IV-B.Tech I-Semester II-Mid Term Examinations, November 2025
Key

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

Q.No. 1	Question & Answer	Marks
a.	**What is a support vector in SVM?** Support vectors are the data points that lie closest to the decision boundary (hyperplane) in Support Vector Machines.	**1**
b.	**Define Bayes optimal classifier.** Bayes optimal classifier is the best possible classifier that makes predictions by selecting the hypothesis with the maximum posterior probability.	**1**
c.	**Expand PAC learning.** PAC stands for Probably Approximately Correct learning.	**1**
d.	**Define sample complexity.** Sample complexity represents the minimum number of samples needed to ensure that the learned hypothesis has error at most ε with probability at least (1-δ).	**1**
e.	**Write the distance formula used in K-Nearest Neighbor algorithm.** Euclidean Distance: $d(x, y) = \sqrt{[\Sigma(xi - yi)^2]}$ For 2D: $d = \sqrt{[(x_1-x_2)^2 + (y_1-y_2)^2]}$	**1**
f.	**What is the curse of dimensionality in K-NN?** The curse of dimensionality refers to the problem that as the number of features (dimensions) increases, the volume of the feature space increases exponentially, making data sparse.	**1**
g.	**What is chromosome representation in genetic algorithms?** Chromosome representation is the encoding of a solution (individual) in a genetic algorithm. For example, Binary string (e.g., 11010101).	**1**
h.	**Define population in the context of genetic algorithms.** Population is a set of individuals (chromosomes) that represent possible solutions to the optimization problem in a given generation.	**1**
i.	**What is fitness function?** A fitness function is an objective function used to evaluate how good or fit a solution (chromosome) is in solving the given problem.	**1**
j.	**What is the purpose of fitness function in genetic algorithms?** The fitness function serves to: (1) Evaluate the quality of each solution, (2) Guide the selection process by identifying better solutions, (3) Determine which individuals should reproduce, (4) Drive the evolution towards optimal or near-optimal solutions, (5) Provide stopping criteria when desired fitness is achieved.	**1**
Q.2(a)	**What is a kernel function in SVM?** Kernel function enables SVM to handle non-linearly separable data by implicitly mapping input features to higher dimensions. Common kernel functions: • Linear kernel: $K(x,y) = x \cdot y$ • Polynomial kernel: $K(x,y) = (x \cdot y + c)^d$ • RBF/Gaussian kernel: $K(x,y) = \exp(-\gamma\|x-y\|^2)$ • Sigmoid kernel: $K(x,y) = \tanh(\alpha x \cdot y + c)$	**1**

(b)	**Using the dataset below, classify the test instance (Student = Yes, Credit Score = Good, Income = Low) with Naïve Bayes. Show probability calculations.**	**4**

Student	Credit Score	Income	Buy?
No	Good	High	No
No	Good	Low	Yes
Yes	Good	Low	Yes
Yes	Bad	Low	No
No	Bad	Low	No
Yes	Good	High	Yes
No	Bad	High	No
Yes	Bad	High	Yes
Yes	Good	Low	Yes
No	Good	High	No

Prior Probabilities:

P(Buy = Yes) = 5/10 = 0.5

P(Buy = No) = 5/10 = 0.5

Conditional Probabilities:

Feature	Value	P(Value|Buy=Yes)	P(Value|Buy=No)
Student	Yes	4/5 = 0.8	1/5 = 0.2
Student	No	1/5 = 0.2	4/5 = 0.8
Credit Score	Good	4/5 = 0.8	2/5 = 0.4
Credit Score	Bad	1/5 = 0.2	3/5 = 0.6
Income	Low	3/5 = 0.6	2/5 = 0.4
Income	High	2/5 = 0.4	3/5 = 0.6

Test Instance: Student = Yes, Credit Score = Good, Income = Low

Naive Bayes Formula: P(Buy|Features) = P(Buy) × P(Student|Buy) × P(Credit|Buy) × P(Income|Buy)

For Buy = Yes:

P(Buy=Yes|Test) = P(Buy=Yes) × P(Student=Yes|Buy=Yes) × P(Credit=Good|Buy=Yes) × P(Income=Low|Buy=Yes) = 0.5 × 0.8 × 0.8 × 0.6 = 0.192

For Buy = No:

P(Buy=No|Test) = P(Buy=No) × P(Student=Yes|Buy=No) × P(Credit=Good|Buy=No) × P(Income=Low|Buy=No) = 0.5 × 0.2 × 0.4 × 0.4 = 0.016

Since P(Buy=Yes|Test) > P(Buy=No|Test), **Buy = YES**

Q.3(a)	**State PAC hypothesis and discuss sample complexity for finite hypothesis spaces.**	**3**

A concept class C is PAC-learnable if there exists an algorithm that, for any target concept c in C, any distribution D over instances, and parameters ε (error) and δ (confidence), will output a hypothesis h such that: With probability at least (1-δ): error(h) ≤ ε.

Sample Complexity for Finite Hypothesis Spaces: Sample complexity is the number of training examples m required to learn successfully.

$m \geq (1/\varepsilon)[\ln|H| + \ln(1/\delta)]$

Where:

- |H| = size of hypothesis space
- ε = error parameter (how approximately correct)
- δ = confidence parameter (how probably correct)

(b)	**Explain the mistake-bound model of learning.**	**2**

The mistake-bound model is an online learning framework where:

- Learning proceeds in trials/rounds
- In each trial, learner receives an instance, makes a prediction, then receives true label
- If prediction is wrong, it is counted as a mistake and learner updates hypothesis
- Goal: Minimize total number of mistakes made during learning

Mistake bound: Maximum number of mistakes an algorithm makes before learning the target concept.

Q.4(a)	**Explain K-nearest neighbor algorithm with an example.** K-NN is a lazy, instance-based learning algorithm that classifies new instances based on the majority class of their K nearest neighbors. **Algorithm Steps:** • Choose value of K (number of neighbors) • Calculate distance between test instance and all training instances • Select K instances with minimum distance • For classification: Assign majority class among K neighbors • For regression: Assign average value of K neighbors **Example:** **Training Data:** • Point A(1,1) - Class: Red • Point B(2,2) - Class: Red • Point C(6,6) - Class: Blue • Point D(7,7) - Class: Blue **Test Point: P(3,3), K=3** **Distance Calculations (Euclidean):** • $d(P,A) = \sqrt{[(3-1)^2 + (3-1)^2]} = \sqrt{8} = 2.83$ • $d(P,B) = \sqrt{[(3-2)^2 + (3-2)^2]} = \sqrt{2} = 1.41$ • $d(P,C) = \sqrt{[(3-6)^2 + (3-6)^2]} = \sqrt{18} = 4.24$ • $d(P,D) = \sqrt{[(3-7)^2 + (3-7)^2]} = \sqrt{32} = 5.66$ 3 Nearest Neighbors: **B(Red), A(Red), C(Blue)** Majority class: **Red** Prediction: **P** is classified as **Red**	**3**
(b)	**Discuss locally weighted regression with an example.** LWR is a non-parametric regression method that fits a different simple model at each query point using only nearby training data. Points closer to the query point are given higher weights. **Example:** **Scenario:** Predicting house price at location x=5 **Training data:** (x, price) • (1, 100K), (3, 150K), (4, 180K), (7, 250K), (9, 300K) **For query point x=5 with τ=2:** • Points near x=5 (like x=3,4,7) get higher weights • Points far from x=5 (like x=1,9) get lower weights • Fit weighted linear regression: price = $w_0 + w_1 \times x$ • Use this local model only to predict at x=5	**2**
Q.5(a)	**Discuss hypothesis space search in Genetic Algorithms.** Genetic Algorithms perform hypothesis space search through evolutionary process: **Population-based Search:** • Multiple hypotheses (population) searched simultaneously • Each chromosome represents a point in hypothesis space • Parallel exploration of different regions **Search Mechanism:** • Selection: Fitter hypotheses have higher probability of reproduction • Crossover: Combines parts of good hypotheses to create new ones • Mutation: Introduces random changes for exploration • Successive generations move toward better regions of hypothesis space **Advantages:** • Less likely to get stuck in local optima • Explores multiple promising regions simultaneously • No gradient information needed • Robust to noisy fitness evaluations	**2**

(b) **Write a short note on the following terms:** **3**

(i) Selection (ii) Crossover (iii) Mutation

(i) Selection: Selection is the process of choosing individuals from the population for reproduction based on their fitness values.

Types of Selection:

- Roulette Wheel Selection: Probability proportional to fitness
- Tournament Selection: Select best from random subset
- Rank-based Selection: Based on fitness rank rather than absolute value
- Elitism: Best individuals directly copied to next generation

Purpose: Ensure fitter individuals have higher chance of passing genes to next generation.

(ii) Crossover: Crossover (or recombination) combines genetic material from two parent chromosomes to create offspring.

Types of Crossover:

- Single-point: Split at one point, exchange tails
- Parent1: 11010|101, Parent2: 00110|010
- Child1: 11010|010, Child2: 00110|101
- Two-point: Split at two points, exchange middle
- Uniform: Each bit randomly selected from either parent

Purpose: Combine good traits from parents; exploitation of good solutions.

(iii) Mutation: Mutation introduces random changes to individual chromosomes to maintain genetic diversity.

Types of Mutation:

- Bit-flip: Flip binary bit (0→1 or 1→0)
- Before: 11010101
- After: 11011101 (bit 5 flipped)
- Swap: Exchange positions of two genes
- Inversion: Reverse substring

Q.7(a) **Distinguish between lazy learning and eager learning.** **3**

Aspect	**Lazy Learning**	**Eager Learning**
Training Phase	No explicit training; simply stores training data	Builds explicit model during training
Prediction Time	Computationally expensive; needs to search training data	Fast prediction using pre-built model
Memory	High memory requirement; stores all training data	Low memory; only stores model parameters
Model Updates	Easy to add new data incrementally	Requires retraining for new data
Examples	K-NN, Case-Based Reasoning, Locally Weighted Regression	Decision Trees, Neural Networks, Naive Bayes, SVM
Adaptability	Naturally adapts to local data distributions	Global model may not capture local patterns
Interpretability	Less interpretable (depends on neighbors)	More interpretable (has explicit rules/structure)
When to Use	Complex decision boundaries, local patterns, incremental learning	Need fast predictions, limited memory, interpretable models

(b)	Discuss parallelizing Genetic Algorithm in detail. Parallelizing Genetic Algorithms exploits the inherently parallel nature of GA to improve performance. **Why Parallelize GA?** • Fitness evaluation can be done independently for each individual • Speeds up computation for large populations • Maintains multiple subpopulations exploring different regions • Reduces premature convergence **Parallel GA Models:** a) Master-Slave Model (Global Parallelization): b) Island Model (Coarse-Grained): c) Cellular Model (Fine-Grained): d) Hybrid Models: **Advantages:** • Faster convergence through parallel evaluation • Better exploration through multiple populations • Reduced risk of premature convergence • Scalable to very large problems	**2**
Q.6	**Consider two strings as initial population in a GA:** **String 1: 11101001000** **String 2: 00001010101** **Generate all possible offspring using crossover and mutation operators.** Given: String 1 (P1): 11101001000; String 2 (P2): 00001010101 **CROSSOVER OPERATIONS** 1. Single-Point Crossover at Position 3: • P1: 111|01001000 • P2: 000|01010101 • → Offspring C1: 111|01010101 • → Offspring C2: 000|01001000 2. Single-Point Crossover at Position 6: • P1: 111010|01000 • P2: 000010|10101 • → Offspring C3: 111010|10101 • → Offspring C4: 000010|01000 3. Two-Point Crossover at Positions 3 and 7: • P1: 111|0100|1000 • P2: 000|0101|0101 • → Offspring C5: 111|0101|1000 (swap middle) • → Offspring C6: 000|0100|0101 (swap middle) 4. Uniform Crossover (example): • Mask: 11001010110 • P1: 11101001000 • P2: 00001010101 • → C7: 11001001100 (1=from P1, 0=from P2) **MUTATION OPERATIONS** 5. Bit-Flip Mutation on C1: Original C1: 11101010101 • Mutate at position 4: 11111010101 (0→1) • Mutate at position 8: 11101010001 (1→0) 6. Multiple Bit-Flip Mutations on C3: Original C3: 11101010101 • Mutate positions 2,5,9: 10100010001 7. Mutation on C5: Original C5: 11101011000 • Mutate at position 10: 11101011001	**5**