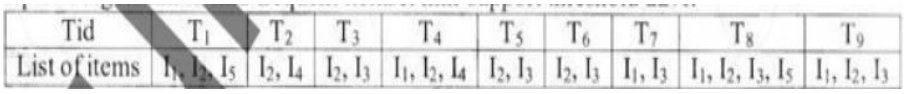
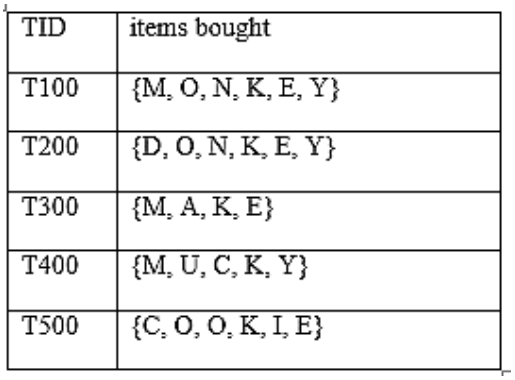
Scheme of Evaluation
Internal Assessment Test 2 – May 2023

Sub:	DATA MINING AND DATA WAREHOUSING							Code:	18CS641
Date:	24/5/2023	Duration:	90mins	Max Marks:	50	Sem:	VI	Branch:	ISE

Note: Answer Any five full questions.

Question #		Description	Marks Distribution		Max Marks
1	a)	Explain Data preprocessing steps Data Mining with example. Aggregation • Sampling • Dimensionality Reduction • Feature subset selection • Feature creation • Discretization and Binarization • Attribute Transformation Any 5	2M*5	10M	10M
2	a)	Briefly explain the similarity Dissimilarity between the objects. Find the SMC and Jacquard coefficient of two binary vectors. 10M X=(1,0,0,0,0,0,0,0,0,0) Y=(0,0,0,0,0,0,0,0,0,1) Similarity Dissimilarity SMC Jacquard coefficient	2.5M 2.5M 2.5M 2.5M	10M	10M
3	a)	Briefly explain the candidate generation procedure using $F_{k-1} \times F_{k-1}$ merging strategy. Diagram Explanation	4M 5M	10M	10M

4	a)	Consider the following transaction data set and list the items using Apriori Algorithm to find frequent itemset. Minimum support threshold= 22%. Also write valid association rule. Confidence = 50%  Candidate generation Association Rules	5M 5M	10M	10M
5	a)	What is frequent itemset generation? Explain frequent itemset generation of Apriori algorithm. Definition frequent item set Apriori Algorithm Pseudo code Example	2M 5M 3M	10M	10M
6	a)	Construct the FP tree showing the trees separately after reading each transaction and generate the frequent item set using FP growth algorithm.  Support count FP growth tree after every transaction Conditional pattern Frequent item set	1M 5M 2M 2M	10M	10M

Tid	T_1	T_2	T_3	T_4	T_5	T_6	T_7	T_8	T_9
List of items	I_1, I_2, I_5	I_2, I_4	I_2, I_3	I_1, I_2, I_4	I_2, I_3	I_2, I_3	I_1, I_3	I_1, I_2, I_3, I_5	I_1, I_2, I_3

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}

Scheme Of Evaluation
Internal Assessment Test 2 – May 2023

Sub:	DATA MINING AND DATA WAREHOUSING							Code:	18CS641
Date:	24/5/2023	Duration:	90mins	Max Marks:	50	Sem:	VI	Branch:	ISE

Note: Answer Any full five questions

Q. 1 Explain Data preprocessing steps Data Mining with example.
Aggregation • Sampling • Dimensionality Reduction • Feature subset selection • Feature creation • Discretization and Binarization • Attribute Transformation

Sampling

- Sampling is the main technique employed for data selection.
- Selecting a subset of the data objects to be analyzed
- It is often used for both the preliminary investigation of the data and the final data analysis.
- Statisticians sample because obtaining the entire set of data of interest is too expensive or time consuming.
- The key principle for effective sampling is the following:
  - using a sample will work almost as well as using the entire data sets, if the sample is representative
  - A sample is representative if it has approximately the same property (of interest) as the original set of data

Types of Sampling

- Simple Random Sampling

There is an equal probability of selecting any particular item

- Sampling without replacement

As each item is selected, it is removed from the population

□ Sampling with replacement

Objects are not removed from the population as they are selected for the sample. In sampling with replacement, the same object can be picked up more than once

□ Stratified sampling

Split the data into several partitions; then draw random samples from each partition

Dimensionality Reduction:

- A key benefit is that many data mining algorithms work better if the dimensionality the number of attributes in the data-is lower.
- This is partly because dimensionality reduction can eliminate irrelevant features and reduce noise

Purpose:

o Avoid curse of dimensionality
o Reduce amount of time and memory required by data mining algorithms
o Allow data to be more easily visualized
o May help to eliminate irrelevant features or reduce noise

□ **The Curse of Dimensionality:** the curse of dimensionality refers to the phenomenon that many types of data analysis become significantly harder as the dimensionality of the data increases. Specifically, as dimensionality increases, the data becomes increasingly sparse in the space that it occupies.

- Principal Component Analysis and Singular valued decomposition(linear algebra Technique) are the techniques for Dimensionality reduction
- PCA- for continuous attributes that finds new attributes (principal components) that (1) are linear combinations of the original attributes, (2) are orthogonal (perpendicular) to each other, and (3) capture the maximum amount of variation in the data.

Feature Subset Selection:

- Another way to reduce dimensionality of data
- Use only a subset of the features
- Redundant and irrelevant features can reduce classification accuracy and the quality of the clusters that are found.

Redundant features

- Duplicate much or all of the information contained in one or more other attributes
- Example: purchase price of a product and the amount of sales tax paid almost same

Irrelevant features

- Contain no information that is useful for the data mining task at hand
- Example: students' ID is often irrelevant to the task of predicting students' GPA

Many techniques developed, especially for classification- There are three standard approaches to feature selection: embedded, filter, and wrapper.

□ Techniques of feature subset selection: Brute-force approach:

- Try all possible feature subsets as input to data mining algorithm
- **Embedded approaches:**

  Feature selection occurs naturally as part of the data mining algorithm. Specifically, during the operation of the data mining algorithm, the algorithm itself decides which attributes to use and which to ignore.
- **Filter approaches:**

  Features are selected before data mining algorithm is run using some approach that is independent of the data mining task. For example, we might select sets of attributes whose pair wise correlation is as low as possible.

Q. 2 a) Briefly explain the similarity Dissimilarity between the objects. Find the SMC and Jacquard coefficient of two binary vectors. 10M

X=(1,0,0,0,0,0,0,0,0,0) Y=(0,0,0,0,0,0,0,0,0,1)

Similarity and Dissimilarity: Definition

- Similarity between two objects is a numerical measure of how alike two data objects are.
- Similarities are higher for pairs of objects that are more alike.
- Similarities are usually non-negative and are often between 0 (no similarity) and 1 (complete similarity).
- **Dissimilarity** between two objects is a Numerical measure of how different are two data objects.
- Minimum dissimilarity is often 0, Upper limit varies

Similarity/Dissimilarity for Simple Attributes

p and q are the attribute values for two data objects

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 is the number of values)	$s = 1 - d$
Interval or Ratio	$d = \lvert x - y \rvert$	$s = -d$, $s = \frac{1}{1+d}$, $s = e^{-d}$, $s = 1 - \frac{d - min\_d}{max\_d - min\_d}$

Dissimilarities between Data Objects

Euclidean Distance

The Euclidean distance, d, between two points, x and y, in one-, two-, three-, or higher dimensional space, is given by the following familiar formula:

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

where n is the number of dimensions (attributes) and x_k and y_k are, respectively, the k^{th} attributes (components) or data objects $\mathbf{x}$ and $\mathbf{y}$.

Example:

Minkowski Distance

- Minkowski Distance is a generalization of Euclidean Distance

$$d(\mathbf{x}, \mathbf{y}) = \left(\sum_{k=1}^{n} |x_k - y_k|^r \right)^{1/r}$$

-
- Where r is a parameter. Where n is the number of dimensions attributes) and x_k and y_k are,respectively, the kth attributes (components) or data objects x and y.

Cosine Similarity

- We need a similarity measure for documents needs to ignores 0-0 matches like the Jaccard measure, but also must be able to handle non-binary vectors.
- x and y are two document vectors, then

$$\cos(\mathbf{x}, \mathbf{y}) = \frac{\mathbf{x} \cdot \mathbf{y}}{\|\mathbf{x}\| \, \|\mathbf{y}\|}, \qquad (2.7)$$

Similarity Measures for Binary Data-examples of Proximity Measures

- Let x and y be two objects that consist of n binary attributes. The comparison of two such objects, i.e., two binary vectors, Leads to the following four quantities (frequencies:)

  f_{00} = the number of attributes where **x** is 0 and **y** is 0
  f_{01} = the number of attributes where **x** is 0 and **y** is 1
  f_{10} = the number of attributes where **x** is 1 and **y** is 0
  f_{11} = the number of attributes where **x** is 1 and **y** is 1

- **Simple Matching Coefficient (SMC):**

$$SMC = \frac{\text{number of matching attribute values}}{\text{number of attributes}} = \frac{f_{11} + f_{00}}{f_{01} + f_{10} + f_{11} + f_{00}}.$$

- **Jaccard Coefficient:**
- The Jaccard coefficient is frequently used to handle objects consisting of asymmetric binary attributes. The Jaccard coefficient, which is often symbolized by J is given by the following equation:

$$J = \frac{\text{number of matching presences}}{\text{number of attributes not involved in 00 matches}} = \frac{f_{11}}{f_{01} + f_{10} + f_{11}}.$$

Example 2.17 (The SMC and Jaccard Similarity Coefficients). To illustrate the difference between these two similarity measures, we calculate SMC and J for the following two binary vectors.

$\mathbf{x} = (1, 0, 0, 0, 0, 0, 0, 0, 0, 0)$
$\mathbf{y} = (0, 0, 0, 0, 0, 0, 1, 0, 0, 1)$

$f_{01} = 2$ the number of attributes where **x** was 0 and **y** was 1
$f_{10} = 1$ the number of attributes where **x** was 1 and **y** was 0
$f_{00} = 7$ the number of attributes where **x** was 0 and **y** was 0
$f_{11} = 0$ the number of attributes where **x** was 1 and **y** was 1

$SMC = \frac{f_{11}+f_{00}}{f_{01}+f_{10}+f_{11}+f_{00}} = \frac{0+7}{2+1+0+7} = 0.7$

$J = \frac{f_{11}}{f_{01}+f_{10}+f_{11}} = \frac{0}{2+1+0} = 0$ ■

Q. 3 a) Briefly explain the candidate generation procedure using $F_{k-1} \times F_{k-1}$ merging strategy.

$\mathbf{F}_{k-1} \times \mathbf{F}_{k-1}$ Method The candidate generation procedure in the apriori-gen function merges a pair of frequent $(k-1)$-itemsets only if their first $k-2$ items are identical. Let $A = \{a_1, a_2, \ldots, a_{k-1}\}$ and $B = \{b_1, b_2, \ldots, b_{k-1}\}$ be a pair of frequent $(k-1)$-itemsets. A and B are merged if they satisfy the following conditions:

$$a_i = b_i \text{ (for } i = 1, 2, \ldots, k-2) \text{ and } a_{k-1} \neq b_{k-1}.$$

In Figure 6.8, the frequent itemsets {Bread, Diapers} and {Bread, Milk} are merged to form a candidate 3-itemset {Bread, Diapers, Milk}. The algorithm does not have to merge {Beer, Diapers} with {Diapers, Milk} because the first item in both itemsets is different. Indeed, if {Beer, Diapers, Milk} is a viable candidate, it would have been obtained by merging {Beer, Diapers} with {Beer, Milk} instead. This example illustrates both the completeness of the candidate generation procedure and the advantages of using lexicographic ordering to prevent duplicate candidates. However, because each candidate is obtained by merging a pair of frequent $(k-1)$-itemsets, an additional candidate pruning step is needed to ensure that the remaining $k-2$ subsets of the candidate are frequent.

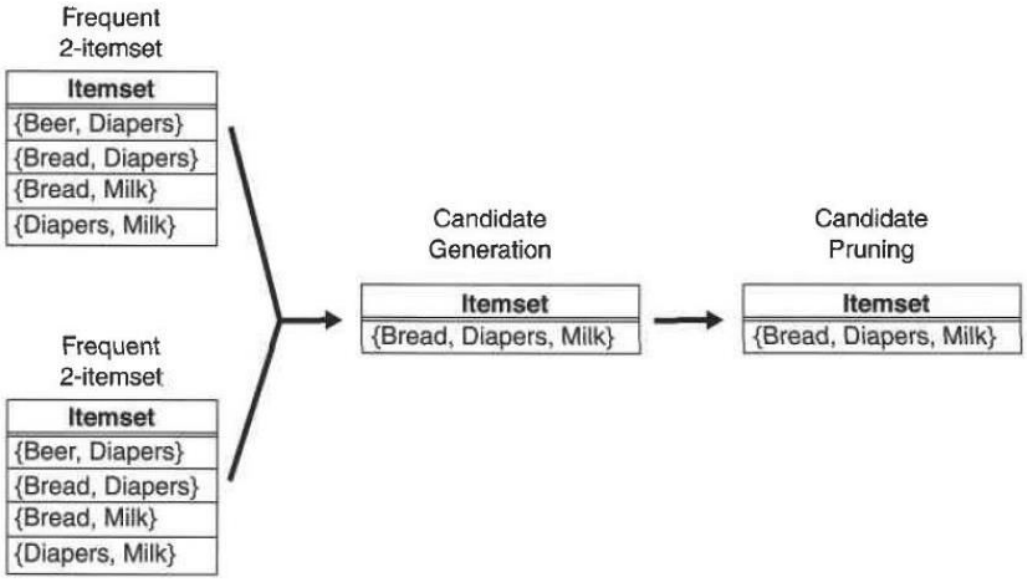


Figure 6.8. Generating and pruning candidate k-itemsets by merging pairs of frequent $(k-1)$-itemsets.

Q. 4 Consider the following transaction data set and list the items using Apriori Algorithm to find frequent itemset. Minimum support threshold= 22%. Also write valid association rule. Confidence = 50%

Tid	T_1	T_2	T_3	T_4	T_5	T_6	T_7	T_8	T_9
List of items	I_1, I_2, I_5	I_2, I_4	I_2, I_3	I_1, I_2, I_4	I_2, I_3	I_2, I_3	I_1, I_3	I_1, I_2, I_3, I_5	I_1, I_2, I_3

Transactions List

TID	List of Items IDs
T100	I1, I2, I5
T200	I2, I4
T300	I2, I3
T400	I1, I2, I4
T500	I1, I3
T600	I2, I3
T700	I1, I3
T800	I1, I2, I3, I5
T900	I1, I2, I3

1-item Sets	Frequency
I1	6
I2	7
I3	5
I4	4
I5	2

Frequent 1-item Sets	Frequency
I1	6
I2	7
I3	5
I4	4

Transactions List

I1, I2, I3, I4, I5

TID	List of Items IDs
T100	I1, I2, I5
T200	I2, I4
T300	I2, I3
T400	I1, I2, I4
T500	I1, I3
T600	I2, I3
T700	I1, I3
T800	I1, I2, I3, I5
T900	I1, I2, I3

3-item Sets	Frequency
I1, I2, I3	2
I1, I2, I4	0
I1, I2, I5	2
I1, I3, I4	1
I1, I3, I5	1
I1, I4, I5	0
I2, I3, I4	0
I2, I3, I5	1
I2, I4, I5	0
I3, I4, I5	0

Frequent 3-item Sets	Frequency
I1, I2, I3	2
I1, I2, I5	2

Association Rule Mining - Subset Creation

- Frequent 3-Item Set = I => {1, 2, 3} and {1, 2, 5}
- Min_Support = 2 = 2/9 = 22.22% and Min_Confidence = 50%

- Non-Empty subset are
  - {{1}, {2}, {3}, {1, 2}, {1, 3}, {2, 3}}
  - {{1}, {2}, {5}, {1, 2}, {1, 5}, {2, 5}}

- How to form Association Rule…?
  - For every non-empty subset S of I, the association rule is,

Association Rule Mining - Subset Creation

- Frequent 3-Item Set = I => {1, 2, 3}
- Non-Empty subset are
  - {{1}, {2}, {3}, {1, 2}, {1, 3}, {2, 3}}

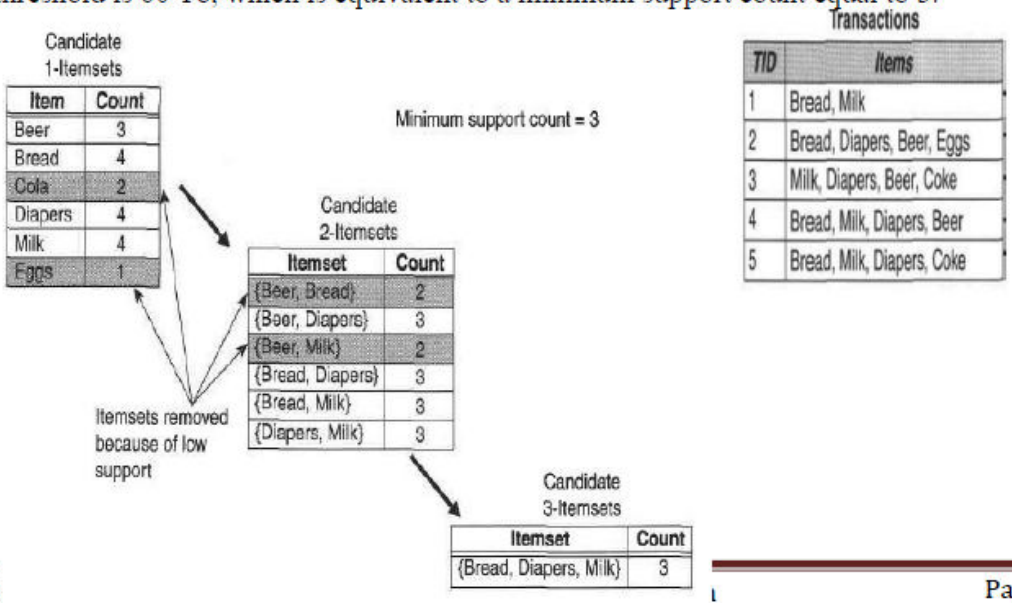
- Rule 1: {1} → {2, 3} {S= 22.22 %, C=33.34%}
  - Support = 2/9 = 22.22%
  - Confidence = Support (1, 2, 3)/Support(1) = $\frac{2/9}{6/9}$ = 2/6 = 33.34% < 50%
  - Invalid Rule

- Rule 2: {2} → {1, 3} {S= 22.22 %, C=28.57%}
  - Support = 2/9 = 22.22 %

Q. 5 a) What is frequent itemset generation? Explain frequent itemset generation of Apriori algorithm.

Frequent Itemset Generation in the Apriori Algorithm: Illustration with example.

- Apriori, is the first association rule mining algorithm that pioneered the use of support-based pruning to systematically control the exponential growth of candidate itemsets
- Figure 6.5 provides a high-level illustration of the frequent item set generation part of the Apriori algorithm for the transactions shown in Table 6.1. We assume that the support threshold is 60 To, which is equivalent to a minimum support count equal to 3.



- Initially, every item is considered as a candidate 1-itemset. After counting their supports, the candidate itemsets {Co1a} and {Eggs} are discarded because they appear in fewer than three transactions.
- In the next iteration, candidate 2-itemsets are generated using only the frequent 1-itemsets because the **Apriori principle ensures that all supersets of the infrequent 1-itemsets must be infrequent.**
- Because there are only four frequent 1-itemsets, the number of candidate 2-itemsets generated by the algorithm is 6. Two of these six candidates, {Beer, Bread} and {Beer, Milk}, are subsequently found to be infrequent after computing their support values. The remaining four candidates are frequent, and thus will be used to generate candidate 3-itemsets.
- Without support-based pruning, there are 20 candidate 3-itemsets that can be formed using the six items given in this example. With the Apriory principle, we only need to keep candidate 3- itemsets whose subsets are frequent. The only candidate that has this property is {Bread, Diapers, Milk}

  If every subset is considered,
  ${}^6C_1 + {}^6C_2 + {}^6C_3 = 41$
  With support-based pruning,
  $6 + 6 + 1 = 13$

- The effectiveness of the Apriori pruning strategy can be shown by counting the number of candidate itemsets generated.
- A brute-force strategy of enumerating all itemsets(up to size3) as candidates will produce 41 candidates.
- With the Apriori principle, this number decreases to 13 candidates, which represents a 68% reduction in the number of candidate itemsets even in this simple example.

Apriori Algorithm:

The pseudocode for the frequent itemset generation part of the *Apriori* algorithm is shown in Algorithm 6.1. Let C_k denote the set of candidate k-itemsets and F_k denote the set of frequent k-itemsets:

- 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.

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

1: $k = 1$.
2: $F_k = \{ i \mid i \in I \wedge \sigma(\{i\}) \geq N \times minsup\}$. {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 $t \in T$ **do**
7: C_t = subset(C_k, t). {Identify all candidates that belong to t}
8: **for** each candidate itemset $c \in C_t$ **do**
9: $\sigma(c) = \sigma(c) + 1$. {Increment support count}
10: **end for**
11: **end for**
12: $F_k = \{ c \mid c \in C_k \wedge \sigma(c) \geq N \times minsup\}$. {Extract the frequent k-itemsets}
13: **until** $F_k = \emptyset$
14: Result = $\bigcup F_k$.

- 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).

Q. 6 a)

Construct the FP tree showing the trees separately after reading each transaction and generate the frequent item set using FP growth algorithm.

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}