IV-B.Tech I-Semester I-Mid Term Examinations, August 2025
Key / Scheme of Evaluation

Subject: Machine Learning
Subject Code: 127CX **Max. Marks: 30**

Q.No. 1	Question & Answer	Marks
a.	Define a well-posed problem with one example. A problem is well-posed if: its performance at tasks ***T***, as measured by ***P***, improves with experience ***E***. A Handwriting Recognition Learning Problem: ■ **Task *T*:** recognizing and classifying handwritten words within images ■ **Performance measure *P***: percent of words correctly classified ■ **Training experience *E***: a database of handwritten words with given classifications	1
b.	List types of learning in ML. Supervised, Unsupervised, Semi-supervised, Reinforcement learning.	1
c.	State one design issue in ML. Overfitting or underfitting while training a model.	1
d.	What is a maximally specific hypothesis? A hypothesis that is consistent with training data and cannot be made more specific without contradicting the data.	1
e.	What is a version space in concept learning? The set of all hypotheses that are consistent with the training examples.	1
f.	What are different impurity measures used in decision trees? Gain, Gain Ratio, Gini Gain	1
g.	What is a perceptron? A simple linear binary classifier that maps input features to an output using weights, bias, and activation function.	1
h.	What is an activation function? A function applied to the weighted sum of inputs in a neural network neuron to introduce non-linearity. Examples: Sigmoid, ReLU, Tanh.	1
i.	Give definition of Support Vector Machine (SVM). A supervised ML algorithm that finds the optimal hyperplane separating data points of different classes with maximum margin.	1
j.	What is a linear classifier? A classifier that separates classes using a linear decision boundary of the form: $w^T x + b = 0$.	1
Q.2(a)	Explain the steps involved in designing a learning system. 1. Problem Description 2. Choosing the Training Experience 3. Choosing the Target Function 4. Choosing a Representation for the Target Function 5. Choosing a Function Approximation Algorithm 6. Final Design Brief explanation of each step.	3
(b)	Differentiate between supervised, unsupervised learning with example. **Supervised:** Learns mapping from input to output using labeled data. Example: spam detection. **Unsupervised:** Finds patterns in unlabeled data. Example: customer segmentation	2
Q.3(a)	Discuss concept learning task along with general to specific ordering of hypothesis. **Concept learning task:** Inferring a Boolean-valued function from training examples. **General-to-specific ordering:** Hypotheses ordered such that general hypotheses cover more examples, while specific hypotheses cover fewer.	2

(b) Explain the Candidate Elimination Algorithm. Show how the S and G boundaries update for a given small training set. — 3

Maintains S (specific boundary) and G (general boundary).

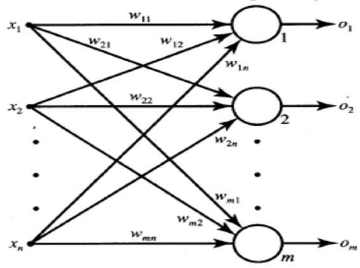
- **Positive example:** Generalize S minimally, restrict G.
- **Negative example:** Specialize G minimally, restrict S.

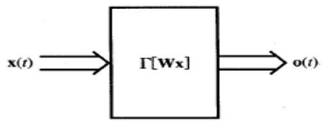
Considering EnjoySport dataset:

Example	Sky	AirTemp	Humidity	Wind	EnjoySport
1	Sunny	Warm	Normal	Strong	Yes
2	Sunny	Warm	High	Strong	Yes
3	Rainy	Cold	High	Strong	No
4	Sunny	Warm	High	Strong	Yes

Start: S = <Ø,Ø,Ø,Ø>, G = <?,?,?,?>
After 1st positive: S = <Sunny, Warm, Normal, Strong>
After 2nd positive: S = <Sunny, Warm, ?, Strong>
After negative: G specialized to exclude negative instances
Final: S = <Sunny, Warm, ?, Strong>

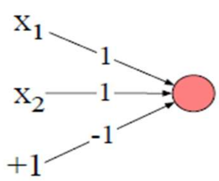
Q.5(a) Draw and explain the architecture of a single-layer feed-forward neural network. — 2



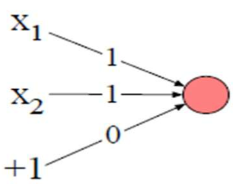


- A single-layer feed-forward neural network connects input features directly to output neurons through weighted connections, where each weight indicates the importance of an input.
- Each neuron computes a weighted sum of inputs plus bias, which is then passed through an activation function (step, sigmoid, ReLU, tanh) to generate its output.
- The output layer consists of one or more neurons that provide the network's final predictions (class labels or continuous values).
- Data flows strictly forward from inputs to outputs with no feedback loops, hence the name *feed-forward*.

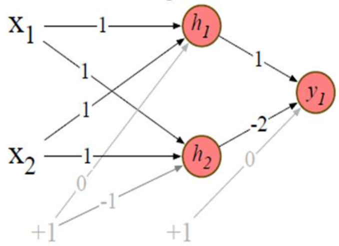
(b) Show how a perceptron can implement logic gates (AND / OR / XOR) with neat diagrams. — 3



AND



OR



XOR

AND

x1	x2	y
0	0	0
0	1	0
1	0	0
1	1	1

OR

x1	x2	y
0	0	0
0	1	1
1	0	1
1	1	1

XOR

x1	x2	y
0	0	0
0	1	1
1	0	1
1	1	0

$$y = \begin{cases} 0, & \text{if } \mathbf{w}\cdot\mathbf{x}+b \leq 0 \\ 1, & \text{if } \mathbf{w}\cdot\mathbf{x}+b > 0 \end{cases}$$

Q.4 Consider the training examples given below: **5**

Weekend	Weather	Parents	Money	Decision
W1.	Sunny	Yes	Rich	Cinema
W2	Sunny	No	Rich	Tennis
W3	Windy	Yes	Rich	Cinema
W4	Rainy	Yes	Poor	Cinema
W5	Rainy	No	Rich	Stay-in
W6	Rainy	Yes	Poor	Cinema
W7	Windy	No	Poor	Cinema
W8	Windy	No	Rich	Shopping
W9	Windy	Yes	Rich	Cinema
W10	Sunny	No	Rich	Tennis

i. What is the best attribute for the root node child?
ii. Using ID3 algorithm construct decision tree.

i. Target classes and counts:
Cinema = 6, Tennis = 2, Stay-in = 1, Shopping = 1.
H(S) = $-\sum p_c \log_2 p_c \approx 1.57095$ bits
1) Weather (values: Sunny, Windy, Rainy)

- Sunny (3) → {Tennis 2, Cinema 1} H = 0.91830
- Windy (4) → {Cinema 3, Shopping 1} H = 0.81128
- Rainy (3) → {Cinema 2, Stay-in 1} H = 0.91830

H(S|Weather) = 0.87549
IG(Weather) = 1.57095 − 0.87549 = 0.69546
2) Parents (values: Yes, No)

- **Yes (5)** → {Cinema 5} → H = 0
- **No (5)** → {Tennis 2, Stay-in 1, Cinema 1, Shopping 1} H = 1.92193

H(S|Parents) = 0.96096
IG(Parents) = 1.57095 − 0.96096 = 0.60999
3) Money (values: Rich, Poor)

- **Rich (7)** → {Cinema 3, Tennis 2, Stay-in 1, Shopping 1} H = 1.84237
- **Poor (3)** → {Cinema 3} → H = 0

H(S|Money) = 1.28966
IG(Money) = 1.57095 − 1.28966 = 0.28129

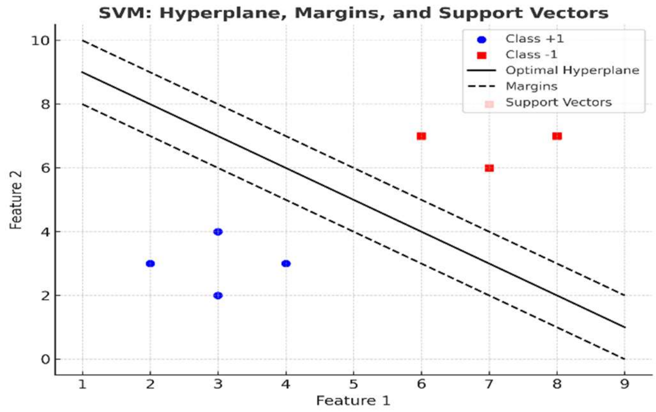
Weather (IG =0.69546) is the best root.

ii. Grow the tree recursively
Split on Weather and compute the best attribute inside each branch using the remaining attributes {Parents, Money}.

```
                 Weather
             /          |         \
        Rainy       Sunny     Windy
       / \               /   \            /  \
   Parents?        Parents?       Parents?
   /       \            /        \            /      \
Yes      No      Yes       No      Yes       No
Cinema Stay-in Cinema Tennis Cinema  Money?
                                                  /    \
                                            Poor     Rich
                                            Cinema Shopping
```

Q.6(a) Explain the role of support vectors and margin in Support Vector Machines (SVMs). Illustrate with a neat sketch. **[3]**

- **Support vectors** are the training points lying on margin boundaries; they alone define the optimal hyperplane.
- **Margin** is the distance from the hyperplane to the closest points; SVM maximizes this margin → better generalization.



(b) List four applications of SVMs. **[2]**

- Text categorization / spam filtering (high-dimensional sparse features).
- Image classification / handwritten digit recognition.
- Bioinformatics (protein/DNA classification).
- Sentiment analysis, fault detection, medical diagnosis

Q.7(a) Apply the Find-S algorithm on the EnjoySport dataset given below and determine the most specific hypothesis. **[2]**

Example	Sky	AirTemp	Humidity	Wind	EnjoySport
1	Sunny	Warm	Normal	Strong	Yes
2	Sunny	Warm	High	Strong	Yes
3	Rainy	Cold	High	Strong	No
4	Sunny	Warm	High	Strong	Yes

Start S=⟨∅,∅,∅,∅⟩.

1. (Sunny, Warm, Normal, Strong) **Yes** → S = ⟨Sunny, Warm, Normal, Strong⟩
2. (Sunny, Warm, High, Strong) **Yes** → mismatch at Humidity ⇒ generalize ⇒ S = ⟨Sunny, Warm, ?, Strong⟩
3. (Rainy, Cold, High, Strong) **No** → ignored by Find-S (negatives don't change S).
4. (Sunny, Warm, High, Strong) **Yes** → already covered; S unchanged.

Final most specific hypothesis:
S = ⟨Sunny, Warm, ?, Strong⟩.

(b) Discuss the issues in decision tree learning. **[3]**

- **Overfitting:** trees grow deep and memorize noise. Remedies: pre-pruning (min samples per leaf, max depth), post-pruning (cost-complexity), cross-validation, MDL penalties.
- **Attribute selection bias:** information gain favors many-valued attributes. Remedy: Gain Ratio (C4.5) or Gini.
- **Continuous attributes:** need thresholding (choose split that maximizes IG/Gini).
- **Missing values:** surrogate splits or fractional instance assignment.
- **Class imbalance:** class-weighted splits or resampling.
- **Computational cost:** choose efficient split scans, feature subsampling (Random Forests).
- **Stability/interpretability trade-off:** ensembles vs single interpretable tree.