USN | | | | | | | | | | |

CMRIT — CMR INSTITUTE OF TECHNOLOGY, BENGALURU. ACCREDITED WITH A+ GRADE BY NAAC

Internal Assessment Test 2 – June 2021

Sub:	Data Mining and Data warehouse					Sub Code:	18CS641/17 CS651	Branch:	ISE
Date:	23/06/2021	Duration:	90 min’s	Max Marks:	50	Sem/Sec:	VI A,B&C		OBE

	Answer any FIVE FULL Questions	MARKS	CO	RBT
1a)	Compare the data mining tasks clustering and classification with example.	[4]	CO1	L2

At least 4 points- 4 marks

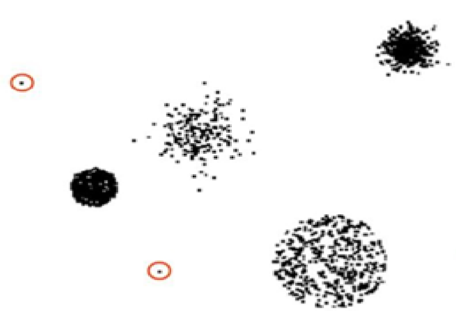
Classification	Clustering
Find a model for class attribute as a function of the values of other attributes.	Given a set of data points, each having a set of attributes, and a similarity measure among them, find clusters such that Data points in one cluster are more similar to one another and Data points in separate clusters are less similar to one another.
previously unseen records should be assigned a class as accurately as possible.	Similarity measures like Euclidean Distance, Manhattan are used
Mode is built based on training data set and tested with testing data set.	Data set does not have class labels
Decision Tree, Naive Bayes, KNN are examples Classifying bank customer, Predict fraudulent cases in credit card transactions.	K-Means, DBSCAN, Agglomerative Clustering Market Segmentation, Grouping the documents, and News

		MARKS	CO	RBT
1b)	Explain the various data quality problems and how to handle them.	[6]	CO1	L2

Data mining focuses on
(1) the detection and correction of data quality problems (called data cleaning.)
(2) the use of algorithms that can tolerate poor data quality.

- **Examples of data quality problems:[1 Mark]**
  - Noise and outliers
  - missing values
  - duplicate data

Noise: Noise refers to modification of original values **[1 Mark]**
Examples: distortion of a person’s voice when talking on a poor phone and “snow” on television screen
Outliers : Outliers are data objects with characteristics that are considerably different than most of the other data objects in the data set**[1 Mark]**



	Missing Values: [2 Marks] □ **Reasons for missing values** • Information is not collected (e.g., people decline to give their age and weight) • Attributes may not be applicable to all cases (e.g., annual income is not applicable to children) □ **Handling missing values** • Eliminate Data Objects • Estimate Missing Values • Ignore the Missing Value During Analysis • Replace with all possible values (weighted by their probabilities) **Duplicate Data: [1 Mark]** □ Data set may include data objects that are duplicates, or almost duplicates of one another □ Major issue when merging data from heterogeneous sources □ Examples: Same person with multiple email addresses □ **Data cleaning** Process of dealing with duplicate data issues			
2a)	Why data preprocessing is significant step data mining? Explain Discretization and Binarization techniques with example.	[2+4]	CO1	L2

Data preprocessing makes the data suitable for data mining

It reduces noise, duplicates, processing time and memory requirements and removes unwanted data. **[2 Marks]**

Binarization: [2 Marks]

- **Binarization** maps a continuous or categorical attribute into one or more binary variables. Typically used for association analysis
- **Often convert a continuous attribute to a categorical attribute and then convert a categorical attribute to a set of binary attributes**
- Association analysis needs asymmetric binary attributes

Table 2.5. Conversion of a categorical attribute to three binary attributes.

Categorical Value	Integer Value	x_1	x_2	x_3
awful	0	0	0	0
poor	1	0	0	1
OK	2	0	1	0
good	3	0	1	1
great	4	1	0	0

Table 2.6. Conversion of a categorical attribute to five asymmetric binary attributes.

Categorical Value	Integer Value	x_1	x_2	x_3	x_4	x_5
awful	0	1	0	0	0	0
poor	1	0	1	0	0	0
OK	2	0	0	1	0	0
good	3	0	0	0	1	0
great	4	0	0	0	0	1

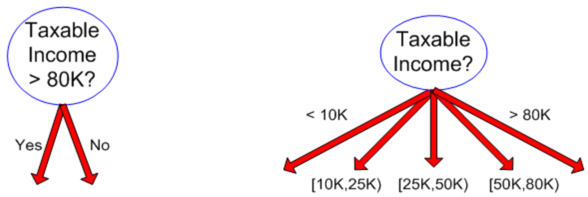
Discretization[2 Marks]

- Discretization is the process of converting a continuous attribute into an ordinal attribute
- Discretization is typically applied to attributes that are used in **classification**

<table>
<tr><td></td><td>

or association analysis.

- Transformation of a continuous attribute to a categorical attribute involves two subtasks:
- **deciding how many categories to have and determining how to map the values of the continuous attribute to these categories.**
-



(i) Binary split (ii) Multi-way split

</td><td></td><td></td><td></td></tr>
<tr><td>2b)</td><td>

Compute the cosine similarity of the following two objects that represent two document vectors:

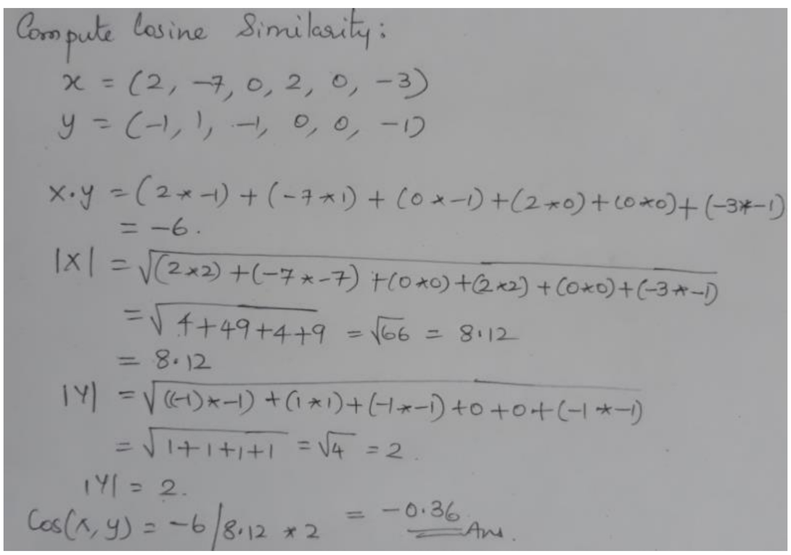
x = (2, -7,0,2,0, -3) , y =(-1, 1,- 1,0,0, -1)

x.y - [1 mark]

|x| - [1 mark]

|y | - [1 mark]

Cosine Similarity = [1 mark]



Compute Cosine Similarity:

$x = (2, -7, 0, 2, 0, -3)$

$y = (-1, 1, -1, 0, 0, -1)$

$x \cdot y = (2*-1) + (-7*1) + (0*-1) + (2*0) + (0*0) + (-3*-1)$

$= -6.$

$|X| = \sqrt{(2*2) + (-7*-7) + (0*0) + (2*2) + (0*0) + (-3*-1)}$

$= \sqrt{4+49+4+9} = \sqrt{66} = 8.12$

$= 8.12$

$|Y| = \sqrt{((-1)*-1) + (1*1) + (-1*-1) + 0 + 0 + (-1*-1)}$

$= \sqrt{1+1+1+1} = \sqrt{4} = 2$

$|Y| = 2.$

$Cos(x, y) = -6 / 8.12 * 2 = -0.36$ Ans.

</td><td>[4]</td><td>CO1</td><td>L3</td></tr>
<tr><td>3</td><td>

Write and Explain Apriori Algorithm for frequent itemset generation with example.

Algorithm with Explanation -[5 marks]

- The algorithm initially makes a single pass over the data set to determine the support of each item. Upon completion of this step, the set of all frequent 1-itemsets, F_1, will be known (steps 1 and 2).
- Next, the algorithm will iteratively generate new candidate k-itemsets using the frequent $(k-1)$-itemsets found in the previous iteration (step 5). Candidate generation is implemented using a function called apriori-gen, which is described in Section 6.2.3.

</td><td>[5+5]</td><td>CO3</td><td>L2</td></tr>
</table>

- To count the support of the candidates, the algorithm needs to make an additional pass over the data set (steps 6–10). The subset function is used to determine all the candidate itemsets in C_k that are contained in each transaction t. The implementation of this function is described in Section 6.2.4.
- After counting their supports, the algorithm eliminates all candidate itemsets whose support counts are less than *minsup* (step 12).
- The algorithm terminates when there are no new frequent itemsets generated, i.e., $F_k = \emptyset$ (step 13).

Algorithm 6.1 Frequent itemset generation of the *Apriori* algorithm.

```
1:  k = 1.
2:  F_k = { i | i ∈ I ∧ σ({i}) ≥ N × minsup}.    {Find all frequent 1-itemsets}
3:  repeat
4:    k = k + 1.
5:    C_k = apriori-gen(F_{k-1}).    {Generate candidate itemsets}
6:    for each transaction t ∈ T do
7:      C_t = subset(C_k, t).    {Identify all candidates that belong to t}
8:      for each candidate itemset c ∈ C_t do
9:        σ(c) = σ(c) + 1.    {Increment support count}
10:     end for
11:   end for
12:   F_k = { c | c ∈ C_k ∧ σ(c) ≥ N × minsup}.    {Extract the frequent k-itemsets}
13: until F_k = ∅
14: Result = ∪ F_k.
```

Example with explanation- [5 marks]

Transactions

TID	Items
1	Bread, Milk
2	Bread, Diapers, Beer, Eggs
3	Milk, Diapers, Beer, Coke
4	Bread, Milk, Diapers, Beer
5	Bread, Milk, Diapers, Coke

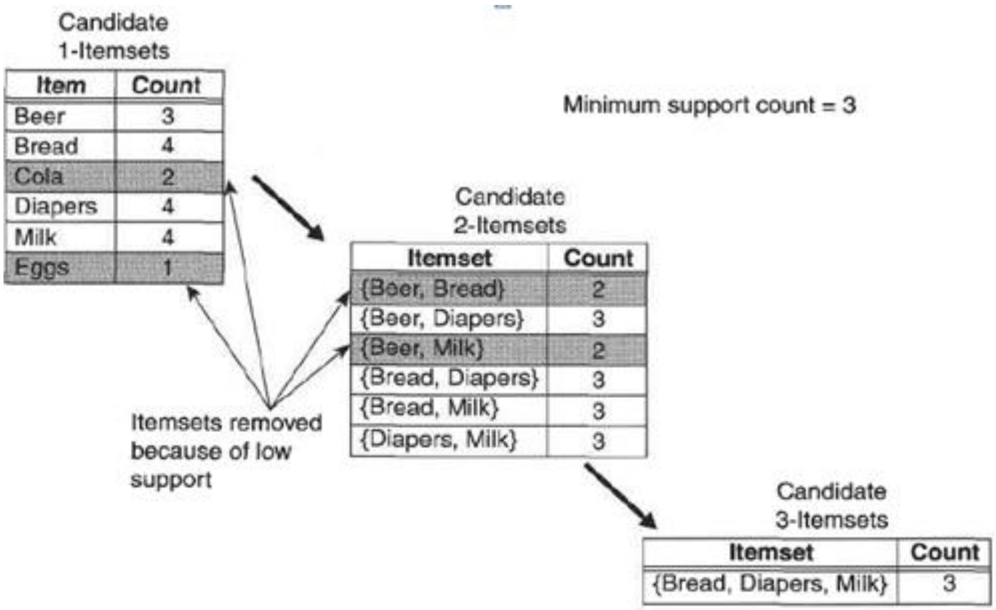


Figure 6.5. Illustration of frequent itemset generation using the *Apriori* algorithm.

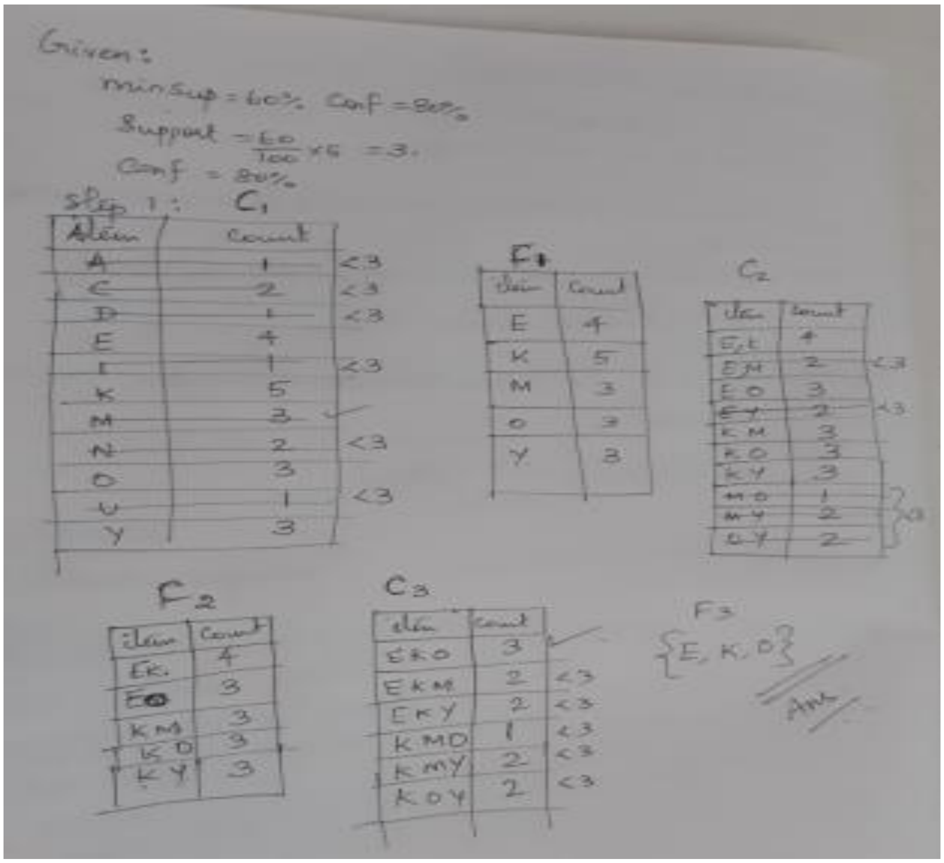
4a) A database has five transactions. Let min sup=60% and min conf =80%. [8] CO3 L3

TID	items bought

T100	{M, O, N, K, E, Y}
T200	{D, O, N, K, E, Y}
T300	{M, A, K, E}
T400	{M, U, C, K, Y}
T500	{C, O, O, K, I, E}

Find all frequent item sets using Apriori algorithm.

Computation of C1 = 2 marks
Computation of F1 = 1 marks
Computation of C2 = 1 marks
Computation of F2 = 1 marks
Computation of C3 = 1 marks
Computation of F3 = 2 marks


Given:
minSup = 60% Conf = 80%
Support = 60/100 × 5 = 3.
Conf = 80%
Step 1: C_1

Item	Count	
A	1	<3
C	2	<3
D	1	<3
E	4	
I	1	<3
K	5	
M	3	
N	2	<3
O	3	
U	1	<3
Y	3	

F_1

Item	Count
E	4
K	5
M	3
O	3
Y	3

C_2

Item	Count	
E,K	4	
E,M	2	<3
E,O	3	
E,Y	2	<3
K,M	3	
K,O	3	
K,Y	3	
M,O	1	<3
M,Y	2	<3
O,Y	2	<3

F_2

Item	Count
EK	4
EO	3
KM	3
KO	3
KY	3

C_3

Item	Count	
EKO	3	
EKM	2	<3
EKY	2	<3
KMO	1	<3
KMY	2	<3
KOY	2	<3

F_3
{E, K, O}
Ans

4b)	What is anti-monotone property with respect to support of an itemset? Explain with example	[2]	CO3	L2

Property: 1 mark
Example: 1 mark

Apriori principle holds due to the following property of the support measure:

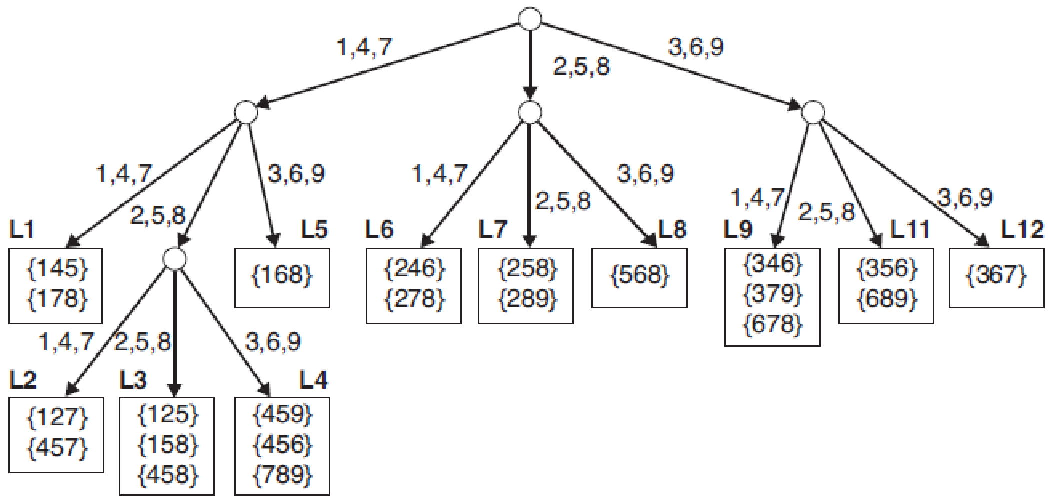
$$\forall X, Y : (X \subseteq Y) \Rightarrow s(X) \geq s(Y)$$

– Support of an itemset never exceeds the support of its subsets
– This is known as the anti-monotone property of support

Example:
Consider {a,b,c} itemset. Assume Support count (a,b,c)=5 support count(a,b) >=Support count(a,b,c) Number of times a,b,c occuring in Transactions never exceeds Support count(a,b)

Q. No.	Question	Marks	CO	Level
5	Construct the Hash tree structure for the given set of candidate item sets. Given transaction {1, 3, 4, 5, 8}, which of the hash tree leaf nodes will be visited when finding the candidates of the transaction? {145}{178}{127}{457}{125}{158}{458}{459}{456}{789}{168}{246}{278}{258}{289}{568}{346}{379}{678}{356}{689}{367}	[10]	CO3	L3

Construction of Hash Tree: 5 marks



Given a transaction that contains items {1, 3, 4, 5, 8}, The leaf nodes visited are L1, L3, L5, L9, and L11 will be visited when finding the candidates of the transaction. **5 marks**

Q. No.	Question	Marks	CO	Level
6	Explain the Apriori rule Generation algorithm with example	[10]	CO2	L2

Algorithm: 5 marks

Example: 5 marks

Algorithm 6.2 Rule generation of the *Apriori* algorithm.

1: **for** each frequent k-itemset f_k, $k \geq 2$ **do**
2: $H_1 = \{i \mid i \in f_k\}$ {1-item consequents of the rule.}
3: **call** ap-genrules(f_k, H_1.)
4: **end for**

Algorithm 6.3 Procedure ap-genrules(f_k, H_m).

$k = |f_k|$ {size of frequent itemset.}
$m = |H_m|$ {size of rule consequent.}
if $k > m + 1$ **then**
  H_{m+1} = apriori-gen(H_m).
  for each $h_{m+1} \in H_{m+1}$ **do**
    $conf = \sigma(f_k)/\sigma(f_k - h_{m+1})$.
    if $conf \geq minconf$ **then**
      output the rule $(f_k - h_{m+1}) \longrightarrow h_{m+1}$.
    else
      delete h_{m+1} from H_{m+1}.
    end if
  end for
  call ap-genrules(f_k, H_{m+1}.)
end if

Rule generation: Min conf = 50%

Assume k = 3.

$f_3 = \{B, C, E\}$ frequent item set.

Alg 6.2 produces

$H_1 = \{B\}\{C\}\{E\}$

Alg 6.3 ap-genrules

① $k = |f_k| = 3$ ② $m = |H_m| = 1$.

$k = 3$.

$\sigma(B,C,E) = 3$

$\sigma(E) = 5$

$\sigma(C) = 4$

$\sigma(B) = 5$.

③ $3 > 2$

for $H_{m+1} = H_2 = $ apriori_gen(H_1)

④ $H_2 = \{B, C\}\{B, E\}\{C, E\}$.

⑤ for $\{BC\}$ → $conf = \sigma(BCE) / \sigma(BCE - BC) = 3/\sigma(E) = \frac{3}{5} = 60\% > minconf$ (50%)

($f_k - h_{m+1}$)

∴ E → BC is valid rule. — ①

for (B,E) $conf = \sigma(BCE) / \sigma(BCE - BE) = \frac{\sigma(BCE)}{\sigma(C)} = 3/4 = 75\% > min\ conf\%$ (50%)

C → BE is valid rule. → ②

for (CE) $conf = \sigma(BCE) / \sigma(BCE - CE) = \frac{\sigma(BCE)}{\sigma(B)} = 3/5 = 60\% > minconf$ (50%)

∴ B → CE is valid rule → ③

Recursive call.

ap-genrules($\{B, C, E\}$, $\{\{B,C\}, \{B,E\}\{C,E\}\}$)

(f_k, H_{m+1})

2nd call

$k = |f_k| = 3$.

$m = |H_m| = 2$ (size of the rule consequent)

$3 > 3$

$k > 2+1$

false

End if.