1 a) Give the definition of data warehousing.Discuess the need of data warehousing(06 Marks)

A data warehouse is a subject-oriented, integrated, time-variant and non-volatile collection of data in support of management's decision making process.

Subject-Oriented: A data warehouse can be used to analyze a particular subject area. For example, "sales" can be a particular subject.

Integrated: A data warehouse integrates data from multiple data sources. For example, source A and source B may have different ways of identifying a product, but in a data warehouse, there will be only a single way of identifying a product.

Time-Variant: Historical data is kept in a data warehouse. For example, one can retrieve data from 3 months, 6 months, 12 months, or even older data from a data warehouse. This contrasts with a transactions system, where often only the most recent data is kept. For example, a transaction system may hold the most recent address of a customer, where a data warehouse can hold all addresses associated with a customer.

Non-volatile: Once data is in the data warehouse, it will not change. So, historical data in a data warehouse should never be altered.

NEED OF DW:

- To provide a single version of truth about enterprise information. This may appear rather obvious but it is not uncommon in an enterprise for two database systems to have two different versions of the truth. In many years of working in universities, I have rarely found a university in which everyone agrees with financial figures of income and expenditure at each reporting time during the year.
- To speed up ad hoc reports and queries that involve aggregations across many attributes (that is, may GROUP BY's) which are resource intensive. The managers require trends, sums and aggregations that allow, for example, comparing this year's performance to last year's or preparation of forecasts for next year.
- To provide a system in which managers who do not have a strong technical background are able to run complex queries. If the managers are able to access the information they require, it is likely to reduce the bureaucracy around the managers.
- To provide a database that stores relatively clean data. By using a good ETL process, the data warehouse should have data of high quality. When errors are discovered it may be desirable to correct them directly in the data warehouse and then propagate the corrections to the OLTP systems.
- To provide a database that stores historical data that may have been deleted from the OLTP systems. To improve response time, historical data is usually not retained in OLTP systems other than that which is required to respond to customer queries. The data warehouse can then store the data that is purged from the OLTP systems

1 b) Give the diffrence between OLTP and DW

PROPERTY	OLTP	DATA WAREHOUSE
Nature of the database	3NF	multidimensional
Indexes	few	many
Joins	many	some
Duplicated data	Normalized data	Denormalized data
Derived data and aggregations	rare	commonly
Queries	Mostly predefined	Mostly adhoc
Nature of queries	Mostly simple	Mostly complex
updates	All the time	Not allowed
Historical data	Often not available	essential

1 C) Discuess the characteristics of operational data store with its design and implementaion issues.

An operational data store is used for operational reporting and as a source of data for the Enterprise Data Warehouse (EDW). It is used for operational reporting, controls and decision making, as opposed to the EDW, which is used for tactical and strategic decision support.

An operational data store will take transactional data from one or more production system and loosely integrate it, in some respects it is still subject oriented, integrated and time variant, but without the volatility constraints. This integration is mainly achieved through the use of EDW structures and content.

1. *Instance identity problem:* The same customer or client may be represented slightly different in different source systems. For example, my name is represented as Gopal Gupta in some systems and as GK Gupta in others. Given that the name is unusual for data entry staff in Western countries, it is sometimes misspelled as Gopal Gopta or Gopal Gupta or some other way. The name may also be represented as Professor GK Gupta, Dr GK Gupta or Mr GK Gupta. There is thus a possibility of mismatching between the different systems that needs to be identified and corrected.

2. *Data errors:* Many different types of data errors other than identity errors are possible. For example: · Data may have some missing attribute values.

· Coding of some values in one database may not match with coding in other databases (i.e. different codes with the same meaning or same code for different meanings)

· Meanings of some code values may not be known.

· There may be duplicate records.

· There may be wrong aggregations.

· There may be inconsistent use of nulls, spaces and empty values.

· Some attribute values may be inconsistent (i.e. outside their domain)

· Some data may be wrong because of input errors.

· There may be inappropriate use of address lines.

· There may be non-unique identifiers.

The ETL process needs to ensure that all these types of errors and others are resolved using a sound Technology.

3. *Record linkage problem:* Record linkage relates to the problem of linking information from different databases that relate to the same customer or client. The problem can arise if a unique identifier is not available in all databases that are being linked. Perhaps records from a database are being linked to records from a legacy system or to information from a spreadsheet. Record linkage can involve a large number of record comparisons to ensure linkages that have a high level of accuracy.

4. *Semantic integration problem:* This deals with the integration of information found in heterogeneous OLTP and legacy sources. Some of the sources may be relational, some may not be. Some may be even in text documents. Some data may be character strings while others may be integers.

5. *Data integrity problem:* This deals with issues like referential integrity, null values, domain of values, etc. Overcoming all these problems is often a very tedious work. Many errors can be difficult to identify. In some cases one may be forced to ask the question how accurate the data ought to be since improving the accuracy is always going to require more and more resources and completely fixing all problems may be unrealistic. Checking for duplicates is not always easy. The data can be sorted and duplicates removed although for large files this can be expensive. In some cases the duplicate records are not identical. In these cases checks for primary key may be required. If more than one record has the same primary key then it is likely to be because of duplicates. A sound theoretical background is being developed for data cleaning techniques. It has been suggested that data cleaning should be based on the following five steps:

1. *Parsing:* Parsing identifies various components of the source data files and then establishes relationships between those and the fields in the target files. The classical example of parsing is identifying the various components of a person's name and address.

2. *Correcting:* Correcting the identified components is usually based on a variety of sophisticated techniques including mathematical algorithms. Correcting may involve use of other related information that may be available in the enterprise.

3. *Standardizing:* Business rules of the enterprise may now be used to transform the data to standard form. For example, in some companies there might be rules on how name and address are to be represented.

4. *Matching:* Much of the data extracted from a number of source systems is likely to be related. Such data needs to be matched.

5. *Consolidating:* All corrected, standardized and matched data can now be consolidated to build a single version of the enterprise data.

2 a) Describe the operations of data cube(10 Marks)

DATA CUBE OPERATIONS

The common data cubes operations are:

· Roll-p

· Drill-down

· Slice and dice

· Pivot

Roll-up

Roll-up is like zooming out on the data cube. It is required when the user needs further abstraction or less detail. This operation performs further aggregations on the data, for example, from single degree programs to all programs offered by a School or department, from single countries to a collection of countries, and from individual semesters to academic years. Often a hierarchy defined on a dimension is useful in the roll-up operation as suggested by the example of countries and regions.

Drill-down

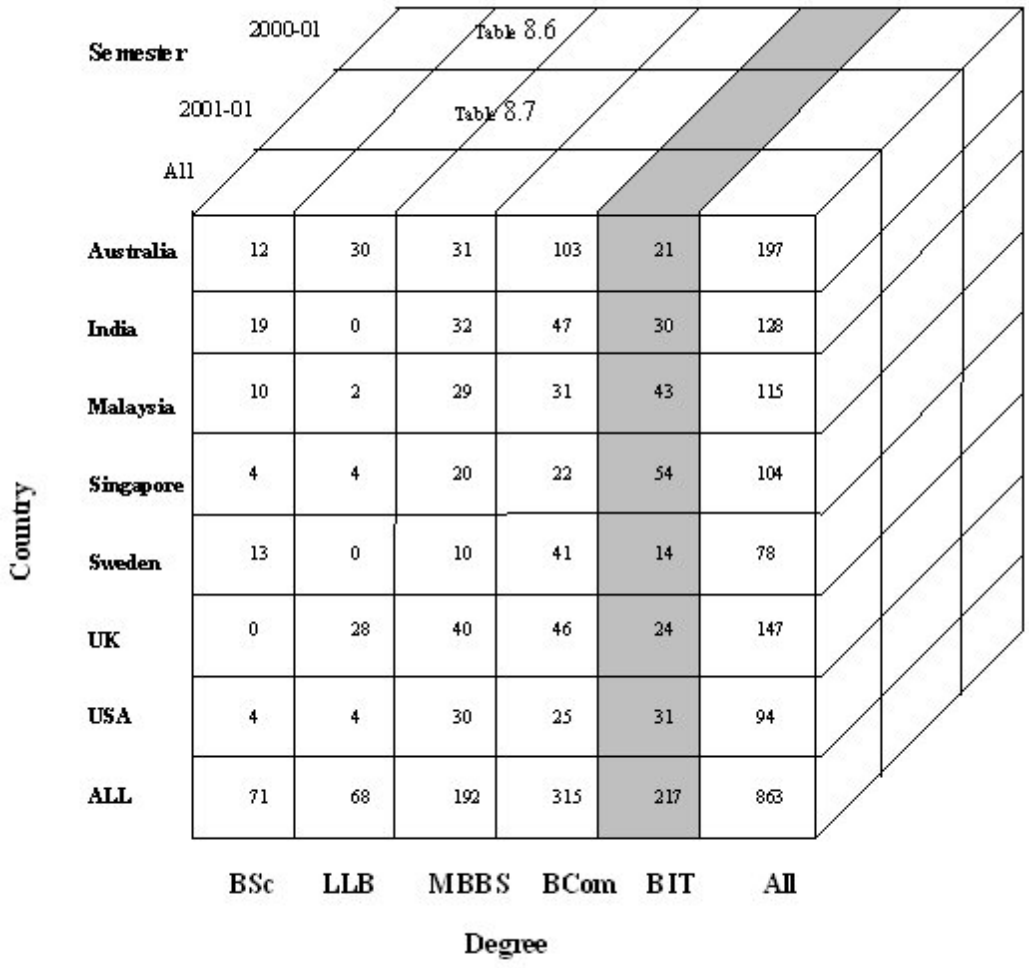
Drill-down is like zooming in on the data and is therefore the reverse of roll-up. It is an appropriate operation when the user needs further details or when the user wants to partition more finely or wants to focus on some particular values of certain dimensions. Drill-down adds more details to the data. Hierarchy defined on a dimension may be involved in drill-down. For example, a higher level views of student data,

Slice and dice

Slice and dice are operations for browsing the data in the cube. The terms refer to the ability to look at information from different viewpoints. A slice is a subset of the cube corresponding to a

single value for one or more members of the dimensions. For example, a slice operation is performed when the user wants a selection on one dimension of a three-dimensional cube resulting in a two- dimensional site. Let the degree dimension be fixed as degree = BIT. The slice will not include any information about other degrees. The information retrieved therefore is more like a two-dimensional cube for degree = BIT

The dice operation is similar to slice but dicing does not involve reducing the number of dimensions. A dice is obtained by performing a selection on two or more dimensions. For example, one may only be interested in degrees BIT and BCom and countries Australia, India, and Malaysia for semesters 2000-01 and 2000-01. The result is a three-dimensional cube and we show it by placed on top of each other. For example one may only be interested in the degrees BIT and BCom and the countries, Australia, India



Pivot or Rotate

The pivot operation is used when the user wishes to re-orient the view of the data cube. It may involve swapping the rows and columns, or moving one of the row dimensions into the column dimension. For example, the cube gives the dimension degree along the x-axis, country along the y-axis and starting semester along the z-axis (or the vertical axis). One may want to swap the dimensions country and starting semester. The cube will then consist of several tables.

2 b) present five major characteristics from codd's rule

1. Multidimensional conceptual view: As noted above, this is central characteristic of an OLAP system. By requiring a multidimensional view, it is possible to carry out operations like slice and dice.

2. Accessibility (OLAP as a mediator): The OLAP software should be sitting between data sources (e.g data warehouse) and an OLAP front-end.

3. Batch extraction vs interpretive: An OLAP system should provide multidimensional data staging plus precalculation of aggregates in large multidimensional databases.

4. Multi-user support: Since the OLAP system is shared, the OLAP software should provide many normal database operations including retrieval, update, concurrency control, integrity and security.

5. Storing OLAP results: OLAP results data should be kept separate from source data. Read-write OLAP applications should not be implemented directly on live transaction data if OLAP source systems are supplying information to the OLAP system directly.

6. Extraction of missing values: The OLAP system should distinguish missing values from zero values. A large data cube may have a large number of zeros as well as some missing values. If a distinction is not made between zero values and missing values, the aggregates are likely to be computed incorrectly

7. Treatment of missing values: An OLAP system should ignore all missing values regardless of their source. Correct aggregate values will be computed once the missing values are ignored.

8. Uniform reporting performance: Increasing the number of dimensions or database size should not significantly degrade the reporting performance of the OLAP system. This is a good objective although it may be difficult to achieve in practice.

9. Generic dimensionality: An OLAP system should treat each dimension as equivalent in both is structure and operational capabilities. Additional operational capabilities may be granted to selected dimensions but such additional functions should be grantable to any dimension.

10. Unlimited dimensions and aggregation levels: An OLAP system should allow unlimited dimensions and aggregation levels. In practice, the number of dimensions is rarely more than 10 and the number of hierarchies rarely more than six.

2 c) Explain the diffrence between MOLAP and ROLAP

Property	MOLAP	ROLAP
Data structure	**Multidimensional database using sparse arrays**	**Relational tables (each cell is a row)**
Disk space	**Separate database for data cube; large for large data cubes**	**May not require any space other than that available in the data warehouse**
Retrieval	**Fast(pre-computed)**	**Slow(computes on-the-fly)**
Scalability	**Limited (cubes can be very large)**	**Excellent**
Best suited for	**Inexperienced users, limited set of queries**	**Experienced users, queries change frequently**
DBMS facilities	**Usually weak**	**Usually very strong**

3 a) Explain various tasks of data mining with example for each

Data mining functionalities are used to specify the kind of patterns to be found in data mining tasks. In general, data mining tasks can be classified into two categories:
• Descriptive

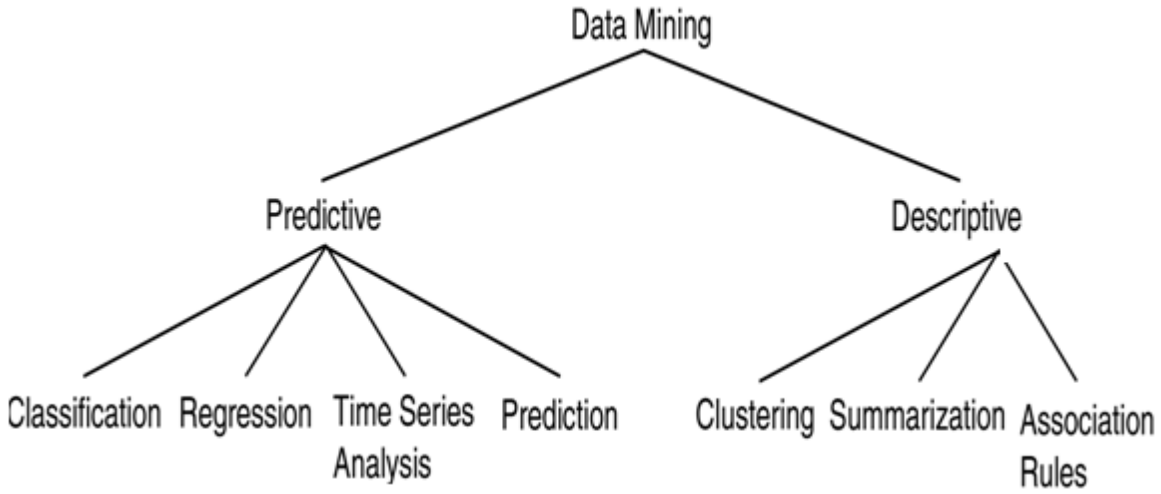
• predictive

• Predictive tasks. The objective of these tasks is to predict the value of a particular attribute based on the values of other attribute.

– Use some variables (independent/explanatory variable) to predict unknown or future values of other variables (dependent/target variable).
• Description Methods: Here the objective is to derive patterns that summarize the underlying relationships in data.

– Find human-interpretable patterns that describe the data. There are four core tasks in Data Mining

There are four core tasks in Data Mining: i. Predictive modeling ii. Association analysis iii. Clustering analysis, iv. Anomaly detection
Descriptive mining tasks characterize the general properties of the data in the database. Predictive mining tasks perform inference on the current data in order to make predictions.



1). predictive method
Find some missing or unavailable data values rather than class labels referred to as prediction. 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. Prediction also encompasses the identification of distribution trends based on the available data. Example: Predicting flooding is difficult problem. One approach is uses monitors placed at various points in the river. These monitors collect data relevant to flood prediction: water level, rain amount, time, humidity etc. These water levels at a potential flooding point in the river can be predicted based on the data collected by the sensors upriver from this point. The prediction must be made with respect to the time the data were collected

Classification:
• It predicts categorical class labels

• It classifies data (constructs a model) based on the training set and the values (class labels) in a classifying attribute and uses it in classifying new data

• Typical Applications

o credit approval

o target marketing

o medical diagnosis

o treatment effectiveness analysis

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 rule
2) Decision trees
3) Neural network.

3 b) Explain (i) Data mining applications (ii) issues in proximity calculation

Essentially most data mining techniques that we are concerned about are designed to discover and match profiles. The aims of the majority of such data mining activities are laudable but the techniques are not always perfect. What happens if a person matches the profile but does not belong to the category? Perhaps it is not a matter of great concern if a telecommunications company labels a person as one that is likely to switch and then decides to target that person with a special campaign designed to encourage the person to stay. On the other hand, if the Customs department identifies a person as fitting the profile of a drug smuggler then that person is likely to undergo a special search whenever he/she returns home from overseas and perhaps at other airports if the customs department of one country shares information with other countries. This would be a matter of much more concern to governments. Knowledge about the classification or profile of an individual who has been so classified or profiled may lead to disclosure of personal information with some given probability. The characteristics that someone may be able to deduce about a person with some possibility may include sensitive information, for example, race, religion, travel history, and level of credit card expenditure.
Data mining is used for many purposes that are beneficial to society, as the list of some of the common aims of data mining below shows.
· The primary aim of many data mining applications is to understand the customer better and improve customer services.

Some applications aim to discover anomalous patterns in order to help identify, for example, fraud, abuse, waste, terrorist suspects, or drug smugglers.

- In many applications in private enterprises, the primary aim is to improve the profitability of an enterprise
- The primary purpose of data mining is to improve judgement, for example, in making diagnoses, in resolving crime, in sorting out manufacturing problems, in predicting share prices or currency movements or commodity prices.

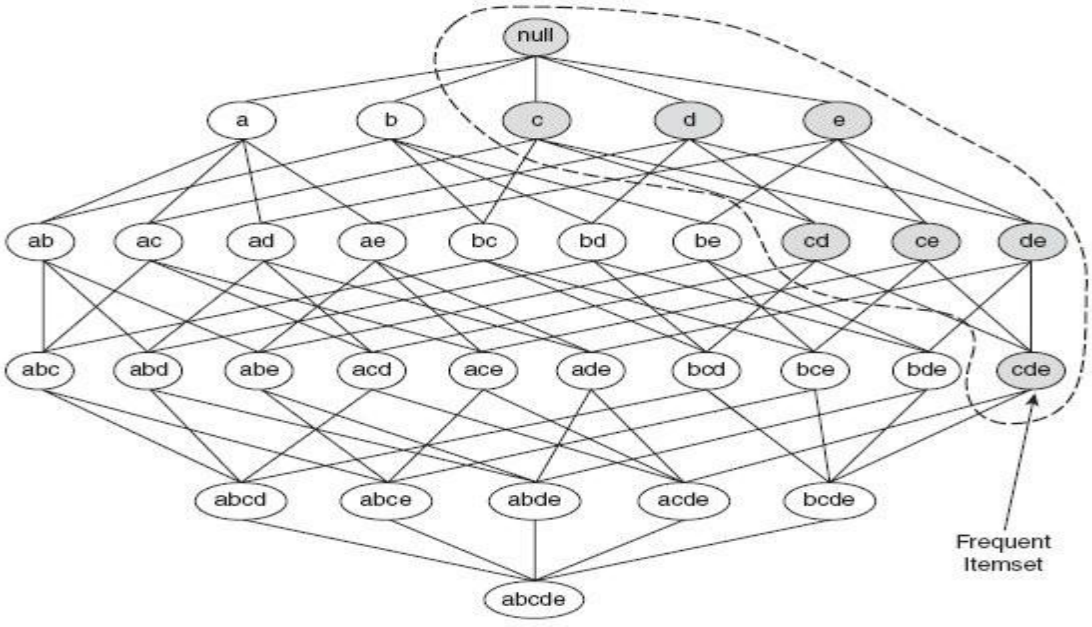
- In some government applications, one of the aims of data mining is to identify criminal and fraud activities.
- In some situations, data mining is used to find patterns that are simply not possible without the help of data mining, given the huge amount of data that must be processed.

4 a) What is Frequent Itemset Generation .Explain Frequent Itemset Generation using Apriori algorithm.

Frequent Itemset Generation, whose objective is to find all the item-sets that satisfy the minsup threshold. These itemsets are called frequent itemsets.

The Apriori Principle

This section describes how the support measure helps to reduce the number of candidate itemsets explored during frequent itemset generation. The use of support for pruning candidate itemsets is guided by the following principle. Theorem 4.1 (Apriori Principle). If an itemset is frequent, then all of its subsets must also be frequent. To illustrate the idea behind the Apriori principle, consider the itemset lattice shown in Figure 4.3. Suppose {c, d, e} is a frequent itemset. Clearly, any transaction that contains {c, d, e} must also contain its subsets, {c, d},{c, e}, {d, e}, {c}, {d}, and {e}. As a result, if {c, d, e} is frequent, then all subsets of {c, d, e} (i.e., the shaded itemsets in this figure) must also be frequent.

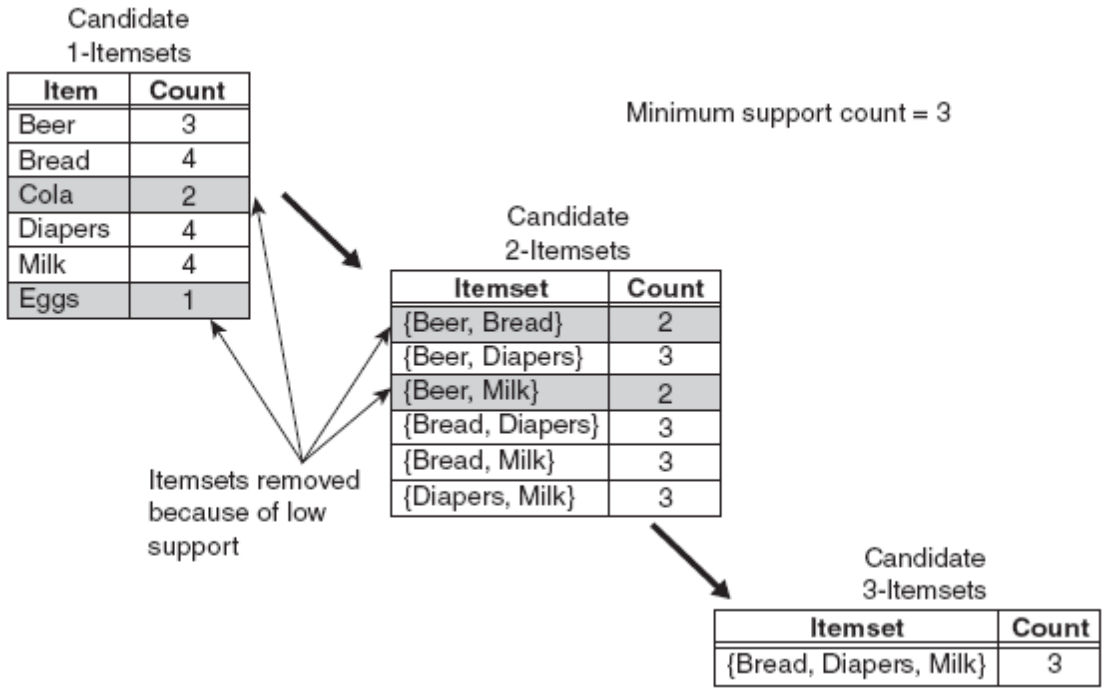


If {c, d, e} is frequent, then all subsets of this itemset are frequent. Conversely, if an itemset such as {a, b} is infrequent, then all of its supersets must be infrequent too. As illustrated in Figure 6.4, the entire subgraph containing the supersets of {a, b} can be pruned immediately once {a, b}

is found to be infrequent. This strategy of trimming the exponential search space based on the support measure is known as support-based pruning. Such a pruning strategy is made possible by a key property of the support measure, namely, that the support for an itemset never exceeds the support for its subsets. This property is also known as the anti-monotone property of the support measure.

Definition 4.2 (Monotonicity Property). Let I be a set of items, and J =2I be the power set of I. A measure f is monotone (or upward closed)

$$\forall X, Y \in J : \ (X \subseteq Y) \longrightarrow f(X) \leq f(Y),$$



Given table We assume that the support threshold is 60%, which is equivalent to a minimum support count equal to 3. Apriori principle ensures that all supersets of the infrequent 1-itemsets must be infrequent. Because there are only four frequent 1-itemsets, the number of candidate 2-itemsets generated by the algorithm is = 6. Two of these six candidates, {Beer, Bread} and {Beer, Milk}, are subsequently found to be infrequent after computing their support values. The remaining four candidates are frequent, and thus will be used to generate candidate 3-itemsets. Without support- based pruning, there are = 20 candidate 3-itemsets that can be formed using the six items given in this example. With the Apriori principle, we only need to keep candidate 3-itemsets whose subsets are frequent. The only candidate that has this property is {Bread,Diapers,Milk}. The effectiveness of the Apriori pruning strategy can be shown by counting the number of candidate itemsets generated. A brute-force strategy of enumerating all itemsets (up to size 3) as candidates will produce

$$\binom{6}{1} + \binom{6}{2} + \binom{6}{3} = 6 + 15 + 20 = 41$$

candidates.With the Apriori principle, this number decreases to

$$\binom{6}{1} + \binom{4}{2} + 1 = 6 + 6 + 1 = 13$$

candidates, which represents a 68% reduction in the number of candidate itemsets even in this simple example. The pseudocode for the frequent itemset generation part of the Apriori algorithm is shown in Algorithm 4.1. Let Ck denote the set of candidate k-itemsets and Fk denote the set of frequent k- itemsets:

• The algorithm initially makes a single pass over the data set to determine the support of each item. Upon completion of this step, the set of all frequent 1-itemsets, F1, will be known (steps 1 and 2).

• Next, the algorithm will iteratively generate new candidate k-itemsets using the frequent (k

- 1)-itemsets found in the previous iteration (step 5). Candidate generation is implemented using a function called apriori-gen

Algorithm 6.1 Frequent itemset generation of the *Apriori* algorithm.

1: $k = 1$.
2: $F_k = \{ i \mid i \in I \wedge \sigma(\{i\}) \geq N \times minsup\}$. {Find all frequent 1-itemsets}
3: **repeat**
4: $k = k + 1$.
5: $C_k = \text{apriori-gen}(F_{k-1})$. {Generate candidate itemsets}
6: **for** each transaction $t \in T$ **do**
7: $C_t = \text{subset}(C_k, t)$. {Identify all candidates that belong to t}
8: **for** each candidate itemset $c \in C_t$ **do**
9: $\sigma(c) = \sigma(c) + 1$. {Increment support count}
10: **end for**
11: **end for**
12: $F_k = \{ c \mid c \in C_k \wedge \sigma(c) \geq N \times minsup\}$. {Extract the frequent k-itemsets}
13: **until** $F_k = \emptyset$
14: Result $= \bigcup F_k$.

• To count the support of the candidates, the algorithm needs to make an additional pass over the data set (steps 6–10). The subset function is used to determine all the candidate itemsets in Ck that are contained in each transaction t.

• After counting their supports, the algorithm eliminates all candidate itemsets whose support counts are less than minsup (step 12).

• The algorithm terminates when there are no new frequent itemsets generated.

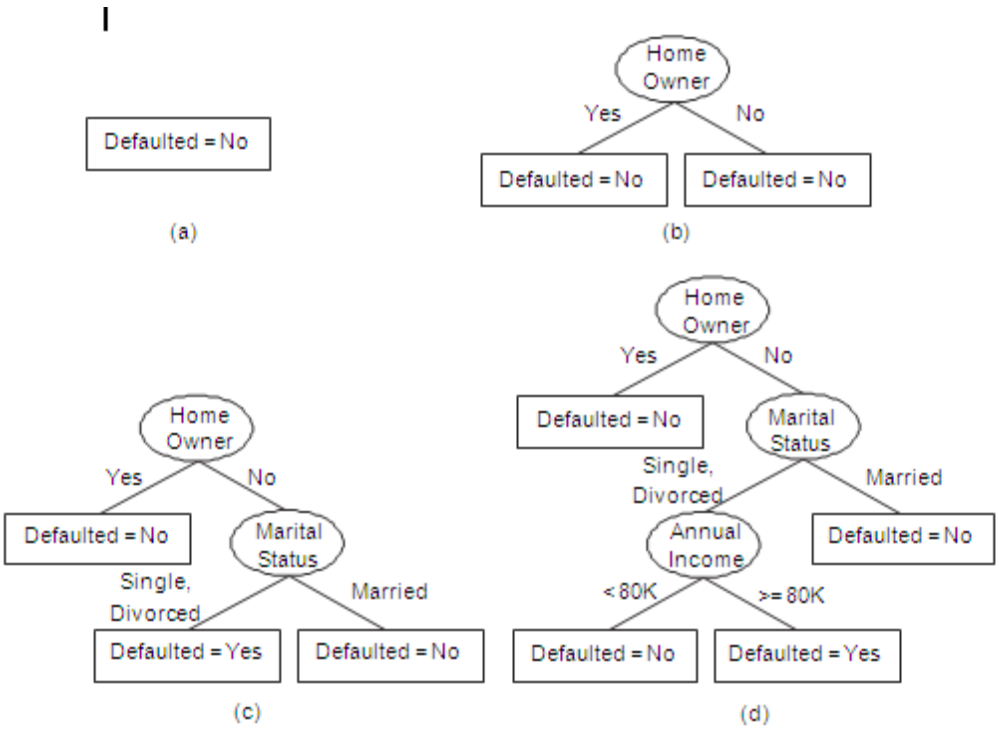
5 a) Explain Hunt's algorithm.Using Hunt's algorithm write decision for the following data

Tid	Home Owner	Marital Status	Annual Income	Defaulted Borrower
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

Hunt's Algorithm
In Hunt's algorithm, a decision tree is grown in a recursive fashion by partitioning the training records into successively purer subsets. Let Dtbe the set of training records that are associated with node t and y = {y1, y2, . . . , yc} be the class labels. The following is a recursive definition of Hunt's algorithm.
Step 1: If all the records in Data belong to the same class yt, then t is a leaf node labeled as yt.
Step 2: If Data contains records that belong to more than one class, an attribute test condition is selected to partition the records into smaller subsets. A child node is created for each outcome of the test condition and the records in Dt are distributed to the children based on the outcomes. The algorithm is then recursively applied to each child node.

To illustrate how the algorithm works, consider the problem of predicting whether a loan applicant will repay her loan obligations or become delinquent, subsequently defaulting on her loan. A training set for this problem can be constructed by examining the records of previous borrowers. In the example shown in Figure 4.6, each record contains the personal information of a borrower along with a class label indicating whether the borrower has defaulted on loan payments. The initial tree for the classification problem contains a single node with class label Defaulted = No (see Figure 3.7(a)), which means that most of the borrowers successfully repaid their loans. The tree, however, needs to be redefined since the root node contains records from both classes. The records are
Subsequently divided into smaller subsets based on the outcomes of the Home Owner test condition, as shown in Figure 3.7(b). The justification for choosing this attribute test condition will be discussed later. For now, we will assume that this is the best criterion for splitting the data at this point. Hunt's algorithm is then applied recursively to each child of the root node. From the training set given in Figure 3.6, notice that all borrowers who are home owners

successfully repaid their loans. The left child of the root is therefore a leaf node labelled Defaulted = No (see Figure 3.7(b)). For the right child, we need to continue applying the recursive step of Hunt's algorithm until all the records belong to the same class. The trees resulting from each recursive step are shown in Figures 3.7(c) and (d).



Hunt's algorithm will work if every combination of attribute values is present in the training data and each combination has a unique class label. These assumptions are too stringent for use in most practical situations. Additional conditions are needed to handle the following cases:
1. It is possible for some of the child nodes created in Step 2 to be empty; i.e., there are no records associated with these nodes. This can happen if none of the training records have the combination of attribute values associated with such nodes. In this case the node is declared a leaf node with the same class label as the majority class of training records associated with its parent node.

2. In Step 2, if all the records associated with D_t have identical attribute values (except for the class label), then it is not possible to split these records any further. In this case, the node is declared a leaf node with the same class label as the majority class of training records associated with this node.

5 b) Explain the various meassures for selecting the best split.

There are many measures that can be used to determine the best way to split the records. These measures are defined in terms of the class distribution of the records before and after splitting. Let $p(i|t)$ denote the fraction of records belonging to class i at a given node t. We sometimes omit the reference to node t and express the fraction as pi. In a two-class problem, the class distribution at any node can be written as (p0, p1), where p1 = 1 − p0. The class distribution before splitting is (0.5, 0.5) because there are an equal number of records from each class. If we split the data using the Gender attribute, then the class distributions of the child nodes are (0.6, 0.4) and (0.4, 0.6), respectively. Although the classes are no longer evenly distributed, the child nodes still contain records from both

classes. Splitting on the second attribute, Car Type will result in purer partitions. The measures developed for selecting the best split are often based on the degree of impurity of the child nodes. The smaller the degree of impurity, the more skewed the class distribution. For example, a node with class distribution (0, 1) has zero impurity, whereas a node with uniform class distribution (0.5, 0.5) has the highest impurity. Examples of impurity measures include

$$\text{Entropy}(t) = -\sum_{i=0}^{c-1} p(i|t)\log_2 p(i|t),$$

$$\text{Gini}(t) = 1 - \sum_{i=0}^{c-1} [p(i|t)]^2,$$

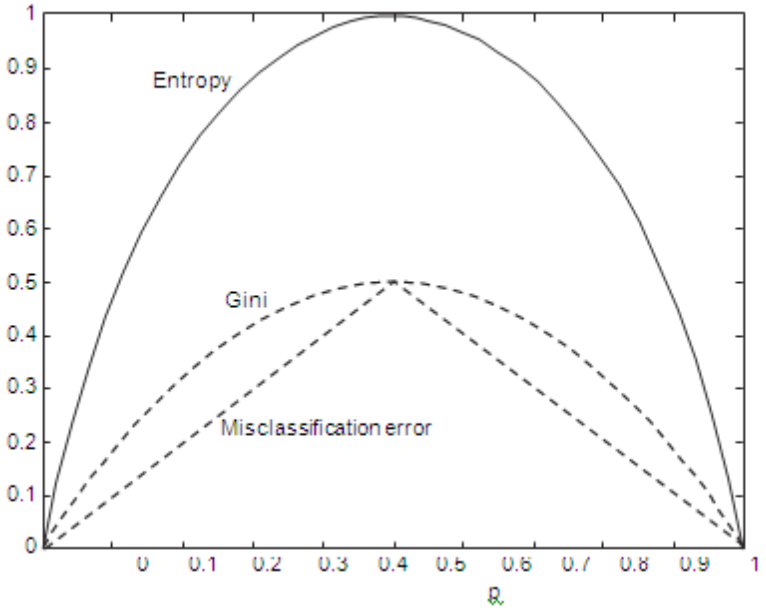


Figure 3.**13** Comparison among the impurity measures for binary classification problems.

Figure 3.13 compares the values of the impurity measures for binary classification problems. p refers to the fraction of records that belong to one of the two classes. Observe that all three measures attain their maximum value when the class distribution is uniform (i.e., when p = 0.5). The minimum values for the measures are attained when all the records belong to the same class (i.e., when p equals 0 or 1). We next provide several examples of computing the different impurity measures.

Node N_1	Count
Class=0	0
Class=1	6

$\text{Gini} = 1 - (0/6)^2 - (6/6)^2 = 0$
$\text{Entropy} = -(0/6)\log_2(0/6) - (6/6)\log_2(6/6) = 0$
$\text{Error} = 1 - \max[0/6, 6/6] = 0$

Node N_2	Count
Class=0	1
Class=1	5

$\text{Gini} = 1 - (1/6)^2 - (5/6)^2 = 0.278$
$\text{Entropy} = -(1/6)\log_2(1/6) - (5/6)\log_2(5/6) = 0.650$
$\text{Error} = 1 - \max[1/6, 5/6] = 0.167$

Node N_3	Count
Class=0	3
Class=1	3

$\text{Gini} = 1 - (3/6)^2 - (3/6)^2 = 0.5$
$\text{Entropy} = -(3/6)\log_2(3/6) - (3/6)\log_2(3/6) = 1$
$\text{Error} = 1 - \max[3/6, 3/6] = 0.5$

Gain Ratio

Impurity measures such as entropy and Gini index tend to favor attributes that have a large number of distinct values. Comparing the first test condition, Gender, with the second, Car Type, it is easy to see that Car Type seems to provide a better way of splitting the data since it produces purer descendent nodes. However, if we compare both conditions with Customer ID, the latter appears to produce purer partitions. Yet Customer ID is not a predictive attribute because its value is unique for each record. Even in a less extreme situation, a test condition that results in a large number of outcomes may not be desirable because the number of record associated with each partition is too small to enable us to make any reliable predictions. There are two strategies for overcoming this problem. The first strategy is to restrict the test conditions to binary splits only. This strategy is employed by decision tree algorithms such as CART. Another strategy is to modify the splitting criterion to take into account the number of outcomes produced by the attribute test condition. For example, in the C4.5 decision tree algorithm,a splitting criterion known as gain ratio is used to determine the goodness of a split. This criterion is defined as follows:

GAIN RATIO = ΔINFO Split info Here, Split Info = − ki=1 P (v_i) $\log_2$P (v_i) and k is the total number of splits. For example, if each attribute value has the same number of records, then $\forall_i$: P (vi) = 1/k and the split information would be equal to $\log_2 k$. This example suggests that if an attribute produces a large number of splits, its split information will also be large, which in turn reduces its gain ratio.

5 c) Explain the rule evaluation criteria for classification

SPEED:

speed involves not just the time or computation cost of constructing a model,it involves the time required to learn to use the model.Obviously, a user wish to minimize both times although it has

to be understood theat any significant data mining project will take time to plan and prepare the data.

ROBUSTNESS:

Data errors are common ,in particular when data is being collected from number of sources and errors may remain same even after data cleaning. It is therefore desirable that a method be able to produce good results in spite of some errors and missing values in dataset.

SCALABILITY:

Many data mining techniques were originally designed for small datasets.many have been modified to deal with large problems.Given that large datasets are becoming common,It is desirable that a method continues to work efficently for large disk-resident database as well.

INTERPRETABILITY:

An important task of a data mining professional is to ensure that the results of data mining are explained to the decision makers.It is therefore desirable that the end-user be able to understand and gain insight from the results produced by the classification method.

GOODNESS OF THE MODEL:

For a model to be effective,it needs to fit the problem that is being solved.For example in a decision tree classification,it is desirable to find a decision tree of the "right" size and compactness.

6 a) What are Bayesian classifiers ?Explain Baye's theorem for classification.

The Bayesian Classification represents a supervised learning method as well as a statistical method for classification. Assumes an underlying probabilistic model and it allows us to capture uncertainty about the model in a principled way by determining probabilities of the outcomes. It can solve diagnostic and predictive problems. This Classification is named after Thomas Bayes (1702-1761), who proposed the ayesTheorem.Bayesian classification provides practical learning algorithms and prior knowledge and observed data can be combined. Bayesian

Classification provides a useful perspective for understanding and evaluating many learning algorithms. It calculates explicit probabilities forhypothesis and it is robust to noise in input data.Uses of Naive Bayes classification: 1. Naive Bayes text classification) The Bayesian classification is used as a probabilistic learning method (Naive Bayes text classification). Naive Bayes classifiers are among the most successful known algorithms forlearning to classify text documents.
2. Spam filtering (http://en.wikipedia.org/wiki/Bayesian_spam_filtering)Spam filtering is the best known use of Naive Bayesian text classification. It makes use of anaive Bayes classifier to

identify spam e-mail.Bayesian spam filtering has become a popular mechanism to distinguish illegitimate

spamemail from legitimate email (sometimes called "ham" or "bacn").[4] Many modern mail clientsimplement Bayesian spam filtering. Users can also install separate email filtering programs

IF-THEN Rules

Rule-based classifier makes use of a set of IF-THEN rules for classification. We can express a rule in the following from –

IF condition THEN conclusion

Let us consider a rule R1,

```
R1: IF age = youth AND student = yes
   THEN buy_computer = yes
```

Points to remember −

- The IF part of the rule is called rule antecedent or precondition.
- The THEN part of the rule is called rule consequent.
- The antecedent part the condition consist of one or more attribute tests and these tests are logically ANDed.
- The consequent part consists of class prediction.

Note − We can also write rule R1 as follows −

```
R1: (age = youth) ^ (student = yes))(buys computer = yes)
```

If the condition holds true for a given tuple, then the antecedent is satisfied.

Rule Extraction

Here we will learn how to build a rule-based classifier by extracting IF-THEN rules from a decision tree.

Points to remember −

To extract a rule from a decision tree −

- One rule is created for each path from the root to the leaf node.
- To form a rule antecedent, each splitting criterion is logically ANDed.
- The leaf node holds the class prediction, forming the rule consequent.

Rule Induction Using Sequential Covering Algorithm

Sequential Covering Algorithm can be used to extract IF-THEN rules form the training data. We do not require to generate a decision tree first. In this algorithm, each rule for a given class covers many of the tuples of that class.

Some of the sequential Covering Algorithms are AQ, CN2, and RIPPER. As per the general strategy the rules are learned one at a time. For each time rules are learned, a tuple covered by the rule is removed and the process continues for the rest of the tuples. This is because the path to each leaf in a decision tree corresponds to a rule.

Note − The Decision tree induction can be considered as learning a set of rules simultaneously.

The Following is the sequential learning Algorithm where rules are learned for one class at a time. When learning a rule from a class Ci, we want the rule to cover all the tuples from class C only and no tuple form any other class.

```
Algorithm: Sequential Covering

Input:
D, a data set class-labeled tuples,
Att_vals, the set of all attributes and their possible values.

Output:  A Set of IF-THEN rules.
Method:
Rule_set={ }; // initial set of rules learned is empty

for each class c do

   repeat
      Rule = Learn_One_Rule(D, Att_valls, c);
      remove tuples covered by Rule form D;
   until termination condition;

   Rule_set=Rule_set+Rule; // add a new rule to rule-set
end for
return Rule_Set;
```

Rule Pruning

The rule is pruned is due to the following reason −

- The Assessment of quality is made on the original set of training data. The rule may perform well on training data but less well on subsequent data. That's why the rule pruning is required.
- The rule is pruned by removing conjunct. The rule R is pruned, if pruned version of R has greater quality than what was assessed on an independent set of tuples.

FOIL is one of the simple and effective method for rule pruning. For a given rule R,

FOIL_Prune = pos - neg / pos + neg

where pos and neg is the number of positive tuples covered by R, respectively.

Note − This value will increase with the accuracy of R on the pruning set. Hence, if the FOIL_Prune value is higher for the pruned version of R, then we prune R.

6 b) Explain how the predictive accuracy of classification method be estimated.

Confusion Matrix

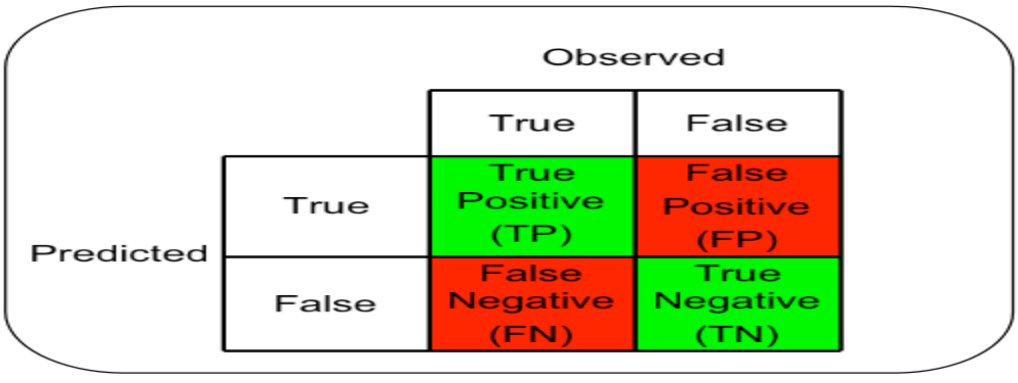
A binary classification model classifies each instance into one of two classes say a true and a false class. This gives rise to four possible classifications for each instance: a true positive, a true negative, a false positive, or a false negative. This situation can be depicted as a confusion matrix (also called contingency table) given in figure

1. The confusion matrix juxtaposes the observed classifications for a phenomenon (columns) with the predicted classifications of a model (rows). the classifications that lie along the major diagonal of the table are the correct classifications, that is, the true positives and the true negatives. The other fields signify model errors. For a perfect modelwe would only see the true positive and true negative fields filled out, the other fields would be set to zero. It is common to call true positives hits, true negatives correct rejections, false positive false alarms, and false negatives misses. A numberof model performance metrics can be derived from the confusion matrix. Perhaps, the most common metric is accuracy defined by the following formula:

accuracy=TP+TN/TP+TN+FP+FN

Other performance metrics include precision and recall defined as follows:

precision =TP/TP+FP



Evaluation Methods for a Classification Model

For classification problems it is natural to measure a classifier's performance in terms of the error rate. The classifier predicts the class of each instance; if it is correct, it is counted as "Success", if not, it is an error. The error rate is just the proportion of the errors made over the whole set of instances and it measures the overall performance of the classifier.The most popular method for the performance evaluation of a classifier are described below:

Cross-validation

Cross-validation is a technique for estimating the generalization performance of a predictive model. The main idea behind CV is to split data, once or several times, for estimating the risk of each algorithm: Part of data (the training sample) is used for training each algorithm, and the remaining part (the validation sample) is used for estimating the risk of the algorithm. Then, CV selects the algorithm with the smallest estimated risk.Cross validation is an alternative to random subsampling.

Holdout Method

Hold-out or (simple) validation relies on a single split of data. The holdout method is the simplest kind of cross validation. The data set is separated into two sets, called the training set and the testing set. The function approximator fits a function using the training set only. Then the function approximator is asked to predict the output values for the data in the testing set (it has never seen these output values before). The errors it makes are accumulated as before to give the mean absolute test set error, which is used to evaluate the model. The advantage of this method is that it is usually preferable to theresidual method and takes no longer to compute. However, its evaluation can have a high variance. The evaluation may depend heavily on which data points end up in the training set and which end up in the test set, and thus the evaluation may be significantly different depending on how the division is made.

Random Sub-sampling

The hold out method can be repeated several times to improve the estimation of a classifier's performance. This approach is known as random sub-sampling. Random sub-sampling encounters some of the problems associated with the holdout method because it does not utilize as mush data as possible for training. It has also no control over the number if times each record is used for testing and training. Consequently, some records might be used for training more often than others.

k-fold Cross-validation

It is one way to improve over the holdout method. The data set is divided into K subsets, and the holdout method is repeated k times. Each time, one of the K subsets is used as the test set and the other k-1 subsets are put together to form a training set. Then the average error across all ktrials is computed. The advantage of this method is that it matters less how the data gets divided. Every data point gets to be in a test set exactly once, and gets to be in a trainin g set k-1times. The variance of the resulting estimate is reduced as k is increased. The disadvantage of this method is that the training algorithm has to be rerun from scratch k times, which means it takes ktimes as much computation to make an evaluation. A variant of this method is to randomly divide the data into a test and training set k different times. The advantage of doing this is that you can independently choose how large each test set is and how many trials you average over.

Leave-one-out Method

When K-fold cross-validation taken to its logical extreme, with K equal to N, the number of data points in the set. That means that N separate times, the function approximator is trained on all the data except for one point and a prediction is made for that point. As before the average error is computed and used to evaluate the model. The evaluation given by leave-one-out cross validation error (LOO-XVE) is good, but at first pass it seems very expensive to compute. Fortunately, locally weighted learners can make LOO predictions just as easily as they make regular predictions. That means computing the LOO-XVE takes no more time than computing the residual error and it is a much better way to evaluate models.

Bootstrap
Un like other methods discussed above, in which the sampling is done without replacement, in bootstrap method, the training records are sampled with replacement i.e., a record already chosen for training is put back into the original pool of records so that it is equally likely to be redrawn.

7 a) Give the definition of cluster analysis.Explain desired features of cluster analysis.

Clustering and classification are both fundamental tasks in Data Mining. Classification is used mostly as a supervised learning method, clustering for unsupervised learning (some clustering models are for both). The goal of clus- tering is descriptive, that of classification is predictive (Veyssieres and Plant, 1998). Since the goal of clustering is to discover a new set of categories, the new groups are of interest in themselves, and their assessment is intrinsic. In classification tasks, however, an important part of the assessment is extrinsic, since the groups must reflect some reference set of classes. *"Understanding our world requires conceptualizing the similarities and differences between the entities that compose it"* (Tyron and Bailey, 1970). Clustering groups data instances into subsets in such a manner that similar instances are grouped together, k $S = Ci$ and $Ci \cap Cj = I$ for $i = j$. Consequently, any instance in S while different instances belong to differ-ent groups. The instances are thereby organized into an efficient representation that characterizes the population being sampled. Formally, the clustering structure is represented as a set of subsets $C = C1, \ldots, Ck$ of S, such that: i=1 belongs to exactly one and only one subset. Clustering of objects is as ancient as the human need for describing the salient characteristics of men and objects and identifying them with a type. Therefore, it embraces various scientific disciplines: from mathematics and statistics to biology and genetics, each of which uses different terms to describethe topologies formed using this analysis. From biological "taxonomies", to medical "syndromes" and genetic "genotypes" to manufacturing "group tech-nology" —the problem is identical: forming categories of entities and assigning individuals to the proper groups within it.

DistanceMeasures
Since clustering is the grouping of similar instances/objects, some sort of measure that can determine whether two objects are similar or dissimilar is required. There are two main type of measures used to estimate this relation: distance measures and similarity measures. Many clustering methods use distance measures to determine the similarity or dissimilarity between any pair of objects. It is useful to denote the distance between two instances xi and xj as: $d(xi,xj)$. A valid distance measure should be symmetric and obtains its minimum value (usually zero) in case of identical vectors. The distance measure is called a metric distance measure if it also satisfies the following properties:
1. Triangle inequality $d(xi,xk) \leq d(xi,xj) + d(xj,xk)$ $Ixi,xj,xk\ I\ S$.

2. d(*xi ,xj*)= 0*I xi = xj* *Ixi ,xj I S.*

7.2 Features of cluster analysis

The main reason for having many clustering methods is the fact that the notion of "cluster" is not precisely defined (Estivill-Castro, 2000). Consequently many clustering methods have been developed, each of which uses a different in- diction principle. Farley and Raftery (1998) suggest dividing the clustering methods into two main groups: hierarchical and partitioning methods. Han and Kamber (2001) suggest categorizing the methods into additional three main categories: *density-based methods*, *model-based clustering* and *grid-based methods*. An alternative categorization based on the induction principle of the various clustering methods is presented in (Estivill-Castro, 2000).

Types of Cluster Analysis Methods, Partitional Methods, Hierarchical Methods, Density Based Methods

HierarchicalMethods

These methods construct the clusters by recursively partitioning the instances in either a top-down or bottom- up fashion. These methods can be sub- divided as following:

Agglomerative hierarchical clustering — Each object initially represents a cluster of its own. Then clusters are successively merged until the desired cluster structure is obtained.

Divisive hierarchical clustering — All objects initially belong to one cluster. Then the cluster is divided into sub-clusters, which are successively divided into their own sub-clusters. This process continues until the desired cluster structure is obtained. The result of the hierarchical methods is a dendrogram, representing the nested n grouping of objects and similarity levels at which groupings change. A clustering of the data objects is obtained by cutting the dendrogram at the desired similarity level.

The merging or division of clusters is performed according to some similar-ity measure, chosen so as to optimize some criterion (such as a sum of squares).The hierarchical clustering methods could be further divided according to the manner that the similarity measure is calculated (Jain *et al*., 1999):

Single-link clustering (also called the connectedness, the minimum method or the nearest neighbor method) — methods that consider the distance between two clusters to be equal to the shortest distance from any member of one cluster to any member of the other cluster. If the data consist of similarities, the similarity between a pair of clusters is considered to be equal to the greatest similarity from any member of one cluster to any member of the other cluster (Sneath and Sokal, 1973).

Complete-link clustering (also called the diameter, the maximum method or the furthest neighbor method) - methods that consider the distance between two clusters to be equal to the longest distance from any member of one cluster to any member of the other cluster (King,1967).

Average-link clustering (also called minimum variance method) - meth-ods that consider the distance between two clusters to be equal to the average distance from any member of one cluster to any member of the other cluster. Such clustering algorithms may be found in (Ward, 1963)and (Murtagh, 1984).The disadvantages of the single-link clustering and the average-link clustering can be summarized as follows (Guha *et al*., 1998):Single- link clustering has a drawback known as the "chaining effect": Afew points that form a bridge between two clusters

cause the single-link clustering to unify these two clusters into one. Average-link clustering may cause elongated clusters to split and for portions of neighboring elongated clusters to merge. The complete-link clustering methods usually produce more compact clusters and more useful hierarchies than the single-link clustering methods, yet the single-link methods are more versatile. Generally, hierarchical methods are characterized with characterized with the following strengths: Versatility — The single-link methods, for example, maintain good performance on data sets containing non-isotropic clusters, including well-separated, chain-like and concentric clusters.

Multiple partitions — hierarchical methods produce not one partition, but multiple nested partitions, which allow different users to choose different partitions, according to the desired similarity level. The hierarchical partition is presented using the dendrogram. The main disadvantages of the hierarchical methods are:

Inability to scale well — The time complexity of hierarchical algorithms is at least $O(m2)$ (where m is the total number of instances), which is non-linear with the number of objects. Clustering a large number of objects using a hierarchical algorithm is also characterized by huge I/O costs. Hierarchical methods can never undo what was done previously. Namely there is no back-tracking capability.

PartitioningMethods

Partitioning methods relocate instances by moving them from one cluster to another, starting from an initial partitioning. Such methods typically require that the number of clusters will be pre-set by the user. To achieve global optimality in partitioned-based clustering, an exhaustive enumeration process of all possible partitions is required. Because this is not feasible, certain greedy heuristics are used in the form of iterative optimization. Namely, a relocation method iteratively relocates points between the k clusters. The following subsections present various types of partitioning methods.

7 b) Explain the following clustering techniques with algorithm

i) K-means method
ii) Divisive hierarchial method

Step 1 and 2:

• Let the three seeds be the first three students as shown in Table .

Step 3 and 4:

• Now compute the distances using the 4 attributes and using the sum of absolute differences for simplicity. The distance values for all the objects are given in Table

Table 4.3 Data for Example 4.1

Student	*Age*	*Mark1*	*Mark2*	*Mark3*
S_1	18	73	75	57
S_2	18	79	85	75
S_3	23	70	70	52
S_4	20	55	55	55
S_5	22	85	86	87
S_6	19	91	90	89
S_7	20	70	65	60
S_8	21	53	56	59
S_9	19	82	82	60
S_{10}	47	75	76	77

Table 4.5 First iteration—allocating each object to the nearest cluster

				Distances from clusters			
C_1 *18.0*	*73.0*	*75.0*	*57.0*	*From* C_1	*From* C_2	*From* C_3	*Allocation to the nearest cluster*
C_2 *18.0*	*79.0*	*85.0*	*75.0*				
C_3 *23.0*	*70.0*	*70.0*	*52.0*				
S_1	18.0	73.0	75.0	57.0	0.0	34.0	18.0	C_1
S_2	18.0	79.0	85.0	75.0	34.0	0.0	52.0	C_2
S_3	23.0	70.0	70.0	52.0	18.0	52.0	0.0	C_3
S_4	20.0	55.0	55.0	55.0	42.0	76.0	36.0	C_3
S_5	22.0	85.0	86.0	87.0	57.0	23.0	67.0	C_2
S_6	19.0	91.0	90.0	89.0	66.0	32.0	82.0	C_2
S_7	20.0	70.0	65.0	60.0	18.0	46.0	16.0	C_3
S_8	21.0	53.0	56.0	59.0	44.0	74.0	40.0	C_3
S_9	19.0	82.0	82.0	60.0	20.0	22.0	36.0	C_1
S_{10}	47.0	75.0	76.0	77.0	52.0	44.0	60.0	C_2

Step 5:

• Table 4.6 compares the cluster means of clusters found in Table 4.5 with the original seeds.

Table 4.6 Comparing new centroids and the seeds

	Age	*Mark1*	*Mark2*	*Mark3*
C_1	18.5	77.5	78.5	58.5
C_2	26.5	82.5	84.3	82.0
C_3	21	61.5	61.5	56.5
Seed1	18	73	75	57
Seed2	18	79	85	75
Seed3	23	70	70	52

Step 3 and 4:

• Use the new cluster means to re-compute the distance of each object to each of the means, again allocating each object to the nearest cluster. Table 4.7 shows the second iteration.

Table 4.7 Second iteration—allocating each object to the nearest cluster

					Distances from clusters			
C_1	*18.5*	*77.5*	*78.5*	*58.5*	*From*	*From*	*From*	*Allocation*
C_2	*26.5*	*82.5*	*84.3*	*82.0*	C_1	C_2	C_3	*to the nearest*
C_3	*21.0*	*62.0*	*61.5*	*56.5*				*cluster*
S_1	18.0	73.0	75.0	57.0	10.0	52.3	28.0	C_1
S_2	18.0	79.0	85.0	75.0	25.0	19.8	62.0	C_2
S_3	23.0	70.0	70.0	52.0	27.0	60.3	23.0	C_3
S_4	20.0	55.0	55.0	55.0	51.0	90.3	16.0	C_3
S_5	22.0	85.0	86.0	87.0	47.0	13.8	79.0	C_2
S_6	19.0	91.0	90.0	89.0	56.0	28.8	92.0	C_2
S_7	20.0	70.0	65.0	60.0	24.0	60.3	16.0	C_3
S_8	21.0	53.0	56.0	59.0	50.0	86.3	17.0	C_3
S_9	19.0	82.0	82.0	60.0	10.0	32.3	46.0	C_1
S_{10}	47.0	75.0	76.0	77.0	52.0	41.3	74.0	C_2

• Number of students in cluster 1 is again 2 and the other two clusters still have 4 students each.
• A more careful look shows that the clusters have not changed at all. Therefore, the method has converged rather quickly for this very simple dataset.
• The cluster membership is as follows
Cluster C1= {S1, S9} Cluster C2= {S2, S5, S6, S10} Cluster C3= {S3, S4, S7, S8}

ii)agglomerative method

Table 4.10 Data for Example 4.3

Student	*Age*	*Mark1*	*Mark2*	*Mark3*
S_1	18	73	75	57
S_2	18	79	85	75
S_3	23	70	70	52
S_4	20	55	55	55
S_5	22	85	86	87
S_6	19	91	90	89
S_7	20	70	65	60
S_8	21	53	56	59
S_9	19	82	82	60
S_{10}	47	75	76	77

Steps 1 and 2:
• Allocate each point to a cluster and compute the distance-matrix using the centroid method.
• The distance-matrix is symmetric, so we only show half of it in Table.
• Table 4.11 gives the distance of each object with every other object.

Table 4.11 Distance matrix for data in Example 4.3

	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	S_9
S_1	0								
S_2	34	0							
S_3	18	52	0						
S_4	42	76	36	0					
S_5	57	23	67	95	0				
S_6	66	32	82	106	15	0			
S_7	18	46	16	30	65	76	0		
S_8	44	74	40	8	91	104	28	0	
S_9	20	22	36	60	37	46	30	115	0
S_{10}	52	44	60	90	55	70	60	98	58

Steps 3 and 4:

• The smallest distance is 8 between objects S4 and S8. They are combined and removed and we put
the combined cluster (C1) where the object S4 was.

• Table is now the new distance-matrix. All distances except those with cluster C1 remain unchanged.

Table 4.12 Distance matrix after merging S_4 and S_8 into cluster C_1

	S_1	S_2	S_3	C_1	S_5	S_6	S_7	S_9
S_1	0							
S_2	34	0						
S_3	18	52	0					
C_1	41	75	38	0				
S_5	57	23	67	93	0			
S_6	66	32	82	105	15	0		
S_7	18	46	16	29	65	76	0	
S_9	20	22	36	59	37	46	30	0
S_{10}	52	44	60	88	55	70	60	58

• The result of using the agglomerative method could be something like that shown in Figure

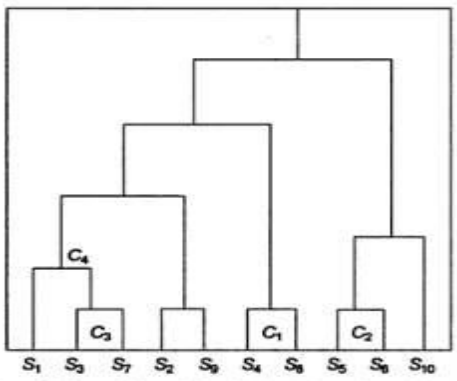


Figure 4.6 A possible result of using the agglomerative method.

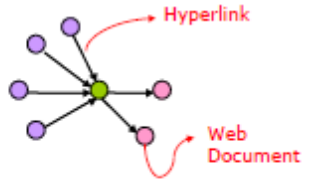
8 a) What is Web data mining?Explain Web document clustering

WEB DATA MINING

• This is the process of extracting useful information from the contents of web-documents.
• We see more & more government information are gradually being placed on the web in recent years.
• We have
→ digital libraries which users can access from the web
→ web-applications which users can access through web-interfaces
• Some of the web-data are hidden-data, and
some are generated dynamically as a result of queries and reside in the DBMSs.
• The web-content consists of different types of data such as text, image, audio, video as well as hyperlinks.
• Most of the research on web-mining is focused on the text or hypertext contents.
• The textual-parts of web-data consist of
→ unstructured-data such as free texts
→ semi structured-data such as HTML documents &
→ structured-data such as data in the tables
• Much of the web-data is unstructured, free text-data.
As a result, text-mining techniques can be directly employed for web-mining.
• Issues addressed in text mining are, topic discovery, extracting association patterns, clustering of
web documents and classification of Web Pages.
• Research activities on this topic have drawn heavily on techniques developed in other disciplines
such as IR(Information Retrieval) and NLP(Natural Language Processing).

WEB DOCUMENT CLUSTERING

• The structure of a typical web-graph consists of web-pages as nodes, and hyperlinks as edges connecting related pages.
• Web-Structure mining is the process of discovering structure information from the web.



Web Graph Structure

• This type of mining can be performed either at the (intra-page) document level or at the (interpage)
hyperlink level.
• This can be used to classify web-pages.
• This can be used to generate information such as the similarity & relationship between different web-sites.

PageRank

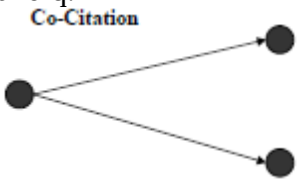
• PageRank is a metric for ranking hypertext documents based on their quality.
• The key idea is that a page has a high rank if it is pointed to by many highly ranked pages

Clustering & Determining Similar Pages

• For determining the collection of similar pages, we need to define the similarity measure between the pages. There are 2 basic similarity functions:

1) Co-citation

For a pair of nodes p and q, the co-citation is the number of nodes that point to both p and q.



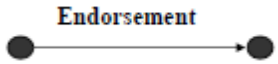
2) Bibliographic coupling

For a pair of nodes p and q, the bibliographic coupling is equal to the number of nodes that have links from both p and q.

Social Network Analysis

• This can be used to measure the relative standing or importance of individuals in a network.

• The basis idea is that if a web-page points a link to another web-page, then the former is, in some sense, endorsing the importance of the latter.



• Links in the network may have different weights, corresponding to the strength of endorsement.

8 B) Explain different text mining approach

TEXT MINING

• This is concerned with various tasks, such as

→ extraction of information implicitly contained in the collection of documents or

→ similarity-based structuring

• Text-collection lacks the imposed structure of a traditional database.

• The text expresses a vast range of information, but encodes the information in a form that is difficult to decipher automatically.

• Traditional data-mining techniques have been designed to operate on structured-databases.

• In structured-databases, it is easy to define the set of items and hence, it becomes easy to employ

the traditional mining techniques.

In textual-database, identifying individual items (or terms) is not so easy.

• Text-mining techniques have to be developed to process the unstructured textual-data to aid in knowledge discovery.

• The inherent nature of textual-data, namely unstructured characteristics, motivates the development of separate text-mining techniques.

• Following approaches can be used for text-mining:

1) One way is to impose a structure on the textual-database and use any of the known datamining techniques meant for structured-databases.

2) The other approach would be to develop a very specific technique for mining that exploits the inherent characteristics of textual-databases.

INFORMATION RETRIEVAL (IR)

• This is concerned with finding and ranking documents that match the users' information needs.
• The way of dealing with textual-information is a keyword-based document representation.
• A body of text is analyzed by its constituent-words, and various techniques are used to build the
core words for a document.
• The goals are
→ to find documents that are similar based on some specification of the user
→ to find right index terms in a collection, so that querying will return appropriate document
• Recent trends in IR research include

document classification
→ data visualization
→ filtering

INFORMATION EXTRACTION (IE)

• This is concerned with transforming a collection of documents, usually with the help of an IR system, into information that is more readily digested and analyzed.
• IE extracts relevant facts from the documents
whereas IR selects relevant documents.
• In general, IE works at a finer granularity level than IR does on the documents.
• Most IE systems use data-mining(or machine-learning) techniques to learn the extraction patterns
(or rules) for documents semi-automatically or automatically.
• The results of the IE process could be in the form of
→ structured database or
→ summary/compression of the original document

UNSTRUCTURED TEXT

• Unstructured-documents are free texts, such as news stories.
• To convert an unstructured-document to a structured form, following features can be extracted:

Word Occurrences

• The vector-representation (or bag of words) takes single words found in the
training-set as features, ignoring the sequence in which the words occur.
• The feature is said to be boolean, if we consider whether a word either occurs or
does not occur in a document.
• The feature is said to be frequency-based, if the frequency of the word in a
document is taken into consideration.

Stopword Removal

• Stopwords are frequently occurring and insignificant words in a language that help
construct sentences but do not represent any content of the documents.
• Common stopwords in English include:
a, about, an, are, as, at, be, by, for, from, how, in, is, of, on, or,
that, the, these, this, to, was, what, when, where, who, will, with
• Such words can be removed before documents are indexed and stored.

Stemming

• *Stemming* refers to the process of reducing words to their morphological roots or
stems

• A stem is the portion of a word that is left after removing its prefixes and suffixes.
• For example, "informing", "information", "informer" & "informed" are reduced to "inform".

POS

• POS stands for Part of Speech.
• There can be 25 possible values for POS tags.
• Most common tags are noun, verb, adjective and adverb.
• Thus, we can assign a number 1,2,3,4 or 5, depending on whether the word is a noun, verb, adjective, adverb or any other, respectively.

Positional Collocations

• The values of thisfeature are the words that occur one or two position to the right or left of the given word.

n-gram

• An n-gram is a contiguous sequence of n items from a given sequence of text.
• This can be used for predicting the next item in a sequence.

Sample sequence	1-gram sequence	2-gram sequence	3-gram sequence
... to be or not to be ...	..., to, be, or, not, to, be, ...	..., to be, be or, or not, not to, to be, ...	..., to be or, be or not, or not to, not to be, ...

LSI

• LSI stands for Latent Semantic Indexing.
• *LSI* is an indexing and retrieval method to identify the relationships between the terms and concepts contained in an unstructured collection of text.

8 c) Describe sequential mining techniques with an example.

Sequential pattern mining is a topic of data mining concerned with finding statistically relevant patterns between data examples where the values are delivered in a sequence.It is usually presumed that the values are discrete, and thus time series mining is closely related, but usually considered a different activity. Sequential pattern mining is a special case of structured data mining.

String mining typically deals with a limited alphabet for items that appear in a sequence, but the sequence itself may be typically very long. Examples of an alphabet can be those in the ASCII character set used in natural language text, nucleotide bases 'A', 'G', 'C' and 'T' in DNA sequences, or amino acids for protein sequences. In biology applications analysis of the arrangement of the alphabet in strings can be used to examine gene and protein sequences to determine their properties. Knowing the sequence of letters of a DNA or a protein is not an ultimate goal in itself. Rather, the major task is to understand the sequence, in terms of its structure and biological function. This is typically achieved first by identifying individual regions or structural units within each sequence and then assigning a function to each structural unit. In many cases this requires comparing a given sequence with previously studied ones. The comparison between the strings becomes complicated when insertions, deletions and mutations occur in a string.

A survey and taxonomy of the key algorithms for sequence comparison for bioinformatics is presented by Abouelhoda & Ghanem (2010), which include

- **Repeat-related problems:** that deal with operations on single sequences and can be based on exact string matching or approximate string matching methods for finding dispersed fixed length and maximal length repeats, finding tandem repeats, and finding unique subsequences and missing (un-spelled) subsequences.
- **Alignment problems:** that deal with comparison between strings by first aligning one or more sequences; examples of popular methods include BLAST for comparing a single sequence with multiple sequences in a database, and ClustalW for multiple alignments. Alignment algorithms can be based on either exact or approximate methods, and can also be classified as global alignments, semi-global alignments and local alignment. See sequence alignment.