Qn. Set Code-7(SOT) Semester: Programme: 5'"B. Tech

Branch: Comp. Engg_

AUTUMN END SEMESTER EXAMINATION-2025

5$^{rd}$ Semester B.Tech

DATA MINING DATA WAREHOUSING

CS30013 1 IT 3031

(For 2023 & Previous Admitted Batches)

Time: 2 Hours 30 Minutes Full Marks: 50

## SECTION-A

1. Answer the following questions. [1 x 10]

**(a) Define supervised and unsupervised learning in data mining.**

Solution:

□ Supervised**:** model trained on labeled data (input → known output). Used for classification/regression (e.g. spam detection, house price prediction).

□ Unsupervised**:** model finds structure in unlabelled data (clustering, dimensionality reduction, association rules). Used for segmentation, anomaly detection.

**(b) Give the five-number summary of the data, where the age values for the data tuples are (in increasing order) 13, 15, 16, 16, 19, 20, 20, 21, 22, 22, 25, 25, 25, 25, 30, 33, 33, 35, 35, 35, 35, 36, 40, 45, 46, 52, 70.**

Solution: 13, 15, 16, 16, 19, 20, 20, 21, 22, 22, 25, 25, 25, 25, 30, 33, 33, 35, 35, 35, 35, 36, 40, 45, 46, 52, 70 (27 values).

- **Min** = 13
- **Q1** = 20.5
- **Median** = 25.0
- **Q3** = 35.0
- **Max** = 70.0.

**(c) What is the importance of conelation analysis in understanding relationships between variables, and how can it support decision-making and predictive modeling?**

Solution:

□ Measures linear association between variables (strength & direction).

□ Helps identify which features are related (reduces multicollinearity), informs feature selection, and suggests predictors for regression/prediction.

□ Correlation supports decision-making (e.g., if X and Y highly correlated, changes in X indicate changes in Y) and helps build better predictive models and diagnostics.

(d) **What is the role of entropy in decision tree algorithms?**

Solution:

□ Entropy measures impurity/uniformity of class labels in a node. Lower entropy = more pure node.

□ Decision tree algorithm picks splits that **maximize information gain** (reduction in entropy) so child nodes are more homogeneous — leads to simpler, more accurate trees.

(e) **How many 3-item candidate sets are possible from 6 unique items?**

Solution: Combinations C(6,3) = 20.

(f) **How many steps are there in the back propagation algorithm? What learning rule is used to implement gradient descent in back propagation networks?**

Solution:

□ Forward pass: compute network outputs.

□ Compute error (loss) at output.

□ Backward pass: propagate error gradients back through layers using chain rule.

□ Update weights using gradient descent (or variant).
**Learning rule:** gradient descent (stochastic / batch / mini-batch) — weights updated by $w \leftarrow w - \eta \, \partial \text{Loss}/\partial w$.

(g) **Compute the Jaccard similarity between the following two binary vectors:**
**X= {0011001100}, Y = {1110001110}**

Solution: **Jaccard similarity** between binary vectors X = 0011001100, Y = 1110001110. Positions with 1s: X = {2,3,6,7}, Y = {0,1,2,6,7,8}. Intersection size = 3 (positions 2,6,7). Union size = 7. **Jaccard = 3/7 ≈ 0.4286.**

(h) **Suppose the optimal separating hyperplane is given by 2x1 + 4x2 + x3 — 4 = O and the class labels are +1 and —I. For the training example (1, 0.5, I), what will be the class label and check is it a support vector?**

Solution: **SVM hyperplane / class label / support vector**
Hyperplane: $2x_1 + 4x_2 + x_3 - 4 = 0$. For example (1, 0.5, 1):
Compute $g = 2 \cdot 1 + 4 \cdot 0.5 + 1 \cdot 1 - 4 = 2 + 2 + 1 - 4 = 1$.

- Since $g > 0$, predicted class = **+1**.
- For a (hard-margin) SVM with margin scaled so support vectors satisfy $y(w \cdot x+b)=1$: here $y \cdot g = +1 \Rightarrow$ this point lies **on the margin**, so **it is a support vector**.

(i) **Draw flowchart of Genetic Algorithm.**

Solution:

□ Initialize population (random chromosomes).

□ Evaluate fitness of each individual.

□ Selection (pick parents by fitness).

□ Crossover (recombine parents → offspring).

□ Mutation (random small changes).

□ Replace population (form new generation).

□ If termination criterion met (max generations or target fitness) → stop; else go to step 2.

(j) **Given: TP=40, FP=IO, FN-5, TN=45. Calculate Fl score.**

Solution: **F1 score** (given TP=40, FP=10, FN=5, TN=45)

- Precision = TP/(TP+FP) = 40/50 = 0.80.
- Recall = TP/(TP+FN) = 40/45 ≈ 0.8889.
- **F1 = 2·(P·R)/(P+R) ≈ 0.8421.**

## SECTION-B

2. (a) Explain the concept of hierarchy in data warehousing. The Global Mart, a multinational retail company that sells products both online and in physical stores. The management wants to design a data warehouse to analyze business performance across sales, customers, products, and regions. Attributes:
Sales 112 Product ID, Customer ID, store ID, Date ID,
Revenue, Quantity Sold, Discount.

(Tables: Sales, Product, Customer, Store, Date)

Design three possible schema models (Star Schema, Snowflake Schema, Galaxy Schema)

i. Provide labeled diagrams for each.

ii. Explain the structure, characteristics, and use cases of each schema.

Solution: A **hierarchy** in a data warehouse is an ordered arrangement of dimension attributes that represent levels of detail from **lowest to highest**. Hierarchies support OLAP operations such as **drill-down** (move to more detailed level) and **roll-up** (aggregate to higher level).

**Examples of hierarchies in a retail warehouse:**

- **Date hierarchy:** Day → Month → Quarter → Year
- **Product hierarchy:** Product → Subcategory → Category
- **Geographical hierarchy:** City → State → Country
  These hierarchies help users analyse data at different granularities.

Given business attributes:
**Sales_ID, Product_ID, Customer_ID, Store_ID, Date_ID, Revenue, Quantity_Sold, Discount**

---

(i) Star Schema

A **Star Schema** has a central **Fact Table** connected to denormalized **Dimension Tables**.

**Fact Table (Sales):**

- Sales_ID, Date_ID, Product_ID, Customer_ID, Store_ID, Revenue, Quantity_Sold, Discount

**Dimension Tables:**

- **Product:** Product_ID, Product_Name, Subcategory, Category, Brand
- **Customer:** Customer_ID, Name, Age_Group, City, State, Country
- **Store:** Store_ID, Store_Name, Region, Zone
- **Date:** Date_ID, Day, Month, Quarter, Year

**Features:**

- Simple design

- Fast query performance
- Suitable for OLAP reporting

---

### (ii) Snowflake Schema

A **Snowflake Schema** normalizes the dimension tables to reduce redundancy.

**Example normalization of Product dimension:**

- **Product(Product_ID, Product_Name, Subcategory_ID)**
- **Subcategory(Subcategory_ID, Subcategory_Name, Category_ID)**
- **Category(Category_ID, Category_Name)**

**Fact table remains same as Star Schema.**

**Features:**

- Reduced redundancy
- More complex joins
- Saves storage

---

### (iii) Galaxy Schema (Fact Constellation)

A **Galaxy Schema** contains **multiple fact tables** that share common dimension tables.

**Examples:**

- **Sales Fact:** Sales_ID, Date_ID, Product_ID, Store_ID, Customer_ID, Revenue
- **Returns Fact:** Return_ID, Date_ID, Product_ID, Store_ID, Customer_ID, Units_Returned
- **Inventory Fact:** Inv_ID, Date_ID, Product_ID, Store_ID, Stock_Level

**Shared Dimension Tables:** Product, Store, Date, Customer

**Features:**

- Supports multiple business processes
- Uses conformed dimensions
- Suitable for large enterprise warehouses

(b) Discuss at least two techniques that overcome the limitations of the Apriori algorithm. Consider the given transaction database. Find-out the frequent item sets and strong association rules with minimum support 50% and minimum confidence 75 % using FP-Growth tree.

| Transaction ID | Items Purchased |
|---|---|
| | Pen, Notebook, Sta ler |
| | Pen, Notebook |
| | Pen. Sta ler, Marker |
| | Notebook. Sta ler, Envelo |
| | Pen, Notebook, Sta ler_ Envelo e |
| | Notebook, Sta ler |

Solution: Overcoming Apriori limitations & FP-Growth solution

**Limitations of Apriori:** expensive candidate generation, multiple database scans, costly for large itemsets.

**Two techniques that overcome these:**

1. **FP-Growth:** builds an FP-tree (compact prefix tree) and extracts frequent itemsets without candidate generation; scans DB only twice.
2. **ECLAT (vertical format):** uses item-tid lists (vertical representation) and performs intersections to find frequent itemsets — avoids candidate explosion.

**Given transactions (T1–T6)** (from paper):
T1: Pen, Notebook, Stapler
T2: Pen, Notebook

T3: Pen, Stapler, Marker
T4: Notebook, Stapler, Envelope
T5: Pen, Notebook, Stapler, Envelope
T6: Notebook, Stapler

There are 6 transactions → **min support 50%** ⇒ support threshold = 3 transactions. Compute frequent itemsets (using FP-concept):

- **Singleton supports:** Notebook=5, Stapler=5, Pen=4, Envelope=2, Marker=1 → frequent singletons: Notebook(5), Stapler(5), Pen(4).
- **Frequent pairs (support ≥3):**
  - (Notebook, Stapler): support = 4
  - (Notebook, Pen): support = 3
  - (Pen, Stapler): support = 3
- No triplet reaches support ≥3 (Pen+Notebook+Stapler has support 2).

**Strong association rules with min confidence 75% (confidence = support(X∪Y)/support(X))**:

- Notebook → Stapler : support 4, support(Notebook)=5 → confidence = 4/5 = **0.80** (strong).
- Stapler → Notebook : 4/5 = **0.80** (strong).
- Pen → Notebook : 3/4 = **0.75** (meets threshold).
- Pen → Stapler : 3/4 = **0.75** (meets threshold).

3. (a**) Why k-NN is called as Lazzy Learner? Consider a set [51 of five training examples given as ((Xi, YD, CD values, where and are the two attribute values (positive integers) and Ci is the binary class label: {((l, l), -1),**

   ((1, 7), +1), ((3, 3), +1), ((5, 4), −1), ((2, 5), −1)}.

   **Classify a test example at coordinates (3, 6) using a kNN classifier with k 3 and Manhattan distance defined by d((u, v), (p, q)) = Iu — pl + Iv ql.**

Solution:

k-NN (lazy learner) and classify (3,6) with k=3 using Manhattan distance

**Why k-NN is called a lazy learner:** it stores training data and waits till prediction time to compute nearest neighbors — no explicit training phase (no model parameter fitting).

**Training examples (xi, yi, ci):**
((1,1), -1), ((1,7), +1), ((3,3), +1), ((5,4), -1), ((2,5), -1).

Compute Manhattan distances to test (3,6):

- to (1,1): |3−1|+|6−1| = 2+5 = 7
- to (1,7): 2+1 = 3
- to (3,3): 0+3 = 3
- to (5,4): 2+2 = 4
- to (2,5): 1+1 = 2

The 3 nearest neighbors: (2,5, −1) distance 2 ; (1,7, +1) distance 3 ; (3,3, +1) distance 3. Votes: +1 (2 votes), −1 (1 vote). **Classify test point as +1.**

(b) **For the given set of objects, apply k-means clustering method to find the final two clusters and their centers assuming initial cluster centers to be Al and A3.**

| OWects | Al | A2 | | | | |
|---|---|---|---|---|---|---|
| | 2 | 2 | 8 | 5 | 7 | 6 |
| | 10 | 5 | 4 | 8 | 5 | 4 |

Solution:

k-means clustering for given points (A1...A6) and initial centers A1 and A3

Points :

- A1 = (2,10)
- A2 = (2,5)
- A3 = (8,4)
- A4 = (5,8)
- A5 = (7,5)
- A6 = (6,4)

Initial centers: A1=(2,10) and A3=(8,4).

**Run k-means (Euclidean)** — after convergence clusters are:

- **Cluster 1:** A1 (2,10), A2 (2,5), A4 (5,8)
  - Center ≈ (3.0, 7.6667)
- **Cluster 2:** A3 (8,4), A5 (7,5), A6 (6,4)
  - Center ≈ (7.0, 4.3333)

4.(a) What is Naive Bayes Classifier? Consider the following [5 ] hypothetical data concerning student characteristics and whether or not each student should be hired.

| Name | | GPA | ffort | Hirable |
|---|---|---|---|---|
| Sarah | | oor | lots | No |
| Dana | | average | some | No |
| | | verage | some | Yes |
| Annie | | average | Ots | Yes |
| mily | | excellent | Ots | 0 |
| ete | | excellent | Ots | |
| Ohn | | excellent | Ots | |
| ath | | | some | |

Use a Naive Bayes classifier to determine whether or not someone with excellent attendance, poor GPA, and lots of effort should be hired.

Solution: Naive Bayes example

Training set (extracted from the paper table):

- Sarah: GPA=poor, Effort=lots → Hirable=Yes
- Dana: average, some → No
- Alex: average, some → No
- Annie: average, lots → Yes
- Emily: excellent, lots → Yes
- Pete: excellent, lots → No
- John: excellent, lots → No
- Kathy: poor, some → No

We are asked: **Should someone with excellent attendance, poor GPA, and lots of effort be hired?**

**Assumption:** the question text mentions “excellent attendance, poor GPA, lots of effort.” The provided table contains **GPA** and **Effort**, and the nearest matching attribute for “excellent attendance” is taken here as **Effort=lots** (so we use GPA=poor and Effort=lots). If you want an alternate interpretation (attendance as a distinct attribute), give me that attribute values and I’ll recompute.

Compute priors and likelihoods (no smoothing):

- Prior P(Hirable=Yes) = 3/8 (Sarah, Annie, Emily).
- Prior P(Hirable=No) = 5/8.

Likelihoods given **Hirable=Yes** (3 examples):

- P(GPA=poor | Yes) = 1/3 (Sarah)
- P(Effort=lots | Yes) = 3/3 = 1

Likelihoods given **Hirable=No** (5 examples):

- P(GPA=poor | No) = 1/5 (Kathy)
- P(Effort=lots | No) = 2/5 (Pete, John)

Posterior (unnormalized):

- Score(Yes) = P(Yes) * P(GPA=poor|Yes) * P(Effort=lots|Yes) = (3/8)*(1/3)*1 = 1/8 = 0.125
- Score(No) = P(No) * P(GPA=poor|No) * P(Effort=lots|No) = (5/8)*(1/5)*(2/5) = 0.05

Since Score(Yes) > Score(No), **predict: Hirable = YES** for the described person.

(b) **How to handle missing values in dataset? Discuss the [5] different type of missing values with a suitable examples.**

Solution: **Types of missingness:**

- **MCAR (Missing Completely at Random):** missingness unrelated to data.
- **MAR (Missing at Random):** missingness depends on observed data.
- **MNAR (Not at Random):** missingness depends on unobserved values.

**Techniques to handle missing values:**

- **Deletion:** listwise or pairwise deletion (simple but may bias or reduce data).
- **Simple imputation:** mean/median for numeric, mode for categorical (fast, but underestimates variance).
- **KNN imputation:** use nearest neighbors to impute based on similar rows.
- **Regression imputation:** predict missing value using regression on other features.
- **Multiple imputation:** create multiple plausible completed datasets and combine (accounts for uncertainty).
- **Model-based methods:** EM algorithm, or integrated approaches in algorithms (e.g., tree methods that handle missing values).

**Example:** If age missing for some patients: mean imputation (if MCAR) or regression on other covariates if MAR; for categorical like occupation, use mode or predictive model.

5.(a) Predict the sales based on the expenditure on advertisement using simple linear regression. Calculate MSE and MAE

| X S end in Lakhs (Advertisin | Y (Sales in Lakhs) |
|---|---|
| | 1.8 |
| 2 | 2.4 |
| 3 | 3.2 |
| 4 | 3.8 |
| 5 | 4.5 |

Solution: Simple linear regression — predict sales from advertising spend

Data (X: advertising in ₹ Lakhs, Y: sales in ₹ Lakhs):
(1, 1.8), (2, 2.4), (3, 3.2), (4, 3.8), (5, 4.5).

Compute least-squares regression:

- **Fitted line:** $\hat{y} = 1.10 + 0.68 \cdot X$ (rounded: intercept = 1.10, slope = 0.68).
- Predictions $\hat{y}$ at X=1..5: [1.78, 2.46, 3.14, 3.82, 4.50].
- **MSE** = mean squared error = **0.0016** (very small)
- **MAE** = mean absolute error = **0.0320**

(b)Why do we use DBSCAN? Apply DBSCAN clustering [51 algorithm to cluster the following points with epsilon—I .5 and minimum points=2.

| Point | x | |
|---|---|---|
| | 1.0 | 1.0 |
| | 1.2 | 1.1 |
| c | 0.8 | 1.2 |
| | 5.0 | 5.0 |
| | 9.0 | 9.0 |

Solution: **Why use DBSCAN:** finds clusters of arbitrary shape, robust to noise, does not require pre-specifying number of clusters (uses density parameters eps and minPts).

**Apply DBSCAN ($\varepsilon = 1.5$, minPts = 2)** to points:
A(1.0, 1.0), B(1.2, 1.1), C(0.8, 1.2), D(5.0, 5.0), E(9.0, 9.0).

- A, B, C are within ε of each other and form a dense region → **Cluster 1 = {A, B, C}**.
  (Each of A, B, C has at least one neighbor within 1.5; density reachable → same cluster.)

- D and E are isolated (no other points within ε), and minPts=2 → **D and E are noise** (or singleton clusters if minPts=1). So DBSCAN yields one cluster {A,B,C} and treats D,E as noise.

6(a) Predict whether a customer will subscribe to a mobile[51 plan based on this following factors using Decision tree algorithm.

I. Carry out the first split using information gain Il. Next level onwards carry out the decision tree construction using heuristic method. Ill. Infer all the decision rules.

| Customer | Age Grou | Data Usa e | Contract | Subscribed |
|---|---|---|---|---|
|  | Youn | Hi h | Pre aid | Yes |
| 2 | Youn | Low | Pos aid | No |
| 3 | Middle | Medium | Pre aid | Yes |
| 4 | Senior | Low | Pre aid | No |
| 5 | Senior |  | Post aid | Yes |
| 6 | Middle | Hi h | Pos aid | Yes |

Solution:

- Target entropy H(S) ≈ 0.9183 (4 Yes, 2 No).
- **Info gain(Data Usage)** is highest (≈ 0.9183) because splits produce pure or nearly pure nodes:
    - High → all Yes (pure)
    - Low → all No (pure)
    - Medium → single Yes

      So **first split = Data Usage**.

**Resulting tree (simple):**

- Root: **Data Usage**
    - High → Subscribed = **Yes**
    - Medium → Subscribed = **Yes**

- Low → Subscribed = **No**

**Decision rules:**

- If Data Usage = High or Medium → Subscribe = Yes.
- If Data Usage = Low → Subscribe = No.

(b)Cluster the following data points with agglomerative clustering with average linkage.

| Points | x | |
|---|---|---|
| | 2 | 3 |
| | 3 | 4 |
| c | 5 | 8 |
| | 6 | 9 |
| | 8 | 10 |

Solution:

- A = (2,3)
- B = (3,4)
- C = (5,8)
- D = (6,9)
- E = (8,10)

Perform average-link agglomerative clustering (stepwise merges using average distance):

**Merging steps (average linkage):**

1. Merge **A** & **B** (closest): distance ≈ 1.414. → cluster {A,B}
2. Merge **C** & **D** (closest): distance ≈ 1.414. → cluster {C,D}
3. Merge **E** with {C,D}: average distance ≈ 2.921 → cluster {C,D,E}

4. Merge {A,B} and {C,D,E} last.

If you stop at **2 clusters**, you get:

- **Cluster1** = {A,B} and **Cluster2** = {C,D,E}.