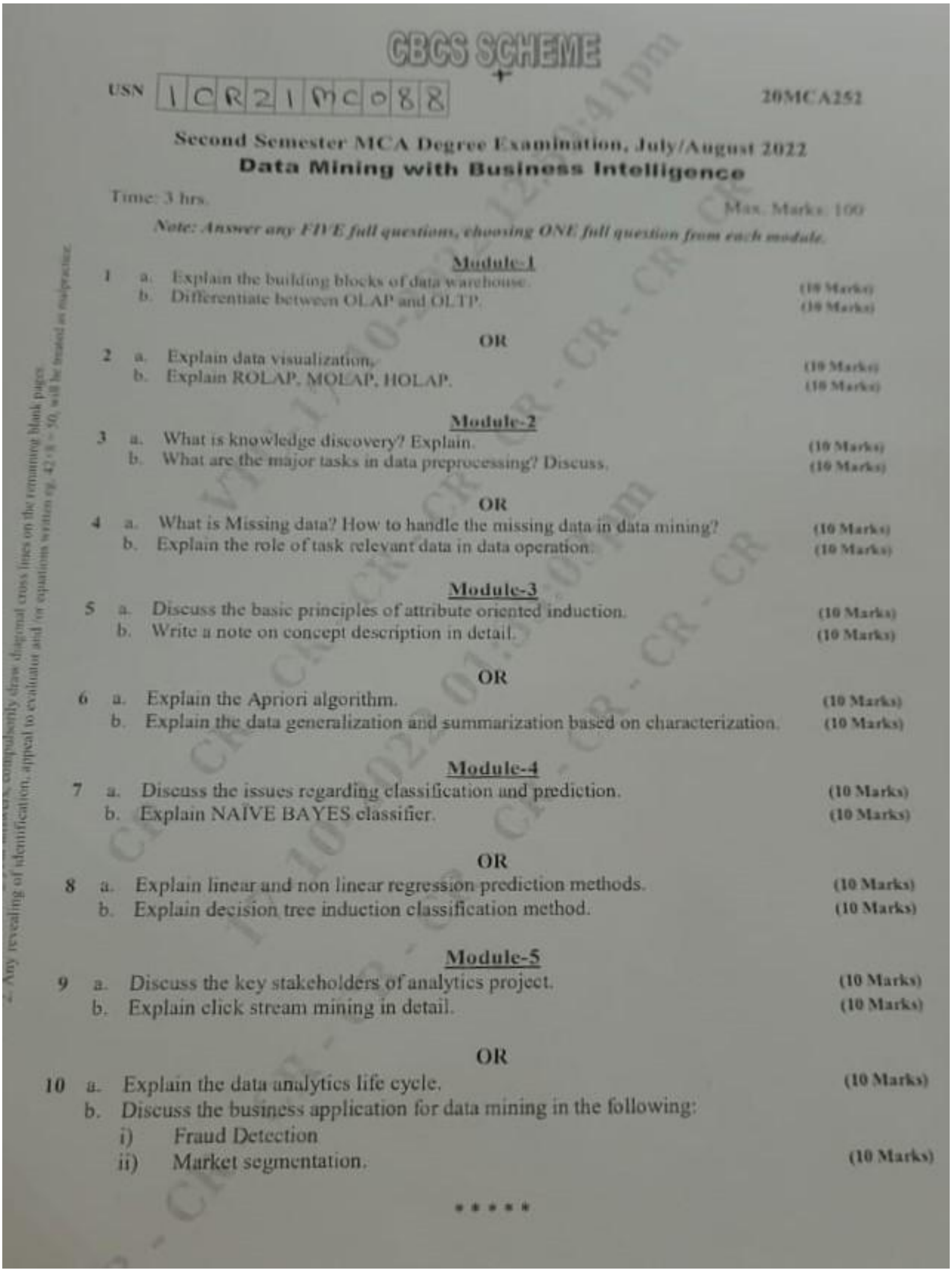


CBCS SCHEME

USN 1CR21MC088

20MCA252

Second Semester MCA Degree Examination, July/August 2022

Data Mining with Business Intelligence

Time: 3 hrs.

Max. Marks: 100

Note: Answer any FIVE full questions, choosing ONE full question from each module.

...swers, compulsorily draw diagonal cross lines on the remaining blank pages.

2. Any revealing of identification, appeal to evaluator and /or equations written eg, 42+8 = 50, will be treated as malpractice.

Module-1

1 a. Explain the building blocks of data warehouse. (10 Marks)

b. Differentiate between OLAP and OLTP. (10 Marks)

OR

2 a. Explain data visualization. (10 Marks)

b. Explain ROLAP, MOLAP, HOLAP. (10 Marks)

Module-2

3 a. What is knowledge discovery? Explain. (10 Marks)

b. What are the major tasks in data preprocessing? Discuss. (10 Marks)

OR

4 a. What is Missing data? How to handle the missing data in data mining? (10 Marks)

b. Explain the role of task relevant data in data operation. (10 Marks)

Module-3

5 a. Discuss the basic principles of attribute oriented induction. (10 Marks)

b. Write a note on concept description in detail. (10 Marks)

OR

6 a. Explain the Apriori algorithm. (10 Marks)

b. Explain the data generalization and summarization based on characterization. (10 Marks)

Module-4

7 a. Discuss the issues regarding classification and prediction. (10 Marks)

b. Explain NAIVE BAYES classifier. (10 Marks)

OR

8 a. Explain linear and non linear regression prediction methods. (10 Marks)

b. Explain decision tree induction classification method. (10 Marks)

Module-5

9 a. Discuss the key stakeholders of analytics project. (10 Marks)

b. Explain click stream mining in detail. (10 Marks)

OR

10 a. Explain the data analytics life cycle. (10 Marks)

b. Discuss the business application for data mining in the following:

i) Fraud Detection

ii) Market segmentation. (10 Marks)

* * * * *

1.a. Explain the building blocks of data warehouse.

The Building Blocks of Data Warehouse:

A) A data warehouse is a relational database that is designed for query and analysis.

It separates an analysis workload from a transaction workload and enables an organization to consolidate data from several sources.

B) Dimensional modeling: It is developed to be oriented around query performance and ease of use. The dimensional modeling handle approach is at a logical level.

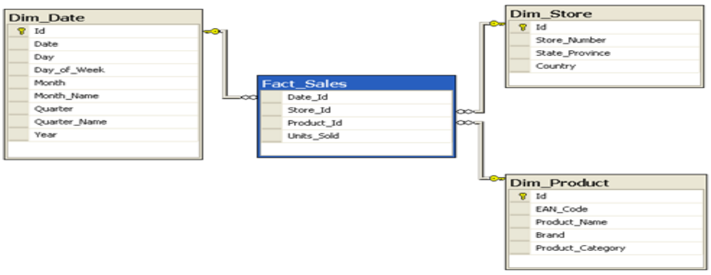
Facts or Business measurement-numeric values

Dimensions or Descriptors specify the facts- text values

C)Star Scheme: The fact table is at the center of the schema surrounded by dimensions tables.

Eg. At the center of the schema there is fact table FACT-SALES.

The fact table is sourrounded by the dimension tables Dim-Data, Dim-Store, Dim-Product.



D)Fact Table: It is a dimensional model in data warehouse design. Facts are also known as measurements.

Types of Fact table are Transactional, periodic and accumulating tables.

Transactional –Transactional fact table is the most basic one that each grain associated with it indicated as “one row per line in a transaction”, e.g.,Price- every line item appears on an invoice.

Periodic snapshots – Periodic snapshots fact table stores the data that is a snapshot in a period of time. Ex. Sales period

Accumulating snapshots – The accumulating snapshots fact table describes the activity of a business process that has a clear beginning and end. Eg. Purchasing: Requisition, Purchase order, Vendor Invoice, Delivery, Payment.

b. Differentiate between OLAP and OLTP.

Feature	**OLTP**	**OLAP**
Characteristic	operational processing	informational processing
Orientation	transaction	analysis
User	clerk, DBA, database professional	knowledge worker (e.g., manager, executive, analyst)
Function	day-to-day operations	long-term informational requirements, decision support
DB design	ER based, application-oriented	star/snowflake, subject-oriented

Data	current; guaranteed up-to-date	historical; accuracy maintained over time
Summarization	primitive, highly detailed	summarized, consolidated
View	detailed, flat relational	summarized, multidimensional
Unit of work	short, simple transaction	complex query
Access	read/write	mostly read
Focus	data in	information out
Operations	index/hash on primary key	lots of scans
No. of records accessed	tens	millions
Number of users	thousands	hundreds
DB size	100 MB to GB	100 GB to TB
Priority	high performance, high availability	high flexibility, end-user autonomy
Metric	transaction throughput	query throughput, response time

2. a. Explain data visualization.

Data visualization is defined as a graphical representation that contains the information and the data.

By using visual elements like charts, graphs, and maps, data visualization techniques provide an accessible way to see and understand trends, outliers, and patterns in data.

In modern days we have a lot of data in our hands i.e, in the world of Big Data, data visualization tools, and technologies are crucial to analyze massive amounts of information and make data-driven decisions.

It is used in many areas such as:

To model complex events.

Visualize phenomenons that cannot be observed directly, such as weather patterns, medical conditions, or mathematical relationships.

Different Types of Analysis for Data Visualization

Mainly, there are three different types of analysis for Data Visualization:

Univariate Analysis: In the univariate analysis, we will be using a single feature to analyze almost all of its properties.

Bivariate Analysis: When we compare the data between exactly 2 features then it is known as bivariate analysis.

Multivariate Analysis: In the multivariate analysis, we will be comparing more than 2 variables.

Univariate Analysis Techniques for Data Visualization

1. Distribution Plot

- It is one of the best univariate plots to know about the distribution of data.
- When we want to analyze the impact on the target variable(output) with respect to an independent variable(input), we use distribution plots a lot.
- This plot gives us a combination of both probability density functions(pdf) and histogram in a single plot.

Implementation:

- The distribution plot is present in the **Seaborn** package.

Some conclusions inferred from the above distribution plot:

From the above distribution plot we can conclude the following observations:

- We have observed that we created a distribution plot on the feature **'Age'**(input variable) and we used different colors for the **Survival status**(output variable) as it is the class to be predicted.
- There is a huge overlapping area between the PDFs for different combinations.
- In this plot, the sharp block-like structures are called histograms, and the smoothed curve is known as the Probability density function(PDF).

NOTE:

The Probability density function(PDF) of a curve can help us to capture the underlying distribution of that feature which is one major takeaway from Data visualization or Exploratory Data Analysis(EDA).

2. Box and Whisker Plot

- This plot can be used to obtain more **statistical details** about the data.
- The straight lines at the maximum and minimum are also called **whiskers**.
- Points that lie outside the whiskers will be considered as an outlier.
- The box plot also gives us a description of the **25th, 50th,75th quartiles**.
- With the help of a box plot, we can also determine the **Interquartile range(IQR)** where maximum details of the data will be present. Therefore, it can also give us a clear idea about the outliers in the dataset.

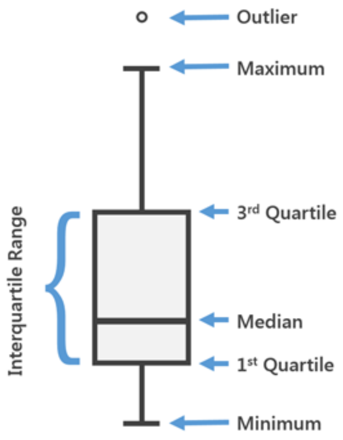
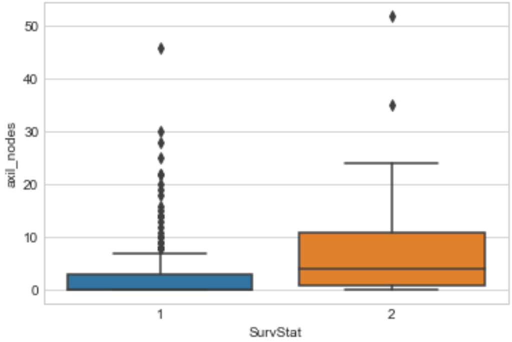


Fig. General Diagram for a Box-plot

- Boxplot is available in the **Seaborn** library.
- Here x is considered as the dependent variable and y is considered as the independent variable. These box plots come under **univariate analysis**, which means that we are exploring data only with one variable.
- Here we are trying to check the impact of a feature named **“axil_nodes”** on the class named **“Survival status”** and not between any two independent features.



Some conclusions inferred from the above box plot:

From the above box and whisker plot we can conclude the following observations:

- How much data is present in the 1st quartile and how many points are outliers etc.
- For class 1, we can see that it is very little or no data is present between the median and the 1st quartile.
- There are more outliers for class 1 in the feature named **axil_nodes**.

NOTE:

We can get details about outliers that will help us to well prepare the data before feeding it to a model since outliers influence a lot of Machine learning models.

3. Violin Plot

- The violin plots can be considered as a combination of Box plot at the middle and distribution plots**(Kernel Density Estimation)** on both sides of the data.
- This can give us the description of the distribution of the dataset like whether the distribution is **multimodal**, **Skewness**, etc.
- It also gives us useful information like a **95% confidence interval**.

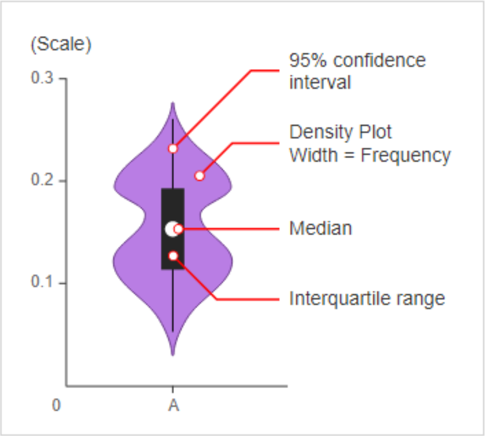
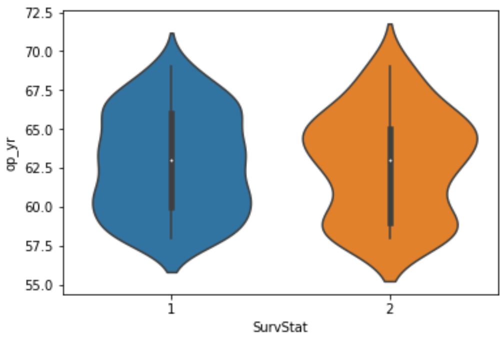


Fig. General Diagram for a Violin-plot



Some conclusions inferred from the above violin plot:

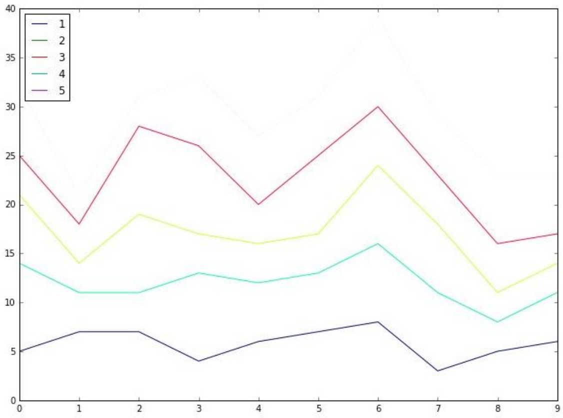
From the above violin plot we can conclude the following observations:

- The median of both classes is close to 63.
- The maximum number of persons with class 2 has an **op_yr** value of 65 whereas, for persons in class1, the maximum value is around 60.
- Also, the 3rd quartile to median has a lesser number of data points than the median to the 1st quartile.

Bivariate Analysis Techniques for Data Visualization

1. Line Plot

- This is the plot that you can see in the nook and corners of any sort of analysis between 2 variables.
- The line plots are nothing but the values on a series of data points will be connected with straight lines.
- The plot may seem very simple but it has more applications not only in machine learning but in many other areas.



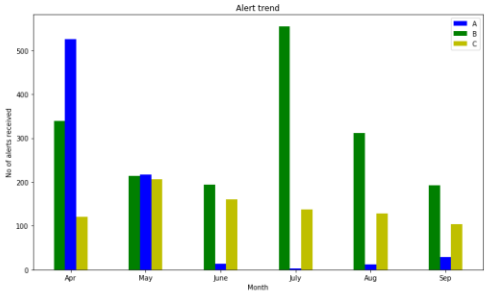
Some conclusions inferred from the above line plot:

From the above line plot we can conclude the following observations:

- These are used right from performing distribution Comparison using **Q-Q plots** to CV tuning using the **elbow method**.
- Used to analyze the performance of a model using the **ROC- AUC curve**.

2. Bar Plot

- This is one of the widely used plots, that we would have seen multiple times not just in data analysis, but we use this plot also wherever there is a trend analysis in many fields.
- Though it may seem simple it is powerful in analyzing data like **sales figures every week, revenue from a product**, **Number of visitors to a site on each day of a week**, etc.



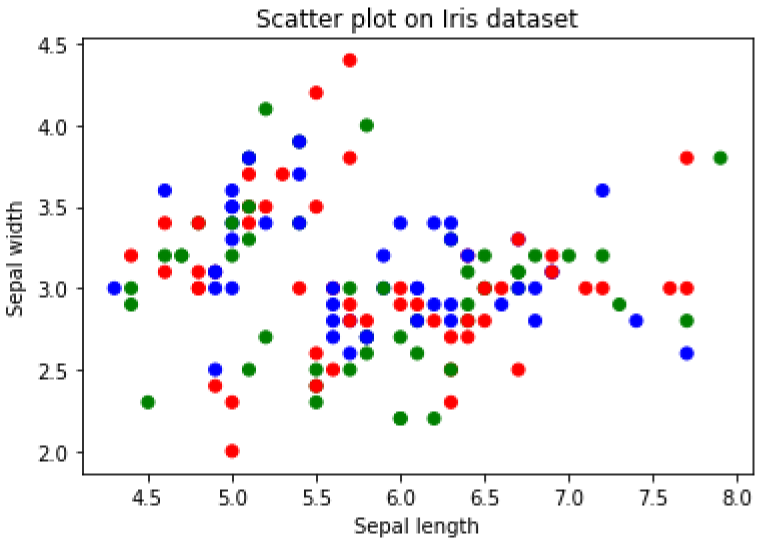
Some conclusions inferred from the above bar plot:

From the above bar plot we can conclude the following observations:

- We can visualize the data in a cool plot and can convey the details straight forward to others.
- This plot may be simple and clear but it’s not much frequently used in Data science applications.

3. Scatter Plot

- It is one of the most commonly used plots used for visualizing simple data in Machine learning and Data Science.
  - This plot describes us as a representation, where each point in the entire dataset is present with respect to any 2 to 3 features(Columns).
- Scatter plots are available in both 2-D as well as in 3-D. The 2-D scatter plot is the common one, where we will primarily try to find the patterns, clusters, and separability of the data.



Some conclusions inferred from the above Scatter plot:

From the above Scatter plot we can conclude the following observations:

- The colors are assigned to different data points based on how they were present in the dataset **i.e, target column representation.**
- We can color the data points as per their class label given in the dataset.

b. Explain ROLAP, MOLAP and HOLAP.

We have four types of OLAP servers:

1. *Relational OLAP*

- ROLAP servers are placed between relational back-end server and client front-end tools.
- To store and manage warehouse data, ROLAP uses relational or extended-relational DBMS.
- ROLAP includes the following:
  - Implementation of aggregation navigation logic.
  - Optimization for each DBMS back end.
  - Additional tools and services.

2. *Multidimensional OLAP*

- MOLAP uses array-based multidimensional storage engines for multidimensional views of data.
- With multidimensional data stores, the storage utilization may be low if the data set is sparse.
- Many MOLAP server use two levels of data storage representation to handle dense and sparse data sets.

3. *Hybrid OLAP (HOLAP)*

- Hybrid OLAP is a combination of both ROLAP and MOLAP.
- It offers higher scalability of ROLAP and faster computation of MOLAP.
- HOLAP servers allows to store the large data volumes of detailed information.
- The aggregations are stored separately in MOLAP store.

4. *Specialized SQL Servers*

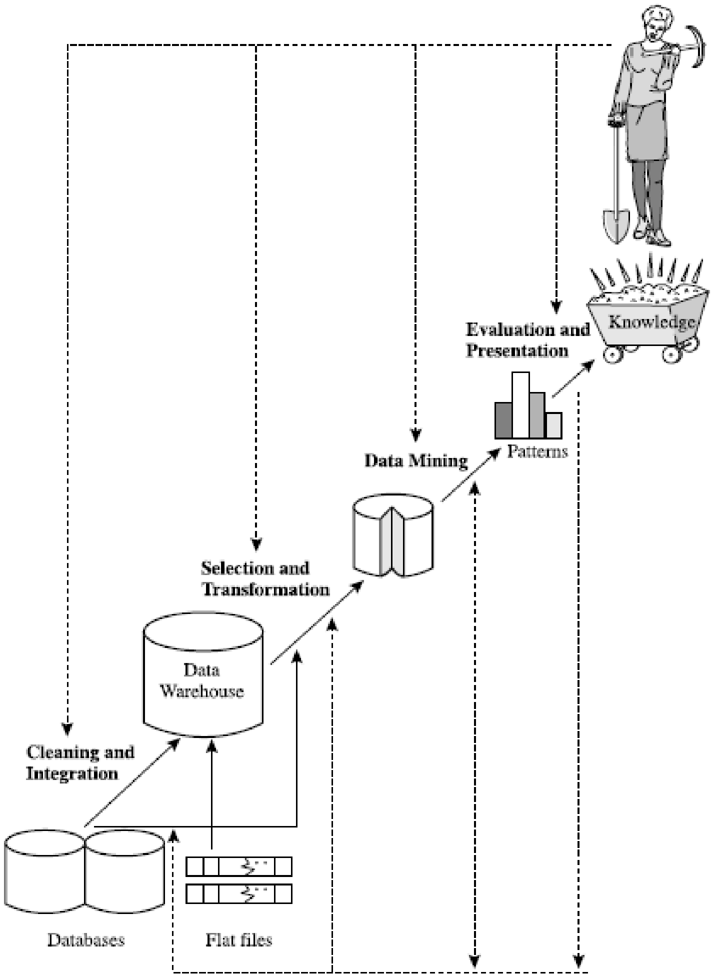
- Specialized SQL servers provide advanced query language and query processing support for SQL queries over star and snowflake schemas in a read-only environment.

3. a. What is knowledge discovery? Explain

KDD (Knowledge Discovery from Data) Process

- KDD stands for knowledge discoveries from database. There are some pre-processing operations which are required to make pure data in data warehouse before use that data for Data Mining processes.
- A view data mining as simply an essential step in the process of knowledge discovery. Knowledge discovery as a process is depicted in Figure 2 and consists of an iterative sequence of the following steps:
  - ✓ **Data cleaning:** To remove noise and inconsistent data.

- ✓ **Data integration:** where multiple data sources may be combined.
- ✓ **Data selection:** where data relevant to the analysis task are retrieved from the database.
- ✓ **Data transformation**: where data are transformed or consolidated into forms appropriate for mining by performing summary or aggregation operations, for instance.
- ✓ **Data mining**: An essential process where intelligent methods are applied in order to extract data patterns.
- ✓ **Pattern evaluation**: To identify the truly interesting patterns representing knowledge based on some interestingness measures.



- ✓ **Knowledge presentation**: where visualization and knowledge representation techniques are used to present the mined know ledge to the user

.

Fig. 2 Data mining as a step in the process of knowledge discovery

- KDD refers to the overall process of discovering useful knowledge from data. It involves the evaluation and possibly interpretation of the patterns to make the decision of what qualifies as knowledge. It also includes the choice of encoding schemes, preprocessing, sampling, and projections of the data prior to the data mining step.

- Data mining refers to the application of algorithms for extracting patterns from data without the additional steps of the KDD process.
- Objective of Pre-processing on data is to remove noise from data or to remove redundant data.
- There are mainly 4 types of Pre-processing Activities included in KDD Process that is shown in fig. as Data cleaning, Data integration, Data transformation, Data reduction.

b. What the major tasks in data preprocessing.

- Today's real-world databases are highly susceptible to noisy, missing, and inconsistent data due to their typically huge size (often several gigabytes or more) and their likely origin from multiple, heterogeneous sources.
- Low-quality data will lead to low-quality mining results. How can the data be preprocessed in order to help improve the quality of the data and, consequently, of the mining results? How can the data be preprocessed so as to improve the efficiency and ease of the mining process?
- Data have quality if they satisfy the requirements of the intended use. There are many factors comprising data quality, including accuracy, completeness, consistency, timeliness, believability, and interpretability.
- **Example**
    - Imagine that you are a manager at **AllElectronics** and have been charged with analyzing the company's data with respect to your branch's sales.
    - You immediately set out to perform this task. You carefully inspect the company's database and data warehouse, identifying and selecting the attributes or dimensions (e.g., item, price, and units sold) to be included in your analysis.
    - Alas! You notice that several of the attributes for various tuples have no recorded value. For your analysis, you would like to include information as to whether each item purchased was advertised as on sale, yet you discover that this information has not been recorded.
    - Furthermore, users of your database system have reported errors, unusual values, and inconsistencies in the data recorded for some transactions.
    - In other words, the data you wish to analyze by data mining techniques are incomplete (lacking attribute values or certain attributes of interest, or containing only aggregate data); inaccurate or noisy (containing errors, or values that deviate from the expected); and inconsistent (e.g., containing discrepancies in the department codes used to categorize

items).

- Above example illustrates three of the elements defining data quality: **accuracy, completeness,** and ***consistency***.
- Inaccurate, incomplete, and inconsistent data are commonplace properties of large real-world databases and datawarehouses.
- There are many possible reasons for inaccurate data (i.e., having incorrect attribute values). The data collection instruments used may be faulty.
- There may have been human or computer errors occurring at data entry. Users may purposely submit incorrect data values for mandatory fields when they do not wish to submit personal information (e.g., by choosing the default value "January 1" displayed for birthday). This is known as disguised missing data. Errors in data transmission can also occur.
- There may be technology limitations such as limited buffer size for coordinating synchronized data transfer and consumption. Incorrect data may also result from inconsistencies in naming conventions or data codes, or inconsistent formats for input fields (e.g., date).
- Incomplete data can occur for a number of reasons. Attributes of interest may not always be available, such as customer information for sales transaction data.
- Other data may not be included simply because they were not considered important at the time of entry. Relevant data may not be recorded due to a misunderstanding or because of equipment malfunctions. Data that were inconsistent with other recorded data may have been deleted.
- Furthermore, the recording of the data history or modifications may have been overlooked. Missing data, particularly for tuples with missing values for some attributes, may need to be inferred.
- **Data Preprocessing Methods/Techniques:**
  - **Data Cleaning** routines work to "clean" the data by filling in missing values, smoothing noisy data, identifying or removing outliers, and resolving inconsistencies.
  - **Data Integration** which combines data from multiple sources into a coherent data store, as in data warehousing.
  - **Data Transformation**, the data are transformed or consolidated into forms appropriate for mining
  - **Data Reduction** obtains a reduced representation of the data set that is much smaller in volume, yet produces the same (or almost the same) analytical results.

4. a. What is missing data? How to handle the missing data?

- Ignore the tuple (record/row):

Usually done when class label is missing.

Fill missing value manually:

Use the attribute mean (average) to fill in the missing value and also use the attribute mean (average) for all samples belonging to the same class.

- Fill in the missing value automatically:

Predict the missing value by using a learning algorithm:

Consider the attribute with the missing value as a dependent variable and run a learning algorithm (usually regression,Naive Bayes or Decision tree) to predict the missing value.

- Use a global constant to fill in the missing value

Replace all missing attribute values by the same constant such as a label like “Unknown” or NAN

b. Explain the role of task relevant data in data operation.

- A data mining task can be specified in the form of a **data mining query**, which is input to the data mining system.
- A data mining **query** is defined in terms of data mining task primitives.
- These primitives **allow the user to inter-actively communicate** with the **data mining system** during discovery of knowledge.
- The data mining task primitives includes the following:
    - Task-relevant data
    - Kind of knowledge to be mined
    - Background knowledge
    - Interestingness measurement

Presentation for visualizing the discovered patterns.

- **Task-relevant data**
    - This specifies the **portions of the database or the dataset** of data in which the **user is interested**.
    - This includes the **database attributes** or data warehouse dimensions of interest (referred to as the relevant attributes or dimensions).
- **The kind of knowledge to be mined**
    - This specifies the data mining functions to be performed.

  - Such as **characterization, discrimination, association or correlation analysis, classification, prediction, clustering, outlier analysis**, or evolution analysis.

- **The background knowledge to be used in the discovery process**
  - The **knowledge** about the **domain** is useful for **guiding the knowledge discovery process** for evaluating the interesting patterns.
  - **Concept hierarchies** are a **popular form of background knowledge**, which allow data to be mined at multiple levels of abstraction.
  - An example of a concept hierarchy for the attribute (or dimension) age is shown in **user beliefs** regarding relationships in the data are another form of background knowledge.

5.a. Discuss basic principles of attribute oriented induction.

An attribute selection measure is a heuristic for selecting the splitting criterion that "best" separates a given data partition, *D*, of class-labeled training tuples into individual classes.

Information gain:

- ID3 uses information gain as its attribute selection measure.
- This measure is based on pioneering work by Claude Shannon on information theory, which studied the value or "information content" of messages.
- Let node *N* represents or hold the tuples of partition *D*. The attribute with the highest information gain is chosen as the splitting attribute for node *N*.
- This attribute minimizes the information needed to classify the tuples in the resulting partitions and reflects the least randomness or "impurity" in these partitions.
- Such an approach minimizes the expected number of tests needed to classify a given tuple and guarantees that a simple (but not necessarily the simplest) tree is found.
- The expected information needed to classify a tuple in *D* is given by

$$Info(D) = -\sum_{i=1}^{m} p_i \log_2(p_i)$$

- Where *pi* is the probability that an arbitrary tuple in *D* belongs to class*Ci* and is estimated by **|***Ci*,*D***|/|***D***|**.
- A log function to the base 2 is used, because the information is encoded in bits.
- *Info(D)* is just the average amount of information needed to identify the class label of a tuple in *D*.
- Note that, at this point, the information we have is based solely on the proportions of tuples of each class. *Info(D)* is also known as the entropy of *D*. Now, suppose we were to partition the tuples in *D* on some attribute *A* having *v* distinct values, {*a*1, *a*2, , *av*}, as observed from the training data.
- If *A* is discrete-valued, these values correspond directly to the *v* outcomes of a test on *A*. Attribute *A* can be used to split*D*into *v* partitions or subsets, {*D*1, *D*2,. , *Dv*}, where*Dj* contains those tuples in *D* that
  have outcome *aj* of *A*.
- These partitions would correspond to the branches grown from node *N*.
- Ideally, we would like this partitioning to produce an exact classification of the tuples. That is, we would like for each partition to be pure. However, it is quite likely that the partitions will be impure (e.g., where a partition may contain a collection of tuples from different classes rather than from a single class).

How much more information would we still in order to arrive at an exact classification?

- This amount is measured by

$$Info_A(D) = \sum_{j=1}^{v} \frac{|D_j|}{|D|} \times Info(D_j).$$

- The term **|***Dj***|** / **|***D***|** acts as the weight of the *j*th partition. *InfoA*(*D*) is the expected information required to classify a tuple from *D* based on the partitioning by *A*.
- The smaller the expected information (still) required, the greater the purity of the partitions.
- Information gain is defined as the difference between the original information requirement (i.e., based on just the proportion of classes) and the newrequirement (i.e., obtained after partitioning on *A*). That is,

Gain(*A*) = *Info*(*D*) - *InfoA*(*D*)

- In other words, *Gain(A)* tells us how much would be gained by branching on *A*. It is the expected reduction in the information requirement caused by knowing the value of *A*.
- The attribute *A* with the highest information gain, (*Gain*(*A*)), is chosen as the splitting attribute at node *N*.

- This is equivalent to saying that we want to partition on the attribute *A* that would do the "best classification," so that the amount of information still required to finish classifying the tuples is minimal (i.e., minimum *InfoA*(*D*)).

Gain ratio:

- The information gain measure is biased toward tests with many outcomes.
- That is, it prefers to select attributes having a large number of values. For example, consider an attribute that acts as a unique identifier, such as *product ID*.
- A split on *product ID* would result in a large number of partitions (as many as there are values), each one containing just onetuple.

- Because each partition is pure, the information required to classify data set *D* based on this partitioning would be *Infoproduct_ID*(*D*) = 0.
- Therefore, the information gained by partitioning on this attribute is maximal. Clearly, such a partitioning is useless forclassification.
- C4.5, a successor of ID3, uses an extension to information gain known as *gain ratio*, which attempts to overcome this bias.
- It applies a kind of normalization to information gain using a **"split information"** value defined analogously with *Info*(*D*) as

$$SplitInfo_A(D) = -\sum_{j=1}^{v} \frac{|D_j|}{|D|} \times \log_2\left(\frac{|D_j|}{|D|}\right)$$

- This value represents the potential information generated by splitting the training data set, *D*, into *v* partitions, corresponding to the *v* outcomes of a test on attribute *A*.
- Note that, for each outcome, it considers the number of tuples having that outcome with respect to the

total number of tuples in *D*.

- It differs from information gain, which measures the information with respect to classification that is acquired based on the same partitioning.
- The **gain ratio** is defined as

$$GainRatio(A) = \frac{Gain(A)}{SplitInfo(A)}$$

- The attribute with the maximum gain ratio is selected as the splitting attribute. Note, however, that as the split information approaches 0, the ratio becomes unstable.
- A constraint is added to avoid this, whereby the information gain of the test selected must be large at least as great as the average gain over all tests examined.

<u>Gini Index:</u>

- The Gini index is used in CART. Using the notation described above, the Gini index measures the impurity of *D*, a data partition or set of training tuples, as

$$Gini(D) = 1 - \sum_{i=1}^{m} p_i^2,$$

- where *pi* is the probability that a tuple in *D* belongs to class *Ci* and is estimated by $|C_{i,D}|/|D|$. The sum is computed over *m* classes.s
- The Gini index considers a binary split for each attribute. Let's first consider the case where *A* is a discrete-valued attribute having *v* distinct values, {*a*1, *a*2, ... , *av*}, occurring in *D*.
- To determine the best binary split on *A*, we examine all of the possible subsets that can be formed using known values of*A*.
- Each subset, *SA*, can be considered as a binary test for attribute *A* of the form "*A* є *SA*?".
- Given a tuple, this test is satisfied if the value of *A* for the tuple is among the values listed in *SA*.
- If *A* has *v* possible values, then there are 2^v possible subsets.
- For example, if *income* has three possible values, namely {*low, medium, high*} then the possible subsets are {*low, medium, high*}, {*low, medium*}, {*low, High*},{*medium, high*}, {*low*}, {*medium*}, {*high*}, and {}.
- We exclude the power set, {*low, medium, high*}, and the empty set from consideration since, conceptually, they do not represent a split. Therefore, there are 2^v-2 possible ways to form two partitions of the data, *D*, based on a binary split on *A*.
- When considering a binary split, we compute a weighted sum of the impurity of each resulting partition. For example, if a binary split on *A* partitions *D* into *D*1 and *D*2, the gini index of *D* given that partitioning is

$$Gini_A(D) = \frac{|D_1|}{|D|}Gini(D_1) + \frac{|D_2|}{|D|}Gini(D_2)$$

- For each attribute, each of the possible binary splits is considered.
- For a discrete-valued attribute, the subset that gives the minimum gini index for that attribute is selected as its splitting subset.
- For continuous-valued attributes, each possible split-point must be considered. The strategy is similar to that described above for information gain, where the midpoint between each pair of (sorted) adjacent values is taken as a possible split-point.
- The point giving the minimum Gini index for a given (continuous-valued) attribute is taken as the split-

point of that attribute.

- Recall that for a possible split-point of A, D_1 is the set of tuples in D satisfying $A \leq split\_point$, and D_2 is the set of tuples in D satisfying $A > split\_point$.
- The reduction in impurity that would be incurred by a binary split on a discrete- or continuous-valued attribute A is

$$\Delta Gini(A) = Gini(D) - Gini_A(D)$$

- The attribute that maximizes the reduction in impurity (or, equivalently, has the minimum Gini index) is selected as the splitting attribute.
- This attribute and either its splitting subset (for a discrete-valued splitting attribute) or split-point (for a continuous valued splitting attribute) together form the splitting criterion.

b. Write a note on concept description in detail.

The simplest kind of descriptive data mining is called concept description. A concept usually refers to a collection of data such as frequent_buyers, graduate_students and so on.

As data mining task concept description is not a simple enumeration of the data. Instead, concept description generates descriptions for characterization and comparison of the data.

It is sometimes called class description when the concept to be described refers to a class of objects

- **Characterization**: It provides a concise and succinct summarization of the given collection of data.
- **Comparison**: It provides descriptions comparing two or more collections of data.

6.a Explain the Apriori Algorithm.

□ **Purpose**: The Apriori Algorithm is an influential algorithm for mining frequent itemsets for boolean association rules.

□ ***Key Concepts:***

- **Frequent Itemsets**: The sets of item which has minimum support (denoted by L_i for ith-Itemset).
- **Apriori Property**: Any subset of frequent itemset must be frequent.
- **Join Operation**: To find L_k, a set of candidate k-itemsets is generated by joining L_{k-1} itself.

  ○ Find the frequent itemsets: the sets of items that have minimum support – A subset of a frequent itemset must also be a frequent itemset **(Apriori Property)**

  ○ i.e., if {AB} is a frequent itemset, both {A} and {B} should be a frequent itemset – Iteratively find frequent itemsets with cardinality from 1 to k (k-itemset)

  ○ Use the frequent itemsets to generate association rules.

□ ***The Apriori Algorithm : Pseudo code***

  ○ **Join Step**: C k is generated by joining Lk-1with itself

- ○ **Prune Step**: Any (k-1)-itemset that is not frequent cannot be a subset of a frequent k-itemset

- Pseudo-code:

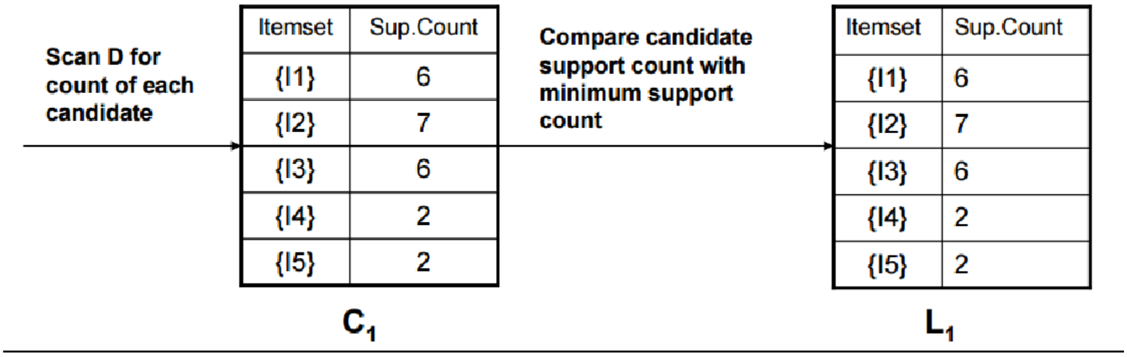
```
$C_k$: Candidate itemset of size
k $L_k$: frequent itemset of size
k $L_1$= {frequent items};
for (k = 1; $L_k$ != ∅; k++) do begin
        $C_{k+1}$ = candidates generated from $L_k$;
for each transaction t in database do
        Increment the count of all candidates in
        $C_{k+1}$ That are contained in t
        $L_{k+1}$ = candidates in $C_{k+1}$ with min_support
        end
        return $\cup_k L_k$;
```

TID	List of Items
T100	I1, I2, I5
T100	I2, I4
T100	I2, I3
T100	I1, I2, I4
T100	I1, I3
T100	I2, I3
T100	I1, I3
T100	I1, I2 ,I3, I5
T100	I1, I2, I3

Example

- Consider a database, **D**, consisting of 9 transactions.
- Suppose min. support count required is **2** (i.e. min_sup = 2/9 = 22 %)
- Let minimum confidence required is **70%**.
- We have to first find out the frequent itemset using Apriori algorithm.
- Then, Association rules will be generated using min. support & min. confidence.

Step 1: Generating 1-itemset Frequent Pattern


Scan D for count of each candidate
Compare candidate support count with minimum support count

Itemset	Sup.Count
{I1}	6
{I2}	7
{I3}	6
{I4}	2
{I5}	2

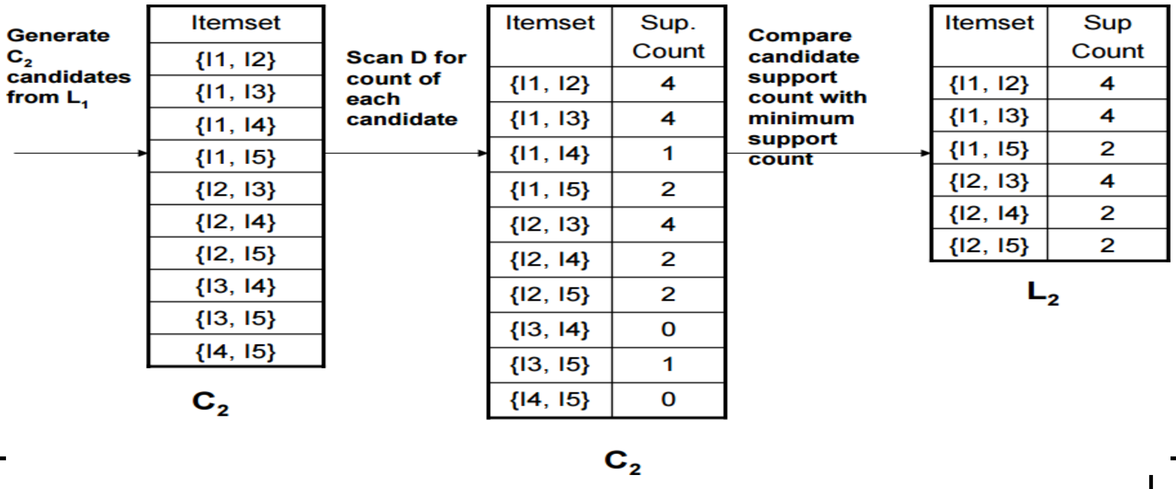
C_1

Itemset	Sup.Count
{I1}	6
{I2}	7
{I3}	6
{I4}	2
{I5}	2

L_1

- **The set of frequent 1-itemsets, L1**, consists of the candidate 1- itemsets satisfying minimum support.
- In the first iteration of the algorithm, each item is a member of the set of candidate.

Step 2: Generating 2-itemset Frequent Pattern


Generate C_2 candidates from L_1
Scan D for count of each candidate
Compare candidate support count with minimum support count

Itemset
{I1, I2}
{I1, I3}
{I1, I4}
{I1, I5}
{I2, I3}
{I2, I4}
{I2, I5}
{I3, I4}
{I3, I5}
{I4, I5}

C_2

Itemset	Sup. Count
{I1, I2}	4
{I1, I3}	4
{I1, I4}	1
{I1, I5}	2
{I2, I3}	4
{I2, I4}	2
{I2, I5}	2
{I3, I4}	0
{I3, I5}	1
{I4, I5}	0

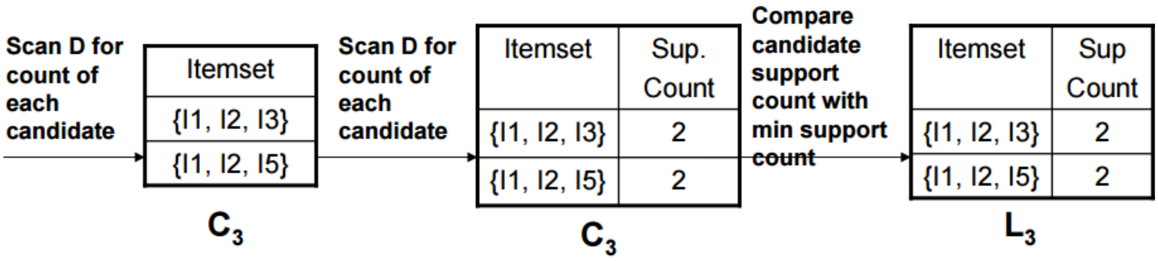
C_2

Itemset	Sup Count
{I1, I2}	4
{I1, I3}	4
{I1, I5}	2
{I2, I3}	4
{I2, I4}	2
{I2, I5}	2

L_2

- To discover the set of frequent 2-itemsets, L2, the algorithm uses L1 Join L1 to generate a candidate set of 2-itemsets, C2.
- Next, the transactions in D are scanned and the support count for each candidate itemset in C2 is accumulated (as shown in the middle table).
- The set of frequent 2-itemsets, L2, is then determined, consisting of those candidate 2-itemsets in C2 having minimum support.
- **Note**: We haven't used Apriori Property yet.

- ***Step 3: Generating 3-itemset Frequent Pattern***


Scan D for count of each candidate
Scan D for count of each candidate
Compare candidate support count with min support count
C_3
C_3
L_3

Itemset
{I1, I2, I3}
{I1, I2, I5}

Itemset	Sup. Count
{I1, I2, I3}	2
{I1, I2, I5}	2

Itemset	Sup Count
{I1, I2, I3}	2
{I1, I2, I5}	2

- The generation of the set of candidate 3-itemsets, C3 , involves use of the Apriori Property.
- In order to find C3, we compute L2 Join L2.
- C3 = L2 join L2 = {{I1, I2, I3}, {I1, I2, I5}, {I1, I3, I5}, {I2, I3, I4}, {I2, I3, I5}, {I2, I4, I5}}.
- Now, Join step is complete and Prune step will be used to reduce the size of C3. Prune step helps to avoid heavy computation due to large Ck.
- Based on the **Apriori property** that all subsets of a frequent itemset must also be frequent, we can determine that four latter candidates cannot possibly be frequent. How ?
- For example, lets take **{I1, I2, I3}**. The 2-item subsets of it are {I1, I2}, {I1, I3} & {I2, I3}. Since all 2-item subsets of {I1, I2, I3} are members of L2, We will keep {I1, I2, I3} in C3.
- Lets take another example of **{I2, I3, I5}** which shows how the pruning is performed. The 2-item subsets are {I2, I3}, {I2, I5} & {I3,I5}.
- But, {I3, I5} is not a member of L2 and hence it is not frequent **violating Apriori Property**. Thus We will have to remove {I2, I3, I5} from C3.
- Therefore, C3 = {{I1, I2, I3}, {I1, I2, I5}} after checking for all members of result of Join operation for Pruning.
- Now, the transactions in D are scanned in order to determine **L3, consisting of those candidates 3- itemsets in C3 having minimum support.**

Step 4: Generating 4-itemset Frequent Pattern

- The algorithm uses L3 Join L3 to generate a candidate set of 4-itemsets, **C4**. Although the join results in {{I1, I2, I3, I5}}, this itemset is pruned since its subset {{I2, I3, I5}} is not frequent.
- Thus, **C4 = φ**, and algorithm terminates, **having found all of the frequent items. This completes our Apriori Algorithm.** What’s Next?
- These frequent itemsets will be used to generate **strong association rules** (where strong association rules satisfy both minimum support & minimum confidence).

Step 5: Generating Association Rules from Frequent Itemsets

Procedure:

- For each frequent itemset “l”, generate all nonempty subsets of l.
- For every nonempty subset s of l, output the rule “s -> (l-s)” if support_count(l) /

  support_count(s) >= min_conf where min_conf is minimum confidence threshold.

Back to Example:

- We had L = {{I1}, {I2}, {I3}, {I4}, {I5}, {I1, I2}, {I1, I3}, {I1, I5}, {I2, I3}, {I2, I4}, {I2, I5}, {I1, I2, I3}, {I1, I2, I5}}.
- Let’s take l = {I1, I2, I5}. – It’s all nonempty subsets are {I1, I2}, {I1, I5}, {I2, I5}, {I1}, {I2}, {I5}.
- Let **minimum confidence threshold** is, say 70%.
- The resulting association rules are shown below, each listed with its confidence.
- R1: I1 ^ I2 -> I5 Confidence = sc{I1, I2, I5}/sc{I1,I2} = 2/4 = 50% (R1 is Rejected)
- R2: I1 ^ I5 -> I2 Confidence = sc{I1, I2, I5}/sc{I1,I5} = 2/2 = 100% (**R2 is Selected**)
- R3: I2 ^ I5 -> I1 Confidence = sc{I1, I2, I5}/sc{I2,I5} = 2/2 = 100% (**R3 is Selected**)
- R4: I1 -> I2 ^ I5 Confidence = sc{I1, I2, I5}/sc{I1} = 2/6 = 33% (R4 is Rejected)
- R5: I2 -> I1 ^ I5 Confidence = sc{I1, I2, I5}/{I2} = 2/7 = 29% (R5 is Rejected)
- R6: I5 -> I1 ^ I2 Confidence = sc{I1, I2, I5}/ {I5} = 2/2 = 100% (**R6 is Selected**)
- In this way, we have found **three strong association rules**.

b. Explain the data generalization and summarization based on characterization.

Data Generalization & Summarization

Data and objects in databases contain detailed information at the primitive concept level.
For example, the item relation in a sales database may contain attributes describing low-level item information such as item_ID, name, brand, category, supplier, place_made and price.

It is useful to be able to summarize a large set of data and present it at a high conceptual level.

For example, summarizing a large set of items relating to Christmas season sales provides a general description of such data, which can be very helpful for sales and marketing managers.

This requires an important functionality called data generalization.

Data Generalization

A process that abstracts a large set of task-relevant data in a database from a low conceptual level to higher ones.

Data Generalization is a summarization of general features of objects in a target class and produces what is called characteristic rules.

The data relevant to a user-specified class are normally retrieved by a database query and run through a summarization module to extract the essence of the data at different levels of abstractions.

For example, one may want to characterize the "OurVideoStore" customers who regularly rent more than 30 movies a year. With concept hierarchies on the attributes describing the target class, the **attribute-oriented induction** method can be used, for example, to carry out data summarization.

Note that with a data cube containing a summarization of data, simple OLAP operations fit the purpose of data characterization.

Approaches:

- Data cube approach(OLAP approach).
- Attribute-oriented induction approach.

Presentation Of Generalized Results

Generalized Relation:

- Relations where some or all attributes are generalized, with counts or other aggregation values accumulated.

Cross-Tabulation:

- Mapping results into cross-tabulation form (similar to contingency tables).

Visualization Techniques:

- Pie charts, bar charts, curves, cubes, and other visual forms.

Quantitative characteristic rules:

- Mapping generalized results in characteristic rules with quantitative information associated with it.

7.a. Discuss the issues regarding classification and prediction.

Issues Regarding Classification and Prediction

This section describes issues regarding preprocessing the data of classification and prediction. Criteria for the comparison and evaluation of classification methods are also described.

1 Preparing the Data for Classification and Prediction

The following preprocessing steps may be applied to the data in order to help improve the accuracy, efficiency, and scalability of the classification or prediction process.

Data Cleaning: This refers to the preprocessing of data in order to remove or reduce noise (by applying smoothing techniques) and the treatment of missing values (e.g., by replacing a missing value with the most commonly occurring value for that attribute, or with the most probable value based on statistics.) Although most classification algorithms have some mechanisms for handling noisy or missing data, this step can help reduce confusion during learning.

Relevance Analysis: Many of the attributes in the data may be irrelevant to the classification or prediction task. For example, data recording the day of the week on which a bank loan application was filed is unlikely to be relevant to the success of the application. Furthermore, other attributes may be redundant. Hence, relevance analysis may be performed on the data with the aim of removing any irrelevant or redundant attributes from the learning process. In machine learning, this step is known as feature selection. Including such attributes may otherwise slow down, and possibly mislead, the learning step.

Ideally, the time spent on relevance analysis, when added to the time spent on learning from the resulting “reduced” feature subset should be less than the time that would have been spent on learning from the original set of features. Hence, such analysis can help improve classification efficiency and scalability.

Data Transformation: The data can be generalized to higher – level concepts. Concept hierarchies may be used for this purpose. This is particularly useful for continuous – valued attributes. For example, numeric values for the attribute income may be generalized to discrete ranges such as low, medium, and high. Similarly, nominal – valued attributes like street, can be generalized to higher – level concepts, like city. Since generalization compresses the original training data, fewer input / output operations may be involved during learning.

The data may also be normalized, particularly when neural networks or methods involving distance measurements are used in the learning step. **Normalization** involves scaling all values for a given attribute so that they fall within a small specified range, such as – 1.0 to 1.0, or 0.0 to 1.0. In methods that use distance measurements, for example, this would prevent attributes with initially large ranges (like, say, income) from outweighing attributes with initially smaller ranges (such as binary attributes).

2 Comparing Classification Methods

Classification and prediction methods can be compared and evaluated according to the following criteria:

Predictive Accuracy: This refers to the ability of the model to correctly predict the class label of new or previously unseen data.

Speed: This refers to the computation costs involved in generating and using the model.

Robustness: This is the ability of the model to make correct predictions given noisy data or data with missing values.

Scalability: This refers to the ability to construct the model efficiently given large amount of data.

Interpretability: This refers to the level of understanding and insight that is provided by the model.

4 Classification by Decision Tree Induction

“What is a decision tree”? A **decision tree** is a flow – chart – like tree structure, where each internal node denotes a test on an attribute, each branch represents an outcome of the test, and leaf nodes represent classes or class distributions. The top – most node in a tree is the root node.

A typical decision tree is shown in Figure 8.2. It represents the concept buys_ computer, that is, it predicts whether or not a customer at All Electronics is likely to purchase a computer. Internal nodes are denoted by rectangles, and leaf nodes are denoted by ovals.

age?

< = 30

31 … 40

> 40

student

yes

credit _ rating?

no

yes

excellent

fair

no

yes

no

yes

In order to classify an unknown sample, the attribute values of the sample are tested against the decision tree. A path is traced from the root to a leaf node that holds the class prediction for that sample. Decision trees can easily be converted to classification rules.

In Section 8.4.1, we describe a basic algorithm for learning decision trees. When decision trees are built, many of the branches may reflect noise or outliers in the training data. Tree pruning attempts to identify and remove such branches, with the goal of improving classification accuracy on unseen data. **Algorithm:** Generate _ decision _ tree. Generate a decision tree from the given training data.

Input: The training samples, samples, represented by discrete – valued attributes; the set of candidate attributes, attribute – list.

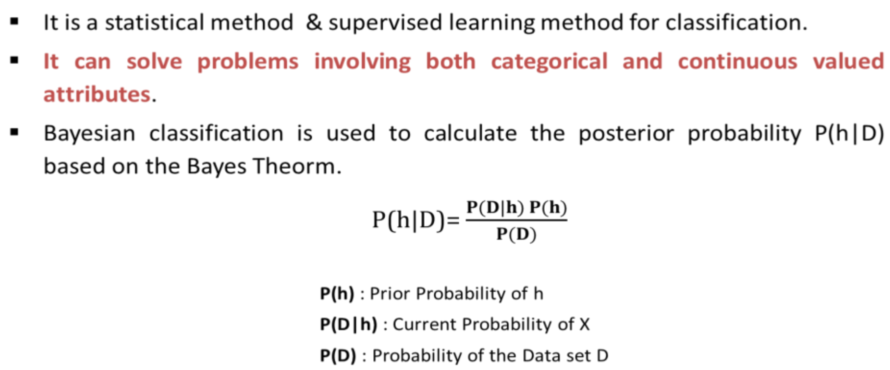
Output: A decision tree **Method:**

```
(1) create a node N;
(2) if samples are all of the same class, C then
(3) return N as a leaf node labeled with the class C;
(4) if attribute – list is empty then


(5) return N as a leaf node labeled with the most common class in samples ; // majority voting
(6) select test – attribute, the attribute among attribute – list with the highest information gain;
(7) label node N with test – attribute;
```

```
(8) for each known value ai of test – attribute // partition the sample
(9) grow a branch from node N for the condition test – attribute = ai;
(10) let si be the set of samples in samples for which test – attribute = ai; // a partition
(11) if si is empty then
(12) attach a leaf labeled with the most common class in samples;
```

b. Explain NAÏVE BAYES classifier.



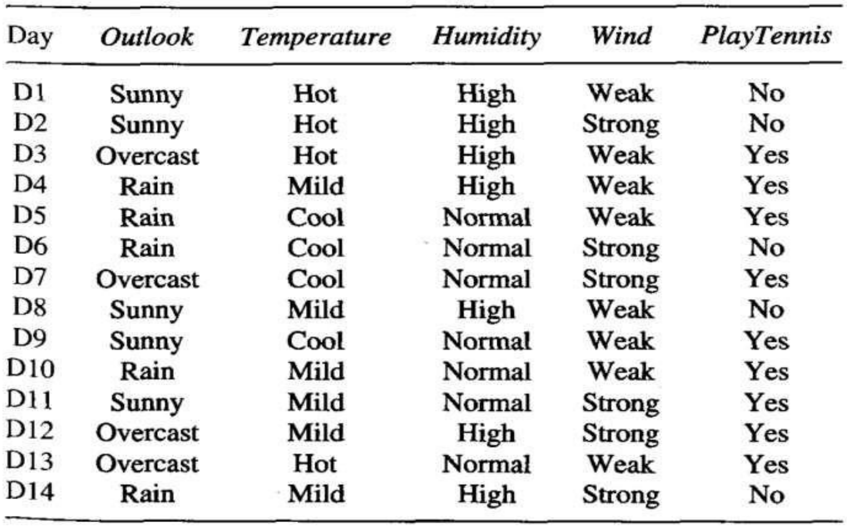
- It is a statistical method & supervised learning method for classification.
- **It can solve problems involving both categorical and continuous valued attributes**.
- Bayesian classification is used to calculate the posterior probability P(h|D) based on the Bayes Theorm.

$$P(h|D)=\frac{P(D|h)\,P(h)}{P(D)}$$

P(h) : Prior Probability of h

P(D|h) : Current Probability of X

P(D) : Probability of the Data set D



Day	*Outlook*	*Temperature*	*Humidity*	*Wind*	*PlayTennis*
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

- Here there are 14 training examples of the target concept PlayTennis, where each

day is described by the attributes Outlook, Temperature, Humidity, and Wind.

- Here we use the naive Bayes classifier and the training data from this table to classify the following novel instance:

$$\langle Outlook = sunny, Temperature = cool, Humidity = high, Wind = strong \rangle$$

$$v_{NB} = \underset{v_j \in \{yes, no\}}{\operatorname{argmax}} P(v_j) \prod_i P(a_i | v_j)$$

$$= \underset{v_j \in \{yes, no\}}{\operatorname{argmax}} P(v_j) \quad P(Outlook = sunny | v_j) P(Temperature = cool | v_j)$$

$$\cdot P(Humidity = high | v_j) P(Wind = strong | v_j$$

$$P(PlayTennis = yes) = 9/14 = .64$$

$$P(PlayTennis = no) = 5/14 = .36$$

$$v_{NB} = P(yes)\ P(sunny|yes)\ P(cool|yes)\ P(high|yes)\ P(strong|yes) = .0053$$

=0.64* 2/9 *3/9*3/9*3/9 = 0.0053

After Normalizing,

$$v_{NB} = P(no)\ P(sunny|no)\ P(cool|no)\ P(high|no)\ P(strong|no) \quad = .0206$$

=0.36* 3/5 *1/5*4/5*3/5 = 0.0206

After Normalizing,

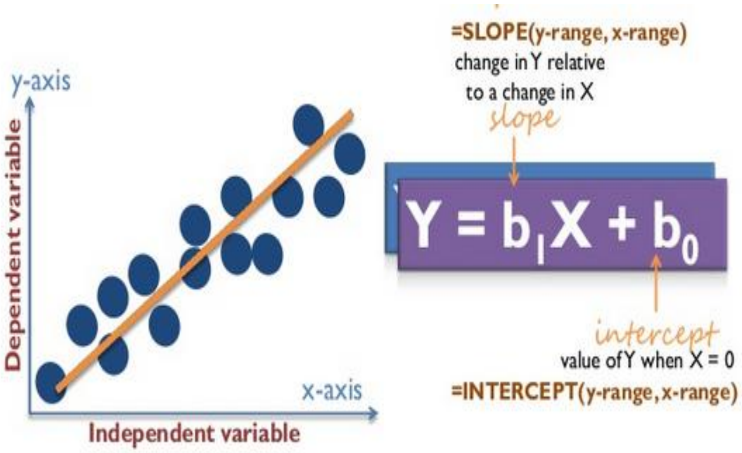
- Thus, the naive Bayes classifier assigns the target value ***PlayTennis = no*** *to this* new instance

8.a. Explain linear and non linear regression prediction methods.

Linear Regression:

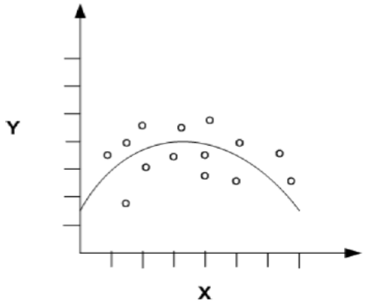
- The simplest form of regression to visualize is linear regression with a **single predictor**.
- A linear regression technique can be used **if the relationship between x and y can be approximated with a straight line**.
- Linear regression with a single predictor can be expressed with the following equation with one dependent and one independent variable is defined by the given formula
  - Where y = Dependent variable which we are trying to predict, b_1 = The Slope, and X = independent variable, b_0 = The Intercept, u = Random Error/Residual

$$\boxed{Y = b_1X + b_0 + u}$$



Non Linear Regression:

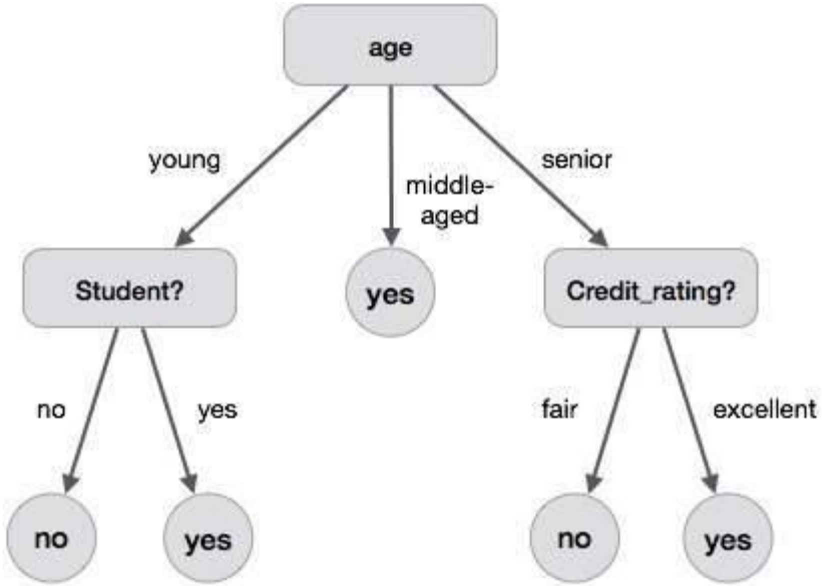
- Often the relationship between x and y cannot be approximated with a straight line or curve for that nonlinear regression technique may be used.
- Alternatively, the data could be preprocessed to make the relationship linear.



b. Explain decision tree induction classification method.

A decision tree is a structure that includes a root node, branches, and leaf nodes. Each internal node denotes a test on an attribute, each branch denotes the outcome of a test, and each leaf node holds a class label. The topmost node in the tree is the root node.

The following decision tree is for the concept buy_computer that indicates whether a customer at a company is likely to buy a computer or not. Each internal node represents a test on an attribute. Each leaf node represents a class.



The benefits of having a decision tree are as follows −

- It does not require any domain knowledge.
- It is easy to comprehend.
- The learning and classification steps of a decision tree are simple and fast.

Decision Tree Induction Algorithm

A machine researcher named J. Ross Quinlan in 1980 developed a decision tree algorithm known as ID3 (Iterative Dichotomiser). Later, he presented C4.5, which was the successor of ID3. ID3 and C4.5 adopt a greedy approach. In this algorithm, there is no backtracking; the trees are constructed in a top-down recursive divide-and-conquer manner.

Generating a decision tree form training tuples of data partition D

Algorithm : Generate_decision_tree

Input:
Data partition, D, which is a set of training tuples
and their associated class labels.
attribute_list, the set of candidate attributes.
Attribute selection method, a procedure to determine the
splitting criterion that best partitions that the data
tuples into individual classes. This criterion includes a
splitting_attribute and either a splitting point or splitting subset.

Output:
A Decision Tree

Method
create a node N;

```
if tuples in D are all of the same class, C then
  return N as leaf node labeled with class C;

if attribute_list is empty then
  return N as leaf node with labeled
  with majority class in D;|| majority voting

apply attribute_selection_method(D, attribute_list)
to find the best splitting_criterion;
label node N with splitting_criterion;

if splitting_attribute is discrete-valued and
  multiway splits allowed then  // no restricted to binary trees

attribute_list = splitting attribute; // remove splitting attribute
for each outcome j of splitting criterion

  // partition the tuples and grow subtrees for each partition
  let Dj be the set of data tuples in D satisfying outcome j; // a partition

  if Dj is empty then
    attach a leaf labeled with the majority
    class in D to node N;
  else
    attach the node returned by Generate
    decision tree(Dj, attribute list) to node N;
  end for
return N;
```

Tree Pruning

Tree pruning is performed in order to remove anomalies in the training data due to noise or outliers. The pruned trees are smaller and less complex.

Tree Pruning Approaches

There are two approaches to prune a tree −

- **Pre-pruning** − The tree is pruned by halting its construction early.
- **Post-pruning** - This approach removes a sub-tree from a fully grown tree.

9.a. Discuss the key stakeholders of analytics project.

Core Deliverables

The data products that result from developing a big data product are in most of the cases some of the following −

 Machine learning implementation − This could be a classification algorithm, a regression model or a segmentation model.

 Recommender system − The objective is to develop a system that recommends choices based on user behavior. Netflix is the characteristic example of this data product, where based on the ratings of users, other movies are recommended.

 Dashboard − Business normally needs tools to visualize aggregated data. A

dashboard is a graphical mechanism to make this data accessible.

 Ad-Hoc analysis − Normally business areas have questions, hypotheses or myths

that can be answered doing ad-hoc analysis with data.

b. Explain click stream mining in detail.

Clickstream Mining

1. The approach which is used by most of the people for surfing information on Websites is

difficult to analyze and understand.

2. Quantitative data can lack information about what a user actually intends to do, while

qualitative data tends to be localized and is impractical to gather for large samples.

3. Once a website is made public, the user is in ultimate control of their own navigation, often

employing a variety of different strategies for browsing.

4. These strategies also vary over time depending, not only on the user’s goals, but also on

factors such as expertise, familiarity with the site, time pressures and perceive cost of

information.

5. Given this continually shifting nature of browsing strategies, the question arises how can these strategies be identified in the use made of an existing Website.

6. One solution is to use the clickstream logs, which contain the address of each page visited, then date and time of the visit and the referring page and are potentially rich source of data on Internet user activity.

7. Clickstream logs can be generate either by software hosted by the client application or directly from the server logs.

8. Collection and Restoration of Clickstream Data:

 A common tool for collecting data on the pages visited by Website users is the use of

server-side clickstream data.

 This identifies the pages delivered by a server in response to a client’s request. However, these clickstream data logs are often large and unwieldy and present an incomplete picture of activity.

- For example, server-side logs do not record activities that involve browser caching, network caching, or the navigation of pages that are internal to the site but are held on another server.
- Despite these server-side limitations, there are some aspects of user behavior, such as use of the back button or the opening of new/additional windows within the same Website, that can be captured by such techniques such as the Pattern Restore Method(PRM) algorithm.

9. Visualization and Categorization of Clickstream Data:

- Once the clickstream data have been processed, a technique for analyzing and categorizing these data into usage patterns is required.
- The visualization techniques facilitate this by producing ‘Footstep’ graphs.
- These are based on the use of a 2-D x-y plot, where x-axis represents the browsing time between two Web pages and the y-axis the Web page in the users browsing route.
- Thus, the distance travelled on the x-axis represents the time the user has spent browsing and a change in the y-axis represents a transition from one Web page to another

10. a.Explain the data analytics life cycle.

Life Cycle Phases of Data Analytics

Data Analytics Lifecycle :

The Data analytic lifecycle is designed for Big Data problems and data science projects. The cycle is iterative to represent real project. To address the distinct requirements for performing analysis on Big Data, step – by – step methodology is needed to organize the activities and tasks involved with acquiring, processing, analyzing, and repurposing data.

Phase 1: Discovery –

- The data science team learn and investigate the problem.
- Develop context and understanding.
- Come to know about data sources needed and available for the project.
- The team formulates initial hypothesis that can be later tested with data.

Phase 2: Data Preparation –

- Steps to explore, preprocess, and condition data prior to modeling and analysis.
- It requires the presence of an analytic sandbox, the team execute, load, and transform, to get data into the sandbox.
- Data preparation tasks are likely to be performed multiple times and not in predefined order.
- Several tools commonly used for this phase are – Hadoop, Alpine Miner, Open Refine, etc.

Phase 3: Model Planning –

- Team explores data to learn about relationships between variables and subsequently, selects key variables and the most suitable models.
- In this phase, data science team develop data sets for training, testing, and production purposes.
- Team builds and executes models based on the work done in the model planning phase.
- Several tools commonly used for this phase are – Matlab, STASTICA.

Phase 4: Model Building –

- Team develops datasets for testing, training, and production purposes.
- Team also considers whether its existing tools will suffice for running the models or if they need more robust environment for executing models.
- Free or open-source tools – Rand PL/R, Octave, WEKA.
- Commercial tools – Matlab , STASTICA.

Phase 5: Communication Results –

- After executing model team need to compare outcomes of modeling to criteria established for success and failure.
- Team considers how best to articulate findings and outcomes to various team members and stakeholders, taking into account warning, assumptions.
- Team should identify key findings, quantify business value, and develop narrative to summarize and convey findings to stakeholders.

Phase 6: Operationalize –

- The team communicates benefits of project more broadly and sets up pilot project to deploy work in controlled way before broadening the work to full enterprise of users.
- This approach enables team to learn about performance and related constraints of the model in production environment on small scale , and make adjustments before full deployment.
- The team delivers final reports, briefings, codes.
- Free or open source tools – Octave, WEKA, SQL, MADlib.

b.Discuss the business application for data mining.

i) Fraud detection

ii)Market segmentation

Fraud Detection

Fraud Detection for Telecommunications Industry

1. The telecommunications industry has expanded dramatically in the last few years with the development of affordable mobile phone technology.

2. With the increasing number of mobile phone users, global mobile phone fraud is also set to rise.

3. There are many different types of telecom fraud and these can occur at various levels.

4. The two most prevalent types are subscription fraud and superimposed or surfing.

5. Subscription fraud occurs when the fraudster obtains a subscription to a service, often with false identity details, with no intention of paying. This is thus at the level of a phone number – all transactions from this number will be fraudulent.

6. Superimposed fraud is the use of a service without having the necessary authority and is usually detected by the appearance of phantom calls on a bill.

7. There are several ways to carry out superimpose fraud, including mobile phone cloning and obtaining calling card authorization details.

8. Superimposed fraud will generally occur at the level of individual calls – the fraudulent calls will be mixed in with the legitimate ones.

9. Subscription fraud will generally be detected at some point through the billing process – although the aim is to detect it well before that, since large costs can quickly be run up.

10. Superimpose fraud can remain undetected for a long time.

11. Telecommunications networks generate vast quantities of data, sometimes on the order of several GBs per day, so that data mining techniques are of particular importance.

12. At a low level, simple rule-based detection systems use rules such as the apparent use of the same phone in two very distant geographical locations in quick succession, calls which appear to overlap in time, and very high value and long calls.

13. At a higher level, statistical summaries of call distributions are compare with thresholds determined either by experts or by application of supervised learning methods to known fraud/non-fraud cases.

Market Segmentation

1. Market Segmentation is a process that segments a market into smaller sub-markets, called segments.

2. Segments are to be homogeneous or have similar attributes.

3. Purchasing patterns and trends can appear prominently in certain segments.

4. Good market segmentation is to create segments where prominent patters can emerge.

5. Market segmentation may be use to analyze the followings:

Market responsiveness analysis: Useful in direct marketing since market responsiveness of product offerings can be readily available.

Market trend Analysis: Analyzing segment-by-segment changes of sales revenues can reveal market trends. Trending information is vital in preparing for ever-changing

markets.

It may use one of the following attributes to generate market segments:

- Geographical Regions: Regions, countries, states, zip-codes, countries , etc.
- Demographics: gender , age, income, education etc
- Psychographics: Life style classification
- Sales channels, branches and departments
- Sales representatives
- Product and service types (or product categories)
- Products
- Offer types

6. Segmentation provides opportunities for trend analysis. Trends and patterns embedded in

changes of sales revenues can be useful indicators for market shifts. Trend analysis may analyse the following types of segment trend information:

- What are the projected sales revenues for the next three months?
- Which segments are having the highest growth and which segments are having the

highest revenue decline?

- Which segments are having the highest growth rates in percentage terms?

7. Sales Trend Analysis:

Timely identification of newly emerging trends is very important to businesses.

Sales patterns of customer segments indicate market trends. Upward and downwards

trends in sales signify new market trends. Time-series predictive modeling can be used to identify trends embedded in changes of sales revenues. Understanding of sales trends is important for marketing as well as for customer retention. Typical sales trend analysis includes:

- Which customer segments are having highest growth and highest revenue decline ?
- Which customer segments are having highest growth rates in percentage terms?

8. Trends may be categorized as:

- Short term trends capture rapidly emerging trends
- Mid-term trends capture trends developing in between
- Long term trends capture trends developing over long periods.