SRM Institute of Science and Technology
College of Engineering and Technology
School of Computing

SRM Nagar, Kattankulathur – 603203, Chengalpattu District, Tamilnadu

Academic Year: 2021-22 (ODD)

Test: CLA-T2	**Date:**
Course Code & Title: 18CSC355T Data Mining and Analytics	**Duration:** 1 Hour
Year & Sem: III Year / V Sem	**Max. Marks:** 50

Course Articulation Matrix: ***(to be placed)***

S.No.	Course Outcome	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
1	**CO1**	L	H		H	L				L	L		H
2	**CO2**	M	H		H	L				M	L		H
3	**CO3**	M	H		H	L				M	L		H
4	**CO4**	M	H		H	L				M	L		H
5	**CO5**	H	H		H	L				M	L		H

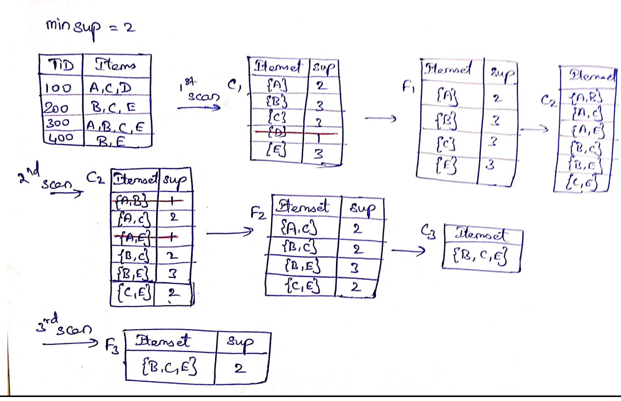
Part - A
(10 x 1 = 10 Marks)

Instructions: Answer all

Q. No	Question	Marks	BL	CO	PO	PI Code
1	In some cases, telecommunication companies desire to segment their clients into distinct groups in order to send suitable and related subscription offer. This can be considered as an example of which of the following methods? a. Supervised learning b. **Unsupervised learning** c. Serration d. Data extraction	**1**	**L1**	**3**	**1**	**1.7.1**
2	What is the relation between a candidate and frequent itemsets? (a) A candidate itemset is always a frequent itemset **(b) A frequent itemset must be a candidate itemset** (c) No relation between these two (d) Strong relation with transactions	**1**	**L2**	**2**	**1**	**1.7.1**
3	Which algorithm requires fewer scans of data? **(a) FP Growth** (b) Naïve Bayes (c) Apriori (d) Decision Tree	**1**	**L2**	**2**	**2**	**1.7.1**
4	For the question given below consider the data Transactions : 1. I1, I2, I3, I4, I5, I6 2. I7, I2, I3, I4, I5, I6 3. I1, I8, I4, I5 4. I1, I9, I10, I4, I6 5. I10, I2, I4, I11, I5 With support as 0.6 find all frequent itemsets? **(a)<I1>, <I2>, <I4>, <I5>, <I6>, <I1, I4>, <I2, I4>, <I2, I5>, <I4, I5>, <I4, I6>, <I2, I4, I5>** (b)<I2>, <I4>, <I5>, <I2, I4>, <I2, I5>, <I4, I5>, <I2, I4, I5> (c)<I11>, <I4>, <I5>, <I6>, <I1, I4>, <I5, I4>, <I11, I5>, <I4,	**1**	**L2**	**2**	**2**	**2.6.3**

	I6>, <I2, I4, I5> (d)<I1>, <I4>, <I5>, <I6>					
5	**Which of the following is a classification algorithm?** a. K-means **b. Decision tree** c. Apriori d. DBSCAN	**1**	**L1**	**3**	**1**	**1.7.1**
6	Frequency of occurrence of an itemset is called as ______ (a) Support (b) Confidence **(c) Support Count** (d) Rules	**1**	**L1**	**2**	**1**	**1.7.1**
7	In the example of predicting number of babies based on storks' population size, number of babies is **(a) Outcome** (b) Feature (c) Attribute (d) observation	**1**	**L2**	**3**	**5**	**1.7.1**
8	**Which of the following is not a type of decision tree node?** a. Root node b. Leaf node c. Decision node **d. Branch node**	**1**	**L2**	**3**	**1**	**1.7.1**
9	How do you calculate Confidence (A -> B)? **(a) Support(A ∩ B) / Support (A)** (b) Support(A ∩ B) / Support (B) (c) Support(A ∪ B) / Support (A) (d) Support(A ∪ B) / Support (B)	**1**	**L2**	**2**	**1**	**1.7.1**
10	The classification or mapping of a class using a predefined class or group is called: a. Data Sub Structure b. Data Set **c. Data Discrimination** d. Data Characterisation	**1**	**L2**	**3**	**1**	**1.7.1**
	Part – B **(5x 5 = 25 Marks)**					
1	**Consider the data set D. Given the minimum support 2, apply Apriori algorithm on this dataset**	**5**	**L3**	**2**	**2**	**2.5.2**

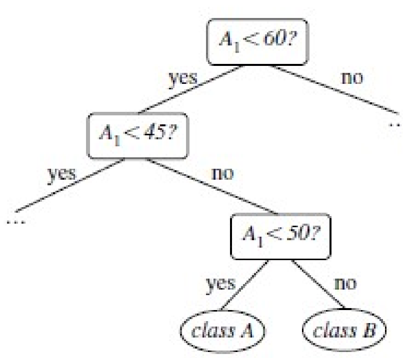
Transaction ID	**Items**
100	**A,C,D**
200	**B,C,E**
300	**A,B,C,E**
400	**B,E**

<table>
<tr><td></td><td>
</td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>2</td><td>How does tree pruning approach works?
There are two common approaches to tree pruning:
-prepruning
-Postpruning
Prepruning approach

A tree is “pruned” by halting its construction early e.g., by deciding not to further split or partition the subset of training tuples at a given node
Upon halting, the node becomes a leaf.
The leaf may hold the most frequent class among the subset tuples or the probability distribution of those tuples.
When constructing a tree, measures such as statistical significance, information gain, Gini index, and so on, can be used to assess the goodness of a split.
If partitioning the tuples at a node would result in a split that falls below a prespecified threshold, then further partitioning of the given subset is halted.
There are difficulties, however, in choosing an appropriate threshold.
High thresholds could result in oversimplified trees, whereas low thresholds could result in very little simplification.

Postpruning Approach

The second and more common approach is which removes subtrees from a “fully grown” tree.
A subtree at a given node is pruned by removing its branches and replacing it with a leaf.
The leaf is labeled with the most frequent class among the subtree being replaced.



(a)</td><td>5</td><td>L3</td><td>3</td><td>2</td><td>1.7.1</td></tr>
<tr><td>3</td><td>Explain the various metrics for evaluating the classifier performance.
Metrics for Evaluating classifier Performance

It presents measures for assessing how good or how “accurate” your classifier is at predicting the class label of tuples
</td><td>5</td><td>L1</td><td>2</td><td>1</td><td>2.6.3</td></tr>
</table>

- Consider the case of where the class tuples are more or less evenly distributed, as well as the case where classes are unbalanced
    - e.g., where an important class of interest is rare such as in medical tests
- They include accuracy (also known as recognition rate), sensitivity (or recall), specificity, precision, *F1, and F.*
- *Note that although accuracy is a specific measure, the word "accuracy" is* also used as a general term to refer to a classifier's predictive abilities.
- Using training data to derive a classifier and then estimate the accuracy of the resulting learned model can result in misleading overoptimistic estimates due to overspecialization of the learning algorithm to the data.

Measure	*Formula*
accuracy, recognition rate	$\frac{TP+TN}{P+N}$
error rate, misclassification rate	$\frac{FP+FN}{P+N}$
sensitivity, true positive rate, recall	$\frac{TP}{P}$
specificity, true negative rate	$\frac{TN}{N}$
precision	$\frac{TP}{TP+FP}$
F, F_1, F-score, harmonic mean of precision and recall	$\frac{2 \times precision \times recall}{precision + recall}$
F_β, where β is a non-negative real number	$\frac{(1+\beta^2) \times precision \times recall}{\beta^2 \times precision + recall}$

Meaning of the various measures

- **True positives** (*TP*): These refer to the positive tuples that were correctly labeled by the classifier. Let *TP* be the number of true positives.
- **True negatives** (*TN*): These are the negative tuples that were correctly labeled by the classifier. Let *TN* be the number of true negatives.
- **False positives** (*FP*): These are the negative tuples that were incorrectly labeled as positive (e.g., tuples of class *buys_computer* = *no* for which the classifier predicted *buys_computer* = *yes*). Let *FP* be the number of false positives.
- **False negatives** (*FN*): These are the positive tuples that were mislabeled as negative (e.g., tuples of class *buys_computer* = *yes* for which the classifier predicted *buys_computer* = *no*). Let *FN* be the number of false negatives.

4	**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}. Show the resulting association rules and its confidence.**	**5**	**L2**	**2**	**2**	**2.5.2**

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

Step 1: Generating 1-Itemset Frequent Pattern

Itemset	Count
{I1}	6
{I2}	7

	{I3} 6 {I4} 2 {I5} 2 The above table is L1. In the first iteration of the algorithm, each item is a member of the set of candidate. The set of frequent 1-itemsets, L1, consists of the candidate 1-itemsets satisfying minimum support. Step 2: Generating 2-Itemset Frequent Pattern To discover the set of frequent 2-itemsets, L2, the algorithm uses L1 Join L1 to generate a candidate set of 2-itemsets, C2. Next, the transactions in D are scanned and the support count for each candidate itemset in C2 is accumulated (as shown in the middle table). The set of frequent 2-itemsets, L2, is then determined, consisting of those candidate 2-itemsets in C2 having minimum support. L1 = {I1,I2,I3,I4,I5}. Since L2 = L1 join L1 then {I1,I2,I3,I4,I5} join {I1,I2,I3,I4,I5}. It becomes -> C2= [{I1,I2} {I1,I3}, {I1,I4}, {I1,I5}, {I2,I3}, {I2,I4}, {I2,I5}, {I3,I4} {I3,I5}, {I4,I4}]. Now we need to check the frequent itemsets with min support count. Then we get -> (C2*C2) L2= [{I1,I2} {I1,I3}, {I1,I5}, {I2,I3}, {I2,I4}, {I2,I5}]. Similarly, We do it for L3. Step 3: Generating 3-Itemset Frequent Pattern L2= [{I1,I2} {I1,I3}, {I1,I5}, {I2,I3}, {I2,I4}, {I2,I5}]. L3 = L2 JOIN L2 i.e. C3 = [{I1,I2,I3},{I1,I2,I5}]. Now, the Join step is complete and the Prune step will be used to reduce the size of C3. Prune step helps to avoid heavy computation due to large Ck. Procedure Step 1: Find Items starting with I2 in B It gives {I1, I2, I3}, {I1, I2, I4}, {I1, I2, I5}. Step 2: Find Items starting with I3 in B It gives NIL, Similarly I4, I5. Step 3: Find out infrequent items sets using min support count and remove them. Based on the Apriori property that all subsets of a frequent itemset must also be frequent, we can determine that four latter candidates cannot possibly be frequent. How? For example, lets take {I1, I2, I3}. The 2-item subsets of it are {I1, I2}, {I1, I3} & {I2, I3}. Since all 2-item subsets of {I1, I2, I3} are members of L2, We will keep {I1, I2, I3} in C3. Lets take another example of {I2, I3, I5} which shows how the pruning is performed. The 2-item subsets are {I2, I3}, {I2, I5} & {I3,I5}. BUT, {I3, I5} is not a member of L2 and hence it is not frequent violating Apriori Property. Thus We will have to remove {I2, I3, I5} from C3. Therefore, C3 = {{I1, I2, I3}, {I1, I2, I5}} after checking for all members of the result of Join operation for Pruning. Now, the transactions in D are scanned in order to determine L3, consisting of those candidates 3-itemsets in C3 having minimum support. Step 4: Generating 4-Itemset Frequent Pattern The algorithm uses L3 Join L3 to generate a candidate set of 4-itemsets, C4. Although the join results in {{I1, I2, I3, I5}}, this itemset is pruned since its subset {{I2, I3, I5}} is not frequent. Thus, C4 = φ(Null), and algorithm terminates, having found all					

	of the frequent items. This completes our Apriori Algorithm. Step 5: Generating Association Rules From Frequent Itemsets Let the minimum confidence threshold is, say 70%. The resulting association rules are shown below, each listed with its confidence. R1: I1 ^ I2 -> I5 Confidence = sc{I1,I2,I5}/sc{I1,I2} = 2/4 = 50%. R1 is Rejected. R2: I1 ^ I5 -> I2 Confidence = sc{I1,I2,I5}/sc{I1,I5} = 2/2 = 100%. R2 is Selected. R3: I2 ^ I5 -> I1 Confidence = sc{I1,I2,I5}/sc{I2,I5} = 2/2 = 100%. R3 is Selected. R4: I1 -> I2 ^ I5 Confidence = sc{I1,I2,I5}/sc{I1} = 2/6 = 33%. R4 is Rejected. R5: I2 -> I1 ^ I5 Confidence = sc{I1,I2,I5}/{I2} = 2/7 = 29% R5 is Rejected. R6: I5 -> I1 ^ I2 Confidence = sc{I1,I2,I5}/ {I5} = 2/2 = 100%. R6 is Selected.					
5	**Explain Naive Baye's Classification.** The **naïve Bayesian** classifier, or **simple Bayesian** classifier, works as follows: **1.** Let D be a training set of tuples and their associated class labels. As usual, each tuple is represented by an n-dimensional attribute vector, $\boldsymbol{X} = (x_1, x_2, \ldots, x_n)$, depicting n measurements made on the tuple from n attributes, respectively, $A_1, A_2, \ldots, A_n$. **2.** Suppose that there are m classes, $C_1, C_2, \ldots, C_m$. Given a tuple, $\boldsymbol{X}$, the classifier will predict that $\boldsymbol{X}$ belongs to the class having the highest posterior probability, conditioned on $\boldsymbol{X}$. That is, the naïve Bayesian classifier predicts that tuple $\boldsymbol{X}$ belongs to the class C_i if and only if $$P(C_i\|\boldsymbol{X}) > P(C_j\|\boldsymbol{X}) \quad \text{for } 1 \le j \le m, j \ne i.$$ Thus we maximize $P(C_i\|\boldsymbol{X})$. The class C_i for which $P(C_i\|\boldsymbol{X})$ is maximized is called the *maximum posteriori hypothesis*. By Bayes' theorem (Equation (6.10)), $$P(C_i\|\boldsymbol{X}) = \frac{P(\boldsymbol{X}\|C_i)P(C_i)}{P(\boldsymbol{X})}. \quad (6.11)$$ **3.** As $P(\boldsymbol{X})$ is constant for all classes, only $P(\boldsymbol{X}\|C_i)P(C_i)$ need be maximized. If the class prior probabilities are not known, then it is commonly assumed that the classes are equally likely, that is, $P(C_1) = P(C_2) = \cdots = P(C_m)$, and we would therefore maximize $P(\boldsymbol{X}\|C_i)$. Otherwise, we maximize $P(\boldsymbol{X}\|C_i)P(C_i)$. Note that the class prior probabilities may be estimated by $P(C_i) = \|C_{i,D}\|/\|D\|$, where $\|C_{i,D}\|$ is the number of training tuples of class C_i in D. 4. In order to reduce computation in evaluating $P(X\text{j}Ci)$, the naïve assumption of class conditional independence is made. $$P(\boldsymbol{X}\|C_i) = \prod_{k=1}^{n} P(x_k\|C_i) = P(x_1\|C_i) \times P(x_2\|C_i) \times \cdots \times P(x_n\|C_i).$$ (a) If A_k is categorical, then $P(x_k\|C_i)$ is the number of tuples of class C_i in D having the value x_k for A_k, divided by $\|C_{i,D}\|$, the number of tuples of class C_i in D. (b) If A_k is continuous-valued, then we need to do a bit more work, but the calculation is pretty straightforward. A continuous-valued attribute is typically assumed to have a Gaussian distribution with a mean μ and standard deviation σ, defined by $$g(x, \mu, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}, \quad (6.13)$$ so that $$P(x_k\|C_i) = g(x_k, \mu_{C_i}, \sigma_{C_i}). \quad (6.14)$$	**5**	**L2**	**3**	**2**	**2.6.3**

5. In order to predict the class label of X, $P(X|C_i)P(C_i)$ is evaluated for each class C_i. The classifier predicts that the class label of tuple X is the class C_i if and only if

$$P(X|C_i)P(C_i) > P(X|C_j)P(C_j) \quad \text{for } 1 \le j \le m, j \ne i. \qquad (6.15)$$

In other words, the predicted class label is the class C_i for which $P(X|C_i)P(C_i)$ is the maximum.

Part – C
(4 x 10 = 10 Marks)

1	**A database has five transactions. Let the minimum support & confidence, min_sup=2, min_confi=80%**	10	L3	2	1	2.5.2

TID	ITEMS
T1	{1,2,3,4,5,6}
T2	{7,2,3,4,5,6}
T3	{1,8,4,5}
T4	{1,9,0,4,6}
T5	{0,2,2,4,5}

Find the frequent itemsets and generate the association rules using Apriori algorithm

TID	Items
T_1	{1,2,3,4,5,6}
T_2	{7,2,3,4,5,6}
T_3	{1,8,4,5}
T_4	{1,9,0,4,6}
T_5	{0,2,2,4,5}

support = 60% = $\frac{60}{100} \times 5 = 3$

confidence = 80%

1. Finding Frequent Itemset

L_1

Item	Count
1	3
2	3
4	5
5	4
6	3

L_2

Item	count
1,4	3
2,4	3
2,5	3
4,5	4
4,6	3

Candidate 1 - Itemset

Item	Count
1	3
2	3
~~3~~	~~2~~
4	5
5	4
6	3
~~7~~	~~1~~
~~8~~	~~1~~
~~9~~	~~1~~
~~0~~	~~2~~

Candidate 2 - Itemset

Itemset	Count
~~1,2~~	~~1~~
1,4	3
~~1,5~~	~~2~~
~~1,6~~	~~2~~
2,4	3
2,5	3
~~2,6~~	~~2~~
4,5	4
4,6	3
~~5,6~~	~~2~~

candidate 3 - Itemset

Item	count
2,4,5	3
~~4,5,6~~	~~2~~

Frequent Itemset
{2,4,5}

2. Generate Association Rule using Min_confi - 80%.

The frequent Itemset - {2,4,5}

write possible subset of frequent itemset

{2,4}, {4,5}, {2,5}, {2}, {4}, {5}, { }, {2,4,5} Eliminate the empty subset

R_1 $2 \wedge 4 \rightarrow 5$ (Accepted)

$$\text{conf} = \frac{\text{sup}(A \cup B)}{\text{sup}(A)} = \frac{\text{sup}(2,4,5)}{\text{sup}(2,4)} = \frac{3}{3} = 1 \; (100\%)$$

R_2 $4 \wedge 5 \rightarrow 2$ (Eliminated)

$$\frac{\text{sup}(4,5,2)}{\text{sup}(4,5)} = \frac{3}{4} = 0.75 \; (75\%)$$

R_3 $2 \wedge 5 \rightarrow 4$ (Accepted)

$$\frac{\text{sup}(2,5,4)}{\text{sup}(2,5)} = \frac{3}{3} = 1 \; (100\%)$$

R_4 $2 \rightarrow 4 \wedge 5$ (Accepted)

$$\frac{\text{sup}(2,4,5)}{\text{sup}(2)} = \frac{3}{3} = 1 \; (100\%)$$

R_5 $4 \rightarrow 2 \wedge 5$ (Eliminated)

$$\frac{\text{sup}(4,2,5)}{\text{sup}(4)} = \frac{3}{5} = 0.6 \; (60\%)$$

R_6 $5 \rightarrow 2 \wedge 4$ (Eliminated)

$$\frac{\text{sup}(5,2,4)}{\text{sup}(5)} = \frac{3}{4} = 0.75 \; (75\%)$$

Final Rules (That satisfies both supports & confidence)

1. $2 \wedge 4 \rightarrow 5$
2. $2 \wedge 5 \rightarrow 4$
3. $2 \rightarrow 4 \wedge 5$

2	**Write the algorithm steps to generate decision tree from the training tuples of the data partition.**	10	L3	3	2	2.6.2

<table>
<tr>
<td></td>
<td>
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:
(1) create a node N;
(2) if tuples in D are all of the same class, C then
(3) return N as a leaf node labeled with the class C;
(4) if attribute_list is empty then
(5) return N as a leaf node labeled with the majority class in D; // majority voting
(6) apply Attribute_selection_method(D, attribute_list) to find the "best" splitting_criterion;
(7) label node N with splitting_criterion;
(8) if splitting_attribute is discrete-valued and
multiway splits allowed then // not restricted to binary trees
(9) attribute_list ← attribute_list − splitting_attribute; // remove splitting_attribute
(10) for each outcome j of splitting_criterion
// partition the tuples and grow subtrees for each partition
(11) let D_j be the set of data tuples in D satisfying outcome j; // a partition
(12) if D_j is empty then
(13) attach a leaf labeled with the majority class in D to node N;
(14) else attach the node returned by Generate_decision_tree(D_j, attribute_list) to node N;
endfor
(15) return N;

Basic algorithm for inducing a decision tree from training tuples.
</td>
<td></td><td></td><td></td><td></td><td></td>
</tr>
<tr>
<td>3</td>
<td>
a) What are the advantages of FP-Growth algorithm?
b) Discuss the applications of association analysis.
Ans a) Advantages of FP Growth Algorithm
o This algorithm needs to scan the database twice when compared to Apriori, which scans the transactions for each iteration.
o The pairing of items is not done in this algorithm, making it faster.
o The database is stored in a compact version in memory.
o It is efficient and scalable for mining both long and short frequent patterns.
o No candidate generation, no candidate test
o Uses compact data structure called FP-Tree
o Eliminates repeated database scan
o Basic operation is counting and FP-tree building
b) Applications of association analysis
The association rule learning is the important technique of machine learning, and it is employed in Market Basket analysis, Web usage mining, continuous production, etc. In market basket analysis, it is an adequate used by several big retailers to find the relations among items.
Association rules were originally transformed from point-of-sale data that represent what products are purchased together. Although its roots are in linking point-of-sale transactions, association rules can be used external the retail market to find relationships among types of "baskets."
There are various applications of Association Rule which are as follows −
• Items purchased on a credit card, such as rental cars and hotel rooms, support insight into the following product that customer are likely to buy.
• Optional services purchased by tele-connection users (call waiting, call forwarding, DSL, speed call, etc.) support decide how to bundle these functions to maximize revenue.
• Banking services used by retail users (money industry accounts, CDs, investment services, car loans, etc.)
</td>
<td>10</td>
<td>L1</td>
<td>2</td>
<td>1</td>
<td>1.7.1</td>
</tr>
</table>

	recognize users likely to needed other services. • Unusual group of insurance claims can be an expression of fraud and can spark higher investigation. • Medical patient histories can supports expressions of likely complications based on definite set of treatments.					
4	**Illustrate any two of the attribute selection measure with an example.** **Attribute Selection Measures** • An attribute selection measure is a heuristic for selecting the splitting criterion that "best" separates a given data partition, *D, of class-labeled training tuples into individual* classes. • If we were to split *D into smaller partitions according to the outcomes of the* splitting criterion, ideally each partition would be pure (i.e., all the tuples that fall into a given partition would belong to the same class). • Attribute selection measures are also known as splitting rules because they determine how the tuples at a given node are to be split. • Three popular attribute selection measures - *information gain* - *gain ratio*, and - *gini index*. **Information gain** • ID3 uses information gain as its attribute selection measure. • The attribute with the highest information gain is chosen as the splitting attribute for node *N*. • Where pi is the probability that an arbitrary tuple in D belongs to class Ci and is estimated by |Ci,D|/|D|. • A log function to the base 2 is used, because the information is encoded in bits. • Info(D) is just the average amount of information needed to identify the class label of a tuple in D. • *Info(D)* is also known as the entropy of *D*. • The expected information needed to classify a tuple in *D is given by* $Info(D) = -\sum_{i=1}^{m} p_i \log_2(p_i),$ • How much more information would we still need (after the partitioning) in order to arrive at an exact classification? This amount is measured by $Info_A(D) = \sum_{j=1}^{v} \frac{\|D_j\|}{\|D\|} \times Info(D_j).$ • The term |*Dj* |/|*D*| acts as the weight of the *j*th partition. • *InfoA*(*D*) is the expected information required to classify a tuple from *D* based on the partitioning by *A*. • The smaller the expected information (still) required, the greater the purity of the partitions. • Information gain is defined as the difference between the original information requirement (i.e., based on just the proportion of classes) and the new requirement (i.e., obtained after partitioning on *A*). $Gain(A) = Info(D) - Info_A(D).$ • The class label attribute, *buys computer*, has two distinct values (namely, {*yes, no*}). • There are two distinct classes (that is, *m* = 2). • Let class *C*1 correspond to *yes* and class *C*2 correspond to *no.* • There are nine tuples of class *yes* and five tuples of class *no*. A (root) node *N* is created for the tuples in *D*.	**10**	**L2**	**3**	**2**	**2.5.2**

$$Info(D) = -\frac{9}{14}\log_2\left(\frac{9}{14}\right) - \frac{5}{14}\log_2\left(\frac{5}{14}\right) = 0.940 \text{ bits.}$$

- Compute the expected information requirement for each attribute.
- *Age* category *youth – 2 yes &3 no, Middle aged – 4 yes & 0 no, Senior – 3 yes & 2 no.*

$$\begin{aligned} Info_{age}(D) = &\frac{5}{14} \times \left(-\frac{2}{5}\log_2\frac{2}{5} - \frac{3}{5}\log_2\frac{3}{5}\right) \\ &+ \frac{4}{14} \times \left(-\frac{4}{4}\log_2\frac{4}{4} - \frac{0}{4}\log_2\frac{0}{4}\right) \\ &+ \frac{5}{14} \times \left(-\frac{3}{5}\log_2\frac{3}{5} - \frac{2}{5}\log_2\frac{2}{5}\right) \\ = &\ 0.694 \text{ bits.} \end{aligned}$$

$$Gain(age) = Info(D) - Info_{age}(D) = 0.940 - 0.694 = 0.246 \text{ bits.}$$

- Compute *Gain*(*income*) = 0.029 bits, *Gain*(*student*) = 0.151 bits, and *Gain*(*credit rating*) = 0.048 bits.
- Age has the highest information gain among the attributes, it is selected as the splitting attribute.

Gain ratio

- It prefers to select attributes having a large number of values.
- C4.5, a successor of ID3, uses an extension to information gain known as *gain ratio.*
- It applies a kind of normalization to information gain using a "split information" value defined analogously with *Info*(*D*) as

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

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

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

$$SplitInfo_A(D) = -\frac{4}{14} \times \log_2\left(\frac{4}{14}\right) - \frac{6}{14} \times \log_2\left(\frac{6}{14}\right) - \frac{4}{14} \times \log_2\left(\frac{4}{14}\right).$$

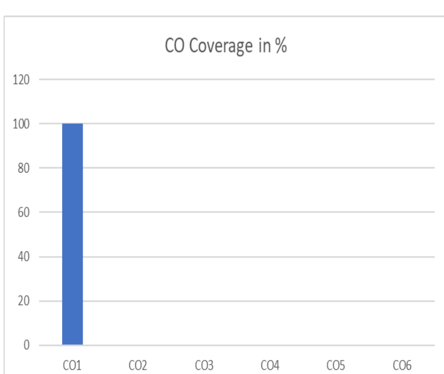
$$= 0.926.$$

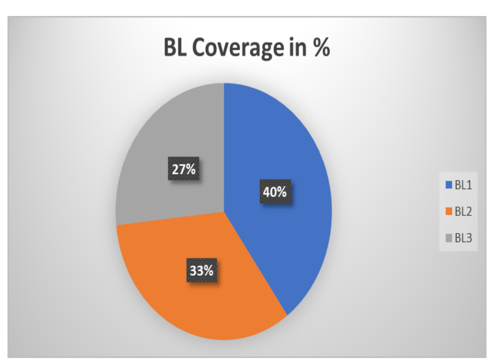
$$Gain(income) = 0.029.$$

$$GainRatio(income) = 0.029/0.926 = 0.031.$$

***Program Indicators are available separately for Computer Science and Engineering in AICTE examination reforms policy.**

Course Outcome (CO) and Bloom's level (BL) Coverage in Questions





Approved by the Audit Professor/Course Coordinator