consistency, and accessibility. Here's a summary of the main data warehousing frameworks:

1. Traditional Data Warehouse Framework (ETL-based)

The traditional approach focuses on the ETL (Extract, Transform, Load) process:

i. **Extract**: Data is extracted from various source systems (databases, APIs, flat files, etc.).
ii. **Transform**: Data is cleaned, enriched, and transformed to fit the structure of the data warehouse.
iii. **Load**: Transformed data is loaded into a data warehouse.

This framework is most suitable for on-premise solutions, large-scale enterprises, or businesses requiring high levels of customization and complex analytics.

2. Modern Cloud-Based Data Warehouse Framework

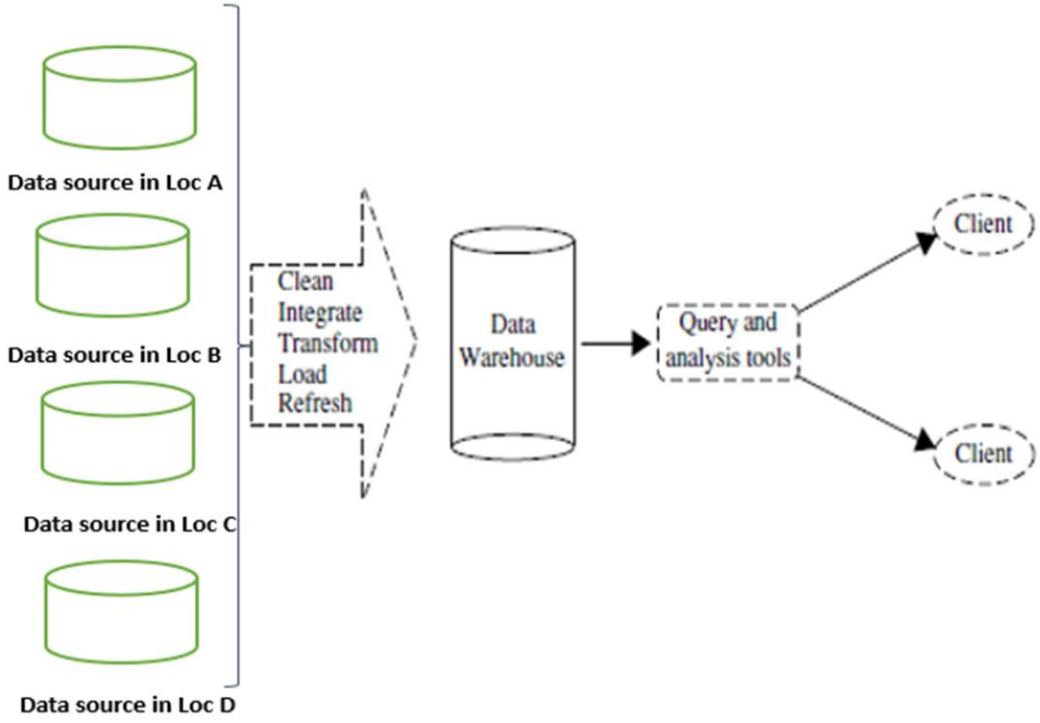
Cloud computing can be referred to as the delivery of computing services (such as storage, processing power, databases, networking, software, and more) over the Internet, With cloud computing, a new generation of data warehouses has emerged,

emphasizing flexibility, scalability, and easier integration with big data tools. This framework is commonly used in cloud-native environments.

Key components:

i. **Cloud Storage**: AWS Redshift, Google BigQuery, Snowflake, Azure Synapse.
ii. **Data Integration Tools**: Fivetran, Stitch, Matillion, or cloud-native connectors.
iii. **Data Transformation**: ELT (Extract, Load, Transform) is more common than ETL in cloud environments because of cloud computing's powerful processing capabilities (e.g., SQL-based transformations).

This framework is best for organizations that want to leverage cloud scalability, quick time-to-market, and lower operational costs.



Typical framework of a data warehouse

Benefits of Data Warehousing:

i. *It provides data consistency*: Standardizing data from multiple sources reduces discrepancies and errors, improving the quality of analysis.
ii. *Improves decision-making*: Access to accurate, timely, and consistent data leads to better business decisions.
iii. *It saves time*: Aggregating and organizing data makes it easier to access and analyze, reducing the time spent searching for information.

iv. *Historical analysis*: Storing historical data allows businesses to track trends over time and perform advanced analytics.

Challenges of Data Warehouse

1. High Initial Costs: Setting up a data warehouse requires significant investment in infrastructure, software, and resources.
2. Complex ETL Process: Extracting, transforming, and loading data from multiple sources can be time-consuming and technically challenging.
3. Data Quality Issues: Data quality issues from source systems may persist if not properly cleaned and transformed during the ETL process.
4. Maintenance and Scalability: Maintaining a data warehouse can be costly and require significant expertise, especially as data grows.

Web Mining

Introduction

The rapid growth of the Web in the past two decades has made it the largest publicly accessible data source in the world. Web mining aims to discover useful information or knowledge from Web hyperlinks, page contents, and usage logs. Based on the primary kinds of data used in the mining process, Web mining tasks can be categorized into three main types: Web structure mining, Web content mining and Web usage mining. Web structure mining discovers knowledge from hyperlinks, which represent the structure of the Web. Web content mining extracts useful information/knowledge from Web page contents. Web usage mining mines user activity patterns from usage logs and other forms of logs of user interactions with Web systems.

The World Wide Web (or the Web for short) has impacted almost every aspect of our lives. It is the biggest and most widely known information source that is easily accessible and searchable. It consists of billions of interconnected documents (called Web pages) which are authored by millions of people. Since its inception, the Web has dramatically changed our information seeking behaviour. Before the Web, finding information meant asking a friend or an expert, or buying/borrowing a book to read. However, with the Web, everything is just a few clicks away from the comfort of our homes or offices. We can not only find needed information on the Web, but also easily share our information and knowledge with others. The Web has also become an important channel for conducting businesses. We can buy almost anything from online stores without needing to go to a physical shop. The Web also provides a convenient means for us to communicate with each other, to express our views and opinions, and to discuss with people from anywhere in the world. The Web is truly a virtual society.

The Web Unique Characteristics

The rapid growth of the Web in the last decade makes it the largest publicly accessible data source in the world. *The Web has many unique characteristics, which make it possible to mine useful information and knowledge a fascinating and challenging task*. All these characteristics present both challenges and opportunities for mining and discovery of information and knowledge from the Web.

Some of these characteristics are:

1. The amount of data/information on the Web is huge and still growing. The coverage of the information is also very wide and diverse. One can find information on almost anything on the Web.

2. Data of all types exist on the Web, e.g., structured tables, semi-structured pages, unstructured texts, and multimedia files (images, audios, and videos).

3. Information on the Web is heterogeneous. Due to diverse authorships of Web pages, multiple pages may present the same or similar information using completely different words and/or formats. This makes integration of information from multiple pages a challenging problem.

4. A significant amount of information on the Web is linked. Hyperlinks exist among Web pages within a site and across different sites. Within a site, hyperlinks serve as an information organization mechanism. Across different sites, hyperlinks represent implicit conveyance of authority to the target pages. That is, those pages that are linked (or pointed) to by many other pages are usually high-quality pages or authoritative pages simply because many people trust them.

5. The information on the Web is noisy. The noise comes from two main sources. First, a typical Web page contains many pieces of information, e.g., the main content of the page, navigation links, advertisements, copyright notices, privacy policies, etc. For a particular application, only part of the information is useful. The rest is considered noise. To perform fine-grained Web information analysis and data mining, the noise should be removed. Second, due to the fact that the Web does not have quality control of information, i.e., one can write almost anything that one likes, a large amount of information on the Web is of low quality, erroneous, or even misleading.

6. The Web is also about businesses and commerce. All commercial Web sites allow people to perform useful operations at their sites, e.g., to purchase products, to pay bills, and to fill in forms. To support such applications, the Web site needs to provide many types of automated services, e.g., recommendation services using recommender systems.

7. The Web is dynamic. Information on the Web changes constantly. Keeping up with the change and monitoring the change are important issues for many applications.

8. The Web is a virtual society. It is not just about data, information and services, but also about interactions among people, organizations and automated systems. One can communicate with people anywhere in the world easily and instantly, and also express one's views and opinions on anything in Internet forums, blogs, review sites and social network

sites. Such information offers new types of data that enable many new mining tasks, e.g., opinion mining and social network analysis.

Types of Web Mining

Web mining involves extracting valuable information and insights from the vast data available on the web. It combines techniques from data mining, machine learning, natural language processing (NLP), and statistics to analyze web content, structure, and user interactions. There are three primary types of web mining and each employs various techniques to achieve its goals.

1. web content mining,
2. web structure mining,
3. web usage mining

1. Web Content Mining

Web content mining focuses on extracting useful information from the content of web pages, such as text, images, videos, and audio. The key techniques used here include:

A. **Text Mining**: The process of extracting meaningful information from text. This can include techniques like:

Natural Language Processing (NLP): Used for sentiment analysis, named entity recognition, and part-of-speech tagging.

Information Retrieval (IR): Finding relevant documents based on search queries, often using keyword-based searching and indexing.

Classification: Assigning categories to documents or data (e.g., spam vs. non-spam emails, sentiment analysis).

Clustering: Grouping similar documents together (e.g., using k-means or hierarchical clustering).

B. **Multimedia Mining**: Extracting patterns and insights from multimedia content, such as images, audio, and video. Techniques involve:

Image Processing: Analyzing images and extracting metadata or features.

Speech Recognition: Converting spoken language into text and analyzing it for insights.

C. **Semantic Web Mining**: Using semantic technologies to interpret the meaning of web content and establish relationships between entities.

2. Web Structure Mining

Web structure mining involves analyzing the structure and links between web pages. It helps in understanding how web pages are related and can be used to discover web page ranking, trends, and network behaviour. Common techniques include:

Link Analysis: Studying the hyperlinks between web pages to understand their structure and influence. Common methods:

i. *PageRank Algorithm*: Used by Google to rank web pages based on their link structure.

ii. *HITS (Hyperlink-Induced Topic Search)*: Identifies hubs and authorities in a web graph.

Graph Mining: Treating the web as a graph where pages are nodes and links are edges. Techniques like graph clustering, centrality measures, and community detection help uncover hidden patterns.

Social Network Analysis (SNA): Examining social media platforms or other online networks to understand user relationships and interactions, often using graph theory and network analysis algorithms.

3. Web Usage Mining

Web usage mining focuses on analyzing user behaviour on the web, often by examining web logs to understand how users interact with websites. The techniques used here include:

Log File Analysis: Analyzing server logs or user interaction logs to gather insights about user behaviour, popular pages, and browsing patterns.

Clickstream Analysis: Studying the sequence of clicks made by users on a website to understand their navigation patterns and identify popular areas of interest.

User Profiling: Building user profiles based on their browsing history to personalize website content, recommend products, or optimize user experience.

Session Analysis: Grouping activities into sessions to better understand how users interact with a site over time.

Additional Techniques in Web Mining

Each of these techniques plays an important role in uncovering valuable insights and patterns from the vast and varied data available on the web.

i. Data Preprocessing: Web data is often noisy or unstructured, so preprocessing steps like cleaning, normalization, and transformation are essential before applying mining techniques.

ii. Machine Learning and AI: Algorithms such as supervised and unsupervised learning, reinforcement learning, and deep learning are often employed for advanced pattern recognition and prediction tasks in web mining.

iii. Web Crawling: A technique used to gather data from websites by systematically browsing through web pages.

iv. Web Scraping: Extracting structured data from websites, often involving techniques like HTML parsing, DOM manipulation, and regular expressions.

TEXT MINING

Text mining can be broadly defined as a knowledge-intensive process in which a user interacts with a document collection over time by using a suite of analysis tools. In a manner analogous to data mining, text mining seeks to extract useful information from data sources through the identification and exploration of interesting patterns. In the case of text mining, however, the data sources are document collections, and interesting patterns are found not among formalized database records but in the unstructured textual data in the documents in these collections.

Certainly, text mining derives much of its inspiration and direction from seminal research on data mining. Therefore, it is not surprising to find that text mining and data mining systems evince many high-level architectural similarities. For instance, both types of systems rely on preprocessing routines, pattern-discovery algorithms, and presentation-layer elements such as visualization tools to enhance the browsing of answer sets. Further, text mining adopts many of the specific types of patterns in its core knowledge discovery operations that were first introduced and vetted in data mining research.

Text mining involves extracting valuable information and patterns from textual data. To do this, various techniques are used, depending on the goals and complexity of the task. Here are some key techniques employed in text mining:

1. Text Preprocessing

Tokenization: Breaking text into smaller components like words, sentences, or phrases. This is a fundamental step for analysis.
Stop-word Removal: Removing common words (e.g., "and", "the", "is") that do not carry significant meaning in the context of analysis.
Stemming and Lemmatization: Reducing words to their root form. Stemming removes prefixes or suffixes (e.g., "running" becomes "run"), while lemmatization converts a word to its base form (e.g., "better" becomes "good").
Normalization: Standardizing text to a common format, like converting all letters to lowercase or removing punctuation and special characters.

2. Text Representation Techniques

Bag-of-Words (BoW): A simple representation where text is treated as a collection of words, ignoring grammar and word order, with each word assigned a frequency count.

TF-IDF (Term Frequency-Inverse Document Frequency): A statistical measure used to evaluate the importance of a word in a document relative to a collection of documents (corpus). It helps highlight unique words in a document while downplaying frequent but less meaningful words.

Latent Semantic Analysis (LSA): A technique for extracting and representing the contextual meaning of words by analyzing the relationships between a set of documents and the terms they contain.

3. Classification

Supervised Learning: Using labelled data to train models to classify new, unseen text data. Algorithms such as Naive Bayes, Support Vector Machines (SVM), and Random Forests are commonly applied in text classification.

Text Categorization: Organizing text into predefined categories (e.g., spam vs. non-spam, sentiment analysis).

4. Clustering

Unsupervised Learning: Grouping similar documents together without predefined labels. Common algorithms include K-means clustering and hierarchical clustering.

5. Sentiment Analysis

Polarity Detection: Determining the sentiment of a piece of text (positive, negative, or neutral). This involves analyzing the text to assess the emotional tone behind words or phrases.

6. Association Rule Mining

Identifying relationships between different elements in large datasets. In text mining, this could be identifying which terms or phrases tend to appear together across different documents.

7. Text Summarization

Extractive Summarization: Identifying the most relevant sentences or phrases in a text and combining them to form a summary.

Abstractive Summarization**:** Generating new sentences that convey the core ideas of the text in a condensed form, often using techniques from deep learning.

Information Retrieval and Web Search

Web search needs no introduction. Due to its convenience and the richness of information on the Web, searching the Web is increasingly becoming the dominant information seeking method. People make fewer and fewer trips to libraries, but more and more searches on the Web. In fact, effective search engines and rich Web contents, is making writing easier by the day.

Web search has its root in information retrieval (or IR for short), a field of study that helps the user find needed information from a large collection of text documents. Traditional IR assumes that the basic information unit is a document, and a large collection of documents is available to form the text database. On the Web, the documents are Web pages.

Retrieving information simply means finding a set of documents that is relevant to the user query. A ranking of the set of documents is usually also performed according to their relevance scores to the query. The most commonly used query format is a list of keywords, which are also called terms. IR is different from data retrieval in databases using SQL queries because the data in databases are highly structured and stored in relational tables, while

information in text is unstructured. There is no structured query language like SQL for text retrieval.

It is safe to say that Web search is the single most important application of IR. To a great extent, Web search also helped IR. Indeed, the tremendous success of search engines has pushed IR to the center stage. Search is, however, not simply a straightforward application of traditional IR models. It uses some IR results, but it also has its unique techniques and presents many new problems for IR research.

First of all, efficiency is a paramount issue for Web search, but is only secondary in traditional IR systems mainly due to the fact that document collections in most IR systems are not very large. However, the number of pages on the Web is huge. For example, at the moment, Google has indexed more than 8 billion pages. Web users also demand very fast responses. No matter how effective an algorithm is, if the retrieval cannot be done efficiently, few people will use it.

Web pages are also quite different from conventional text documents used in traditional IR systems. First, Web pages have hyperlinks and anchor texts, which do not exist in traditional documents (except citations in research publications). Hyperlinks are extremely important for search and play a central role in search ranking algorithms as we will see in the next chapter. Anchor texts associated with hyperlinks too are crucial because a piece of anchor text is often a more accurate description of the page that its hyperlink points to. Second, Web pages are semi-structured.

Finally, spamming is a major issue on the Web, but not a concern for traditional IR. This is so because the rank position of a page returned by a search engine is extremely important. If a page is relevant to a query but is ranked very low (e.g., below top 30), then the user is unlikely to look at the page. If the page sells a product, then this is bad for the business. In order to improve the ranking of some target pages, “illegitimate” means, called spamming, are often used to boost their rank positions. Detecting and fighting Web spam is a critical issue as it can push low quality (even irrelevant) pages to the top of the search rank, which harms the quality of the search results and the user’s search experience.

Basic Concepts of Information Retrieval

Information retrieval (IR) is the study of helping users to find information that matches their information needs. Technically, IR studies the acquisition, organization, storage, retrieval, and distribution of information. Historically, IR is about document retrieval, emphasizing document as the basic unit. Fig. 5.1 gives a general architecture of an IR system. In Fig.5. 1, the user with information need issues a query (user query) to the retrieval system through the query operations module. The retrieval module uses the document index to retrieve those documents that contain some query terms (such documents are likely to be relevant to the query), compute relevance scores for them, and then rank the retrieved documents according to the scores. The ranked documents are then presented to the user. The document collection is also called the text database, which is indexed by the indexer for efficient retrieval.

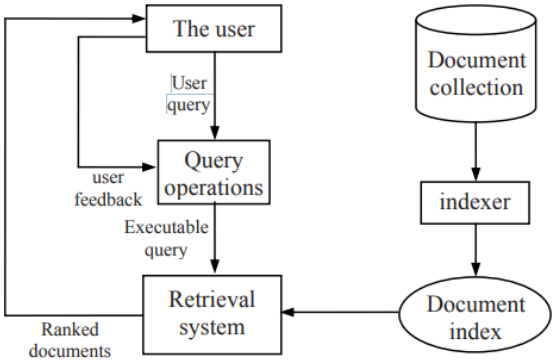


Fig. 5.1. A general IR system architecture

A user query represents the user's information needs, which is in one of the following forms:

1. Keyword queries

The user expresses his/her information needs with a list of (at least one) keywords (or terms) aiming to find documents that contain some (at least one) or all the query terms. The terms in the list are assumed to be connected with a "soft" version of the logical AND. For example, if one is interested in finding information about Web mining, one may issue the query 'Web mining' to an IR or search engine system. 'Web mining' is retreated as 'Web AND mining'. The retrieval system then finds those likely relevant documents and ranks them suitably to present to the user. Note that a retrieved document does not have to contain all the terms in the query. In some IR systems, the ordering of the words is also significant and will affect the retrieval results.

2. Boolean queries

The user can use Boolean operators, AND, OR, and NOT to construct complex queries. Thus, such queries consist of terms and Boolean operators. For example, 'data OR Web' is a Boolean query, which requests documents that contain the word 'data' or 'Web. A page is returned for a Boolean query if the query is logically true in the page (i.e., exact match). Although one can write complex Boolean queries using the three operators. Search engines usually support a restricted version of Boolean queries.

3. Phrase queries

Such a query consists of a sequence of words that makes up a phrase. Each returned document must contain at least one instance of the phrase. In a search engine, a phrase query is normally enclosed with double quotes. For example, one can issue the following phrase query (including the double quotes), "Web mining techniques and applications" to find documents that contain the exact phrase.

4. Proximity queries

The proximity query is a relaxed version of the phrase query and can be a combination of terms and phrases. Proximity queries seek the query terms within close proximity to each other. The closeness is used as a factor in ranking the returned documents or pages. For example, a document that contains all query terms close together is considered more relevant than a page in which the query terms are far apart. Some systems allow the user to specify the

maximum allowed distance between the query terms. Most search engines consider both term proximity and term ordering in retrieval.

5. Full document queries

When the query is a full document, the user wants to find other documents that are similar to the query document. Some search engines (e.g., Google) allow the user to issue such a query by providing the URL of a query page. Additionally, in the returned results of a search engine, each snippet may have a link called "more like this" or "similar pages." When the user clicks on the link, a set of pages similar to the page in the snippet is returned.

6. Natural language questions

This is the most complex case, and also the ideal case. The user expresses his/her information need as a natural language question. The system then finds the answer. However, such queries are still hard to handle due to the difficulty of natural language understanding. Nevertheless, this is an active research area, called question answering. Some search systems are starting to provide question answering services on some specific types of questions, e.g., definition questions, which ask for definitions of technical terms. Definition questions are usually easier to answer because there are strong linguistic patterns indicating definition sentences, e.g., "defined as", "refers to", etc.

The indexer is the module that indexes the original raw documents in some data structures to enable efficient retrieval. The retrieval system computes a relevance score for each indexed document to the query. According to their relevance scores, the documents are ranked and presented to the user. Note that it usually does not compare the user query with every document in the collection, which is too inefficient. Instead, only a small subset of the documents that contains at least one query term is first found from the index and relevance scores with the user query are then computed only for this subset of documents.

Information Retrieval Models

An IR model governs how a document and a query are represented and how the relevance of a document to a user query is defined. There are four main IR models: Boolean model, vector space model, language model and probabilistic model. The most commonly used models in IR systems and on the Web are the first three models earlier stated. Although these three models represent documents and queries differently, they use the same framework. They all treat each document or query as a "bag" of words or terms. Term sequence and position in a sentence or a document are ignored. That is, a document is described by a set of distinctive terms. A term is simply a word whose semantics helps remember the document's main themes.

i. Boolean Model

The Boolean model is one of the earliest and simplest information retrieval models. It uses the notion of exact matching to match documents to the user query. Both the query and the retrieval are based on Boolean algebra.

ii. Vector Space Model

This model is perhaps the best known and most widely used IR model. A document in the vector space model is represented as a weight vector, in which each component weight is computed based on some variation of Term Frequency scheme. The weight wij of term ti in document dj is no longer in {0, 1} as in the Boolean model, but can be any number.

iii. Statistical Language Model

Statistical language models (or simply language models) are based on probability and have foundations in statistical theory. The basic idea of this approach to retrieval is simple. It first estimates a language model for each document, and then ranks documents by the likelihood of the query given the language model.

Evaluation Measures

Evaluation is the key to making real progress in data mining as there is always the need to examine the resulting outputs of a model for correctness. A number of evaluation measures are available to determine the present of error in a model, this depends on the type of model i.e what the model does. In order to evaluate a regression model, i.e a model whose target is a continuous value, several alternative measures, presented in Table 5.1 can be used to evaluate the success of the numeric predictions. Four of the evaluation measures that can be used to compute the accuracy of numeric prediction as enumerated are as follows:

1. ***Mean Squared Error*** *(MSE)**:*** This is the principal and most commonly used measurement; it is sometimes referred to as objective function. The square root is taken to give it the same dimensions as the predicted value itself. Many mathematical techniques such as linear regression use the mean-squared error due to the fact that, it tends to be the easiest measure to manipulate, the mathematicians usually say, "well behaved." The MSE can be used in several instances, but here, it is being used as a performance measure. Generally, most of the performances are easy to calculate, so mean-squared error has no exceptional advantage for this purpose.

2. ***Mean Absolute Error*** *(MAE)*: This is the average of the magnitude of the individual errors regardless of their sign. Mean-Squared Error (MSE) tends to exaggerate the effect of outliers in dataset when the prediction error is larger than the others, but the MAE does not have this effect. All sizes of error are treated evenly according to their magnitude. In terms of importance, sometimes it is the *relative* rather than *absolute* error values that may be seen as vital. For example, if a 10% error is equally important whether it is an error of 50 in a prediction of 500 cases or an error of 0.2 in a prediction of 2 cases, then averages of absolute error will be meaningless, the relative error appears to be appropriate in an instance like this.
3. ***Relative Squared Error*** *(RSE)**:*** This differs a bit from the previous error measurements. Here, the error is made relative to what it would have been if a simple predictor had been used. The simple predictor in question is just the average of the actual values from the training data, which is denoted by '*a'* in Table 5.1. Thus, relative squared error takes the total squared error and normalizes it by dividing by the total squared error of the default predictor. The root relative squared error is obtained in the obvious way.

4. ***Relative Absolute Error*** *(RAE):* This is simply the total absolute error, with the same kind of normalization. In the relative error measures, the errors are normalized by the error of

the simple predictor that predicts average values. These measurements of numeric predictions are further summarized in Table 5.1

Table 5.1 Performance measure of numeric prediction

Evaluating measures	Formula
Mean Square Error	$\frac{(\rho_1 - a_1)^2 + ... + (\rho_n - a_n)^2}{n}$
Mean Absolute Error	$\frac{\lvert\rho_1 - a_1\rvert + ... + \lvert\rho_n - a_n\rvert}{n}$
Relative-Square Error*	$\frac{(\rho_1 - a_1)^2 + ... + (\rho_n - a_n)^2}{(a_1 - \bar{a})^2 + ... + (a_n - \bar{a})^2}$
Relative Absolute Error*	$\frac{\lvert\rho_1 - a_1\rvert + ... + \lvert\rho_n - a_n\rvert}{\lvert a_1 - \bar{a}\rvert + ... + \lvert a_n - \bar{a}\rvert}$

* $\bar{a}$ is the mean value over the training data.

Precision, Recall, F-score and Break-even Point

In some applications, we are only interested in one class. This is particularly true for text and Web applications. For example, we may be interested in only the documents or web pages of a particular topic. Also, in classification involving skewed or highly imbalanced data, e.g., network intrusion and financial fraud detection, we are typically interested in only the minority class. The class that the user is interested in is commonly called the positive class, and the rest negative classes (the negative classes may be combined into one negative class). Accuracy is not a suitable measure in such cases because we may achieve a very high accuracy, but may not identify a single intrusion.

Accuracy, Recognition rate $\frac{TP + TN}{P + N}$

For instance, 99% of the cases are normal in an intrusion detection data set. Then a classifier can achieve 99% accuracy (without doing anything) by simply classifying every test case as “not intrusion”. This is, however, useless.

Precision and recall are more suitable in such applications because they measure how precise and how complete the classification is on the positive class. It is convenient to introduce these measures using a confusion matrix. A confusion matrix contains information about actual and predicted results given by a classifier.

Table 5.2. Confusion matrix of a classifier.

	Classified positive	Classified negative
Actual positive	TP	FN
Actual negative	FP	TN

where

TP: the number of correct classifications of the positive examples (**true positive**)
FN: the number of incorrect classifications of positive examples (**false negative**)
FP: the number of incorrect classifications of negative examples (**false positive**)
TN: the number of correct classifications of negative examples (**true negative**)

Based on the confusion matrix, the precision (p) and recall (r) of the positive class are defined as follows:

$$p = \frac{TP}{TP + FP}. \qquad r = \frac{TP}{TP + FN}.$$

In words, precision p is the number of correctly classified positive examples divided by the total number of examples that are classified as positive. Recall r is the number of correctly classified positive examples divided by the total number of actual positive examples in the test set. The intuitive meanings of these two measures are quite obvious. However, it is hard to compare classifiers based on two measures, which are not functionally related. For a test set, the precision may be very high but the recall can be very low, and vice versa**.**

Example 1: A test data set has 100 positive examples and 1000 negative examples. After classification using a classifier, we have the following confusion matrix (Table 5.2),

Table 5.2 Confusion matrix of a classifier.

	Classified positive	Classified negative
Actual positive	1	99
Actual negative	0	1000

This confusion matrix gives the precision p = 100% and the recall r = 1% because we only classified one positive example correctly and classified no negative examples wrongly.

Although in theory precision and recall are not related, in practice high precision is achieved almost always at the expense of recall and high recall is achieved at the expense of precision. In an application, which measure is more important depends on the nature of the application. If we need a single measure to compare different classifiers, the F-score is often used.

$$F = \frac{2pr}{p + r}.$$

The F-score (also called the F1-score) is the harmonic mean of precision and recall.

$$F = \frac{2}{\frac{1}{p} + \frac{1}{r}}.$$

The harmonic mean of two numbers tends to be closer to the smaller of the two. Thus, for the F-score to be high, both p and r must be high. There is also another measure, called precision and recall breakeven point, which is used in the information retrieval community.

The break- even point is when the precision and the recall are equal. This measure assumes that the test cases can be ranked by the classifier based on their likelihoods of being positive. For instance, in decision tree classification, we can use the confidence of each leaf node as the value to rank test cases.

Example 2: We have the following ranking of 20 text documents. 1 represents the highest rank and 20 represents the lowest rank. "+" ("–") represents an actual positive (negative) document.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
+	+	+	–	+	–	+	–	+	+	–	–	+	–	–	–	+	–	–	+

Assume that the test set has 10 positive examples.

At rank 1:	$p = 1/1 = 100\%$	$r = 1/10 = 10\%$
At rank 2:	$p = 2/2 = 100\%$	$r = 2/10 = 20\%$
...	...	...
At rank 9:	$p = 6/9 = 66.7\%$	$r = 6/10 = 60\%$
At rank 10:	$p = 7/10 = 70\%$	$r = 7/10 = 70\%$

The breakeven point is $p = r = 70\%$. Note that interpolation is needed if such a point cannot be found.

Pitfalls or Challenges of Data Mining

By understanding and addressing these pitfalls, organizations can better leverage data mining tools while minimizing risks and ensuring the validity, fairness, and ethics of their outcomes.. Data mining is known to be very powerful and insightful, the following pitfalls can lead to unintended consequences if not managed properly. Here are some of the key challenges:

1. Data Privacy and Security Issues

Data mining can inadvertently expose personal or confidential information, leading to privacy violations. This is particularly concerning when working with personal, financial, or medical data. Also, if security measures are not sufficiently robust, unauthorized individuals could access sensitive datasets, leading to potential data breaches.

2. Data Quality and Inaccuracy

Data mining models are only as good as the data they are based on. Incomplete, incorrect, or inconsistent data can lead to flawed results. Irrelevant or noisy data is included in the analysis, it can obscure meaningful patterns and reduce the accuracy of predictions or classifications.

3. Overfitting

Overfitting to Training Data: A model that is too closely tailored to a specific dataset may fail to generalize well to unseen data. This results in high performance on training data but poor predictions in real-world applications.

4. Interpretability and Transparency

Black Box Models: Some advanced algorithms (e.g., deep learning) are hard to interpret, making it difficult for users to understand how decisions are made. This lack of transparency can undermine trust in the model's outcomes.

5. Misleading Patterns

Correlation vs Causation: Data mining often uncovers correlations, but these may not imply causality. Relying solely on correlations can lead to erroneous or misleading conclusions.

6. Cost and Resource Intensive

- **High Costs**: Data mining often requires significant investments in hardware, software, and human resources for processing and analyzing large datasets.
- **Time and Effort**: Cleaning and preparing data, tuning models, and validating results can be time-consuming, especially for complex analyses.

7. Legal and Ethical Concerns

Mining data without appropriate permissions or consent can lead to violations of privacy laws. Also, there may be concerns over the ethical implications of using data mining in ways that may harm individuals or communities, such as manipulating consumer behavior or influencing political opinions.

8. Lack of Domain Expertise

Misinterpretation of Results: If data mining is conducted without a solid understanding of the domain, the results may be misinterpreted or used inappropriately. A lack of subject-matter expertise can lead to incorrect or irrelevant conclusions.

Inability to Validate Models: Domain knowledge is also critical for validating and assessing the effectiveness of data mining models in real-world situations.

THE END