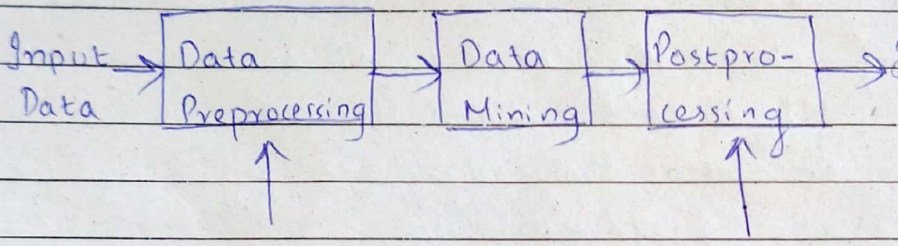
DATA MINING QUES BANK SOLUTION

* UNIT - 1

Q1) Explain the knowledge discovery process in data mining with neat diagram.

⇒ Data mining is an integral part of knowledge discovery in databases, which is the overall process of converting raw data into useful information. This process consists of a series of transformation steps, from data preprocessing to postprocessing of data mining results.



Feature Selection
Dimensionality Reduction
Normalization
Data Subsetting

Filtering Patterns
Visualization
Pattern Interpretation.

The input data can be stored in variety of formats & may reside in a centralized data repository or be distributed across multiple sites. The purpose of preprocessing is to transform the ~~data~~ raw input data into an appropriate format for subsequent analysis. The steps involved in data preprocessing include fusing data from multiple sources, cleaning data to remove noise & duplicate observations, & selecting records & features that are relevant to the data mining task at hand. Because of many ways data can be collected & stored, data preprocessing is perhaps the most laborious & time-consuming step in overall knowledge discovery process. Statistical measures or hypothesis testing methods can also be applied during postprocessing to eleminate spurious data mining results.

Q2) List & explain the challenges that motivate the development of Data Mining.

⇒ The following are some of the specific challenges that motivate the development of data mining:

1) Scalability:

Due to advances in data generation & collection, data sets with sizes of GB, TB or even PB are becomming common. If data mining algorithms are to handle these massive data sets, then they must be scalable. Scalability may also require the implementation of novel data structures to access individual records in an efficient manner. It can also be improved using sampling or developing parallel and distributed algorithm.

2) High Dimensionality:

It is now common to encounter data sets with hundreds of thousands of attributes instead of the handful common a few decades ago. Data sets with temporal or spatial components also tend to have high dimensionality. Consider a data set that contains measurements of temperature at various locations. If temp measurements are taken repeatedly for an extended period, the no. of dimension increases in proportion to the no. of measurements taken. Traditional data analysis techniques that were developed for low-dimensional data often don't work well for such high-dimensional data.

3) Heterogeneous & Complex Data:

Traditional data analysis methods often deal with data sets containing attributes of same type, either continuous or categorical. Data mining in business, science, medicine and other fields has grown, so has the need for techniques that can handle heterogeneous attributes. Techniques developed for mining such complex objects should take into consideration relationships in data.

4) Data ownership & Distribution:

Sometimes, the data needed for an analysis is not stored in one location or owned by one organisation. The key challenges faced by distributed data mining algorithms include: (i) how to reduce amount of communication needed to perform the distributed computation. (ii) how to effectively consolidate the data mining results obtained from multiple sources (iii) how to address data security issues.

5) Non-traditional Analysis:

The traditional statistical approach is based on a hypothesize-& - test paradigm. Hypothesis is proposed, & experiment is designed to gather the data & then the data is analysed wrt the hypothesis. The data sets analysed in data mining are typically not the result of a carefully designed experiment and often represent opportunistic samples of data, rather than random samples. Also, the data sets frequently involve non-traditional types of data & data distributions

Q3) List the properties of numbers that are used to describe the attributes & also different types of attributes.

⇒ A useful way to specify the type of an attribute is to identify the properties of numbers that correspond to underlying properties of the attribute. For eg., an attribute such as length has many of the properties of numbers. It makes sense to compare & order objects by length, as well as to talk about the differences & ratios of length. The following properties of no.s are typically used to describe attributes.

1) Distinctness = and ≠
2) Order <, ≤, >, and ≥
3) Addition + and -
4) Multiplication * and /

Given these properties, we can define 4 types of attributes: nominal, ordinal, interval, & ratio. Each attribute type possesses all of the properties & operations of the attribute types above it. Consequently, any property or operation that is valid for nominal, ordinal & interval attributes is also valid for ratio attributes.

	Attribute Type	Description	Examples	Operations
Categorical (Qualitative)	Nominal	The values of a nominal attribute are just diff names; i.e., nominal values provide only enough info to distinguish one obj from another (=, ≠)	zip codes, employee ID numbers, eye color, gender	mode, entropy, contingency correlation, x^2 test.
	Ordinal	The values of an ordinal attribute provide enough info to order objects (<, >)	hardness of minerals, {good, better, best}, grades, street numbers	median, percentiles, rank correlation, run tests, sign tests.
Numeric (Quantitative)	Interval	For interval attributes, the differences bet^n values are meaningful, i.e., a unit of measurement exits. (+, −)	calander dates, temp in Celsius, or Fahrenheit	mean, standard deviation, Person's correlation, t and F tests.
	Ratio	For ratio variables, both differences & ratios are meaningful (*, /)	temp in Kelvin, monetary quantities, counts, age, mass, length, electrical current	geometric mean, harmonic mean, percent, variation.

Nominal and ordinal attributes are collectively referred to as categorical or quantitative attributes. As the name suggests, qualitative attributes, such as employee ID, lack most of the properties of numbers. Even if they are represented by numbers, i.e., integers, they should be treated more like symbols. The remaining two types of attributes, interval & ratio, are collectively referred to as quantitative or numeric attributes. Quantitative attributes are represented by no.s & have most of the properties of numbers.

Q4) Define the following types of data with an appropriate example for each:

(i) Record Data:

Majority of Data Mining work assumes that data is a collection of records (data objects). The most basic form of record data has no explicit relationship among records or data fields, & every record has the same set of attributes. Record data is usually stored either in flat files or in relational databases. There are a few variations of Record Data, which have some characteristic properties:

a) Transaction or Market Basket Data: It is a special type of record data, in which each record contains a set of items. This type of data is called Market Basket Data. Transaction data is a collection of sets of items, but it can be viewed as a set of records whose fields are asymmetric attributes.

b) The Data Matrix: If the data objects in a collection of data all have the same fixed set of numeric attributes, then the data objects can be thought of as points (vectors) in a multidimensional space, where each dimension represents

a distinct attribute describing the object. The data matrix is the standard data format for most statistical data.

(c) The Sparse Data Matrix: (Document-data matrix) It is a special case of a datamatrix in which the attributes are of the same type and are asymmetric; i.e., only non-zero values are important.

Examples:

Tid	Refund	Marrial Status	Taxable Income	Defaulted Borrower
1	No	Single	70K	No
2	Yes	Married	120K	No
3	No	Divorced	95K	Yes
4	No	Married	60K	No
5	Yes	Divorced	220K	No

Record Data

Tid	Items
1	Bread, Milk
2	Beer, Bread
3	Soda, Milk
4	Bread, Milk
5	Diaper, Soda

Transaction Data

Projection of x load	Projection of y load	~~Dist load~~ Distance	Thickness	Load
10.23	5.27	15.22	1.2	27
12.65	6.25	16.22	1.1	22
13.54	7.23	17.34	1.2	23
14.27	8.43	18.45	0.9	25

Data Matrix

	team	coach	play	ball	score	game	win	lost	time-out	session
Document 1	3	0	5	0	2	6	0	2	0	2
Document 2	0	7	0	2	1	0	0	3	0	0
Document 3	0	1	0	0	1	2	2	0	3	0

Document-term matrix.

(ii) Document Data:

Its a subcategory of record data. Every term, every entry, every data attribute has a numeric value but in this case, we've got counts, we've got discrete values. So, what we have here is that each row, each data object is represented by what we think of as a term vector.
(Example is drawn in previous pg: Document-term matrix)

(iii) Sequence Data:

It consists of a data set that is a sequence of individual entities, such as a sequence of words or letters. It is quite similar to sequential data, except that there are no time stamps; instead, there are positions in an ordered sequence. For example, the genetic info of plants and animals can be represented in the form of sequences of nucleotides that are known as genes.

(iv) Time series data:

It is a special type of sequential data in which each record is a time series, i.e., a series of measurements taken over time. For example, a financial data set might contain objects that are time series of the daily prices of various stocks.

(v) Spatial Data:

Some objects have spatial attributes, such as positions or areas, as well as other types of attributes. An example of spatial data is weather data (precipitation, temp, pressure) that is collected for a variety of geographical locations.

Q5) Define Sampling & explain various sampling approaches followed to obtain the samples from the dataset.

⇒ Sampling is a commonly used approach for selecting a subset of the data objects to be analyzed. However the motivations for sampling in statistics & data mining are often different. Statisticians use sampling coz obtaining the entire set of data of interest is too expensive or time consuming, while data miners sample coz its too expensive or time consuming to process all data.

The key principle for effective sampling is the following: Using a sample will work as well as using the entire dataset if the sample is representative. In turn, a sample is representative if it has approximately the same property (of interest) as the original set of data.

Sampling Approaches:

- Simple Random Sampling:

It is the simplest type of sampling. There is an equal probability of selecting any particular item. There are two variations on random sampling.

a) Sampling without replacement: As each item is selected, it is removed from the set of all objects that together constitute the population.

b) Sampling with replacement: Objects are not removed from the population as they are selected for the sample. Here, the same object can be picked more than once.

- Stratified Sampling:

It starts with prespecified groups of objects. In the simplest version, equal no.s of objects are drawn from each group even though the groups are of different sizes. In another variation, the no. of objects drawn from each group is proportional to the size of that group.

Q6) Write a note on the following with an appropriate example:

a) Aggregation:

It is the process of gathering data and presenting it in a summarized format. The data maybe gathered from multiple sources with the intent of combining these data sources into a summary for data analysis. This is a crucial step, since the accuracy of insights from data analysis depends heavily on the amount and quality of data used. It is imp to gather high-quality accurate data and a large enough amount to create relevant results. Data aggregation is useful for everything from finance or business strategy decisions to product, pricing, operations and marketing strategies.

For example, companies often collect data on their online customers & website visitors. The aggregate data would include statistics on customer demographic & behaviour metrics, such as avg age or no. of transactions. This aggregated data can be used by marketing team to personalize messaging, offers & more in the user's digital experience with the brand.

b) Feature Weighting:

Its an alternative to keeping or eleminating features. More imp features are assigned a higher weight, while less imp features are given a lower weight. These weights are sometimes assigned based on domain knowledge about the relative importance of features. Alternatively, they maybe determined automatically. For example, some classification schemes, such as support vector machines produce classification models in which each feature is given a weight. Features with larger weights play a more imp role in the model. The normalization of objects that takes place when computing the cosine similarity can also be regarded as a type of feature weighting.

c) Feature Creation:

It is frequently possible to create, from the original attributes, a new set of attributes that captures the imp info in a dataset much more effectively. Furthermore, the no. of new attributes can be smaller than the no. of original attributes, allowing us to reap all the previously described benefits of dimensionality reduction. Three related methodologies for creating new attributes are described next: feature extraction, mapping the data to a new space, and feature construction.

d) Feature extraction:

The creation of a new set of features from the original raw data is known as feature extraction. Consider a set of photographs, where each photograph is to be classified according to whether or not it contains a human face. The raw data is a set of pixels, and as such is not suitable for many types of classification algorithm. However, if the data is processed to provide higher-level features, such as the presence or absence of certain types of edges and areas that are highly correlated with the presence of human faces, then a much broader set of classification techniques can be applied to the problem.

e) Feature Construction:

Sometimes the features in the original data sets have the necessary info, but it is not in a form suitable for data mining algorithm. In this situation, one or more new features constructed out of the original features can be more useful than the original features.

For example, consider historical artifacts data set which also contains the info of mass & volume of each artifact. In this case, a density feature constructed from mass & volume features i.e. $d = m/v$, would most directly yield accurate classification. The most common approach is to construct features using domain expertise.

Q7) Write a note on proximity between objects having a single attribute that can take value of different attribute types.

⇒ The proximity between two objects is a function of the proximity between the corresponding attributes of the two objects, we first describe how to measure the proximity between objects having only one simple attribute, and then consider proximity measures for objects with multiple attributes.

- Similarity:
  - → It is a numerical measure of the degree to which the two objects are alike.
  - → Higher for pair of objects that are more alike
  - → Usually non-negative and between 0 & 1.
- Dissimilarity:
  - → It is a numerical measure of the degree to which the two objects are different.
  - → Lower for pair of objects that are more similar.
  - → Range 0 to infinity.
- Transformation function:

  It is the function used to convert similarity to dissimilarity and vice versa, or to transform a proximity measure to fall into a particular range. For instance:

  $s' = (s - min(s)) / (max(s) - min(s))$

  where,

  s' → new transformed proximity measure value,

  s → current proximity measure value,

  $min(s)$ → minimum of proximity measure values,

  $max(s)$ → maximum of proximity measure values
- Similarity & Dissimilarity between Simple Attributes:

  The proximity of objects with a no. of attributes is usually defined by combining the proximities of individual attributes,

so, we first discuss proximity between objs having single attribute

Attribute Type	Dissimilarity	Similarity
Nominal	$d = \begin{cases} 0 & \text{if } x = y \\ 1 & \text{if } x \neq y \end{cases}$	$s = \begin{cases} 1 & \text{if } x = y \\ 0 & \text{if } x \neq y \end{cases}$
Ordinal	$d = \lvert x - y \rvert / (n-1)$ (values mapped to integers 0 to n-1, where n → no.of values)	$s = 1 - d$
Interval & Ratio	$d = \lvert x - y \rvert$	$s = -d,\ s = \frac{1}{1+d},\ s = e^{-d},$ $s = 1 - \frac{d - min\_d}{max\_d - min\_d}$

Q8) Compute the proximity measures for the given vectors:

$x = (1,1,0,1,0,1)$ and $y = (1,1,1,0,0,1)$

(i) Cosine Similarity:

$\cos(x,y) = x \cdot y / \|x\| * \|y\|$

$x \cdot y = 1\cdot1 + 1\cdot1 + 0\cdot1 + 1\cdot0 + 0\cdot0 + 1\cdot1 = 3$

$\|x\| = \sqrt{1^2+1^2+0^2+1^2+0^2+1^2} = \sqrt{4} = 2$

$\|y\| = \sqrt{1^2+1^2+1^2+0^2+0^2+1^2} = \sqrt{4} = 2$

$\therefore \cos(x,y) = 3/(2*2) = 3/4 = 0.75$

(ii) Correlation:

$$corr(x,y) = \frac{covariance(x,y)}{std\_deviation(x) * std\_deviation(y)} = \frac{s_{xy}}{s_x \cdot s_y}$$

$$s_{xy} = \frac{1}{n-1} \sum_{k=1}^{n} (x_k - \bar{x})(y_k - \bar{y})$$

$$s_x = \sqrt{\frac{1}{n-1} \sum_{k=1}^{n} (x_k - \bar{x})^2} \qquad \bar{x} = \frac{1}{n} \sum_{k=1}^{n} x_k \text{ is the mean of } x$$

$$s_y = \sqrt{\frac{1}{n-1} \sum_{k=1}^{n} (y_k - \bar{y})^2} \qquad \bar{y} = \frac{1}{n} \sum_{k=1}^{n} y_k \text{ is the mean of } y.$$

$\bar{x} = 1/6(1+1+0+1+0+1) = 4/6$; $\bar{y} = 1/6(1+1+1+0+0+1) = 4/6$

$$S_{xy} = \frac{1}{6-1}\left[\begin{array}{l}(1-4/6)(1-4/6) + (1-4/6)(1-4/6) + (0-4/6)(1-4/6) + \\ (1-4/6)(0-4/6) + (0-4/6)(0-4/6) + (1-4/6)(1-4/6)\end{array}\right]$$

$$= (1/5)(1/3) = 1/15$$

$$S_x = \sqrt{\frac{1}{6-1}\left[\begin{array}{l}(1-4/6)^2 + (1-4/6)^2 + (0-4/6)^2 + (1-4/6)^2 + \\ (0-4/6)^2 + (1-4/6)^2\end{array}\right]}$$

$$= \sqrt{\left(\frac{1}{5}\right)\left[\frac{4}{3}\right]} = 0.5164$$

$$S_y = \sqrt{\frac{1}{6-1}\left[\begin{array}{l}(1-4/6)^2 + (1-4/6)^2 + (1-4/6)^2 + (0-4/6)^2 \\ + (0-4/6)^2 + (1-4/6)^2\end{array}\right]}$$

$$= \sqrt{\left(\frac{1}{5}\right)\left[\frac{4}{3}\right]} = 0.5164$$

$$\text{Corr}(x,y) = (1/15) / (0.5164 * 0.5164) = 0.25$$

(iv) Jaccard Coefficient

$\Rightarrow J = (f_{11}) / (f_{01} + f_{10} + f_{11})$

$f_{01} = 1$, $f_{10} = 1$, $f_{00} = 1$, $f_{11} = 3$

$\therefore J = 3/(1+1+3)$

$\therefore J = 3/5 = 0.6$

(iii) Euclidean Distance:

$$d(x,y) = \sqrt{\sum_{k=1}^{n} (x_k - y_k)^2}$$

$$= \sqrt{\begin{array}{l}(1-1)^2 + (1-1)^2 + (0-1)^2 + (1-0)^2 + \\ (0-0)^2 + (1-1)^2\end{array}}$$

$$= \sqrt{0+0+1+1+0+0}$$

$$= \sqrt{2}$$

$$= 1.4142$$

ASSIGNMENT - 1 : (Unit-3 Solutions)

Q.) For each of the following measures, determine whether it is monotone, anti-monotone, or non-monotone.

a) A characteristic rule is a rule of form $\{p\} \rightarrow \{q_1, q_2, \cdots, q_n\}$, where the rule antecedent contains only a single item. An itemset of size k can produce upto k characteristic rules. Let ~~ζ ($\{p_1, p_2, \cdots p_k\}$)~~ ζ be the minimum confidence of all characteristic rules generated from a given itemset:

$\zeta(\{p_1, p_2, \cdots, p_k\}) = \min[c(\{p_1\} \rightarrow \{p_2, p_3, \cdots p_k\}), \cdots c(\{p_k\} \rightarrow \{p_1, p_2, \cdots p_{k-1}\})]$

$\Rightarrow \zeta$ is an anti-monotone because

$$\zeta(\{A_1, A_2, \cdots, A_k\}) \geq \zeta(\{A_1, A_2, \cdots A_k, A_{k-1}\})$$

For example, we can compare values of ζ for $\{A, B\}$ & $\{A, B, C\}$

$$\zeta(\{A, B\}) = \min(c(A \rightarrow B), c(B \rightarrow A))$$

$$= \min\left(\frac{s(A,B)}{s(A)}, \frac{s(A,B)}{s(B)}\right)$$

$$= \frac{s(A,B)}{\max(s(A), s(B))}$$

$$\zeta(\{A, B, C\}) = \min(c(A \rightarrow BC), c(B \rightarrow AC), c(C \rightarrow AB))$$

$$= \min\left(\frac{s(A,B,C)}{s(A)}, \frac{s(A,B,C)}{s(B)}, \frac{s(A,B,C)}{s(C)}\right)$$

$$= \frac{s(A,B,C)}{\max(s(A), s(B), s(C))}$$

Since $s(A,B,C) \leq s(A,B)$ and $\max(s(A), s(B), s(C)) \geq \max(s(A), s(B))$, therefore $\zeta(\{A,B\}) \geq \zeta(\{A,B,C\})$.

b) A discriminant rule is a rule of the form $\{p_1, p_2, \cdots, p_n\} \rightarrow \{q\}$, where the rule consequent contains only a single item. An itemset of size k can produce upto k discriminant rules. Let η be the minimum confidence of all discriminant rules generated from a given itemset.

$\eta(\{p_1, p_2, \cdots, p_k\}) = \min[c(\{p_2, p_3, \cdots p_k\} \rightarrow \{p_1\}), \cdots c(\{p_1, p_2, \cdots p_{k-1}\} \rightarrow \{p_k\})]$

Is ζ and η monotone, antimonotone, or non-monotone?

⇒ η is non-monotone. We can show this by comparing $\eta(\{A,B\})$ against $\eta(\{A,B,C\})$.

$$\eta(\{A,B\}) = \min(c(A \to B), c(B \to A))$$
$$= \min\left(\frac{s(A,B)}{s(A)}, \frac{s(A,B)}{s(B)}\right)$$
$$= \frac{s(A,B)}{\max(s(A), s(B))}$$

$$\eta(\{A,B,C\}) = \min(c(AB \to C), c(AC \to B), c(BC \to A))$$
$$= \min\left(\frac{s(A,B,C)}{s(A,B)}, \frac{s(A,B,C)}{s(A,C)}, \frac{s(A,B,C)}{s(B,C)}\right)$$
$$= \frac{s(A,B,C)}{\max(s(A,B), s(A,C), s(B,C))}$$

Since $s(A,B,C) \leq s(A,B)$ and $\max(s(A,B), s(A,C), s(B,C)) \leq \max(s(A), s(B))$, therefore $\eta(\{A,B,C\})$ can be greater than or less than $\eta(\{A,B\})$.

Hence, the measure is non-monotone.

(2) Consider dataset shown in below table:

Cust_Id	Transaction ID	Items Bought
1	0001	{a, d, e}
1	0024	{a, b, c, e}
2	0012	{a, b, d, e}
2	0031	{a, c, d, e}
3	0015	{b, c, e}
3	0022	{b, d, e}
4	0029	{c, d}
4	0040	{a, b, c}
5	0033	{a, d, e}
5	0038	{a, b, e}

(i) Compute the support for itemsets {e}, {b,d}, and {b,d,e} by treating each transaction ID as a market basket.

→ $s(\{e\}) = \frac{8}{10} = 0.8$

$s(\{b,d\}) = \frac{2}{10} = 0.2$

$s(\{b,d,e\}) = \frac{2}{10} = 0.2$

(ii) Use the results in part (a) to compute the confidence for the association rules {b,d} → {e} and {e} → {b,d} is confidence a symmetric measure?

→ $c(bd \rightarrow e) = \frac{0.2}{0.2} = 100\%.$

$c(e \rightarrow bd) = \frac{0.2}{0.8} = 25\%.$

No, confidence is a symmetric measure.

(iii) Repeat part (a) by treating each customer ID as a market basket. Each item should be treated as a binary variable (1 if an item appears in atleast one transaction bought by the customer, 0 otherwise).

→ $s(\{e\}) = \frac{4}{5} = 0.8$

$s(\{b,d\}) = \frac{5}{5} = 1$

$s(\{b,d,e\}) = \frac{4}{5} = 0.8$

(iv) Use the results in part (c) to compute the confidence for the association rules {b,d} → {e} and {e} → {b,d}

→ $c(bd \rightarrow e) = 0.8/1 = 80\%.$

$c(e \rightarrow bd) = 0.8/0.8 = 100\%.$

(v) Suppose s_1 and c_1 are the support & confidence values of an association rule r when treating each transaction ID as a market basket. Also, let s_2 and c_2 be the support & confidence values of r when treating each customer ID as a market basket. Discuss whether there are any relationships between s_1 & s_2 or c_1 & c_2.

→ There are no apparent relationships between s_1, s_2, c_1 & c_2.

Q3) Consider the market basket transactions as shown in below table:

TID	Items Bought
1	{Milk, Beer, Diapers}
2	{Bread, Butter, Milk}
3	{Milk, Diapers, Cookies}
4	{Bread, Butter, Cookies}
5	{Beer, Cookies, Diapers}
6	{Milk, Diapers, Bread, Butter}
7	{Bread, Butter, Diapers}
8	{Beer, Diapers}
9	{Milk, Diapers, Bread, Butter}
10	{Beer, Cookies}

a) What is the maximum no. of association rules that can be extracted from this data (including rules that have zero support)?

→ There are 6 items in dataset. Therefore the total no. of rules is 602.

b) What is the maximum size of frequent itemsets that can be extracted (assuming minsup > 0)?

→ Because the longest transaction contains 4 items, the maximum size of frequent itemset is 4.

c) Write an expression for the maximum no. of size-3 itemsets that can be derived from this data set.

⇒ $\binom{6}{3} = 20$

d) Find an itemset (of size 2 or larger) that has largest support.

⇒ {Bread, Butter}

e) Find a pair of items, a & b, such that rules {a} → {b} and {b} → {a} have the same confidence.

→ (Beer, cookies) or (Bread, Butter).

Q4. Consider the following set of frequent 3-itemsets:

{1,2,3}, {1,2,4}, {1,2,5}, {1,3,4}, {1,3,5}, {2,3,4}, {2,3,5}, {3,4,5}

Assume that there are only 5 items in the dataset.

a) List all candidate 4-itemsets obtained by a candidate generation procedure using the $F_{k-1} \times F_1$ merging strategy.

→ {1,2,3,4}, {1,2,3,5}, {1,2,3,6}
{1,2,4,5}, {1,2,4,6}, {1,2,5,6}
{1,3,4,5}, {1,3,4,6}, {2,3,4,5}
{2,3,4,6}, {2,3,5,6}

b) List all candidate 4-itemsets obtained by the candidate generation procedure in Apriori.

→ {1,2,3,4}, {1,2,3,5}, {1,2,4,5}, {2,3,4,5}, {2,3,4,6}

c) List all candidate 4-itemsets that survive the candidate pruning step of the Apriori algorithm.

→ {1,2,3,4}

Q5) The Apriori algorithm uses a generate-and-count strategy for deriving frequent itemsets. Candidate itemsets of size k+1 are created by joining a pair of frequent itemsets of size k (this is known as

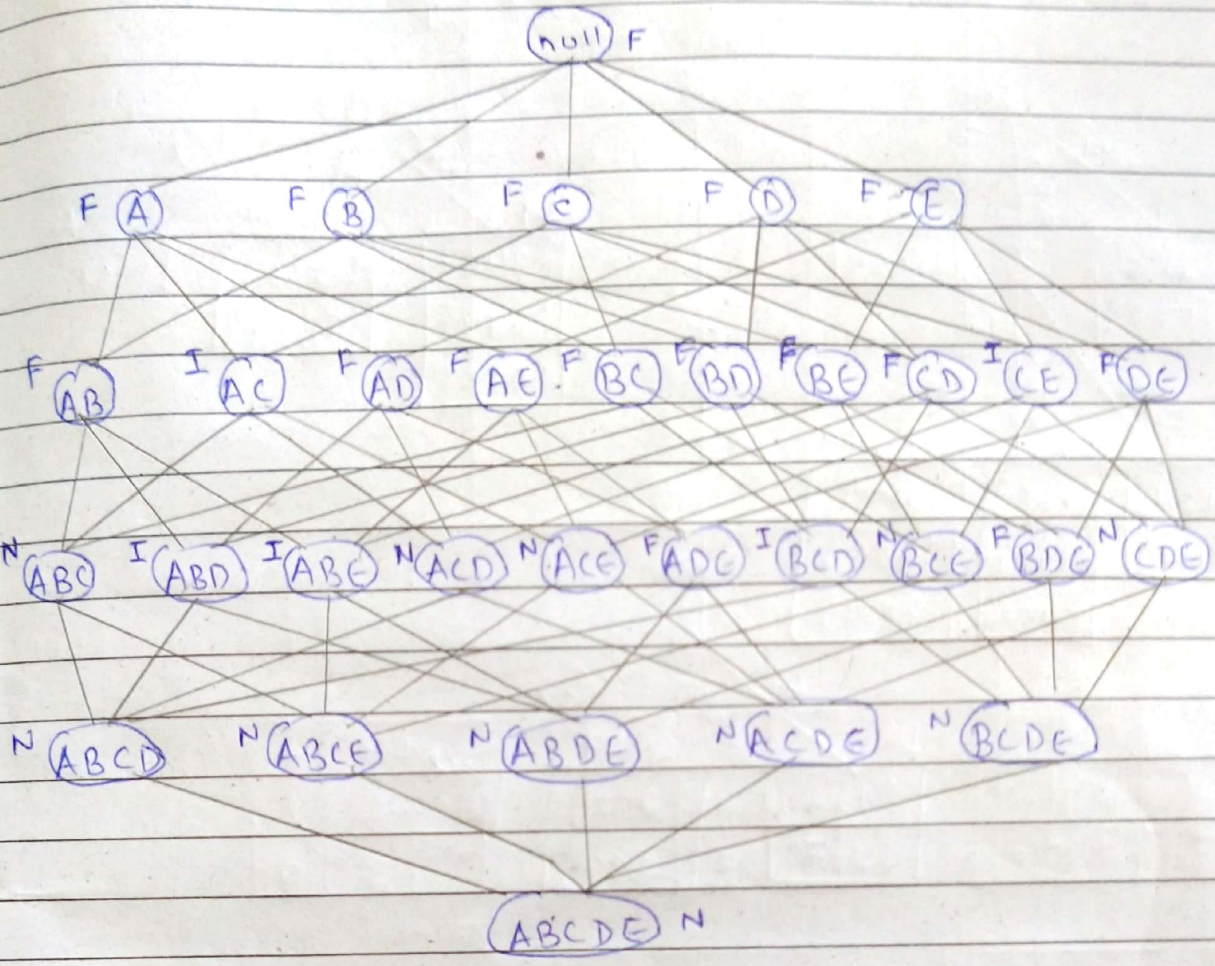
candidate generation step. A candidate is discarded if any one of its subsets is found to be infrequent during the candidate pruning step. Suppose the Apriori algorithm is applied to the dataset given below with minsup = 30%, i.e., any itemset occuring in less than 3 transactions is considered to be infrequent.

TID	Items Bought
1	{a, b, d, e}
2	{b, c, d}
3	{a, b, d, e}
4	{a, c, d, e}
5	{b, c, d, e}
6	{b, d, e}
7	{c, d}
8	{a, b, c}
9	{a, d, e}
10	{b, d}

a) Draw an itemset lattice representing the dataset given in the table. Label each node in the lattice with following letters:

- N: if the itemset is not considered to be a candidate itemset by the Apriori Algorithm. There are two reasons for an itemset not to be considered as a candidate itemset:
  i) It is not generated at all during the candidate generation step, or
  2) It is generated during candidate generation set but is subsequently removed during the candidate pruning step coz one of its subset is found to be infrequent.
- F: if the candidate itemset is found to be frequent by Apriori algorithm
- I: if the candidate itemset is found to be infrequent after support counting

⇒ The lattice structure is shown below:

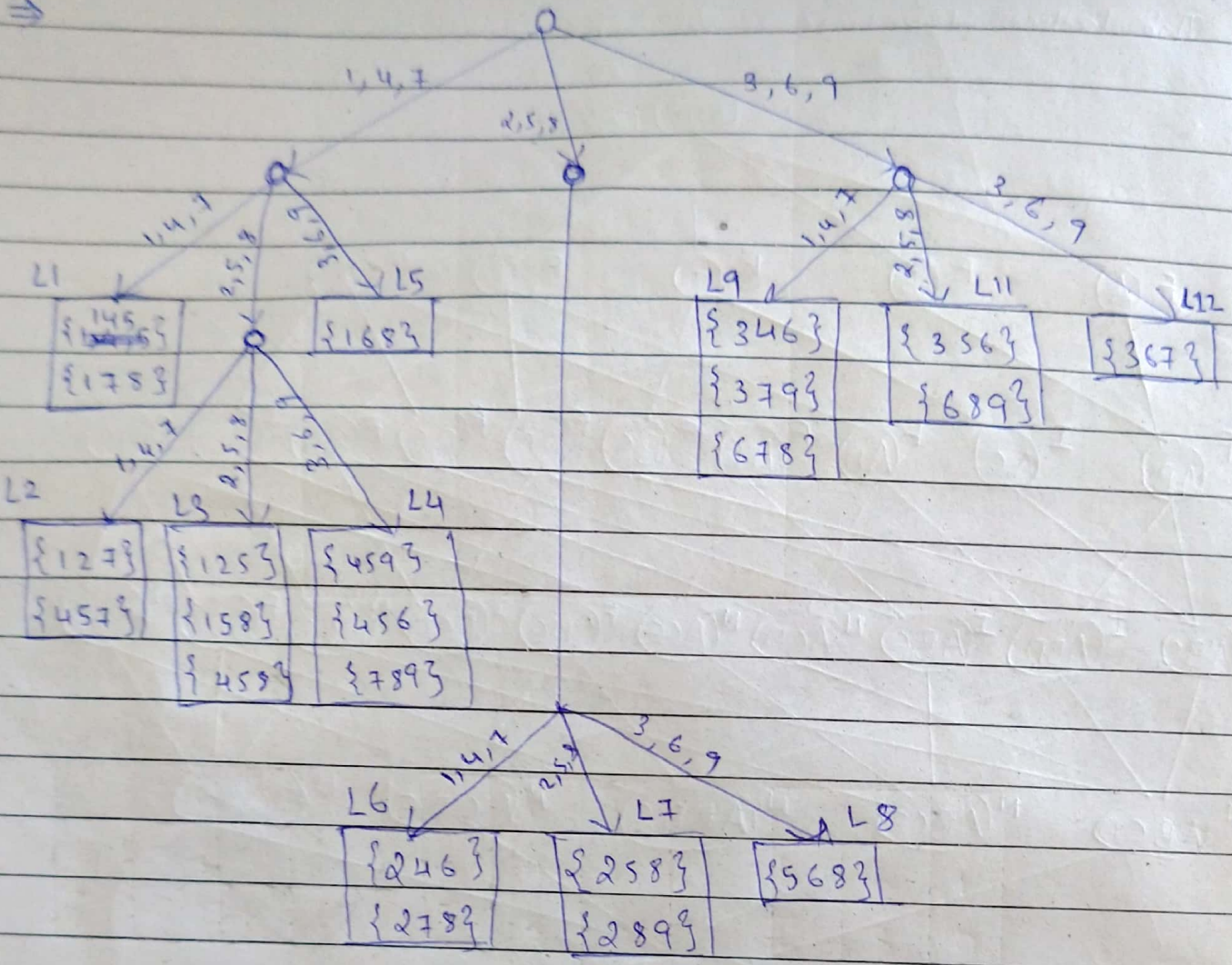


b) What is the percentage of frequent itemsets (with respect to all itemsets in the lattice)?

⇒ Percentage of frequent itemsets = 16/32 = 50%. (including null set)

c) What is the pruning ratio of the Apriori algorithm on this dataset? (Pruning ratio is defined as the percentage of itemsets not considered to be a candidate because (1) they are not generated during candidate generation or (2) they are pruned during the candidate pruning step)

⇒



Pruning ratio is the ratio of N to the total no. of itemsets. Since the count of N = 11, therefore the pruning ratio is 11/32 = 34.4 %.

d) What is the false alarm rate (i.e. percentage of candidate itemsets that are found to be infrequent after performing support counting)?

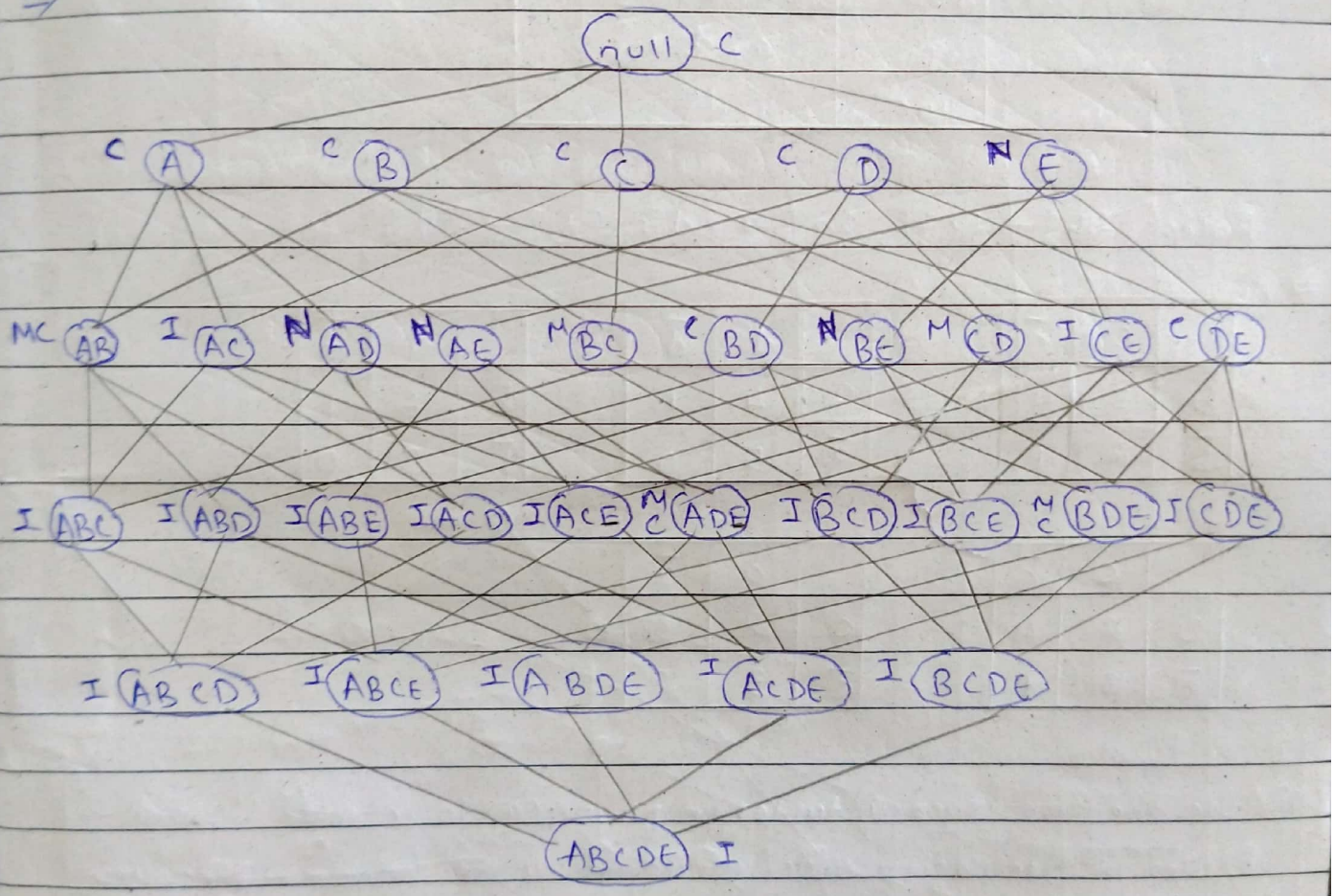
⇒ False alarm rate is the ratio of I to ~~that~~ the total no. of itemsets. Since the count of I = 5, therefore the false alarm rate is 5/32 = 15.6 %.

(06) Given the lattice structure & transaction table in previous question (except N, F, I letters), label each node with the following letters.

- M if the node is a maximal frequent itemset,
- C if it is a closed frequent itemset,
- N if it is frequent but neither maximal nor closed, and
- I if it is infrequent

Assume that the support threshold is equal to 30%.

⇒



(07) The original association rule mining formulation uses the support and confidence measures to prune uninteresting rules.

(a) Draw a contingency table for each of the following rules using the transactions shown in below table:

TID	Items Bought
1	{a, b, d, e}
2	{b, c, d}
3	{a, b, d, e}
4	{a, c, d, e}
5	{b, c, d, e}
6	{b, d, e}
7	{c, d}
8	{a, b, c}
9	{a, d, e}
10	{b, d}

Rules: $\{b\} \rightarrow \{c\}$, $\{a\} \rightarrow \{d\}$, $\{b\} \rightarrow \{d\}$, $\{e\} \rightarrow \{c\}$, $\{c\} \rightarrow \{a\}$

⇒ Answer:

	c	$\bar{c}$
b	3	4
$\bar{b}$	2	1

	d	$\bar{d}$
a	4	1
$\bar{a}$	5	0

	d	$\bar{d}$
b	6	1
$\bar{b}$	3	0

	c	$\bar{c}$
e	2	4
$\bar{e}$	3	1

	a	$\bar{a}$
c	2	3
$\bar{c}$	3	2

b) Use the contingency tables in part (a) to compute & rank the rules in decreasing order according to the following measures:

(i) Support:

Rules	Support	Rank
$b \rightarrow c$	0.3	3
$a \rightarrow d$	0.4	2
$b \rightarrow d$	0.6	1
$e \rightarrow c$	0.2	4
$c \rightarrow a$	0.2	4

(ii) Confidence

Rules	Confidence	Rank
b → c	3/7	3
a → d	4/5	2
b → d	6/7	1
e → c	2/6	5
c → a	2/5	4

(iii) Interest $(X \rightarrow Y) = [P(X,Y)/P(X)] \cdot P(Y)$

Rules	Interest	Rank
b → c	0.214	3
a → d	0.72	2
b → d	0.771	1
e → c	0.167	5
c → a	0.2	4

Q8) Construct the FP-Growth Tree for the following table:

TID	Items Bought
1	{a, b}
2	{b, c, d}
3	{a, c, d, e}
4	{a, d, e}
5	{a, b, c}
6	{a, b, c, d}
7	{a}
8	{a, b, c}
9	{a, b, d}
10	{b, c, e}

⇒ FP-Tree Construction

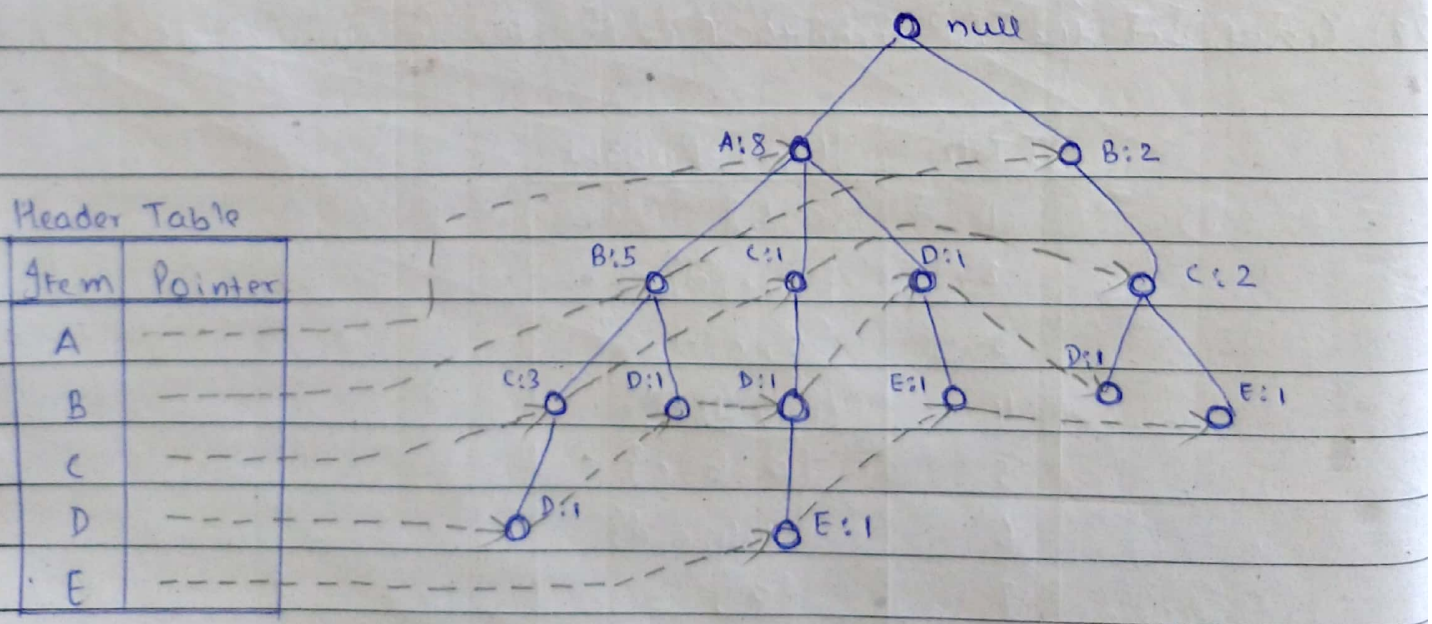
- In the first pass, 1-itemset support is counted & only frequent 1-itemsets (F1) are retained. Infrequent items are discarded.
- items in each transaction are ordered using frequency in descending order.

Most frequent items are likely to be in more prefixes.

- It can also be used in ascending order; Tree constructed will be different & depends on the order in which Txs are used to construct the tree.
- lexicographic order of items is not used.

Item	Frequency
A	8
B	7
C	6
D	5
E	3

Vertical data format is also possible / used in other algorithms.

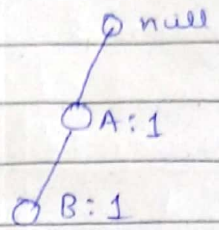


Pointers are used to assist frequent itemset generation.

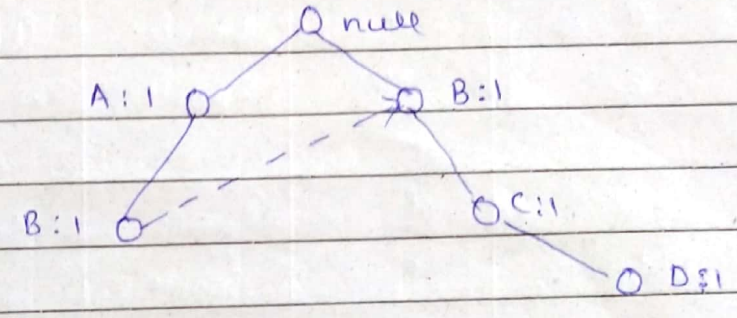
FP-tree construction (Second pass):

TID	Items
1	{A, B}
2	{B, C, D}
3	{A, C, D, E}
4	{A, D, E}
5	{A, B, C}
6	{A, B, C, D}
7	{A}
8	{A, B, C}
9	{A, B, D}
10	{B, C, E}

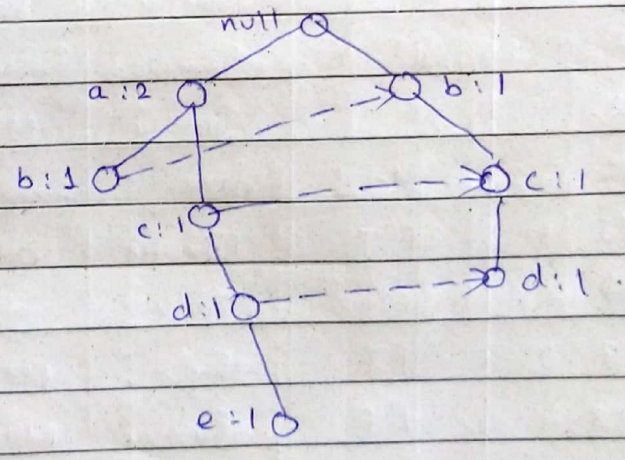
After reading TID = 1:



After reading TID = 2:



After reading TID = 3:



UNIT - 2

Q1) Consider the training examples for a binary classification program:

Customer Id	Gender	Car Type	Shirt Size	Class
1	M	Family	Small	C0
2	M	Sports	Medium	C0
3	M	Sports	Medium	C0
4	M	Sports	Large	C0
5	M	Sports	Extra large	C0
6	M	Sports	Extra large	C0
7	F	Sports	Small	C0
8	F	Sports	Small	C0
9	F	Sports	Medium	C0
10	F	Luxury	Large	C0
11	M	Family	Large	C1
12	M	Family	Extra large	C1
13	M	Family	Medium	C1
14	M	Luxury	Extra large	C1
15	F	Luxury	Small	C1
16	F	Luxury	Small	C1
17	F	Luxury	Medium	C1
18	F	Luxury	Medium	C1
19	F	Luxury	Medium	C1
20	F	Luxury	Large	C1

a) Compute the Gini index for the overall collection of training examples.

$$\Rightarrow Gini(t) = 1 - \sum_{i=0}^{C-1} [p(i|t)]^2 = 1 - \left[\left(\frac{10}{20}\right)^2 + \left(\frac{10}{20}\right)^2\right]$$

$$= 1 - \left[\frac{1}{4} + \frac{1}{4}\right] = \frac{1}{2}$$

b) Compute the Gini index for the customer ID attribute.

⇒ $Gini(t) = 1 - \sum_{i=0}^{c-1} [p(i|t)]^2 = 1 - \left[\left(\frac{0}{1}\right)^2 + \left(\frac{1}{1}\right)^2\right] = 0$

Weighted Average = 0

c) Compute the Gini Index for the Gender attribute.

⇒ Female:

$Gini(t) = 1 - \sum_{i=0}^{c-1} [p(i|t)]^2 = 1 - \left[\left(\frac{6}{20}\right)^2 + \left(\frac{4}{20}\right)^2\right]$

$= 1 - [(0.36) + (0.16)] = 1 - 0.52 = 0.48$

Male:

$Gini(t) = 1 - \sum_{i=0}^{c-1} [p(i|t)]^2 = 1 - \left[\left(\frac{6}{20}\right)^2 + \left(\frac{4}{20}\right)^2\right]$

$= 1 - [(0.36) + (0.16)] = 1 - 0.52 = 0.48$

Weighted Average:

$\left[\text{Female}: \left(\frac{10}{20}\right) * (0.48)\right] + \left[\text{Male}: \left(\frac{10}{20}\right) * (0.48)\right] = 0.48$

d) Compute the Gini index for the Car Type attribute using multiway split.

⇒ Family:

$Gini(t) = 1 - \sum_{i=0}^{c-1} [p(i|t)]^2 = 1 - \left[\left(\frac{1}{4}\right)^2 + \left(\frac{3}{4}\right)^2\right]$

$= 1 - [(0.625) + (0.5625)] = 1 - 0.625 = 0.375$

Luxury:

$Gini(t) = 1 - \sum_{i=0}^{c-1} [p(i|t)]^2 = 1 - \left[\left(\frac{1}{8}\right)^2 + \left(\frac{7}{8}\right)^2\right]$

$= 1 - [(0.0156) + (0.7656)] = 1 - 0.78125 = 0.21875$

Sports:

$$Gini(t) = 1 - \sum_{i=0}^{c-1} [p(i|t)]^2 = 1 - \left[\left(\frac{8}{8}\right)^2 + \left(\frac{0}{8}\right)^2\right]$$

$$= 1 - [(1) + (0)] = 0$$

Weighted Average:

$$\left[\text{Family}: \left(\frac{4}{20}\right) * (0.375)\right] + \left[\text{Luxury}: \left(\frac{8}{20}\right) * (0.21875)\right] + \left[\text{Sports}: \left(\frac{8}{20}\right) * (0)\right] = 0.163$$

c) Explain why Customer Id attribute should not be used as the attribute test condition.

⇒ Customer ID shouldn't be used as the attribute test condition because each attribute is unique.

Q2) Consider the training examples for a binary classification problem:

Instance	a1	a2	Target Class	a3
1	T	T	+	1.0
2	T	T	+	6.0
3	T	F	−	5.0
4	F	F	+	4.0
5	F	T	−	7.0
6	F	T	−	3.0
7	F	F	−	8.0
8	T	F	+	7.0
9	F	T	−	5.0

a) What is the entropy of this collection of training examples?

⇒ There are 4 positive examples & 5 negative examples.

Thus, $P(+) = 4/9$, $P(-) = 5/9$

The entropy of the training examples is

$$-4/9 \log_2(4/9) - 5/9 \log_2(5/9) = 0.9911$$

b) What are the information gain of a1 & a2 relative to these training examples?

⇒ For attribute a1, the corresponding counts & probabilities are:

a_1	+	-
T	3	1
F	1	4

The entropy for a_1 is,

$$\frac{4}{9}\left[-\left(\frac{3}{4}\right)\log_2\left(\frac{3}{4}\right) - \left(\frac{1}{4}\right)\log_2\left(\frac{1}{4}\right)\right]$$

$$+ \frac{5}{9}\left[-\left(\frac{1}{5}\right)\log_2\left(\frac{1}{5}\right) - \left(\frac{4}{5}\right)\log_2\left(\frac{4}{5}\right)\right] = 0.7616$$

Therefore, the info gain for a_1 is $0.9911 - 0.7616 = 0.2294$

For attribute a2, the corresponding counts & probabilities are:

a_2	+	-
T	2	3
F	2	2

The entropy for a_2 is,

$$\frac{5}{9}\left[-\left(\frac{2}{5}\right)\log_2\left(\frac{2}{5}\right) - \left(\frac{3}{5}\right)\log_2\left(\frac{3}{5}\right)\right]$$

$$+ \frac{4}{9}\left[-\left(\frac{2}{4}\right)\log_2\left(\frac{2}{4}\right) - \left(\frac{2}{4}\right)\log_2\left(\frac{2}{4}\right)\right] = 0.9839$$

Therefore, the info gain for a_2 is $0.9911 - 0.9839 = 0.0072$.

c) What is the best split (between a_1 & a_2) according to the classification error rate?

⇒ For attribute a_1 : error rate = 2/9

For attribute a_2 : error rate = 4/9

Therefore, according to error rate, a_1 produces the best split.

Q3) Consider the following set of training examples. Compute a two-level decision tree using the greedy approach. Use the classification error rate as the criterion for splitting. What is the overall error rate of the induced tree?

X	Y	Z	No. of class C1 Examples	No. of class C2 Examples
0	0	0	5	40
0	0	1	0	15
0	1	0	10	5
0	1	1	45	0
1	0	0	10	5
1	0	1	25	0
1	1	0	5	20
1	1	1	0	15

⇒ At level 1

The error rate using attribute X is (60+40)/200 = 0.5; the error rate using attribute Y is (40+40)/200 = 0.4; the error rate using attribute Z is (30+30)/200 = 0.3. Since Z gives the lowest error rate, it is chosen as the splitting attribute at level 1

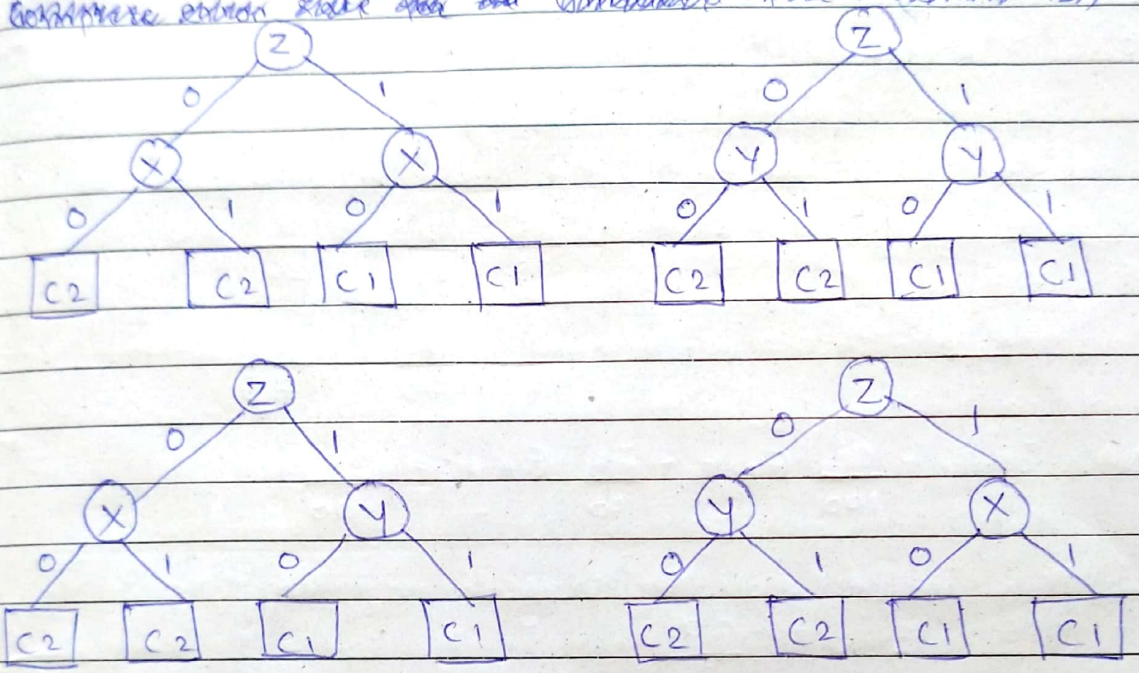
At level 2

For Z = 0, the error rate in both cases (X & Y) are (15+15)/100 = 0.3. For Z = 1, the error rates remain

the same, $(15+15)/100 = 0.3$

Therefore, the corresponding two-level decision tree can be one of the four possibilities shown below and the overall error rate of the induced tree is

$(15+15+15+15)/200 = 0.3$



Q.4) The following table summarizes a dataset with three attributes A, B, C & two class labels +, −. Build 2-level decision tree

A	B	C	No. of Instances	
			+	−
T	T	T	5	0
F	T	T	0	20
T	F	T	20	0
F	F	T	0	5
T	T	F	0	0
F	T	F	25	0
T	F	F	0	0
F	F	F	0	25

a) According to the classification error rate, which attribute would be chosen as the first splitting attribute? For each attribute, show the contingency table & the gains in classification error rate?

=> The error rate for the data without partitioning on any attribute is

$$E_{orig} = 1 - \max\left(\frac{50}{100}, \frac{50}{100}\right) = \frac{50}{100}$$

After splitting on attribute A, the gain in error rate is

	A=T	A=F
+	25	25
-	0	50

$$E_{A=T} = 1 - \max\left(\frac{25}{25}, \frac{0}{25}\right) = \frac{0}{25} = 0$$

$$E_{A=F} = 1 - \max\left(\frac{25}{75}, \frac{50}{75}\right) = \frac{25}{75}$$

$$\Delta A = E_{orig} - \frac{25}{100} E_{A=T} - \frac{75}{100} E_{A=F} = \frac{25}{100}$$

After splitting on attribute B, the gain in error rate is

	B=T	B=F
+	30	20
-	20	30

$$E_{B=T} = \frac{20}{50}$$

$$E_{B=F} = \frac{20}{50}$$

$$\Delta_B = E_{orig} - \frac{50}{100} E_{B=T} - \frac{50}{100} E_{B=F} = \frac{10}{100}$$

After splitting on attribute C, the gain in error rate is

	C=T	C=F
+	25	25
-	25	25

$$E_{C=T} = \frac{25}{50}$$

$$E_{C=F} = \frac{25}{50}$$

$$\Delta_C = E_{orig} - \frac{50}{100} E_{C=T} - \frac{50}{100} E_{C=F} = \frac{0}{100} = 0$$

The algorithm chooses attribute A because it has the highest gain.

b) Repeat for the two children of the root node.

⇒ Because the A = T child node is pure, no further splitting is needed. For the A = F child node, the distribution of training instances is:

B	C	class label	
		+	−
T	T	0	20
F	T	0	5
T	F	25	0
F	F	0	25

The classification error of the A = F child node is:

$$E_{orig} = \frac{25}{75}$$

After splitting on attribute B, the gain in error rate is:

	B = T	B = F
+	25	0
−	20	30

$$E_{B=T} = \frac{20}{45}$$

$$E_{B=F} = 0$$

$$\Delta_B = E_{orig} - \frac{45}{75} E_{B=T} - \frac{20}{75} E_{B=F} = \frac{5}{75}$$

After splitting on attribute C, the gain in error rate is:

	C = T	C = F
+	0	25
−	25	25

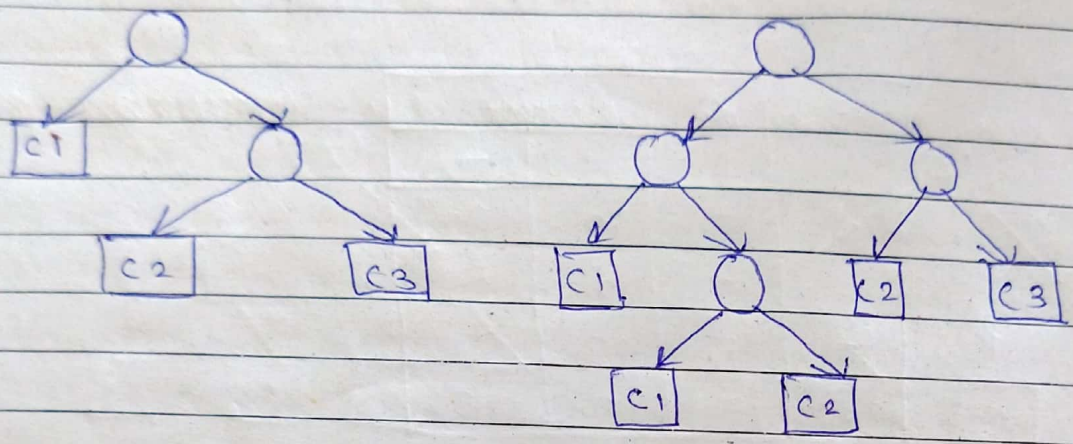
$$E_{C=T} = \frac{0}{25}$$

$$E_{C=F} = \frac{25}{50}$$

$$\Delta_C = E_{orig} - \frac{25}{75} E_{C=T} - \frac{50}{75} E_{C=F} = 0$$

The split will be made on attribute B.

(Q5) Consider the decision trees as shown below. Assume they are generated from a data set D that contains 16 binary attributes and 3 classes C1, C2 and C3.



Compute the total description length of each decision tree according to the minimum description length principle.

- The total description length of a tree is given by:

  Cost (tree, data) = cost (tree) + cost (data | tree)

- Each internal node of the tree is encoded by the ID of the splitting attribute. If there are m attributes, the cost of encoding each attribute is $\log_2 m$ bits.
- Each leaf is encoded using the ID of the class it is associated with. If there are k classes, the cost of encoding a class is node $\log_2 k$ bits.
- Cost (tree) is the cost of encoding all the nodes in the tree. To simplify the computation, you can assume that the total cost of the tree is obtained by adding up the costs of encoding each internal node and each leaf node.
- Cost (data | tree) is encoded using the classification errors the tree commits on the training set. Each error is encoded by $\log_2 n$ bits, where n is the total no. of training instances
- Which decision tree is better, according to the MDL principle?

⇒ Because there are 16 attributes, the cost for each internal node in the decision tree is:

$$\log_2(m) = \log_2(16) = 4$$

Furthermore, because there are 3 classes, the cost for each leaf node is:

$$\lceil \log_2(k) \rceil = \lceil \log_2(3) \rceil = 2$$

The cost for each misclassification error is $\log_2(n)$

The overall cost for the decision tree (a) is $2 \times 4 + 3 \times 2 + 7 \times \log_2 n = 14 + 7 \log_2 n$ and the overall cost for the decision tree (b) is $4 \times 4 + 5 \times 2 + 4 \times 5 = 26 + 4 \log_2 n$. According to the MDL principle, tree (a) is better than (b) if $n < 16$ & is worse than (b) is $n > 16$.

Q6) Explain the following methods that are used to evaluate the performance of a classifier.

a) Holdout Method:

It is the simplest kind of cross validation. The dataset is separated into 2 sets, called the training set and the testing set. The function approximator fits a function using the training set only. Then the function approximator is asked to predict the output values for the data in the testing set. The errors it makes are accumulated as before to give the mean absolute test set error, which is used to evaluate the model. The advantage of this method is that it is usually preferable to the residual method and takes no longer to compute. However, its evaluation can have a high variance. The evaluation may depend heavily on which data points end up in the training set and which end up in the test set, & thus the evaluation may be significantly different depending on how the division is made.

b) Cross-Validation:

Cross validation is a model evaluation method that is better

than residuals. The problem with the residual evaluations is that they don't give an indication of how well the learner will do when it is asked to make new predictions for data it has not already seen. One way to overcome this problem is to not use the entire data set when training a learner. Some of the data is removed before training begins. Then when training is done, the data that was removed can be used to test the performance of the learned model on "new" data. This is the basic idea for a whole class of model evaluation methods called cross validation.

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

Leave-one-out-Cross Validation is K-fold cross validation taken to its logical extreme, with k equal to N, the no. of data points in the set. That means that N separate times, the function

approximator is trained on all the data except for one point & a prediction is made for that point. As before the average error is computed & used to evaluate the model. The evaluation given by leave-one-out cross validation error is good, but at first pass it seems very expensive to compute. Fortunately, locally weighted learners can make LOO predictions just as easily as they make regular predictions. That means computing the leave-one-out cross validation error takes no more time than computing the residual error & it is a much better way to evaluate models. We will see shortly that Vizier relies heavily on leave-one-out cross validation error to choose its metacodes.

Q7. Write a short note on:

a) Tree prepruning: (Early Stopping Rule)

Here, the tree-growing algorithm is halted before generating a fully grown tree that perfectly fits the entire training data. To do this, a more restrictive stopping condition must be used; e.g., stop expanding a leaf node when the observed gain in impurity measure falls below a certain threshold. The advantage of this approach is that it avoids generating overly complex subtrees that overfit the training data. Nevertheless, it is difficult to choose the right threshold for early termination. Too high of a threshold will result in underfitted models, while a threshold that is set too low may not be sufficient to overcome the model overfitting problem. Furthermore, even if no significant gain is obtained using one of the existing attribute test conditions, subsequent splitting may result in better subtrees.

b) Tree post-pruning:

Here, the decision tree is initially grown to its maximum size. This is followed by a tree-pruning step, which proceeds to trim the fully grown tree in a bottom-up fashion. Trimming can be done by replacing a subtree with (1) a new leaf node whose class label is determined from the majority class of records affiliated with the subtree, or (2) the most frequently used branch of the subtree. The tree-pruning step terminates when no further improvement is observed. Post-pruning tends to give better results than prepruning because it makes pruning decisions based on a fully grown tree, unlike prepruning, which can suffer from premature termination of the tree-growing process. However, for post-pruning, the additional computations needed to grow the full tree may be wasted when the subtree is pruned.

c) Bootstrap:

In this approach, the training records are sampled with replacement; i.e., a record already chosen for training is put back into the original pool of records so that it is equally likely to be drawn. If original data has N records, the approximation follows from the fact that the probability a record is chosen by a bootstrap sample is $1-(1-1/N)^N$. When N is sufficiently large, the probability asymptotically approaches $1-e^{-1} = 0.632$. Records that are not included in the bootstrap sample become part of the test set. The model induced from the ~~bootstrap sample become~~ training set is then applied to the test set to obtain an estimate of the accuracy of the bootstrap sample, ϵ_i. The sampling procedure is then repeated b times to generate b bootstrap samples.

Accuracy, $acc_{boot} = \frac{1}{b}\sum_{i=1}^{b}(0.632 \times \epsilon_i + 0.368 \times acc_s)$

Q8) Write a note on following generalization errors estimation.

a) Resubstitution Estimate:

This approach assumes that the training set is a good representation of the overall data. Consequently, the training error, otherwise known as resubstitution error, can be used to provide an optimistic estimate for the generalization error. Under this assumption, a decision tree induction algorithm simply selects the model that produces the lowest training error rate as its final model. However, the training error is usually a poor estimate of generalization error.

b) Pessimistic Error Estimate:

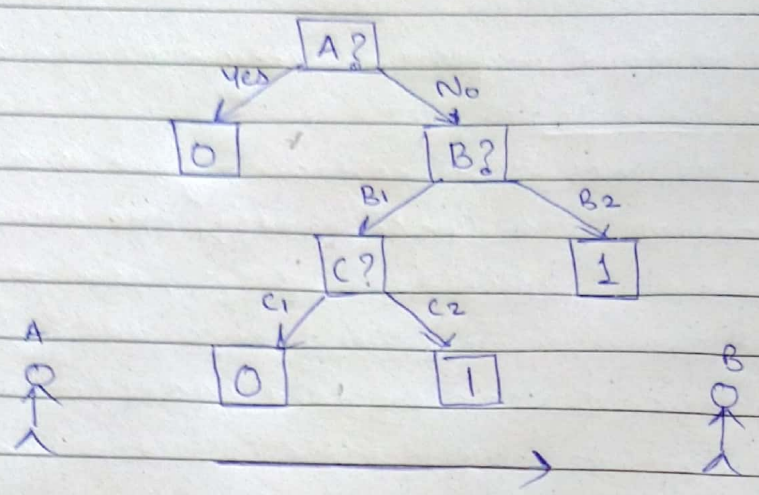
The first approach explicitly computes generalization error as the sum of training error and a penalty term for model complexity. The resulting generalization error ~~for a penalty term for model~~ can be considered its pessimistic error estimate. For instance, let n(t) be no. of training records classified by node t and e(t) be no. of misclassified records. The pessimistic error estimate of a decision tree T, $e_g(T)$, can be computed as:

$$e_g(T) = \frac{\sum_{i=1}^{k} [e(t_i) + \Omega(t_i)]}{\sum_{i=1}^{k} n(t_i)} = \frac{e(T) + \Omega(T)}{N_t}$$

where k → no. of leaf nodes, e(T) is overall training error of decision tree, N_t → no. of training records, and $\Omega(t_i)$ is the penalty term associated with each node t_i.

c) Minimum Description Length Principle:

Another way to incorporate model complexity is based on an information-theoretic approach known as the minimum description length or MDL principle. To illustrate this principle, consider the example:



X	Y → labeled
X_1	1
X_2	0
X_3	0
X_4	1
⋮	⋮
X_n	1

X	Y → Unlabeled
X_1	?
X_2	?
X_3	?
X_4	?
⋮	⋮
X_n	?

In this example, both A & B are given a set of records with known attribute values x. In addition, person A knows the exact class label of each record, while person B knows none of this info. B can contain the classification of each record by requesting that A transmits the class labels sequentially. Such a msg would require $\Theta(n)$ bits of info, where n is the total no. of records.

Alternatively, A may decide to build a classification model that summarizes the relationship between x & y. The model can be encoded in a compact form before being transmitted to B. If the model is 100% accurate, then the cost of the transmission is equivalent to the cost of encoding model. Otherwise, A must also transmit information about which record is classified incorrectly by a model.

Thus, the overall cost of transmission is

$$\text{Cost}(\text{model}, \text{data}) = \text{Cost}(\text{model}) + \text{Cost}(\text{data}|\text{model})$$

where the first term on the RHS is the cost of encoding the model, while the 2nd term represents the cost of encoding the mislabeled records. According to MDL principle, we should seek a model that minimizes the overall cost function. ~~An example show~~