SPRING **MID SEMESTER EXAMINATION-2025**

School of Computer Engineering
Kalinga Institute of Industrial Technology, Deemed to be University
Machine Learning
[CS31002]

Time: 1 1/2 Hours **Full Mark: 20**

Answer Any four questions including question No.1 which is compulsory.
The figures in the margin indicate full marks.
Candidates are required to give their answers in their own words as far as practicable and all parts of a question should be answered at one place only.

1. Answer all the questions. [1 Mark X 5]

a) A dataset has an independent variable X = [2, 3, 5, 7] and a dependent variable Y = [3, 6, 9, 14]. Using simple linear regression, compute the slope (β_1) and intercept (β_0) of the best-fit line.

Sol:

To compute the slope (β_1) and intercept (β_0) for simple linear regression, we use the formulas:

$$\beta_1 = \frac{\sum(X_i - \bar{X})(Y_i - \bar{Y})}{\sum(X_i - \bar{X})^2}$$

$$\beta_0 = \bar{Y} - \beta_1 \bar{X}$$

- Slope (β_1) = 2.1
- Intercept (β_0) = -0.925

So, the best-fit line equation is:

$$Y = 2.1X - 0.925$$

b) Which normalization technique scales values to the range [0, 1] by default? Provide its formula.

Sol: The normalization technique that scales values to the range [0,1] by default is Min-Max Normalization.

$$X' = \frac{X - X_{\min}}{X_{\max} - X_{\min}}$$

c) What is the objective of using the kernel trick in an SVM classifier?

Sol:

The objective of using the kernel trick in an SVM classifier is to transform non-linearly separable data into a higher-dimensional space where it becomes linearly separable, without explicitly computing the transformation. This allows SVM to efficiently find a decision boundary in complex nonlinerly separable datasets.

d) Show that the derivative of the logistic function g(x) is g(x)(1 - g(x)).

Sol:

$$\frac{d}{dx}g(x) = \frac{d}{dx}\left(\frac{1}{1+e^{-x}}\right)$$

Let $u = 1$ and $v = 1 + e^{-x}$, then:

$$\frac{d}{dx}g(x) = \frac{(1+e^{-x})\cdot 0 - 1\cdot(-e^{-x})}{(1+e^{-x})^2}$$

$$= \frac{e^{-x}}{(1+e^{-x})^2}$$

Simplify the above equation:

$$\frac{d}{dx}g(x) = \frac{(1-g(x))g(x)}{1/g(x)^2}$$

$$= g(x)(1-g(x))$$

e) Calculate Accuracy, Precision, Recall and F1 score from the following confusion matrix.

Actual Class\Predicted class	cancer = yes	cancer = no	Total
cancer = yes	**90**	**210**	300
cancer = no	**140**	**9560**	9700
Total	230	9770	10000

Sol:
Accuracy = (TP + TN)/All = (90+9560)/10000 = 96.50%
Error rate = (FP + FN)/All = (140 + 210)/10000 = 3.50%
Precision = TP/(TP + FP) = 90/(90 + 140) = 90/230 = 39.13%
Recall = TP/ (TP + FN) = 90/(90 + 210) = 90/300 = 30.00%
F1 = 2 P × R /(P + R) = 2 × 39.13% × 30.00%/(39.13% + 30%) = 33.96%

2. (a) Parameter k in k-NN algorithm could be a very large value or a very small value. Give the drawbacks, if any, of each choice.

Sol:

i) Very Small k (e.g., k = 1 or 2):

- High variance & sensitivity to noise: Since predictions rely on very few nearest neighbors, the model is highly sensitive to outliers and noisy data.
- Overfitting: The decision boundary becomes too complex, capturing random fluctuations in the training data rather than general patterns.

ii) Very Large k:

- Bias increases (Underfitting): The model becomes too generalized, as it considers a large number of neighbors. It may fail to capture class distinctions in the data.
- Computational inefficiency: A large k increases computation time, as distances to all points need to be computed and sorted.

(b) Consider the two-class classification task that consists of the following data points:
Class 1: $[10, 5]^T$, $[12, 5]^T$, $[15, 8]^T$

Class 2: $[6.5\ 11]^T$, $[7, 15]^T$, $[8, 10]^T$
Using *k*-NN classifier with k=3 and Euclidean distance, predict the test data point $[16, 8]^T$ belongs to which class. [1+4 Marks]
Sol:

1) Compute Euclidean Distances from every data points to test data point.
2) Sort the distnces and find the 3 Nearest Neighbors.

Distance	Point	Class
1.00	(15,8)	Class 1
5.00	(12,5)	Class 1
6.71	(10,5)	Class 1
8.25	(8,10)	Class 2
9.96	(6.5,11)	Class 2
11.40	(7,15)	Class 2

3) The three nearest neighbors are:
   (15,8) → Class 1
   (12,5) → Class 1
   (10,5) → Class 1
4) Since all three neighbors belong to Class 1, the test point (16,8) is classified as Class1.

3. What are Ridge regression and LASSO regression techniques?
(a) How are they different from (ordinary) linear regression?
(b) How is LASSO different from Ridge regression? [5 Marks]
Sol:
Ridge Regression and LASSO (Least Absolute Shrinkage and Selection Operator) are two regularization techniques used in linear regression to prevent overfitting by adding a penalty term to the loss function.
a) Ordinary Least Squares (OLS) regression minimizes the sum of squared residuals to estimate the best-fit line. However, when the dataset has multicollinearity (high correlation between features) or high-dimensional data, OLS can overfit the data, leading to poor generalization.
Ridge and LASSO differ from OLS by introducing a penalty term to the cost function to shrink coefficients and improve model generalization.

1. **Ordinary Linear Regression Cost Function:**

$$J(\theta) = \sum_{i=1}^{n} (y_i - X_i\theta)^2$$

- No regularization is applied, leading to possible overfitting.

2. **Ridge Regression Cost Function (L2 Regularization):**

$$J(\theta) = \sum_{i=1}^{n} (y_i - X_i\theta)^2 + \lambda \sum_{j=1}^{p} \theta_j^2$$

- Adds **L2 norm penalty** (sum of squared coefficients).
- Reduces large coefficients but does **not** set them to zero.
- Helps when features are correlated.

3. **LASSO Regression Cost Function (L1 Regularization):**

$$J(\theta) = \sum_{i=1}^{n}(y_i - X_i\theta)^2 + \lambda \sum_{j=1}^{p} |\theta_j|$$

- Adds **L1 norm penalty** (sum of absolute values of coefficients).
- **Can set some coefficients to zero**, effectively performing **feature selection**.

b) How is LASSO Different from Ridge Regression?

Feature	Ridge Regression	LASSO Regression
Penalty Type	L2 Norm ($\sum \theta_j^2$)	L1 Norm ((\sum
Effect on Coefficients	Shrinks but does **not** set coefficients to zero	Shrinks and **can set some coefficients to exactly zero**
Feature Selection?	**No**, keeps all features	**Yes**, selects only the most relevant features
Best Use Case	When many features contribute to the prediction	When only a few features are important
Behavior with Correlated Features	Distributes weights across correlated features	Selects one feature and ignores the others

4. You are given the following dataset

Color	Legs	Height	Smelly	Species
White	3	Short	Yes	M
Green	2	Tall	No	M
Green	3	Short	Yes	M
White	3	Short	Yes	M
Green	2	Short	No	H
White	2	Tall	No	H
White	2	Tall	No	H
White	2	Short	Yes	H

Use naive bayes to check species for test data X = {Color=Green, Legs=2, Height=Tall, Smelly=No}.

P(Species=M)=4/8=0.5

P(Species=H)=4/8=0.5

P(Color=White/Species=M)=2/4=0.5

P(Color=White/Species=H)=¾=0.75

P(Color=Green/Species=M)=2/4=0.5

P(Color=Green/Species=H)=¼=0.25

P(Legs=2/Species=M)=1/4=0.25

P(Legs=2/Species=H)=4/4=1

P(Legs=3/Species=M)=3/4=0.75

P(Legs=3/Species=H)=0/4=0

P(Height=Tall/Species=M)=3/4=0.75

P(Height=Tall/Species=H)=2/4=0.5

P(Height=Short/Species=M)=1/4=0.25

P(Height=Short/Species=H)=2/4=0.5

P(Smelly=Yes/Species=M)=3/4=0.75

P(Smelly=Yes/Species=H)=1/4=0.25

P(Smelly=No/Species=M)=1/4=0.25

P(Smelly=No/Species=H)=3/4=0.75

P(M/X)=P(Species=M)*P(Color=Green/Species=M)*P(Legs=2/Species=M)*P(Height=Tall/Species=M)*P(Smelly=No/Species=M)

=0.5*0.5*0.25*0.75*0.25

=0.0117

P(H/X)=P(Species=H)*P(Color=Green/Species=H)*P(Legs=2/Species=H)*P(Height=Tall/Species=H)*P(Smelly=No/Species=H)

=0.5*0.25*1*0.5*0.75

=0.0468

So, the probability of X belonging to Species M is 0.0117 and that to Species H is 0.0468. Hence, we will assign the entity X with attributes {Color=Green, Legs=2, Height=Tall, Smelly=No} to species H.

[5 Marks]

5. (a) Describe One-Against-All (OAA) and One-Against-One (OAO) in classification algorithm.

Sol:

i) One-Against-All (OAA) / One-Vs-All (OvA):

In OAA, for a problem with K classes, we train K binary classifiers. Each classifier is trained to distinguish one class vs. all other classes. The class with the highest confidence score is selected for a new input. Requires fewer models (K classifiers for K classes).

ii) One-Against-One (OAO) / One-Vs-One (OvO):

In OAO, for K classes, we train K(K-1)/2 binary classifiers. Each classifier distinguishes between two classes at a time. The class with the most "votes" across all pairwise comparisons is chosen. Requires training more models (K(K-1)/2 classifiers). Computationally expensive for a large number of classes.

(b) For the optimization of the separating hyperplane for linearly separable patterns, formulate the primal and the dual problems. [1+4 Marks]

Sol:

1. Primal Formulation

Let the training dataset be:

$$\{(x_i, y_i) \mid i = 1, 2, \ldots, n\}, \quad y_i \in \{-1, +1\}, \quad x_i \in \mathbb{R}^p$$

where x_i are feature vectors and y_i are class labels.

The goal is to find a **hyperplane** defined by:

$$w^T x + b = 0$$

where w is the weight vector and b is the bias term.

To ensure correct classification with a **maximum margin**, the constraints are:

$$y_i(w^T x_i + b) \geq 1, \quad \forall i$$

The **margin** is given by $\frac{2}{\|w\|}$, so maximizing the margin is equivalent to minimizing $\frac{1}{2}\|w\|^2$.

Thus, the **primal optimization problem** is:

$$\min_{w,b} \quad \frac{1}{2}\|w\|^2$$

subject to:

$$y_i(w^T x_i + b) \geq 1, \quad \forall i$$

2. Dual Formulation

To solve the **primal problem**, we use **Lagrange multipliers** $\alpha_i \geq 0$ to enforce the constraints. The **Lagrangian function** is:

$$L(w, b, \alpha) = \frac{1}{2}\|w\|^2 - \sum_{i=1}^{n} \alpha_i [y_i(w^T x_i + b) - 1]$$

Step 1: Compute the Dual Form

- Taking the derivative of L with respect to w and setting it to zero:

$$\frac{\partial L}{\partial w} = w - \sum_{i=1}^{n} \alpha_i y_i x_i = 0$$

$$w = \sum_{i=1}^{n} \alpha_i y_i x_i$$

- Taking the derivative of L with respect to b and setting it to zero:

$$\frac{\partial L}{\partial b} = -\sum_{i=1}^{n} \alpha_i y_i = 0$$

Substituting w into L, the **dual problem** is:

$$\max_{\alpha} \quad \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j x_i^T x_j$$

subject to:

$$\sum_{i=1}^{n} \alpha_i y_i = 0, \quad \alpha_i \geq 0, \forall i$$

*** Best of Luck ***