**III/IV B.Tech (Supplementary) DEGREE EXAMINATION**

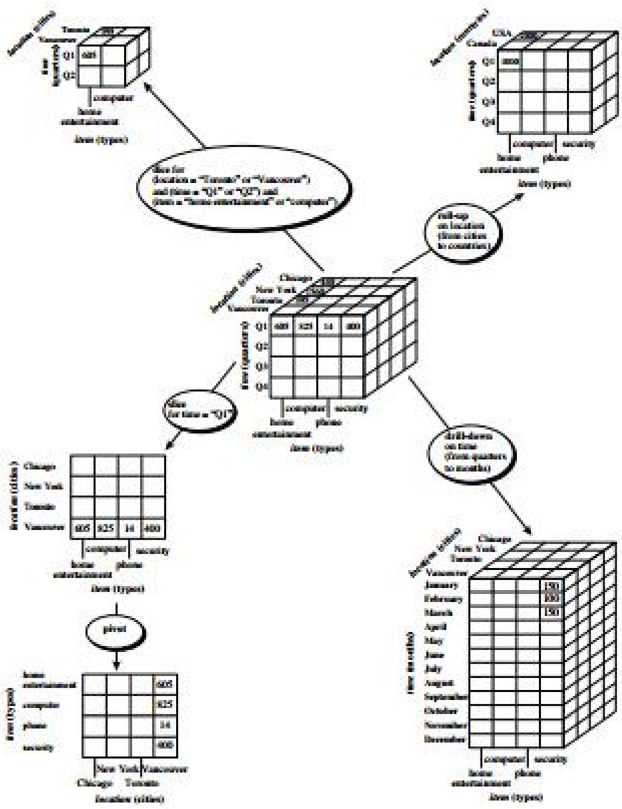
# SCHEME OF EVALUATION

**December, 2024** **Information Technology**

**Sixth Semester** **Data Warehousing & Data Mining**

**Time:** Three Hours **Maximum :** 70 Marks

| Q.No. | QUESTION & ANSWER | MARKS |
|---|---|---|
| **1 A) ANS)** | What is Online Analytical Processing?<br>On-line analytical processing (OLAP) serve users or knowledge workers in the role of data analysis and decision making. Such systems can organize and present data in various formats in order to accommodate the diverse needs of the different users. These systems are known as on-line analytical processing (OLAP) systems. | **1 M** |
| **1 B) ANS)** | What are the ideas of subject-oriented and non-volatile features of data warehouse?<br>Idea of subject-oriented is a data warehouse is organized around major subjects of an organization rather than daily operations.<br>Idea of non-volatile is Once data is entered into the data warehouse, it is not changed or deleted. | **1 M** |
| **1 C) ANS)** | List the data mining functionalities.<br>1)Concept/Class Description: Characterization and Discrimination<br>2) Mining Frequent Patterns, Associations, and Correlations<br>3) Classification and Prediction<br>4) Cluster Analysis<br>5) Outlier Analysis<br>6) Evolution Analysis | **1 M** |
| **1 D) ANS)** | What is data mining?<br>Data mining refers to extracting or "mining" knowledge from large amounts of data. | **1 M** |
| **1 E) ANS)** | Compute the mid-range of following data<br>18, 5, 20, 22, 67, 82, 44, 25, 43, 88, 76, 99, 56, 73, 81, 34, 90, 62, 14, 33<br>The midrange can also be used to assess the central tendency of a data set. It is the average of the largest and smallest values in the set.<br>5+99/2=104/2=52 | **1 M** |
| **1 F) ANS)** | State Bayes Theorem.<br>Bayes' theorem is useful in that it provides a way of calculating the posterior probability, P(H/X), from P(H), P(X/H), and P(X).<br>Bayes' theorem is<br>P(H/X) =P(X/H)*P(H)/P(X) | **1 M** |
| **1 G) ANS)** | List any two examples of discrete data<br>Two examples of discrete data:<br>Number of students in a classroom – You can count them individually (e.g., 25 students).<br>Number of cars in a parking lot – It must be a whole number (e.g., 10 cars).<br>Discrete data consists of countable values, often whole numbers. | **1 M** |
| **1 H) ANS)** | List various methods used for data reduction.<br>Data cube aggregation<br>Attribute subset selection<br>Dimensionality reduction<br>Numerosity reduction<br>Discretization and concept hierarchy generation | **1 M** |
| | | |

| | | |
|---|---|---|
| **1 I)<br>ANS)** | What is a frequent item set?<br>If the relative support of an itemset I satisfies a prespecified minimum support threshold (i.e., the absolute support of I satisfies the corresponding minimum support count threshold), then I is a frequent itemset. | **1 M** |
| **1 J)<br>ANS)** | Recall Rule based system.<br>A Rule-Based System is a system that uses a set of “if-then” rules for classification, prediction, or decision-making based on the data. | **1 M** |
| **1 K)<br>ANS)** | Define the term correlation.<br>Correlation is a statistical measure that describes the strength and direction of a relationship between two variables. | **1 M** |
| **1 L)<br>ANS)** | Define cluster analysis.<br>Cluster analysis is a data analysis technique that explores the naturally occurring groups within a dataset known as cluster. | **1 M** |
| **1 M)<br>ANS)** | What is outlier.<br>The data objects that do not comply with the general behaviour or model of the data is called outlier. | **1 M** |
| **1 N)<br>ANS)** | What is a mediod?<br>Each representative object is actually the medoid, or most centrally located object, of its cluster. This is the basis of the k-medoids method for grouping n objects into k clusters. | **1 M** |
| **2 A<br>ANS)** | What are the various OLAP operations are used in the multidimensional data model? Explain them in detail with an example.<br>Typical OLAP operations on multidimensional data:<br> location (cities)<br>Toronto<br>Vancouver<br>time (quarters)<br>Q1<br>Q2<br>605<br>computer<br>home entertainment<br>item (types)<br>dice for (location = “Toronto” or “Vancouver”) and (time = “Q1” or “Q2”) and (item = “home entertainment” or “computer”)<br>location (countries)<br>USA<br>Canada<br>Q1<br>Q2<br>Q3<br>Q4<br>computer<br>security<br>home entertainment<br>phone<br>item (types)<br>roll-up on location (from cities to countries)<br>location (cities)<br>Chicago<br>New York<br>Toronto<br>Vancouver<br>time (quarters)<br>Q1<br>Q2<br>Q3<br>Q4<br>605<br>825<br>14<br>400<br>computer<br>security<br>home entertainment<br>phone<br>item (types)<br>slice for time = “Q1”<br>drill-down on time (from quarters to months)<br>location (cities)<br>Chicago<br>New York<br>Toronto<br>Vancouver<br>605<br>825<br>14<br>400<br>computer<br>security<br>home entertainment<br>phone<br>item (types)<br>pivot<br>item (types)<br>home entertainment<br>computer<br>phone<br>security<br>605<br>825<br>14<br>400<br>New York<br>Vancouver<br>Chicago<br>Toronto<br>location (cities)<br>location (cities)<br>Chicago<br>New York<br>Toronto<br>Vancouver<br>time (months)<br>January<br>February<br>March<br>April<br>May<br>June<br>July<br>August<br>September<br>October<br>November<br>December<br>150<br>100<br>150<br>computer<br>security<br>home entertainment<br>phone<br>item (types)<br>Roll-up: The roll-up operation (also called the drill-up operation by some vendors) performs aggregation on a data cube, either by climbing up a concept hierarchy for a dimension or by dimension reduction.<br>Drill-down: Drill-down is the reverse of roll-up. It navigates from less detailed data to more detailed data.<br>Slice: The slice operation performs a selection on one dimension of the given cube, resulting in a subcube.<br>Dice: The dice operation defines a subcube by performing a selection on two or more dimensions.<br>Pivot (rotate): Pivot (also called rotate) is a visualization operation that rotates | **5 M**<br>**2 M** |

|  |  |  |
|---|---|---|
|  | the data axes in view in order to provide an alternative presentation of the data.<br>Any relevant explanation of all OLAP operations |  |
| **2 B**<br>**ANS)** | What are the major issues in data mining? Explain.<br>**Mining methodology and user interaction issues**: These reflect the kinds of knowledge mined, the ability to mine knowledge at multiple granularities, the use of domain knowledge, ad hoc mining, and knowledge visualization.<br>1.Mining different kinds of knowledge in databases<br>2.Interactive mining of knowledge at multiple levels of abstraction<br>3.Incorporation of background knowledge<br>4.Data mining query languages and ad hoc data mining<br>5.Presentation and visualization of data mining results<br>6.Handling noisy or incomplete data<br>7.Pattern evaluation<br>**Performance issues:** These include efficiency, scalability, and parallelization of data mining algorithms.<br>1.Efficiency and scalability of data mining algorithms<br>2.Parallel, distributed, and incremental mining algorithms.<br>**Issues relating to the diversity of database types:**<br>1.Handling of relational and complex types of data<br>2.Mining information from heterogeneous databases and global information systems<br>Any relevant explanation of above issues. | **3M**<br>**4M** |
| **3 A**<br>**ANS)** | Explain the data warehouse implementation.<br>Data warehouses contain huge volumes of data. OLAP servers demand that decision support queries be answered in the order of seconds. Therefore, it is crucial for data warehouse systems to support highly efficient cube computation techniques, access methods, and query processing techniques.<br>Methods for the efficient implementation of data warehouse systems:<br>A) Efficient Computation of Data Cubes<br>1. The compute cube Operator and the Curse of Dimensionality<br>2. Partial Materialization: Selected Computation of Cuboids<br>B) Indexing OLAP Data<br>1. Bitmap indexing<br>2. join indexing<br>C) Efficient Processing of OLAP Queries<br>1. Determine which operations should be performed on the available cuboids<br>2. Determine to which materialized cuboid(s) the relevant operations should be applied<br>Any relevant explanation of implementation techniques. | **3 M**<br>**4 M** |
| **3 B**<br>**ANS)** | Explain the different data mining tasks.<br>**Data mining functionalities/Data mining tasks:**<br>Data mining functionalities are used to specify the kind of patterns to be found in data mining tasks. In general, data mining tasks can be classified into two categories: descriptive and predictive. Descriptive mining tasks characterize the general properties of the data in the database. Predictive mining tasks perform inference on the current data in order to make predictions.<br>1)Concept/Class Description: Characterization and Discrimination<br>2) Mining Frequent Patterns, Associations, and Correlations<br>3) Classification and Prediction<br>4) Cluster Analysis<br>5) Outlier Analysis<br>6) Evolution Analysis<br>Any relevant explanation of all functionalities/tasks. | **2 M**<br>**5 M** |

**4 A**

Normalize the following group of data by using the following techniques.

200, 300, 400, 600, 1000

i. min-max normalization technique

ii. z-score normalization

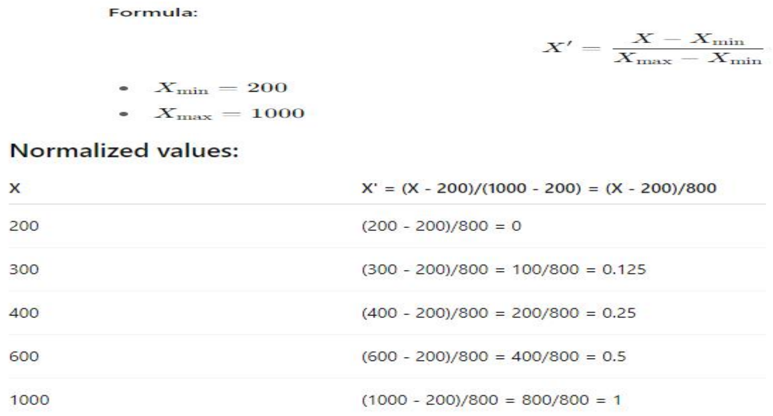
iii. Decimal scaling.

Write your observations on the above techniques.

**ANS)**

i. **Min-Max Normalization:**

**2 M**



Formula:

$$X' = \frac{X - X_{\min}}{X_{\max} - X_{\min}}$$

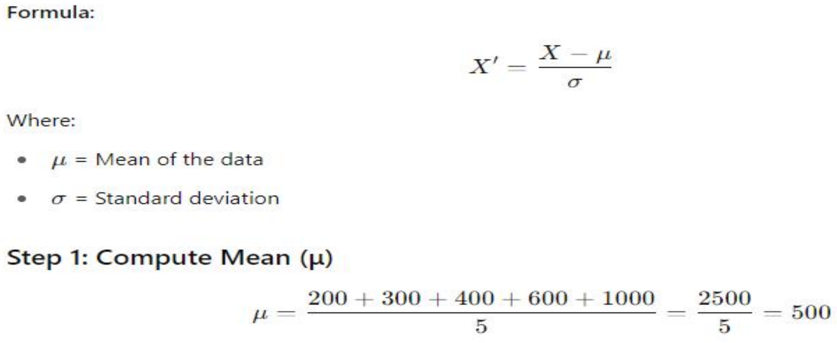
- $X_{\min} = 200$
- $X_{\max} = 1000$

**Normalized values:**

| X | X' = (X - 200)/(1000 - 200) = (X - 200)/800 |
|---|---|
| 200 | (200 - 200)/800 = 0 |
| 300 | (300 - 200)/800 = 100/800 = 0.125 |
| 400 | (400 - 200)/800 = 200/800 = 0.25 |
| 600 | (600 - 200)/800 = 400/800 = 0.5 |
| 1000 | (1000 - 200)/800 = 800/800 = 1 |

ii. **Z-Score Normalization (Standardization)**

**3 M**



Formula:

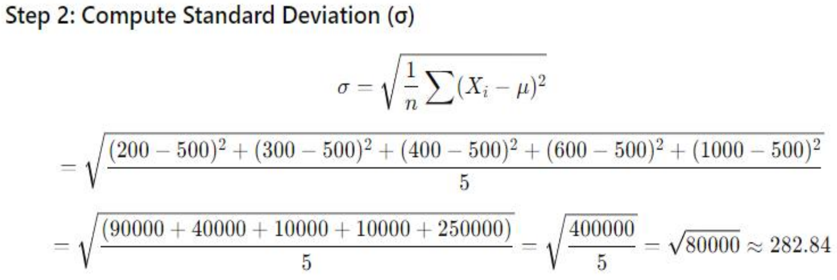
$$X' = \frac{X - \mu}{\sigma}$$

Where:

- $\mu$ = Mean of the data
- $\sigma$ = Standard deviation

**Step 1: Compute Mean (μ)**

$$\mu = \frac{200 + 300 + 400 + 600 + 1000}{5} = \frac{2500}{5} = 500$$

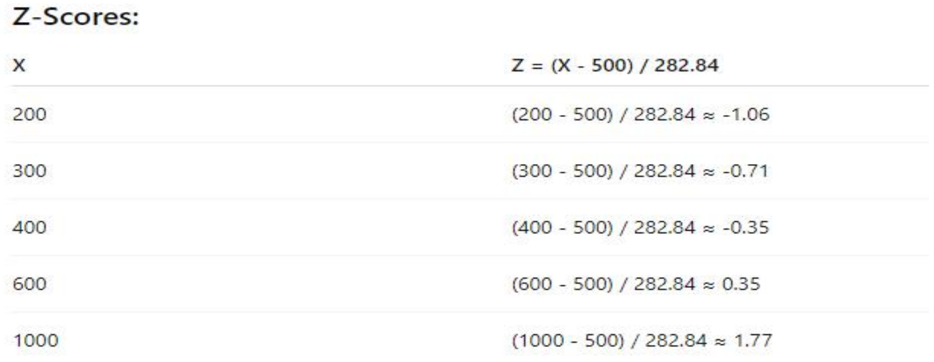


Step 2: Compute Standard Deviation (σ)

$$\sigma = \sqrt{\frac{1}{n}\sum(X_i - \mu)^2}$$

$$= \sqrt{\frac{(200-500)^2 + (300-500)^2 + (400-500)^2 + (600-500)^2 + (1000-500)^2}{5}}$$

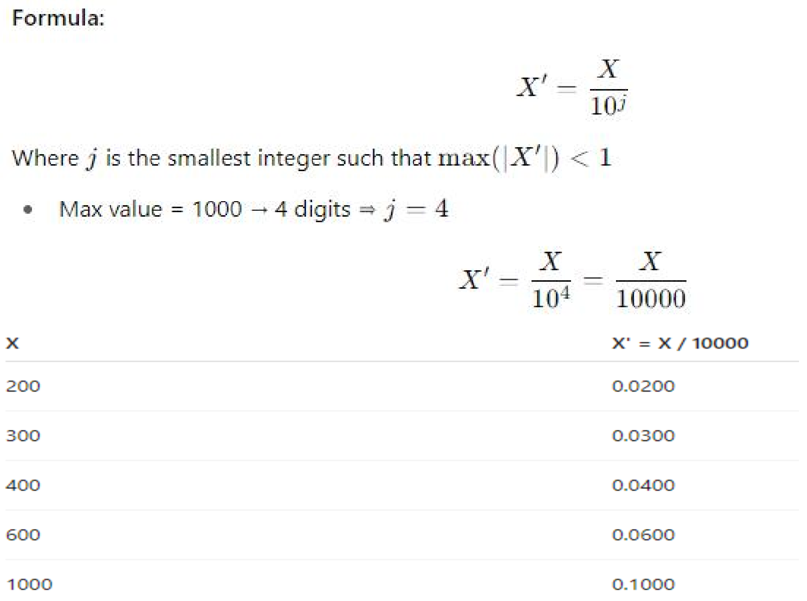
$$= \sqrt{\frac{(90000 + 40000 + 10000 + 10000 + 250000)}{5}} = \sqrt{\frac{400000}{5}} = \sqrt{80000} \approx 282.84$$



**Z-Scores:**

| X | Z = (X - 500) / 282.84 |
|---|---|
| 200 | (200 - 500) / 282.84 ≈ -1.06 |
| 300 | (300 - 500) / 282.84 ≈ -0.71 |
| 400 | (400 - 500) / 282.84 ≈ -0.35 |
| 600 | (600 - 500) / 282.84 ≈ 0.35 |
| 1000 | (1000 - 500) / 282.84 ≈ 1.77 |

iii. **Decimal Scaling:** — 2 M



Formula:

$$X' = \frac{X}{10^j}$$

Where $j$ is the smallest integer such that $\max(|X'|) < 1$

- Max value = 1000 → 4 digits ⇒ $j = 4$

$$X' = \frac{X}{10^4} = \frac{X}{10000}$$

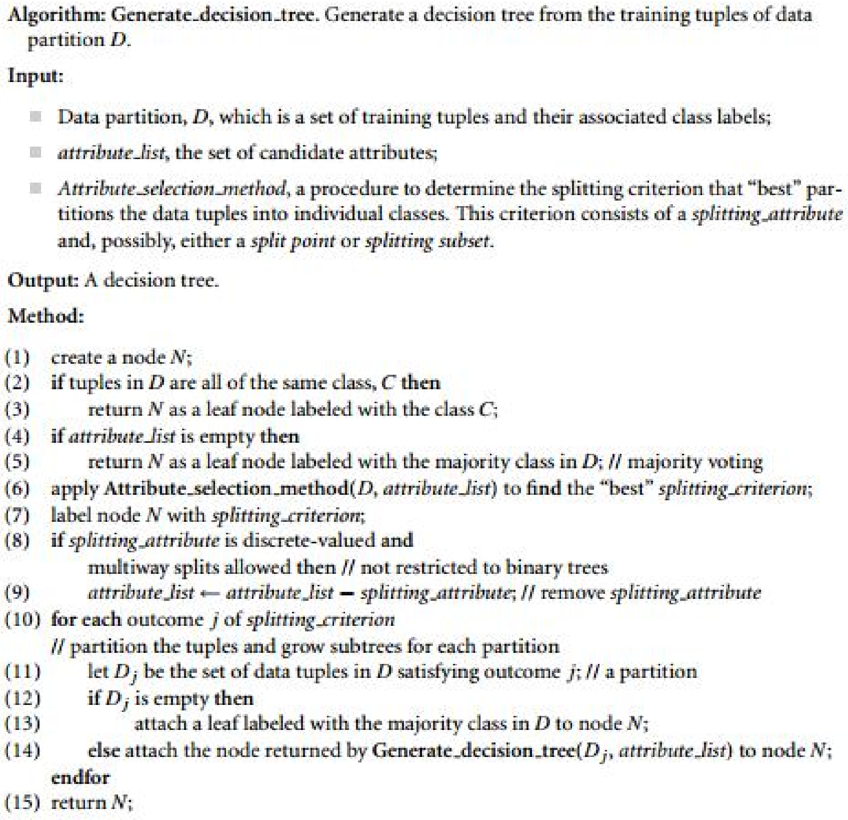
| X | X' = X / 10000 |
|---|---|
| 200 | 0.0200 |
| 300 | 0.0300 |
| 400 | 0.0400 |
| 600 | 0.0600 |
| 1000 | 0.1000 |

**4 B**

**ANS)**

Write and explain decision tree induction algorithm.

Basic algorithm for inducing a decision tree from training tuples. — 5 M



**Algorithm: Generate_decision_tree.** Generate a decision tree from the training tuples of data partition *D*.

**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 the data tuples into individual classes. This criterion consists of a *splitting_attribute* and, possibly, either a *split 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 a leaf node labeled with the class $C$;
if attribute_list is empty then
    return $N$ as a leaf node labeled with the 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 // not restricted to binary trees
    attribute_list ← attribute_list − splitting_attribute; // remove splitting_attribute
for each outcome $j$ of splitting_criterion
// partition the tuples and grow subtrees for each partition
    let $D_j$ be the set of data tuples in $D$ satisfying outcome $j$; // a partition
    if $D_j$ 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($D_j$, attribute_list) to node $N$;
endfor
return $N$;
```

Any relevant explanation of decision tree induction algorithm. — 2 M

**5 A**

**ANS)**

Describe Data Transformation & Data Discretization.

**Data Transformation:** — 4 M

In data transformation, the data are transformed or consolidated into forms appropriate for mining. Data transformation can involve the following:

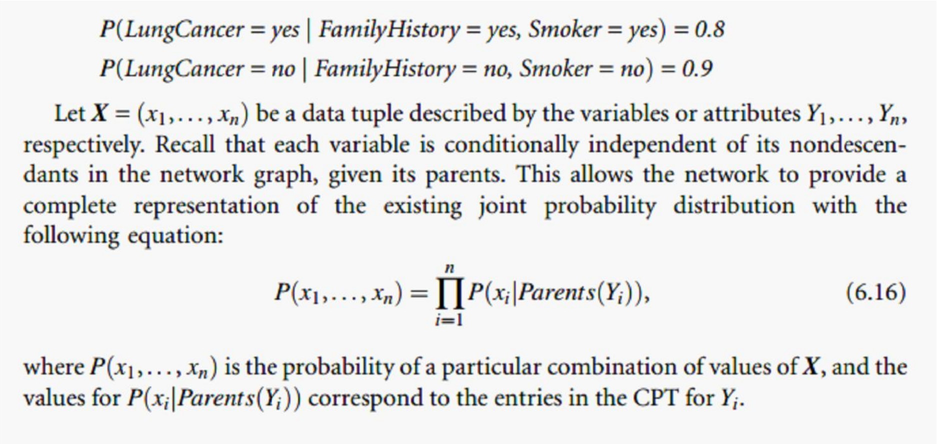
**Smoothing,** which works to remove noise from the data. Such techniques include binning, regression, and clustering.

**Aggregation,** where summary or aggregation operations are applied to the data. For example, the daily sales data may be aggregated so as to compute

| | | |
|---|---|---|
| | monthly and annual total amounts. This step is typically used in constructing a data cube for analysis of the data at multiple granularities.<br>**Generalization** of the data, where low-level or “primitive” (raw) data are replaced by higher-level concepts through the use of concept hierarchies. For example, categorical attributes, like street, can be generalized to higher-level concepts, like city or country.<br>Similarly, values for numerical attributes, like age, may be mapped to higher-level concepts, like youth, middle-aged, and senior.<br>**Normalization**, where the attribute data are scaled so as to fall within a small specified range, such as 1:0 to 1:0, or 0:0 to 1:0.<br>**Attribute construction** (or feature construction), where new attributes are constructed and added from the given set of attributes to help the mining process.<br>**Data Discretization:**<br>Data discretization techniques can be used to reduce the number of values for a given continuous attribute by dividing the range of the attribute into intervals. Interval labels can then be used to replace actual data values. Replacing numerous values of a continuous attribute by a small number of interval labels thereby reduces and simplifies the original data.This leads to aconcise,easy-to-use,knowledge-level representation of mining results.<br>Discretization techniques can be categorized based on how the discretization is performed, such as whether it uses class information or which direction it proceeds (i.e., top-down vs. bottom-up). If the discretization process uses class information, then we say it is supervised discretization. Otherwise, it is unsupervised. If the process starts by first finding one or a few points (called split points or cut points) to split the entire attribute range, and then repeats this recursively on the resulting intervals, it is called top-down discretization or splitting. This contrasts with bottom-up discretization or merging, which starts by considering all of the continuous values as potential split-points, removes some by merging neighborhood values to form intervals, and then recursively applies this process to the resulting intervals. Discretization can be performed recursively on an attribute to provide a hierarchical or multiresolution partitioning of the attribute values, known as a concept hierarchy. Concept hierarchies are useful for mining at multiple levels of abstraction. | **3 M** |
| **5 B**<br>**ANS)** | Explain about Bayesian belief networks.<br><br>Bayesian Belief Networks<br>The naïve Bayesian classifier makes the assumption of class conditional independence, that is, given the class label of a tuple, the values of the attributes are assumed to be conditionally independent of one another. This simplifies computation. When the assumption holds true, then the naïve Bayesian classifier is the most accurate in comparison with all other classifiers. In practice, however, dependencies can exist between variables. Bayesian belief networks specify joint conditional probability distributions. They allow class conditional independencies to be defined between subsets of variables. They provide a graphical model of causal relationships, on which learning can be performed. Trained Bayesian belief networks can be used for classification. Bayesian belief networks are also known as belief networks, Bayesian networks, and probabilistic networks. For brevity, we will refer to them as belief networks.<br>A belief network is defined by two components—a directed acyclic graph and a set of conditional probability tables. Each node in the directed acyclic graph represents a random variable. The variables may be discrete or continuous-valued. They may correspond to actual attributes given in the data or to “hidden variables” believed to form a relationship (e.g., in the case ofmedical data, a hidden variable may indicate a syndrome, representing a number of symptoms that, together, characterize a specific disease). Each arc represents a probabilistic dependence. If an arc is drawn from a node Y to a node Z, then Y | **2M**<br><br><br><br><br>**5 M** |

is a parent or immediate predecessor of Z, and Z is a descendant of Y . Each variable is conditionally independent of its nondescendants in the graph, given its parents. Below figure is a simple belief network, adapted from for six Boolean variables. The arcs in Figure(a) allow a representation of causal knowledge. For example,having lung cancer is influenced by a person's family history of lung cancer, as well as whether or not the person is a smoker. Note that the variable PositiveXRay is independent of whether the patient has a family history of lung cancer or is a smoker, given that we know the patient has lung cancer. In other words, once we know the outcome of the variable LungCancer, then the variables FamilyHistory and Smoker do not provide any additional information regarding PositiveXRay. The arcs also show that the variable LungCancer is conditionally independent of Emphysema, given its parents, FamilyHistory and Smoker.

A belief network has one conditional probability table (CPT) for each variable. The CPT for a variable Y specifies the conditional distribution P( Y jParents( Y )), where Parents( Y ) are the parents of Y . Figure (b) shows a CPT for the variable LungCancer. The conditional probability for each known value of LungCancer is given for each possible combination of values of its parents. For instance, from the upper leftmost and bottom rightmost entries, respectively, we see that



$$P(LungCancer = yes \mid FamilyHistory = yes, Smoker = yes) = 0.8$$

$$P(LungCancer = no \mid FamilyHistory = no, Smoker = no) = 0.9$$

Let $\boldsymbol{X} = (x_1, \ldots, x_n)$ be a data tuple described by the variables or attributes $Y_1, \ldots, Y_n$, respectively. Recall that each variable is conditionally independent of its nondescendants in the network graph, given its parents. This allows the network to provide a complete representation of the existing joint probability distribution with the following equation:

$$P(x_1, \ldots, x_n) = \prod_{i=1}^{n} P(x_i \mid Parents(Y_i)), \qquad (6.16)$$

where $P(x_1, \ldots, x_n)$ is the probability of a particular combination of values of $\boldsymbol{X}$, and the values for $P(x_i \mid Parents(Y_i))$ correspond to the entries in the CPT for $Y_i$.

A node within the network can be selected as an "output" node, representing a class label attribute. There may be more than one output node. Various algorithms for learning can be applied to the network. Rather than returning a single class label, the classification process can return a probability distribution that gives the probability of each class.

**6 A** Apply Apriori Algorithm in tracing all the frequent item datasets

| Transactions | Itemset |
|---|---|
| T100 | 1 2 3 |
| T200 | 2 3 5 |
| T300 | 1 2 3 5 |
| T400 | 2 5 |
| T500 | 1 3 5 |

#Hint: Consider appropriate minimal support and minimal configure values for generating the rules.

**ANS)** Consider **minimal support count = 2** (i.e., an itemset must appear in at least 2 transactions to be frequent).

**7 M**

**Step1: Generate 1-itemsets and their support counts**

| 1-itemset | Support Count |
|---|---|
| {1} | 3 |
| {2} | 4 |
| {3} | 4 |
| {5} | 4 |

**Frequent 1-itemsets(L1): All have support >=2**

**Step 2: Generate Candidate 2-itemsets(C2)**

| 2-Itemsets | Support Count |
|---|---|
| {1,2} | |
| {1,3} | 3 |
| {1,5} | 2 |
| {2,3} | 3 |
| {2,5} | 3 |
| {3,5} | 3 |

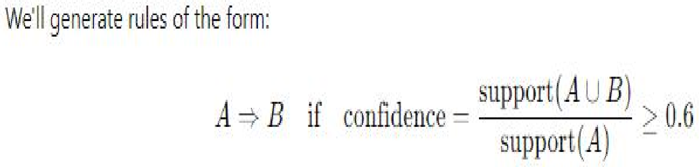
**Frequent 2-itemsets(L2):All 6 itemsets have support >=2**
**Step3: Generate Candidate 3-itemsets(C3)**
Joint frequent 2-itemsets to form 3-itemsets:

| 3-itemset | Support Count |
|---|---|
| {1,2,3} | 2 |
| {1,3,5} | 2 |
| {2,3,5} | 2 |

**Frequent 3-itemsets(L3): All 3 itemsets have support>=2**
**Step 4: Generate Candidate 4-itemsets(C4)**
From L3: only one possible 4 itemset
{1,2,3,5}---> Appears only in T300---> support=1
Not frequent.
**Final list of frequent itemsets:**
L1:{1},{2},{3},{5}
L2:{1,2},{1,3},{1,5},{2,3},{2,5},{3,5}
L3:{1,2,3},{1,3,5},{2,3,5}

**Association rules:**
Let's generate **association rules** from the frequent itemsets you discovered using the **Apriori algorithm**, based on a **minimum confidence of 60%** (0.6).

7 M



We'll generate rules of the form:

$$A \Rightarrow B \quad \text{if} \quad \text{confidence} = \frac{\text{support}(A \cup B)}{\text{support}(A)} \geq 0.6$$

**From 2-itemsets:**

**1. {1,2} (support = 2)**

- Rule: 1 → 2 = 2/3 = 0.667 ✓
- Rule: 2 → 1 = 2/4 = 0.5 ✗

**2. {1,3} (support = 3)**

- Rule: 1 → 3 = 3/3 = 1.0 ✓
- Rule: 3 → 1 = ¾ = 0.75 ✓

**3. {1,5} (support = 2)**

- Rule: 1 → 5 = 2/3 = 0.667 ✓
- Rule: 5 → 1 = 2/4 = 0.5 ❌

**4. {2,3} (support = 3)**

- Rule: 2 → 3 = ¾ = 0.75 ✓
- Rule: 3 → 2 = ¾ = 0.75 ✓

**5. {2,5} (support = 3)**

- Rule: 2 → 5 = ¾ = 0.75 ✓
- Rule: 5 → 2 = ¾ = 0.75 ✓

**6. {3,5} (support = 3)**

- Rule: 3 → 5 = ¾ = 0.75 ✓
- Rule: 5 → 3 = ¾ = 0.75 ✓

**From 3-itemsets:**

**1. {1,2,3} (support = 2)**

- Rule: 1,2 → 3 = 2/2 = 1.0 ✓
- Rule: 1,3 → 2 = 2/3 ≈ 0.667 ✓
- Rule: 2,3 → 1 = 2/3 ≈ 0.667 ✓
- Rule: 1 → 2,3 = 2/3 ≈ 0.667 ✓
- Rule: 2 → 1,3 = 2/4 = 0.5 ❌
- Rule: 3 → 1,2 = 2/4 = 0.5 ❌

**2. {1,3,5} (support = 2)**

- Rule: 1,3 → 5 = 2/3 ≈ 0.667 ✓
- Rule: 1,5 → 3 = 2/2 = 1.0 ✓
- Rule: 3,5 → 1 = 2/3 ≈ 0.667 ✓
- Rule: 1 → 3,5 = 2/3 ≈ 0.667 ✓
- Rule: 3 → 1,5 = 2/4 = 0.5 ❌
- Rule: 5 → 1,3 = 2/4 = 0.5 ❌

**3. {2,3,5} (support = 2)**

- Rule: 2,3 → 5 = 2/3 ≈ 0.667 ✓
- Rule: 2,5 → 3 = 2/3 ≈ 0.667 ✓
- Rule: 3,5 → 2 = 2/3 ≈ 0.667 ✓
- Rule: 2 → 3,5 = 2/4 = 0.5 ❌
- Rule: 3 → 2,5 = 2/4 = 0.5 ❌
- Rule: 5 → 2,3 = 2/4 = 0.5 ❌

**7 A**

Explain the multilevel association mining .

**Mining Multilevel Association Rules**

**ANS)**

For many applications, it is difficult to find strong associations among data items at low or primitive levels of abstraction due to the sparsity of data at those levels. Strong associations discovered at high levels of abstraction may represent commonsense knowledge. Moreover, what may represent common sense to one user may seem novel to another. Therefore, data mining systems should provide capabilities for mining association rules at multiple levels of abstraction, with sufficient flexibility for easy traversal among different abstraction spaces.

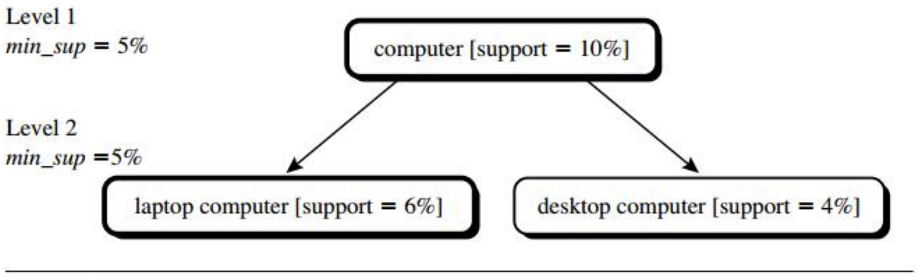
**2 M**

Association rules generated from mining data at multiple levels of abstraction are called multiple-level or multilevel association rules. Multilevel association rules can be mined efficiently using concept hierarchies under a support-confidence framework. In general, a top-down strategy is employed, where counts are accumulated for the calculation of frequent itemsets at each concept level, starting at the concept level 1 and working downward in the hierarchy toward the more specific concept levels, until no more frequent itemsets can be found. For each level, any algorithm for discovering frequent itemsets may be used, such as Apriori or its variations. A number of variations to this approach are described below, where each variation involves "playing" with the support threshold in a slightly different way. The variations are illustrated in Figures given below, where nodes indicate an item or itemset that has been examined, and nodes with thick borders indicate that an examined item or itemset is frequent.

**Using uniform minimum support for all levels (referred to as uniform support):**

**5 M**

The same minimum support threshold is used when mining at each level of abstraction. For example, in Figure given below, a minimum support threshold of 5% is used throughout (e.g., for mining from "computer" down to "laptop computer"). Both "computer" and "laptop computer" are found to be frequent, while "desktop computer" is not. When a uniform minimum support threshold is used, the search procedure is simplified. The method is also simle in that users are required to specify only one minimum support threshold. An Apriori-like optimization technique can be adopted, based on the knowledge that an ancestor is a superset of its descendants: The search avoids examining itemsets containing any item whose ancestors do not have minimum support.

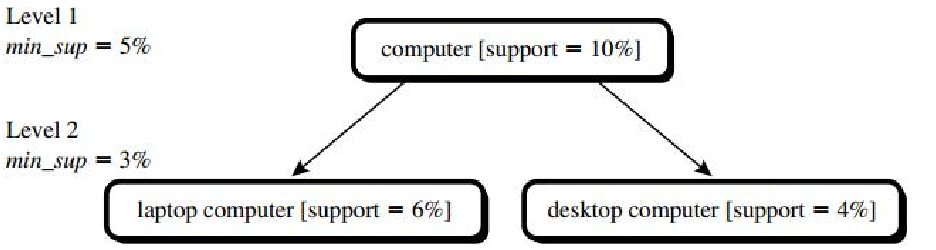

Level 1
min_sup = 5%
computer [support = 10%]
Level 2
min_sup =5%
laptop computer [support = 6%]
desktop computer [support = 4%]

Multilevel mining with uniform support.

The uniform support approach, however, has some difficulties. It is unlikely that items at lower levels of abstraction will occur as frequently as those at higher levels of abstraction. If the minimum support threshold is set too high, it could miss some meaningful associations occurring at low abstraction levels. If the threshold is set too low, it may generate many uninteresting associations occurring at high abstraction levels. This provides the motivation for the following approach.

**Using reduced minimum support at lower levels (referred to as reduced support):**

Each level of abstraction has its own minimum support threshold. The deeper the level of abstraction, the smaller the corresponding threshold is. For example, in the figure given below, the minimum support thresholds for levels 1 and 2 are 5% and 3%, respectively. In this way, "computer," "laptop computer," and "desktop computer" are all considered frequent.



Multilevel mining with reduced support.

**Using item or group-based minimum support (referred to as group-based support):**

Because users or experts often have insight as to which groups are more important than others, it is sometimes more desirable to set up user-specific, item, or groupbased minimal support thresholds when mining multilevel rules. For example, a user could set up the minimum support thresholds based on product price, or on items of interest, such as by setting particularly low support thresholds for laptop computers and flash drives in order to pay particular attention to the association patterns containing items in these categories.

A serious side effect of mining multilevel association rules is its generation of many redundant rules across multiple levels of abstraction due to the "ancestor" relationships among items.

**7 B**

**ANS)**

How do we generate Association Rules from Frequent Itemsets.

Generating Association Rules from Frequent Itemsets Once the frequent itemsets from transactions in a database D have been found,
it is straightforward to generate strong association rules from them (where strong association rules satisfy both minimum support and minimum confidence). This can be done using Equation given below for confidence, which we show again here for completeness:

confidence(A=>B)=P(B/A)=support_count(AUB)/support_count(A) **2M**

The conditional probability is expressed in terms of itemset support count, where support count (A U B) is the number of transactions containing the itemsets A U B, and support count (A) is the number of transactions containing the itemset A. Based on this equation, association rules can be generated as follows:

1) For each frequent itemset l , generate all nonempty subsets of l .
2) For every nonempty subset s of l , output the rule "s => (l - s)" **2M**
If support_count (l )/support count (s) >min conf, where min conf is the minimum confidence threshold.

Because the rules are generated from frequent itemsets, each one automatically satisfies minimum support. Frequent itemsets can be stored ahead of time in hash tables along with their counts so that they can be accessed quickly.

**EXAMPLE:**

Suppose the data contain the frequent itemset l = {I1, I2, I5}. What are the association rules that can be generated from l ? The nonempty subsets of l are {I1, I2}, {I1, I5}, {I2, I5}, {I1}, {I2}, and {I5}. The resulting association rules are as shown below, each listed with its confidence: **3 M**

I 1 ^ I 2 => I 5, confidence = 2/4 = 50%
I 1 ^ I 5 => I 2, confidence = 2/2 = 100%
I 2 ^ I 5 => I 1, confidence = 2/2 = 100%
I 1 => I 2 ^ I 5, confidence = 2/6 = 33%
I 2 => I 1 ^ I 5, confidence = 2/7 = 29%

I 5 => I 1 ^ I 2, confidence = 2/2 = 100%

If the minimum confidence threshold is, say, 70%, then only the second, third, and last rules above are output, because these are the only ones generated that are strong.

**8 A**

**ANS)**

Explain about the k-means clustering algorithm.

A partitioning algorithm organizes the objects into k partitions (k n), where each partition represents a cluster. The clusters are formed to optimize an objective partitioning criterion, such as a dissimilarity function based on distance, so that the objects within a cluster are "similar," whereas the objects of different clusters are "dissimilar" in terms of the data set attributes.

**2 M**

**The k-Means Method**

The k-means algorithm takes the input parameter, k, and partitions a set of n objects into k clusters so that the resulting intracluster similarity is high but the intercluster similarity is low. Cluster similarity is measured in regard to the mean value of the objects in a cluster, which can be viewed as the cluster's centroid or center of gravity.

**Algorithm: k-means.**

The k-means algorithm for partitioning, where each cluster's center is represented by the mean value of the objects in the cluster.

**5 M**

Input:

k: the number of clusters,

D: a data set containing n objects.

Output: A set of k clusters.

Method:

(1) arbitrarily choose k objects from D as the initial cluster centers;

(2) repeat

(3) (re)assign each object to the cluster to which the object is the most similar, based on the mean value of the objects in the cluster;

(4) update the cluster means, i.e., calculate the mean value of the objects for each cluster;

(5) until no change.

**8 B**

**ANS)**

What do you mean by grid-based clustering method? Explain in detail.

The grid-based clustering approach uses a multiresolution grid data structure. It quantizes the object space into a finite number of cells that form a grid structure on which all of the operations for clustering are performed. The main advantage of the approach is its fast processing time, which is typically independent of the number of data objects, yet dependent on only the number of cells in each dimension in the quantized space.

**2 M**

Some typical examples of the grid-based approach include STING, which explores statistical information stored in the grid cells; WaveCluster, which clusters objects using a wavelet transform method; and CLIQUE, which represents a grid-and density-based approach for clustering in high-dimensional data space

**STING: STatistical INformation Grid**

STING is a grid-based multiresolution clustering technique in which the spatial area is divided into rectangular cells. There are usually several levels of such rectangular cells corresponding to different levels of resolution, and these cells form a hierarchical structure: each cell at a high level is partitioned to form a number of cells at the next lower level. Statistical information regarding the attributes in each grid cell (such as the mean,maximum, and minimum values) is precomputed and stored. These statistical parameters are useful for query processing.

**3M**

STING offers several advantages: (1) the grid-based computation is query-

| | | |
|---|---|---|
| | independent, because the statistical information stored in each cell represents the summary information of the data in the grid cell, independent of the query; (2) the grid structure facilitates parallel processing and incremental updating; and (3) the method's efficiency is a major advantage: STING goes through the database once to compute the statistical parameters of the cells, and hence the time complexity of generating clusters is O(n).<br>**WaveCluster: Clustering Using Wavelet Transformation**<br>WaveCluster is a multiresolution clustering algorithm that first summarizes the data by imposing a multidimensional grid structure onto the data space. It then uses a wavelet transformation to transform the original feature space, finding dense regions in the transformed space.<br>In this approach, each grid cell summarizes the information of a group of points that map into the cell. This summary information typically fits into main memory for use by the multiresolution wavelet transform and the subsequent cluster analysis.<br>**Advantages:**<br>1) It provides unsupervised clustering<br>2) The multiresolution property of wavelet transformations can help detect clusters at varying levels of accuracy.<br>3) Wavelet-based clustering is very fast. | **2 M** |
| **9 A**<br>**ANS)** | What are the challenges of outlier detection? Explain in detail.<br>Any 7 challenges with relevant explanation<br>1. Definition of an Outlier is Context-Dependent<br>• Issue: What is considered an outlier in one context may be normal in another.<br>• Example: A credit card transaction of ₹1,00,000 may be an outlier for most users, but normal for a corporate account.<br>• Impact: A universal rule can't be applied across all domains or datasets.<br>2. High Dimensionality of Data<br>• Issue: In high-dimensional data (many features), distance measures become less meaningful (curse of dimensionality).<br>• Consequence: It becomes hard to distinguish between normal and outlier points.<br>• Example: In a dataset with hundreds of features, an outlier may not deviate in any single feature but in a combination of them.<br>3. Lack of Labeled Data<br>• Issue: Most outlier detection is unsupervised because labeled data (what is and isn't an outlier) is rare.<br>• Consequence: Evaluation of the method becomes difficult, and algorithms might miss or mislabel anomalies.<br>4. Imbalanced Data<br>• Issue: Outliers are rare by nature — they form a very small percentage of the dataset.<br>• Consequence: Algorithms may ignore outliers and overly focus on majority patterns (bias toward normal behavior). | **7 M** |

5. Noise vs. Outliers

- Issue: Distinguishing between genuine outliers and noise (random errors) is challenging.
- Example: A data entry error like “age = 999” is noise, but a rare genuine event (like a one-in-a-million medical condition) is an outlier.
- Consequence: Misclassification can lead to wrong decisions or data corruption.

6. Evolving Data (Concept Drift)

- Issue: In real-world systems, data distributions change over time.
- Example: In online behavior, what’s unusual today may become normal tomorrow.
- Consequence: Static models may become outdated and fail to detect new types of outliers.

7. Scalability

- Issue: Outlier detection algorithms may be computationally expensive, especially for large datasets.
- Consequence: Time and resource constraints make real-time or large-scale anomaly detection difficult.

8. Multimodal Distributions

- Issue: Data may come from multiple sources or subpopulations with different distributions.
- Consequence: A data point might look like an outlier in one group but be normal in another.

9. Interpretability

- Issue: Some models (e.g., neural networks, ensemble methods) may detect outliers but can't explain why.
- Consequence: Lack of trust and difficulty in taking corrective actions.

10. Dynamic Threshold Selection

- Issue: Choosing the correct threshold for outlier detection (e.g., how many standard deviations away) is not trivial.
- Consequence: Too strict → many false positives; too lenient → true outliers missed.

**9 B**

**ANS)**

Explain in detail about DBSCAN clustering method.

**3 M**

**DBSCAN (Density-Based Spatial Clustering of Applications with Noise)** is a density based clustering algorithm. The algorithm grows regions with sufficiently high density into clusters and discovers clusters of arbitrary shape in spatial databases with noise. It defines a cluster as a maximal set of density-connected points.

The basic ideas of density-based clustering involve a number of new definitions. We intuitively present these definitions, and then follow up with an example.

1)The neighborhood within a radius ε of a given object is called the **ε-**

|  |  |  |
|---|---|---|
|  | **neighborhood** of the object.<br>2)If the ε-neighborhood of an object contains at least a minimum number, MinPts, of objects, then the object is called a **core object**.<br>3)Given a set of objects, D, we say that an object p is **directly density-reachable** from object q if p is within the ε-neighborhood of q, and q is a core object.<br>4)An object p is **density-reachable** from object q with respect to ε and MinPts in a set of objects, D, if there is a chain of objects p1, ::: , pn, where p1 = q and p n = p such that p i+1 is directly density-reachable from pi with respect to ε and MinPts, for 1  i  n,pi 2 D.<br>5)An object p is **density-connected** to object q with respect to ε and MinPts in a set of objects, D, if there is an object o 2 D such that both p and q are density-reachable from o with respect to ε and MinPts.<br><br>**DBSCAN Pseudocode:**<br>**Step 1:**<br>DBSCAN searches for clusters by checking the ε-neighborhood of each point in the database.<br>**Step 2:**<br>If the ε-neighborhood of a point p contains more than MinPts, a new cluster with p as a core object is created.<br>**Step 3:**<br>DBSCAN then iteratively collects directly density-reachable objects from these core objects, which may involve the merge of a few density-reachable clusters.<br>**Step 4:**<br>The process terminates when no new point can be added to any cluster. | **4M** |

Scheme Prepared By                                                    Signature of the HOD, IT Dept.

Paper Evaluators:

| **S.NO** | **Name of the College** | **Name of the Faculty** | **Signature** |
|---|---|---|---|
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |