UNIT 4 INTRODUCTION TO DATA MINING

Structure **Page Nos.**

4.0 INTRODUCTION

Data mining is emerging as a rapidly growing interdisciplinary field that takes its approach from different areas like, databases, statistics, artificial intelligence and data structures in order to extract hidden knowledge from large volumes of data. The data mining concept is now a days not only used by the research community but also a lot of companies are using it for predictions so that, they can compete and stay ahead of their competitors.

With rapid computerisation in the past two decades, almost all organisations have collected huge amounts of data in their databases. These organisations need to understand their data and also want to discover useful knowledge as patterns, from their existing data.

This unit aims at giving you some of the fundamental techniques used in data mining. This unit emphasises on a brief overview of data mining as well as the application of data mining techniques to the real world. We will only consider structured data as input in this unit. We will emphasise on three techniques of data mining:

(a) Classification,
(b) Clustering, and
(c) Association rules.

4.1 OBJECTIVES

After going through this unit, you should be able to:

- explain what is data mining;
- explain how data mining is applied in the real world;
- define the different approaches to data mining;
- use the classification approach in data mining;

- use the clustering approach;
- explain how association rules are used in data mining, and
- identify some of the leading data mining tools.

4.2 DATA MINING TECHNOLOGY

Data is growing at a phenomenal rate today and the users expect more sophisticated information from this data. There is need for new techniques and tools that can automatically generate useful information and knowledge from large volumes of data. Data mining is one such technique of generating hidden information from the data. Data mining can be defined as: "an automatic process of extraction of non-trivial or implicit or previously unknown but potentially useful information or patterns from data in large databases, data warehouses or in flat files".

Data mining is related to data warehouse in this respect that, a data warehouse is well equipped for providing data as input for the data mining process. The advantages of using the data of data warehouse for data mining are or many some of them are listed below:

- Data quality and consistency are essential for data mining, to ensure, the accuracy of the predictive models. In data warehouses, before loading the data, it is first extracted, cleaned and transformed. We will get good results only if we have good quality data.
- Data warehouse consists of data from multiple sources. The data in data warehouses is integrated and subject oriented data. The data mining process performed on this data.
- In data mining, it may be the case that, the required data may be aggregated or summarised data. This is already there in the data warehouse.
- Data warehouse provides the capability of analysing data by using OLAP operations. Thus, the results of a data mining study can be analysed for hirtherto, uncovered patterns.

As defined earlier, data mining generates potentially useful information or patterns from data. In fact, the information generated through data mining can be used to create knowledge. So let us, first, define the three terms data, information and knowledge.

4.2.1 Data, Information, Knowledge

Before going into details on data mining, let us, first, try to discuss the differences between data, information and knowledge.

1. **Data (Symbols):** It simply exists. It has no significance beyond its existence. It is raw information. For example, "it is raining".
2. **Information:** Information is the processed data. It provides answer to "who", "what", "where", and "when" questions. For example, "The temperature dropped 12 degrees centigrade and then it started raining" is an example of information.
3. **Knowledge:** Knowledge is the application of data and information and it answers the "how" questions. This is not explicit in the database - it is implicit. For example "If humidity is very high and the temperature drops suddenly, then, the atmosphere is often unlikely to be able to hold the moisture so, it rains', is an example of knowledge.

4.2.2 Sample Data Mining Problems

Now that we have defined data, information and knowledge let us define some of the problems that can be solved through the data mining process.

a) Mr Ramniwas Gupta manages a supermarket and the cash counters, he adds transactions into the database. Some of the questions that can come to Mr. Gupta's mind are as follows:

   a) Can you help me visualise my sales?
   b) Can you profile my customers?
   c) Tell me something interesting about sales such as, what time sales will be maximum etc.

   He does not know statistics, and he does not want to hire statisticians.

   The answer of some of the above questions may be answered by data mining.

b) Mr. Avinash Arun is an astronomer and the sky survey has 3 tera-bytes (10^{12}) of data, 2 billion objects. Some of the questions that can come to the mind of Mr. Arun are as follows:

   a) Can you help me recognise the objects?
   b) Most of the data is beyond my reach. Can you find new/unusual items in my data?
   c) Can you help me with basic manipulation, so I can focus on the basic science of astronomy?

   He knows the data and statistics, but that is not enough. The answer to some of the above questions may be answered once again, by data mining.

Please note: The use of data mining in both the questions given above lies in finding certain patterns and information. Definitely the type of the data in both the database as given above will be quite different.

4.2.3 Database Processing Vs. Data Mining Processing

Let us, first, differentiate between database processing and data mining processing: The query language of database processing is well defined and it uses SQL for this, while, the data mining, the query is poorly defined and there is no precise query language. The data used in data processing is operational data, while, in data mining, it is historical data i.e., it is not operational data.

The output of the query of database processing is precise and is the subset of the data, while, in the case of data mining the output is fuzzy and it is not a subset of the data.

Some of the examples of database queries are as follows:

- Find all credit card applicants with the last name Ram.
- Identify customers who have made purchases of more than Rs.10,000/- in the last month.
- Find all customers who have purchased shirt(s).

Some data mining queries may be:

- Find all credit card applicants with poor or good credit risks.
- Identify the profile of customers with similar buying habits.
- Find all items that are frequently purchased with shirt (s).

4.2.4 Data Mining Vs. Knowledge Discovery in Databases (KDD)

Knowledge Discovery in Databases (KDD) is the process of finding useful information, knowledge and patterns in data while data mining is the process of using of algorithms to automatically extract desired information and patterns, which are derived by the Knowledge Discovery in Databases process. Let us define KDD in more details.

Knowledge Discovery in Databases (KDD) Process

The different steps of KDD are as follows:

- **Extraction:** Obtains data from various data sources.
- **Preprocessing:** It includes cleansing the data which has already been extracted by the above step.
- **Transformation:** The data is converted in to a common format, by applying some technique.
- **Data Mining:** Automatically extracts the information/patterns/knowledge.
- **Interpretation/Evaluation:** Presents the results obtained through data mining to the users, in easily understandable and meaningful format.

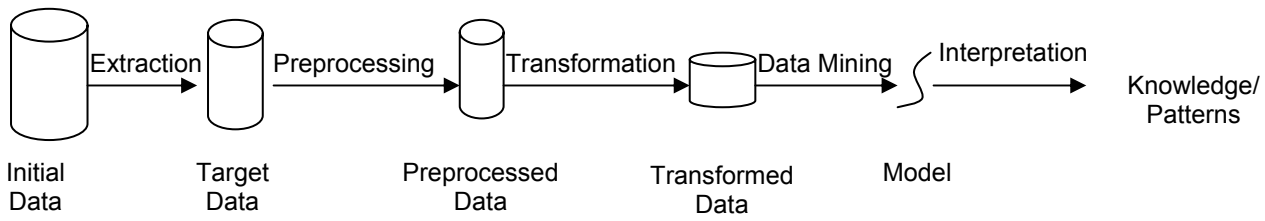

Figure 1: KDD process

Tasks in Knowledge Discovery in Databases (KDD) Process

The different tasks in KDD are as follows:

- **Obtains information on application domain**: It gathers knowledge from the domain relevant to the user.
- **Extracting data set**: It includes extracting required data which will later, be used for analysis.
- **Data cleansing process**: It involves basic operations such as, the removal of noise, collecting necessary information to from noisy data, such as, deciding on strategies for handling missing data fields.
- **Data reduction and projection**: Using dimensionality reduction or transformation methods it reduces the effective number of dimensions under consideration.
- **Selecting data mining task**: In this stage we decide what the objective of the KDD process is. Whether it is classification, clustering, association rules etc.
- **Selecting data mining method**: In this stage, we decide the methods and the parameter to be used for searching for desired patterns in the data.

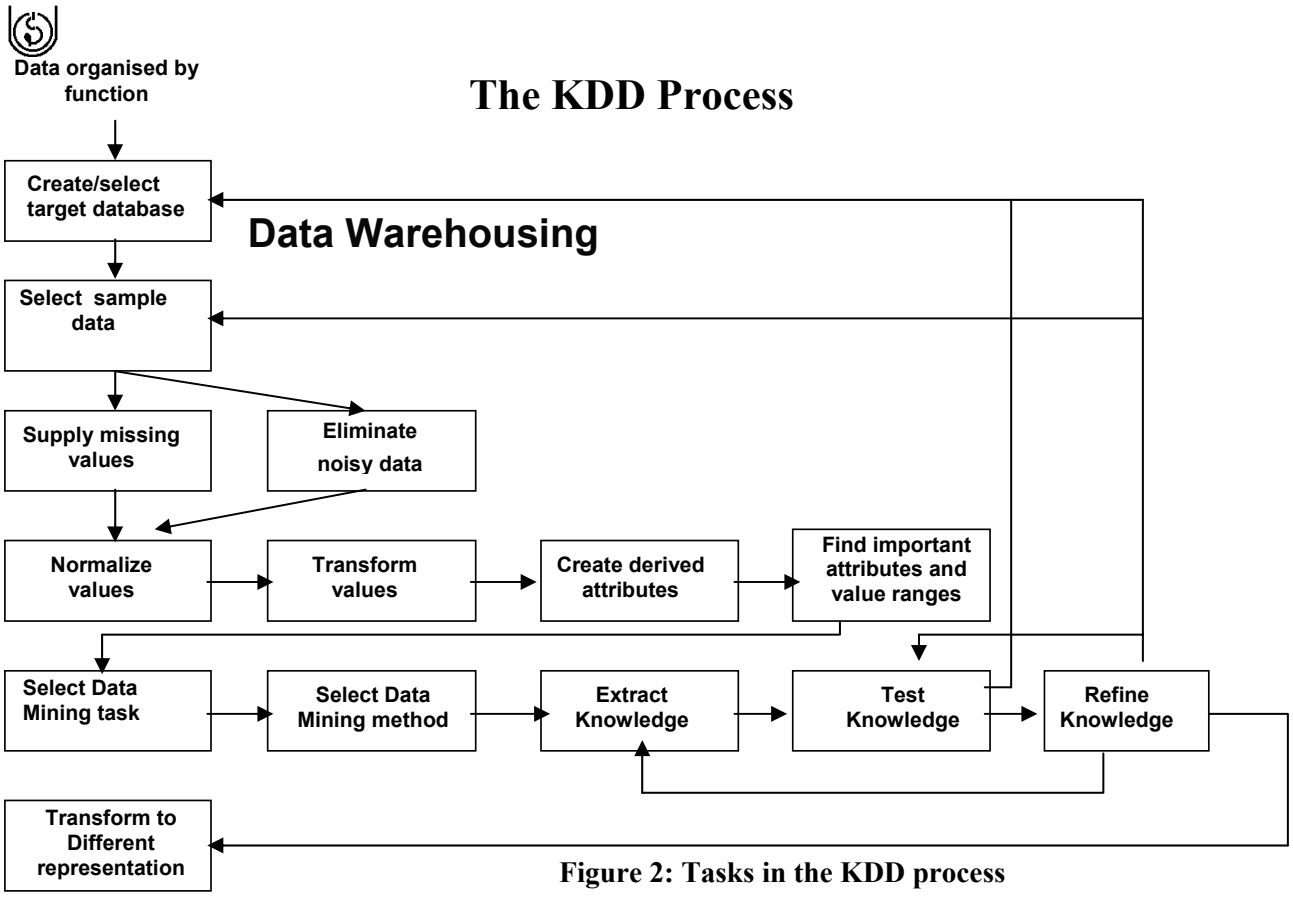

Figure 2: Tasks in the KDD process

- **Extraction of patterns:** It includes searching for desired patterns only, because, the data-mining model may generate a lot of patterns.
- Interpretation and presentation of pattern/model.

4.3 APPROACHES TO DATA MINING PROBLEMS

The approaches to data mining problems are based on the type of information/ knowledge to be mined. We will emphasis on three different approaches: Classification, Clustering, and Association Rules.

The classification task maps data into predefined groups or classes. The class of a tuple is indicated by the value of a user-specified goal attribute. Tuples consists of a set of predicating attributes and a goal attribute. The task, is to discover, some kind of relationship between the predicating attributes and the goal attribute, so that, the discovered information/ knowledge can be used to predict the class of new tuple(s).

The task of clustering is to group the tuples with similar attribute values into the same class. Given a database of tuples and an integer value k, the Clustering is to define a mapping, such that, tuples are mapped to different cluster.

The principle is to maximise intra-class similarity and minimise the interclass similarity. In clustering, there is no goal attribute. So, classification is supervised by the goal attribute, while clustering is an unsupervised classification.

The task of association rule mining is to search for interesting relationships among items in a given data set. Its original application is on “market basket data”. The rule has the form X➔Y, where X and Y are sets of items and they do not intersect. Each

rule has two measurements, support and confidence. Given the user-specified minimum support and minimum confidence, the task is to find, rules with support and confidence above, minimum support and minimum confidence.

The distance measure finds, the distance or dissimilarity between objects the measures that are used in this unit are as follows:

- Euclidean distance: $dis(t_i,t_j)= \sqrt{\sum_{h=1}^{k}(t_{ih} - t_{jh})^2}$
- Manhattan distance: $dis(t_i,t_j)= \sum_{h=1}^{k}|(t_{ih} - t_{jh})|$

where t_i and t_j are tuples and h are the different attributes which can take values from 1 to k

☞ **Check Your Progress 1**

1) What do you mean by data mining?

……

……

……

2) How is data mining different from Knowledge discovery in databases? What are the different steps of KDD?

……

……

……

3) What is the difference between data mining and OLTP?

……

……

……

4) What are different data mining tasks?

……

……

……

4.4 CLASSIFICATION

The classification task maps data into predefined groups or classes.
Given a database/dataset $D=\{t_1,t_2,\ldots,t_n\}$ and a set of classes $C=\{C_1,\ldots,C_m\}$, the classification Problem is to define a mapping $f:D \rightarrow C$ where each t_i is assigned to one class, that is, it divides database/dataset D into classes specified in the Set C.

A few very simple examples to elucidate classification could be:

- Teachers classify students' marks data into a set of grades as A, B, C, D, or F.
- Classification of the height of a set of persons into the classes tall, medium or short.

4.4.1 Classification Approach

The basic approaches to classification are:

- To create specific models by, evaluating training data, which is basically the old data, that has already been classified by using the domain of the experts' knowledge.
- Now applying the model developed to the new data.

Please note that in classification, the classes are predefined.

Some of the most common techniques used for classification may include the use of Decision Trees, Neural Networks etc. Most of these techniques are based on finding the distances or uses statistical methods.

4.4.2 Classification Using Distance (K-Nearest Neighbours)

This approach, places items in the class to which they are "closest" to their neighbour.It must determine distance between an item and a class. Classes are represented by centroid (Central value) and the individual points.

One of the algorithms that is used is K-Nearest Neighbors. Some of the basic points to be noted about this algorithm are:

- The training set includes *classes* along with other attributes. (Please refer to the training data given in the *Table* given below).
- The value of the K defines the number of *near items* (items that have less distance to the attributes of concern) that should be used from the given set of training data (just to remind you again, training data is already classified data). This is explained in point (2) of the following example.
- A new item is placed in the class in which the most number of close items are placed. (Please refer to point (3) in the following example).
- The value of K should be <= $\sqrt{Number\_of\_training\_items}$ However, in our example for limiting the size of the sample data, we have not followed this formula.

Example: Consider the following data, which tells us the person's class depending upon gender and height

Name	Gender	Height	Class
Sunita	F	1.6m	Short
Ram	M	2m	Tall
Namita	F	1.9m	Medium
Radha	F	1.88m	Medium
Jully	F	1.7m	Short
Arun	M	1.85m	Medium
Shelly	F	1.6m	Short
Avinash	M	1.7m	Short
Sachin	M	2.2m	Tall
Manoj	M	2.1m	Tall
Sangeeta	F	1.8m	Medium
Anirban	M	1.95m	Medium
Krishna	F	1.9m	Medium
Kavita	F	1.8m	Medium
Pooja	F	1.75m	Medium

1) You have to classify the tuple <Ram, M, 1.6> from the training data that is given

to you.

2) Let us take only the **height** attribute for distance calculation and suppose K=5 then the following are the near five tuples to the data that is to be classified (using Manhattan distance as a measure on the height attribute).

Name	Gender	Height	Class
Sunita	F	1.6m	Short
Jully	F	1.7m	Short
Shelly	F	1.6m	Short
Avinash	M	1.7m	Short
Pooja	F	1.75m	Medium

3) On examination of the tuples above, we classify the tuple <Ram, M, 1.6> to *Short* class since most of the tuples above belongs to *Short* class.

4.4.3 Decision or Classification Tree

Given a data set D = $\{t_1,t_2 \ldots, t_n\}$ where $t_i=<t_{i1}, \ldots, t_{ih}>$, that is, each tuple is represented by h attributes, assume that, the database schema contains attributes as $\{A_1, A_2, \ldots, A_h\}$. Also, let us suppose that the classes are $C=\{C_1, \ldots., C_m\}$, then:

Decision or Classification Tree is a tree associated with D such that

- Each internal node is labeled with attribute, A_i
- Each arc is labeled with the predicate which can be applied to the attribute at the parent node.
- Each leaf node is labeled with a class, C_j

Basics steps in the Decision Tree are as follows:

- Building the tree by using the training set dataset/database.
- Applying the tree to the new dataset/database.

Decision Tree Induction is the process of learning about the classification using the inductive approach. During this process, we create a decision tree from the training data. This decision tree can, then be used, for making classifications. To define this we need to define the following.

Let us assume that we are given probabilities $p_1, p_2, .., p_s$ whose sum is 1. Let us also define the term Entropy, which is the measure of the amount of randomness or surprise or uncertainty. Thus our basic goal in the classification process is that, the entropy for a classification should be zero, that, if no surprise then, entropy is equal to zero. Entropy is defined as:

$$H(p_1,p_2,\ldots p_s)= \sum_{i=1}^{s} (p_i * \log(1/p_i)) \qquad \ldots\ldots. \quad (1)$$

ID3 Algorithm for Classification

This algorithm creates a tree using the algorithm given below and tries to reduce the expected number of comparisons.

Algorithm: ID3 algorithm for creating decision tree from the given training data.
Input: The *training data* and the *attribute-list*.

Output: A decision tree.

Process:

Step 1: Create a node N

Step 2: If sample data are all of the same class, C (that is probability is 1)
then return N as a leaf node labeled class C

Step 3: If *attribute-list* is empty
then return N as a leaf node label it with the most common class in the training data; // majority voting

Step 4: Select *split-attribute*, which is the attribute in the *attribute-list* with the highest information gain;

Step 5: label node N with *split-attribute*;

Step 6: for each known value Ai, of *split-attribute* // partition the samples
Create a branch from node N for the condition: *split-attribute* = A_i;
// Now consider a partition and recursively create the decision tree:
Let xi be the set of data from training data that satisfies the condition:
split-attribute = A_i
if the set xi is empty then
attach a leaf labeled with the most common class in the prior set of training data;

else
attach the node returned after recursive call to the program with training data as xi and
new attribute list = present attribute-list – split-attribute;

End of Algorithm.

Please note: The algorithm given above, chooses the split attribute with the highest information gain, that is, calculated as follows:

$$\text{Gain (D,S)} = H(D) - \sum_{i=1}^{s} (P(D_i) * H(D_i)) \quad \text{.....(2)}$$

where S is new states = $\{D_1, D_2, D_3 \ldots D_S\}$ and H(D) finds the amount of order in that state

Consider the following data in which *Position* attribute acts as class

Department	Age	Salary	Position
Personnel	31-40	Medium Range	Boss
Personnel	21-30	Low Range	Assistant
Personnel	31-40	Low Range	Assistant
MIS	21-30	Medium Range	Assistant
MIS	31-40	High Range	Boss
MIS	21-30	Medium Range	Assistant
MIS	41-50	High Range	Boss
Administration	31-40	Medium Range	Boss
Administration	31-40	Medium Range	Assistant
Security	41-50	Medium Range	Boss
Security	21-30	Low Range	Assistant

Figure 3: Sample data for classification

We are applying ID3 algorithm, on the above dataset as follows:

The initial entropy of the dataset using formula at (1) is

$$H(initial) = (6/11)\log(11/6) + (5/11)\log(11/5) = 0.29923$$
(Assistant) (Boss)

Now let us calculate gain for the departments using the formula at (2)

Gain(Department) = H(initial) – [P(Personnel) * H(MIS) + P(MIS) * H(Personnel) + P(Administration) * H(Administration) + P(Security) * H(Security)]

= *0.29923*- { (3/11)[(1/3)log3 +(2/3)log(3/2)] + (4/11)[(2/4)log 2 + (2/4)log 2] + (2/11)[(1/2)log 2 + (1/2)log 2] + (2/11)[(1/2)log 2 + (1/2)log 2] }
= 0.29923- 0.2943
= 0.0049

Similarly:

Gain(Age) = 0.29923- { (4/11)[(4/4)log(4/4)] + (5/11)[(3/5)log(5/3) + (2/5)log (5/2)] + (2/11)[(2/2) log (2/2)] }
= 0.29923 - 0.1328
= 0.1664

Gain(Salary) = 0.29923 – { (3/11)[(3/3)log 3] + (6/11)[(3/6) log2 + (3/6)log 2] + (2/11) [(2/2 log(2/2)) }
= 0.29923 – 0.164
= 0.1350

Since age has the maximum gain, so, this attribute is selected as the first splitting attribute. In age range 31-40, class is not defined while for other ranges it is defined.

So, we have to again calculate the spitting attribute for this age range (31-40). Now, the tuples that belong to this range are as follows:

Department	Salary	Position
Personnel	Medium Range	Boss
Personnel	Low Range	Assistant
MIS	High Range	Boss
Administration	Medium Range	Boss
Administration	Medium Range	Assistant

Again the initial entropy = (2/5)log(5/2) + (3/5)log(5/3) = 0.29922
(Assistant) (Boss)

Gain(Department)= 0.29922- { (2/5)[(1/2)log 2 + (1/2)log 2] +1/5[(1/1)log1] + (2/5)[(1/2)log 2 + (1/2)log 2] }
=0.29922- 0.240
=0.05922

Gain (Salary) = 0.29922- { (1/5) [(1/1)log1] +(3/5)[(1/3)log 3 + (2/3)log(3/2)] +

(1/5) [(1/1)log1] }
= 0.29922- 0.1658
=0.13335

The Gain is maximum for salary attribute, so we take salary as the next splitting attribute. In middle range salary, class is not defined while for other ranges it is defined. So, we have to again calculate the spitting attribute for this middle range. Since only department is left, so, department will be the next splitting attribute. Now, the tuples that belong to this salary range are as follows:

Department	Position
Personnel	Boss
Administration	Boss
Administration	Assistant

Again in the Personnel department, all persons are Boss, while, in the Administration there is a tie between the classes. So, the person can be either Boss or Assistant in the Administration department.

Now the decision tree will be as follows:

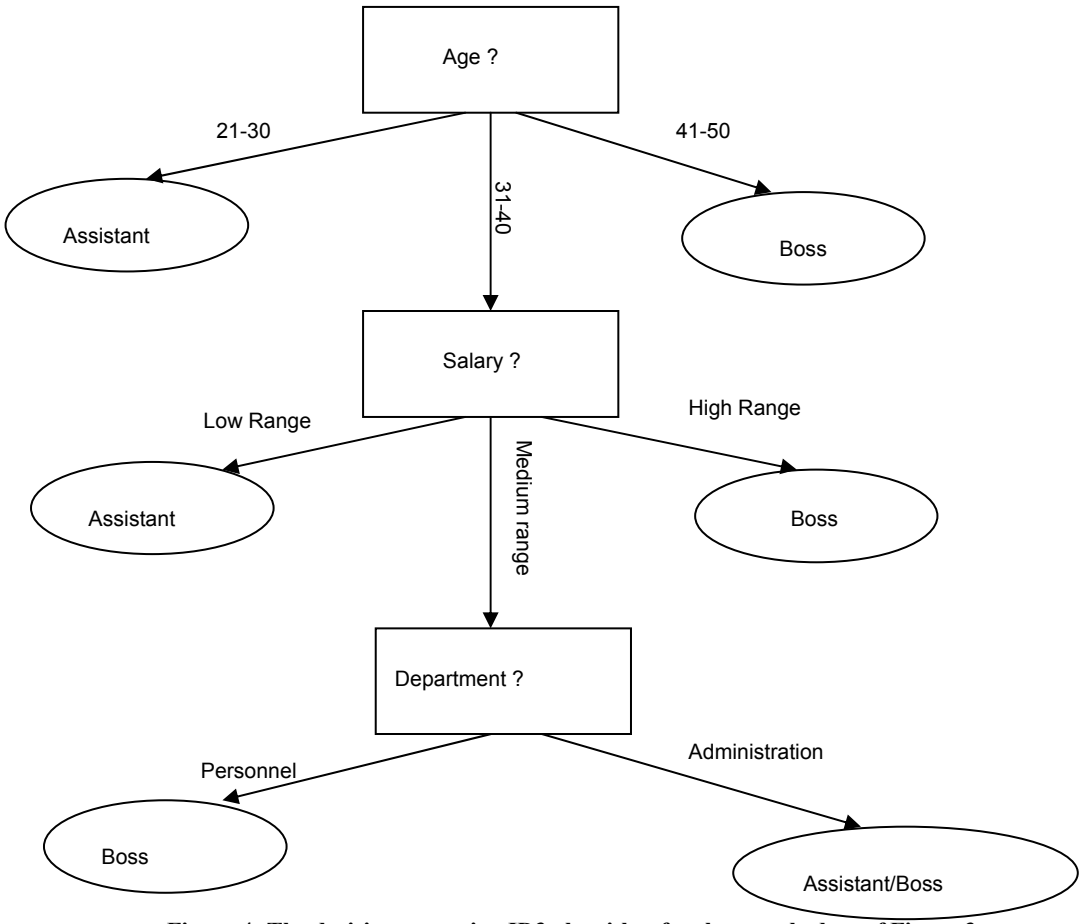

Figure 4: The decision tree using ID3 algorithm for the sample data of Figure 3.

Now, we will take a new dataset and we will classify the class of each tuple by applying the decision tree that we have built above.

Let us discuss, another important classification method called Bayesian classification in the next subsection.

4.4.4 Bayesian Classification

This is a statistical classification, which predicts the probability that a given sample is a member of a particular class. It is based on the Bayes theorem. The Bayesian classification shows better accuracy and speed when applied to large databases. We will discuss here the simplest form of Bayesian classification.

The basic underlying assumptions (also called class conditional independence) for this simplest form of classification known as the native Bayesian classification is:

"The effect of an attribute value on a given class is independent of the values of other attributes"

Let us discuss naive Bayesian classification in more details. But before that, let us, define the basic theorem on which this classification is based.

Bayes Theorem:

Let us assume the following:

X is a data sample whose class is to be determined
H is the hypothesis such that the data sample X belongs to a class C.
P(H | X) is the probability that hypothesis H holds for data sample X . It is also called the posterior probability that condition H holds for the sample X.
P(H) is the prior probability of H condition on the training data.
P(X | H) is the posterior probability of X sample, given that H is true.
P(X) is the prior probability on the sample X.

Please note: We can calculate P(X), P(X | H) and P(H) from the data sample X and training data. It is only P(H | X) which basically defines the probability that X belongs to a class C, and cannot be calculated. Bayes theorem does precisely this function. The Bayer's theorem states:

$$P(H \mid X) = \frac{P(X \mid H)\ P(H)}{P(X)}$$

Now after defining the Bayes theorem, let us explain the Bayesian classification with the help of an example.

i) Consider the sample having an n-dimensional feature vector. For our example, it is 3-dimensional (Department, Age, Salary) vector with training data as given in the Figure 3.

ii) Assume that there are m classes C_1 to C_m. And an unknown sample X. The problem is to data mine which class X belongs to. As per Bayesian classification, the sample is assigned to the class, if the following holds:

$$P(C_i|X) > P(C_j|X) \text{ where j is from 1 to m but } j \neq i$$

In other words the class for the data sample X will be the class, which has the maximum probability for the unknown sample. **Please note:** The $P(C_i|X)$ will be found using:

$$P(C_i| X)= \frac{P(X|C_i)\ P(C_i)}{P(X)} \qquad (3)$$

In our example, we are trying to classify the following data:

X = (Department = "Personal", Age = "31 – 40" and Salary = "Medium Range)
into two classes (based on position) C_1=BOSS OR C_2=Assistant.

iii) The value of P(X) is constant for all the classes, therefore, only $P(X|C_i)\ P(C_i)$ needs to be found to be maximum. Also, if the classes are equally, then,

$P(C_1)=P(C_2)=.....P(C_n)$, then we only need to maximise $P(X|C_i)$.
How is $P(C_i)$ calculated?

$$P(C_i)= \frac{\text{Number of training samples for Class } C_i}{\text{Total Number of Training Samples}}$$

In our example,

$$P(C_1)=\frac{5}{11}$$

and

$$P(C_2)=\frac{6}{11}$$

So $P(C_1) \neq P(C_2)$

iv) $P(X|C_i)$ calculation may be computationally expensive if, there are large numbers of attributes. To simplify the evaluation, in the naïve Bayesian classification, we use the condition of class conditional independence, that is the values of attributes are independent of each other. In such a situation:

$$P(X|C_i)= \prod_{k=1}^{n} P(x_k|C_i) \quad ....(4)$$

where x_k represent the single dimension or attribute.

The $P(x_k|C_i)$ can be calculated using mathematical function if it is continuous, otherwise, if it is categorical then, this probability can be calculated as:

$$P(x_k|C_i)= \frac{\text{Number of training samples of class } C_i \text{ having the value } x_k \text{ for the attribute } A_k}{\text{Number of training samples belonging to } C_i}$$

For our example, we have x_1 as Department= "Personnel"
x_2 as Age="31 – 40" and
x_3 as Salary= "Medium Range"

P (Department= "Personnel" | Position = "BOSS") = 1/5
P (Department= "Personnel" | Position = "Assistant") = 2/6
P (Age="31 – 40" | Position = "BOSS") = 3/5
P (Age="31 – 40" | Position = "Assistant") = 2/6
P (Salary= "Medium Range"| Position = "BOSS") = 3/5
P (Salary= "Medium Range" | Position = "Assistant") = 3/6

Using the equation (4) we obtain:
P (X | Position = "BOSS") = 1/5 * 3/5 * 3/5
P (X | Position = "Assistant") = 2/6 * 2/6 * 3/6

Thus, the probabilities:
P (X | Position = "BOSS") P(Position = "BOSS")
= (1/5 * 3/5 * 3/5) * 5/11
= 0.032727
P (X | Position = "Assistant") P(Position = "Assistant")
= (2/6 * 2/6 * 3/6) * 6/11
= 0.030303

Since, the first probability of the above two is higher, the sample data may be classified into the BOSS position. Kindly check to see that you obtain the same result from the decision tree of *Figure 4.*

4.5 CLUSTERING

Clustering is grouping thing with similar attribute values into the same group. Given a database $D=\{t_1,t_2,\ldots,t_n\}$ of tuples and an integer value k, the Clustering problem is to define a mapping where each tuple t_i is assigned to one cluster K_j, $1<=j<=k$.

A ***Cluster***, K_j, contains precisely those tuples mapped to it. Unlike the classification problem, clusters are not known in advance. The user has to the enter the value of the number of clusters k.

In other words a *cluster* can be defined as the collection of data objects that are similar in nature, as per certain defining property, but these objects are dissimilar to the objects in other clusters.
Some of the clustering examples are as follows:

- To segment the customer database of a departmental store based on similar buying patterns.
- To identify similar Web usage patterns etc.

Clustering is a very useful exercise specially for identifying similar groups from the given data. Such data can be about buying patterns, geographical locations, web information and many more.

Some of the clustering Issues are as follows:

- **Outlier handling:** How will the outlier be handled? (outliers are the objects that do not comply with the general behaviour or model of the data) Whether it is to be considered or it is to be left aside while calculating the clusters?
- **Dynamic data:** How will you handle dynamic data?
- **Interpreting results:** How will the result be interpreted?
- **Evaluating results:** How will the result be calculated?
- **Number of clusters:** How many clusters will you consider for the given data?
- **Data to be used:** whether you are dealing with quality data or the noisy data? If, the data is noisy how is it to be handled?
- **Scalability:** Whether the algorithm that is used is to be scaled for small as well as large data set/database.

There are many different kinds of algorithms for clustering. However, we will discuss only three basic algorithms. You can refer to more details on clustering from the further readings.

4.5.1 Partitioning Clustering

The partitioning clustering algorithms constructs k partitions from a given n objects of the data. Here $k \leq n$ and each partition must have at least one data object while one object belongs to **only one** of the partitions. A partitioning clustering algorithm normally requires users to input the desired number of clusters, k.

Some of the partitioning clustering algorithms are as follows:

- Squared Error
- K-Means

Now in this unit, we will briefly discuss these algorithms.

Squared Error Algorithms

The most frequently used criterion function in partitioning clustering techniques is the squared error criterion. The method of obtaining clustering by applying this approach is as follows:

Squared Error Clustering Method:

(1) Select an initial partition of the patterns with a fixed number of clusters and cluster centers.
(2) Assign each pattern to its closest cluster center and compute the new cluster centers as the centroids of the clusters. Repeat this step until convergence is achieved, i.e., until the cluster membership is stable.
(3) Merge and split clusters based on some heuristic information, optionally repeating step 2.

Some of the parameters that are used in clusters are as follows:

$$\text{Centriod}(C_m) = \sum_{i=1}^{N} t_{mi} / N$$

$$\text{Radius } (R_m) = \sqrt{\sum_{i-1}^{N} (t_{mi} - C_m)^2 / N}$$

$$\text{Diameter } (D_m) = \sqrt{\sum_{i-1}^{N} \sum_{j=1}^{N} (t_{mi} - t_{mj})^2 / (N * (N-1))}$$

A detailed discussion on this algorithm is beyond the scope of this unit. You can refer to more details on clustering from the further readings.

K-Means clustering

In the K-Means clustering, initially a set of clusters is randomly chosen. Then iteratively, items are moved among sets of clusters until the desired set is reached. A high degree of similarity among elements in a cluster is obtained by using this algorithm. For this algorithm a set of clusters $K_i=\{t_{i1},t_{i2},\ldots,t_{im}\}$ is given , the cluster mean is:

$$m_i = (1/m)(t_{i1} + \ldots + t_{im}) \quad \ldots(5)$$

Where t_i represents the tuples and m represents the mean

The K-Means algorithm is as follows:

Input :
D= { $t_1,t_2,\ldots t_n$} //Set of elements
A //Adjacency matrix showing distance between elements.
k //Number of desired clusters.

Output :
K //Set of Clusters

K-Means Algorithm:

```
Assign initial values for means m1,m2..mk;
Repeat
        Assign each item ti to the cluster which has the closest mean;
        Calculate new mean for each cluster;
Until convergence criteria is met.
```

K-Means Example:

Let us take the number of clusters as 2 and the following input set is given to us:
Input set={1,2,3, 5,10,12,22,32,16,18}

- Step 1:We randomly assign means: $m_1=3,m_2=5$
- Step 2:$K_1=\{1,2,3\}$, $K_2=\{5,10,12,22,32,16,18\}$, $m_1=2,m_2=16.43$ (calculated mean using the formula (5)).

Now redefine cluster as per the closest mean:

- Step 3:$K_1=\{1,2,3,5\},K_2=\{10,12,22,32,16,18\}$

Calculate the mean once again:

- $m_1=2.75,m_2=18.33$
- Step 4:$K_1=\{1,2,3,5\},K_2=\{10,12,22,32,16,18\}$, $m_1=2.75,m_2=18.33$
- Stop as the clusters with these means are the same.

4.5.2 Nearest Neighbour Clustering

In this approach, items are iteratively merged into the existing clusters that are closest. It is an incremental method. The threshold, t, used to determine if items are added to existing clusters or a new cluster is created. This process continues until all patterns are labeled or no additional labeling occurs.

The Nearest Neighbour algorithm is as follows:

Input :

D= { $t_1,t_2,\ldots t_n$} //Set of elements
A //Adjacency matrix showing distance between elements.
k //Number of desired clusters.

Output :

K //Set of Clusters

Nearest Neighbour algorithm :

$K_1=\{t_1\}$;
$K=\{K_1\}$;
k=1;
for i=2 to n do
    Find the t_m in some cluster Km in K such that distance(t_i,t_m) is the smallest;
    If dis(t_i,t_m)<= t then
        $K_m=K_m \cup t_i$;
    Else
        k=k +1;
        $K_k=\{t_i\}$;

4.5.3 Hierarchical Clustering

In this method, the clusters are created in levels and depending upon the threshold value at each level the clusters are again created.

An agglomerative approach begins with each tuple in a distinct cluster and successively merges clusters together until a stopping criterion is satisfied. This is the bottom up approach.

A divisive method begins with all tuples in a single cluster and performs splitting until a stopping criterion is met. This is the top down approach.

A hierarchical algorithm yields a dendrogram representing the nested grouping of tuples and similarity levels at which groupings change. Dendrogram is a tree data structure which illustrates hierarchical clustering techniques. Each level shows clusters for that level. The leaf represents individual clusters while the root represents one cluster.

Most hierarchical clustering algorithms are variants of the single-link, average link and complete-link algorithms. Out of these the single-link and complete-link algorithms are the most popular. These two algorithms differ in the way they characterise the similarity between a pair of clusters.

In the single-link method, the distance between two clusters is the minimum of the distances between all pairs of patterns drawn from the two clusters (one pattern from the first cluster, the other from the second).

In the complete-link algorithm, the distance between two clusters is the maximum of all pair-wise distances between patterns in the two clusters.

In either case, two clusters are merged to form a larger cluster based on minimum distance criteria.

You can refer to more detail on the hierarchical clustering algorithms from the further readings.

☞ Check Your Progress 2

1) What is the classification of data? Give some examples of classification.

……

……

……

2) What is clustering?

……

……

……

3) How is clustering different from classification?

……

……

……

4.6 ASSOCIATION RULE MINING

The task of association rule mining is to find certain association relationships among a set of items in a dataset/database. The association relationships are described in association rules. In association rule mining there are two measurements, *support* and *confidence*. The confidence measure indicates the rule's strength, while support corresponds to the frequency of the pattern.

A typical example of an association rule created by data mining often termed to as "market basket data" is: " 80% of customers who purchase bread also purchase butter."

Other applications of data mining include cache customisation, advertisement personalisation, store layout and customer segmentation etc. All these applications try to determine the associations between data items, if it exists to optimise performance.

Formal Definition:

Let $I = \{i_1, i_2, \ldots, i_m\}$ be a set of literals, called items. Let D be a set of transactions, where each transaction T is a set of items such that $T \subseteq I$. TID indicates a unique

transaction identifier. An association rule is an implication of the form X ➔ Y, where $X \subset I$, $Y \subset I$, and $X \cap Y = \varnothing$. X is called the antecedent while Y is called the consequence of the rule.

The rule X ➔ Y has support s in the transaction set D if s% of transactions in D contains $X \cup Y$. The rule has confidence c if c% of transactions in D that contains X also contains Y. Support indicates how frequently the pattern occurs, while confidence indicates the strength of the rule.

Given a user specified minimum support and minimum confidence, the problem of mining association rules is to find all the association rules whose support and confidence are larger than the minimum support and minimum confidence. Thus, this approach can be broken into two sub-problems as follows:

(1) Finding the frequent itemsets which have support above the predetermined minimum support.
(2) Deriving all rules, based on each frequent itemset, which have confidence more than the minimum confidence.

There are a lots of ways to find the large itemsets but we will only discuss the Apriori Algorithm.

Apriori Algorithm: For finding frequent itemsets

The apriori algorithm applies the concept that if an itemset has minimum support, then all its subsets also have minimum support. An itemset having minimum support is called frequent itemset or large itemset. So any subset of a frequent itemset must also be frequent.

Apriori algorithm generates the candidate itemsets to be counted in the pass, by using only the large item set found in the previous pass – without considering the transactions in the database.

It starts by finding all frequent 1-itemsets (itemsets with 1 item); then consider 2-itemsets from these 1-itemsets, and so forth. During each iteration only candidates found to be frequent in the previous iteration are used to generate a new candidate set during the next iteration. The algorithm terminates when there are no frequent k-itemsets.

Notations that are used in Apriori algorithm are given below:

k-itemset	An itemset having k items
L_k	Set of frequent k-itemset (those with minimum support)
C_k	Set of candidate k-itemset (potentially frequent itemsets)

Apriori algorithm function takes as argument L_{k-1} and returns a superset of the set of all frequent k-itemsets. It consists of a join step and a prune step. The Apriori algorithm is given below :

APRIORI

1. k = 1
2. Find frequent set L_k from C_k of all candidate itemsets
3. Form C_{k+1} from L_k; k = k + 1
4. Repeat 2-3 until C_k is empty

Details about steps 2 and 3

Step 2: Scan the data set D and count each itemset in C_k , if it is greater than minimum support, it is frequent

Step 3:

- For k=1, C_1 = all frequent 1-itemsets. (all individual items).
- For k>1, generate C_k from Lk-1 as follows:

The join step

C_k = k-2 way join of L_{k-1} with itself

If both $\{a_1, \ldots, a_{k-2}, a_{k-1}\}$ & $\{a_1, \ldots, a_{k-2}, a_k\}$ are in L_{k-1}, then add $\{a_1, \ldots, a_{k-2}, a_{k-1}, a_k\}$ to C_k

(We keep items sorted).

The prune step

Remove $\{a_1, \ldots, a_{k-2}, a_{k-1}, a_k\}$ if it contains a non-frequent (k-1) subset.

{In the prune step, delete all itemsets $c \in C_k$ such that some (k-1)-subset of C is not in L_{k-1}.}

Example: Finding frequent itemsets:

Consider the following transactions with minimum support s=30% for finding the frequent itemsets by applying Apriori algorithm:

Transaction ID	Item(s) purchased
1	Shirt, Trouser
2	Shirt, Trouser, Coat
3	Coat, Tie, Tiepin
4	Coat, Shirt, Tie, Trouser
5	Trouser, Belt
6	Coat, Tiepin, Trouser
7	Coat, Tie
8	Shirt
9	Shirt, Coat
10	Shirt, Handkerchief

Method of finding the frequent itemset is as follows:

Pass Number	Candidates	Large itemsets (≥ 3)
1	C_1={Belt 1, Coat 6, Handkerchief 1, Shirt 6, Tie 3, Tiepin 2, Trouser 5 }	L_1={Coat 6, Shirt 6, Tie 3, Trouser 5 }
2	C_2= {{Coat, Shirt} 3, {Coat, Tie} 3, {Coat, Trouser} 3, {Shirt, Tie} 1, {Shirt, Trouser} 3, {Tie, Trouser} 1 }	L_2={{Coat, Shirt} 3, {Coat, Tie} 3, {Coat, Trouser} 3, {Shirt, Trouser} 3 }
3	C_3= {{Coat, Shirt, Trouser} 2}	L_3= Ø

The calculation of 3-itemsets is mentioned below:

Join operation yields 3 item sets as: {{Coat, Shirt, Tie}, {Coat, Shirt, Trouser}, {Coat, Tie, Trouser}}

However, the Prune operation removes two of these items from the set due to the following reasons:

{Coat, Shirt, Tie} is pruned as {Shirt, Tie} is not in L_2
{Coat, Shirt, Trouser} is retained as {Coat, Shirt}, {Coat, Trouser} and {Shirt, Trouser} all three are in L_2
{Coat, Tie, Trouser} is pruned as {Tie, Trouser} is not in L_2

The set L={ L_1 , L_2 , L_3}

The following algorithm creates the association rules from the set L so created by the Apriori algorithm.

Algorithm to generate the Association Rules:

Input:

D	//Database of transactions
I	//Items
L	//Large itemsets
s	// Support
c	// Confidence

Output:

R	//Association rules satisfying minimum s and c

AR algorithm:

```
R= Ø
For each l ε L do    // for each large item ( l ) in the set L
        For each x ⊂ I  such that x <> Ø and x <>l do
                if support(l) /support(x)  ≥ c   then
                        R=R ∪  {x ➔ (l - x)};
```

Apriori Advantages/Disadvantages:

The following are the advantages and disadvantages of the Apriori algorithm:

- **Advantages:**
  - It uses large itemset property.
  - It easy to implement.
- **Disadvantages:**
  - It assumes transaction database is memory resident.

4.7 APPLICATIONS OF DATA MINING PROBLEM

Some of the applications of data mining are as follows:

- **Marketing and sales data analysis:** A company can use customer transactions in their database to segment the customers into various types. Such companies may launch products for specific customer bases.
- **Investment analysis:** Customers can look at the areas where they can get good returns by applying the data mining.
- **Loan approval:** Companies can generate rules depending upon the dataset they have. On that basis they may decide to whom, the loan has to be approved.
- **Fraud detection:** By finding the correlation between faults, new faults can be detected by applying data mining.

- **Network management:** By analysing pattern generated by data mining for the networks and its faults, the faults can be minimised as well as future needs can be predicted.
- **Risk Analysis:** Given a set of customers and an assessment of their risk-worthiness, descriptions for various classes can be developed. Use these descriptions to classify a new customer into one of the risk categories.
- **Brand Loyalty:** Given a customer and the product he/she uses, predict whether the customer will change their products.
- **Housing loan prepayment prediction:** Rule discovery techniques can be used to accurately predict the aggregate number of loan prepayments in a given quarter as a function of prevailing interest rates, borrower characteristics and account data.

4.8 COMMERCIAL TOOLS OF DATA MINING

Commercial Tools:

1) **AC2**, provides graphical tools for data preparation and building decision trees.
2) **Business Miner**, data mining product positioned for the mainstream business user.
3) **C4.5**, the "classic" decision-tree tool, developed by **J. R. Quinlan**
4) **C5.0/See5**, constructs classifiers in the form of decision trees and rulesets.
5) **CART**, decision-tree software, combines an easy-to-use GUI with advanced features for data mining, data pre-processing and predictive modeling.
6) **Cognos Scenario**, allows you to quickly identify and rank the factors that have a significant impact on your key business measures.
7) **Decisionhouse**, provides data extraction, management, pre-processing and visualisation, plus customer profiling, segmentation and geographical display.
8) **Kernel Miner**, decision-tree-based classifier with fast DB access.
9) **Knowledge Seeker**, high performance interactive decision tree analytical tool.
10) **SPSS AnswerTree**, easy to use package with four decision tree algorithms - two types of CHAID, CART, and QUEST.
11) **XpertRule Miner** (Attar Software), provides graphical decision trees with the ability to embed as ActiveX components.
12) **AIRA**, a rule discovery, data and knowledge visualisation tool. AIRA for Excel extracts rules from MS-Excel spreadsheets.
13) **Datamite**, enables rules and knowledge to be discovered in ODBC-compliant relational databases.
14) **SuperQuery**, business Intelligence tool; works with Microsoft Access and Excel and many other databases.
15) **WizWhy**, automatically finds all the if-then rules in the data and uses them to summarise the data, identify exceptions, and generate predictions for new cases.
16) **XpertRule Miner** (Attar Software) provides association rule discovery from any ODBC data source.
17) **DMSK: Data-Miner Software Kit :Task**: Collection of tools for efficient mining of big data (Classification, Regression, Summarisation, Deviation Detection multi-task tools).

16) **OSHAM** Task: Task (Clustering) interactive-graphic system for discovering concept hierarchies from unsupervised data

17) **DBMiner** is a data mining system for interactive mining of multiple-level knowledge in large relational databases.

Free Tools:

1) **EC4.5**, a more efficient version of C4.5, which uses the best among three strategies at each node construction.

2) **IND**, provides CART and C4.5 style decision trees and more. Publicly available from NASA but with export restrictions.

3) **ODBCMINE**, shareware data-mining tool that analyses ODBC databases using the C4.5, and outputs simple IF..ELSE decision rules in ASCII.

4) **OC1**, decision tree system continuous feature values; builds decision trees with linear combinations of attributes at each internal node; these trees then partition the space of examples with both oblique and axis-parallel hyper planes.

5) **PC4.5**, a parallel version of C4.5 built with Persistent Linda system.

6) **SE-Learn**, Set Enumeration (SE) trees generalise decision trees. Rather than splitting by a single attribute, one recursively branches on all (or most) relevant attributes. (LISP)

7) **CBA**, mines association rules and builds accurate classifiers using a subset of association rules.

8) **KINOsuite**-PR extracts rules from trained neural networks.

9) **RIPPER**, a system that learns sets of rules from data

☞ Check Your Progress 3

1) What is association rule mining?

……

……

……

2) What is the application of data mining in the banking domain?

……

……

……

3) Apply the Apriori algorithm for generating large itemset on the following dataset:

Transaction ID	Items purchased
T100	$A_1a_3a_4$
T200	$A_2a_3a_5$
T300	$A_1a_2a_3a_5$
T400	A_2a_5

……

……

……

4.9 SUMMARY

1) Data mining is the process of automatic extraction of interesting (non trivial, implicit, previously unknown and potentially useful) information or pattern from the data in large databases.

2) Data mining is one of the steps in the process of Knowledge Discovery in databases.

3) In data mining tasks are classified as: Classification, Clustering and Association rules.

4) The classification task maps data into predefined classes.

5) Clustering task groups things with similar properties/ behaviour into the same groups.

6) Association rules find the association relationship among a set of objects.

7) Data mining is applied in every field whether it is Games, Marketing, Bioscience, Loan approval, Fraud detection etc.

4.10 SOLUTIONS /ANSWERS

Check Your Progress 1

1) Data mining is the process of automatic extraction of interesting (non trivial, implicit, previously unknown and potentially useful) information or patterns from the data in large databases.

2) Data mining is only one of the many steps involved in knowledge discovery in databases. The various steps in KDD are data extraction, data cleaning and preprocessing, data transformation and reduction, data mining and knowledge interpretation and representation.

3) The query language of OLTP is well defined and it uses SQL for it, while, for data mining the query is poorly defined and there is no precise query language. The data used in OLTP is operational data while in data mining it is historical data. The output of the query of OLTP is precise and is the subset of the data while in the case of data mining the output is fuzzy and it is not a subset of the data.

4) The different data-mining tasks are: Classification, Clustering and Association Rule Mining.

Check Your Progress 2

1) The classification task maps data into predefined groups or classes. The class of a tuple is indicated by the value of a user-specified goal attribute. Tuples consists of a set of predicating attributes and a goal attribute. The task is to discover some kind of relationship between the predicating attributes and the goal attribute, so that the discovered knowledge can be used to predict the class of new tuple(s).

   Some of the examples of classification are: Classification of students grades depending upon their marks, classification of customers as good or bad customer in a bank.

2) The task of clustering is to group the tuples with similar attribute values into the same class. Given a database of tuples and an integer value k, Clustering defines mapping, such that, tuples are mapped to different clusters.

3) In classification, the classes are predetermined, but, in the case of clustering the groups are not predetermined. The number of clusters has to be given by the user.

Check Your Progress 3

1) The task of association rule mining is to search for interesting relationships among items in a given data set. Its original application is on "market basket data". The rule has the form X➔Y, where X and Y are sets of items and they do not intersect.

2) The data mining application in banking are as follows:

1. Detecting patterns of fraudulent credit card use.
2. Identifying good customers.
3. Determining whether to issue a credit card to a person or not.
4. Finding hidden correlations between different financial indicators.

3) The dataset D given for the problem is:

Transaction ID	Items purchased
T100	$a_1a_3a_4$
T200	$a_2a_3a_5$
T300	$a_1a_2a_3a_5$
T400	a_2a_5

Assuming the minimum support as 50% for calculating the large item sets. As we have 4 transaction, at least 2 transaction should have the data item.

1. scan D ➔ C_1: a_1:2, a_2:3, a_3:3, a_4:1, a_5:3
➔ L_1: a_1:2, a_2:3, a_3:3, a_5:3
➔ C_2: a_1a_2, a_1a_3, a_1a_5, a_2a_3, a_2a_5, a_3a_5
2. scan D ➔ C_2: a_1a_2:1, a_1a_3:2, a_1a_5:1, a_2a_3:2, a_2a_5:3, a_3a_5:2
➔ L_2: a_1a_3:2, a_2a_3:2, a_2a_5:3, a_3a_5:2
➔ C_3: $a_1a_2a_3$, $a_1a_2a_5$, $a_2a_3a_5$
➔ Pruned C_3: $a_2a_3a_5$
3. scan D ➔ L_3: $a_2a_3a_5$:2

Thus L={L_1,L_2,L_3}

4.11 FURTHER READINGS

1) *Data Mining Concepts and Techniques*, J Han, M Kamber, Morgan Kaufmann Publishers, 2001.

2) *Data Mining*, A K Pujari, 2004.