Department of Computer Science & Engineering
Mid Semester Examination
Monsoon Semester, Session: 2025-2026
Subject: Machine Learning (NCSD519)

Time: 1 Hours
Max. Marks: 30

Instructions: Answer the questions as per the instruction within each section**.**

Sl. No.	Section-I (20 Marks: Attempt All Question)	Marks
1.	**(a)** A hospital wants to predict whether a patient has diabetes (1) or not (0) based on their fasting blood sugar level (in mg/dL). A logistic regression model was built using the following equation: $$P(\text{Diabetes} = 1) = \frac{1}{1 + e^{-(-7+0.08\times X)}}$$ Where X is the fasting blood sugar level. Solution: (i) Calculate the probability that a patient with a fasting blood sugar level of 100 mg/dL has diabetes. Substitute the value X=100 in logistic regression model developed: So, the probability is approximately 0.731. [1 Marks] (ii) Based on a threshold of 0.5, will this patient be classified as diabetic? Since the probability (0.731) is greater than 0.5, the patient will be classified as diabetic (1). [1 Marks] (ii) At what fasting blood sugar level is the predicted probability of diabetes exactly 0.5? Set $P = 0.5$, solve the equation: $$0.5 = \frac{1}{1 + e^{-(-7+0.08X)}}$$ $$\Rightarrow 1 + e^{-(-7+0.08X)} = 2 \Rightarrow e^{7-0.08X} = 1 \Rightarrow 7 - 0.08X = 0 \Rightarrow X = \frac{7}{0.08} = 87.5$$ So, the threshold blood sugar level is **87.5 mg/dL** [2 Marks] (iv) Calculate the log-odds (logit) for X = 95 mg/dL Logit= log-odds = Z Hence Logit= $-7+0.08\times95=-7+7.6=0.6$ [2 Marks]	

(b) A startup wants to predict monthly sales (in $1000s) based on online advertising spend (in $1000s). The dataset contains 3 points: 4

Advertising Spend (x)	Actual Sales
1	3
2	5
3	7

The model is: $\hat{y}=w\times x+b$

Initially, parameters are w=0, b=0, and learning rate α=0.1. Perform one gradient descent update to find new *w* and b.

Solution:-

Step 1 — write the gradient formulas

For MSE (with factor $1/n$) the partial derivatives are

$$\frac{\partial J}{\partial w} = -\frac{2}{n}\sum_{i=1}^{n} x_i(y_i - \hat{y}_i), \qquad \frac{\partial J}{\partial b} = -\frac{2}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i).$$

Step 2 — evaluate residuals at the initial parameters

Initial $\hat{y}_i = wx_i + b = 0 \cdot x_i + 0 = 0$. So residuals $r_i = y_i - \hat{y}_i = y_i$.

Compute the needed sums (do the small multiplications explicitly):

- $x_1y_1 = 1 \cdot 3 = 3$.
- $x_2y_2 = 2 \cdot 5 = 10$.
- $x_3y_3 = 3 \cdot 7 = 21$.

Now add: $3 + 10 = 13$. Then $13 + 21 = 34$.
So $\sum_i x_iy_i = 34$.

Also $\sum_i y_i = 3 + 5 + 7$.
Compute $3 + 5 = 8$. Then $8 + 7 = 15$. So $\sum_i y_i = 15$.

Step 3 — compute the gradients

Plug into the formulas with $n = 3$ and $\hat{y}_i = 0$:

$$\frac{\partial J}{\partial w} = -\frac{2}{3} \cdot \sum_i x_iy_i = -\frac{2}{3} \cdot 34 = -\frac{68}{3}.$$

Compute $\frac{68}{3}$ as a mixed number: $68 \div 3 = 22$ remainder 2, so $68/3 = 22\frac{2}{3} = 22.666666\ldots$
Hence $\dfrac{\partial J}{\partial w} = -22.\overline{6}$.

$$\frac{\partial J}{\partial b} = -\frac{2}{3} \cdot \sum_i y_i = -\frac{2}{3} \cdot 15 = -\frac{30}{3} = -10.$$

So the gradients are: $\nabla_w = -\frac{68}{3}$ and $\nabla_b = -10$.

Step 4 — apply the gradient descent update

Update rule: parameter ← parameter − α × gradient.

For w:

$$w_{\text{new}} = 0 - 0.1 \cdot \left(-\frac{68}{3}\right) = 0.1 \cdot \frac{68}{3}.$$

Compute $0.1 \cdot 68 = 6.8$. Then $6.8 \div 3 = 2 + \frac{0.8}{3} = 2 + 0.266666\ldots = 2.266666\ldots$
As an exact fraction: $0.1 \cdot \frac{68}{3} = \frac{68}{30} = \frac{34}{15}$.

So $\boxed{w_{\text{new}} = \frac{34}{15} \approx 2.2666667.}$

For b:

$$b_{\text{new}} = 0 - 0.1 \cdot (-10) = 0.1 \cdot 10 = 1.$$

So $\boxed{b_{\text{new}} = 1.}$

[2x2=4 Marks]

2. **(a)**A robot needs to navigate inside a warehouse from its starting point to a destination while avoiding obstacles (like shelves, boxes, and walls). The developers collect a dataset of past navigation example where:
Inputs: Robot's sensor readings (e.g. distance to obstacles, current position, orientation)
Output: Correct action taken (e.g. move forward, turn left, turn right , stop)

Solution:

(i) Explain how this navigation task can be formulated as supervised learning problem.

The following steps used.

1.Creation of labelled training dataset from past sensor reading as per following:

- Features (Input): The robot's sensor readings such as distances to obstacles, current position, and orientation. These form the feature vector representing the current state or observation.
- Output (Target Label) : The correct action the robot should take in that state (move forward, turn left, turn right, stop).

[1 Marks]

2. Supervised learning formulation:

To learn, a mapping function f from sensor readings (input) to actions (output) using the dataset of past navigation examples (input-output pairs), the model learns to predict the appropriate action given new sensor readings.

[1 Marks]

(ii) Nature of the problem: regression or classification?

The output is one of several **discrete actions** (move forward, turn left, turn right, stop).This means it's a **multiclass classification problem**, where the goal is to classify sensor readings into one of the multiple predefined action categories.

[1 Marks]

(iii) Which supervised model is suitable and why?

- **Suitable models:** (Any one of the following)
  - **Decision Trees / Random Forests:** Easy to interpret, handle nonlinear relationships well, and robust to noisy sensor data.
  - **Support Vector Machines (SVM) with nonlinear kernels:** Good for complex boundaries between action classes but might be computationally heavier.
  - **Neural Networks (e.g., Multi-layer Perceptron):** Very powerful in modeling complex nonlinear mappings, especially if you have a large dataset.
  - **K-Nearest Neighbors (KNN):** Simple, but might struggle with high-dimensional data or large datasets. [1 Marks]
- **Why:**
  The relationship between sensor inputs and the correct navigation action is **nonlinear and complex** due to the spatial configuration of obstacles and robot dynamics. Hence, models capable of learning **nonlinear decision boundaries** work best are suited in this case. [1 Marks]

(b) Consider the following use cases and mention what type of machine learning technique to be used.

(i) A mall wants to divide its customers into groups based on their shopping behaviour. They do not have predefined categories or labels for the customer.

Type of ML Technique : Unsupervised Learning [1 Marks]

(i) A self-driving car learns to navigate by interacting with environment, steering, braking and receiving rewards (safe driving) or penalties.

Type of ML Technique : Reinforcement Learning [1 Marks]

(ii) A streaming platform suggests movies to users by analyzing what similar users have watched.

Type of ML Technique : Associative Analysis/ Unsupervised Learning

[1 Marks]

(c) Define machine learning model Underfititng and Overfitting case in terms of bias/variance perspective.

Underfititng : High Bias, Low Variance

Overfitting : Low Bias, High Variance [1 x2 =2 Marks]

Section-II (10 Marks : Attempt Any ONE Questions)

3. We will use the dataset below to learn a decision tree which predicts if people pass artificial intelligence (Yes or No), based on their previous GPA (High, Medium, or Low) and whether or not they studied.

GPA	Studied	Passed
L	F	No
L	T	Yes
M	F	No
M	T	Yes
H	F	Yes
H	T	Yes

Solution:

(i)

Entropy(Passed) = 0.9812 [1 Marks]

Entropy(T)= $-(3/3)\log_2(3/3) = 0$

Entropy(F)= $-(1/3)\log_2(1/3) - (2/3)\log_2(2/3) = 0.9183$

GPA :

Entropy(S_L)= 1

Entropy(S_M)= 1

Entropy(S_H)= 0

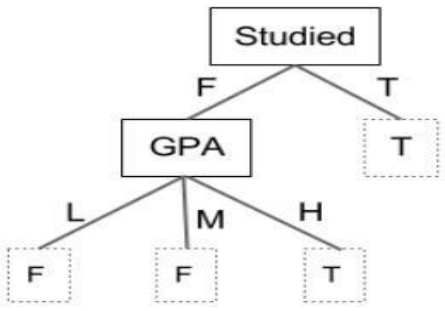
Gain(Passed, Studied)= 0.459145 [1 Marks]

Gain(Passed, GPA)= 0.25162 [1 Marks]

The attribute "Studied" will be selected as root of tree because of having higher information gain with respect to the target class. [1 Marks]

Decision Tree:-

[2 Marks]



[1x4=4 Marks, each leaf node drawing]

Note : Only 1 marks awarded for decision tree drawing in case of any error in part (i)

4. (i)Why is Support Vector Machine (SVM) often effective when dealing with datasets that have a large number of features (high-dimensional space)?

SVM works well in high-dimensional spaces because it focuses on maximizing the margin between classes using only a subset of critical points called support vectors. This means it does not rely on the entire dataset.

[2 Marks]

ii) What is the effect of a small C (regularization on the SVM decision boundary in case of soft margin? How does it affect Overfitting and Underfititng?

A small C allows better generalization since focus on margin maximization

OR

Allows more misclassification, i.e. margin violations, encouraging a wider margin and more tolerance for misclassified points in training.

[1 Marks]

Impact on Underfititng/Overfitting:

This can reduce Overfitting by allowing a simpler decision boundary by focusing on margin maximization, potentially improving generalization [1 Marks]

Risking Underfititng [1 Marks]

Why do we need semi-supervised learning? Mention basic steps involved in self-training semi supervised learning algorithm.

Any one of the following :
(i)Data labelling is expensive and difficult OR Labelling is often unreliable
[1 Marks]

Unlabeled examples easy to obtain in large numbers [1 Marks]

Self traing steps:

- Train the model on the small labeled dataset: Start by training a supervised model on the labeled examples you have.
- Use the trained model to predict labels for the unlabeled data:Run the model on unlabeled data and get predicted labels with confidence scores (like probabilities).
- Select the most confident predictions: Choose unlabelled samples where the model is most confident about the predicted label (above some confidence threshold).
- Add these confidently labeled samples to the labeled set. Treat those confident predictions as if they were true labels.
- Retrain the model on this expanded labeled set. Train again with the original labeled data plus these new pseudo-labeled samples.
- Repeat above steps iteratively. Gradually the model improves as it "teaches itself" using more data.

[0.5 x 6=3 Marks]