INDIAN INSTITUTE OF TECHNOLOGY ROORKEE
End-Term Examination (ETE)
Machine Learning (CSN-382)

Spring Semester 2024-25

Time: 180 minutes **Total Marks: 100**

Instructions: Each problem has a relatively simple and straightforward solution, and we may deduct points for overly complex answers. Therefore, focus on providing clear and concise solutions that directly address the problem at hand.

……

Problem 1 (10 marks)

1. (a)

(Total confusion) The *confusion matrix* is a very useful tool for evaluating classification models. For a C-class problem, this is a $C \times C$ matrix that tells us, for any two classes $c, c' \in [C]$, how many instances of class c were classified as c' by the model. In the example below, $C = 2$, there were $P + Q + R + S$ points in the test set where P, Q, R, S are strictly positive integers. The matrix tells us that there were Q points that were in class $+1$ but (incorrectly) classified as -1 by the model, S points were in class -1 and were (correctly) classified as -1 by the model, etc. **Give expressions for the specified quantities in terms of** P, Q, R, S. No derivations needed. Note that y denotes the true class of a test point and $\hat{y}$ is the predicted class for that point. **(5 x 1 = 5 marks)**

		Predicted class $\hat{y}$	
		$+\mathbf{1}$	$-\mathbf{1}$
True class y	$+\mathbf{1}$	P	Q
	$-\mathbf{1}$	R	S

Confusion Matrix

Accuracy (**ACC**) $\mathbb{P}[\hat{y} = y]$

Precision (**PRE**) $\mathbb{P}[y = 1 | \hat{y} = 1]$

Recall (**REC**) $\mathbb{P}[\hat{y} = 1 | y = 1]$

False discovery rate (**FDR**) $\mathbb{P}[y = -1 | \hat{y} = 1]$

False omission rate (**FOR**) $\mathbb{P}[y = 1 | \hat{y} = -1]$

1. (b)

(Kernel Smash) Melbi has created two Mercer kernels $K_1, K_2 \colon \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ with the feature map for the kernel K_i being $\phi_i \colon \mathbb{R} \to \mathbb{R}^2$. Thus, for any $x, y \in \mathbb{R}$, we have $K_i(x, y) = \langle \phi_i(x), \phi_i(y) \rangle$ for $i \in \{1,2\}$. Melbi knows that $\phi_1(x) = (x, x^3)$ and $\phi_2(x) = (1, x^2)$. Melbo has created a new kernel K_3 using Melbi's kernels so that for any $x, y \in \mathbb{R}$, $K_3(x, y) = \left(K_1(x, y) + 3 \cdot K_2(x, y)\right)^2$. Design a feature map $\phi_3 \colon \mathbb{R} \to \mathbb{R}^7$ for the kernel K_3.

Note that $\boldsymbol{\phi_3}$ must not use more than 7 dimensions.

(5 marks)

$$\phi_3(x) = \mathbf{?}$$

Answer:

		Predicted class $\hat{y}$	
		$+1$	-1
True class y	$+1$	P	Q
	-1	R	S

Confusion Matrix

Accuracy (**ACC**) $\mathbb{P}[\hat{y} = y]$	$\frac{P+S}{P+Q+R+S}$
Precision (**PRE**) $\mathbb{P}[y = 1 \mid \hat{y} = 1]$	$\frac{P}{P+R}$
Recall (**REC**) $\mathbb{P}[\hat{y} = 1 \mid y = 1]$	$\frac{P}{P+Q}$
False discovery rate (**FDR**) $\mathbb{P}[y = -1 \mid \hat{y} = 1]$	$\frac{R}{P+R}$
False omission rate (**FOR**) $\mathbb{P}[y = 1 \mid \hat{y} = -1]$	$\frac{Q}{Q+S}$

$$\phi_3(x) = \left(\boxed{3}, \boxed{x\sqrt{6}}, \boxed{x^2\sqrt{19}}, \boxed{x^3\sqrt{12}}, \boxed{x^4\sqrt{11}}, \boxed{x^5\sqrt{6}}, \boxed{x^6}\right)$$

Problem 2 (10 marks = 5+5)

(Positive Linear Regression) We have data features $\mathbf{x}_1, \ldots, \mathbf{x}_N \in \mathbb{R}^D$ and labels $y_1, \ldots, y_N \in \mathbb{R}$ stylized as $X \in \mathbb{R}^{N \times D}, \mathbf{y} \in \mathbb{R}^N$. We wish to fit a linear model with positive coefficients:

$$\underset{\mathbf{w} \in \mathbb{R}^D,}{\operatorname{argmin}} \frac{1}{2} \|X\mathbf{w} - \mathbf{y}\|_2^2 \ \text{s.t.}\, w_j \geq 0 \text{ for all } j \in [D]$$

1. Write the Lagrangian for this problem by introducing dual variables (no derivation needed).
2. Simplify the dual problem (eliminate $\mathbf{w}$) – show major steps. Assume $X^\top X$ is invertible.

Answer:

Write down the Lagrangian here (you will need to introduce dual variables and give them names)

$$\mathcal{L}(\mathbf{w}, \boldsymbol{\alpha}) = \frac{1}{2} \|X\mathbf{w} - \mathbf{y}\|_2^2 - \boldsymbol{\alpha}^\top \mathbf{w}$$

which can be rewritten for convenience as

$$\mathcal{L}(\mathbf{w}, \boldsymbol{\alpha}) = \frac{1}{2} \mathbf{w}^\top X^\top X \mathbf{w} - \mathbf{w}^\top X^\top \mathbf{y} - \mathbf{w}^\top \boldsymbol{\alpha} + \frac{1}{2} \|\mathbf{y}\|_2^2$$

Derive and simplify the dual. Show major calculations steps.

The dual is $\max_{\boldsymbol{\alpha}\geq\mathbf{0}}\left\{\min_{\mathbf{w}}\{\mathcal{L}(\mathbf{w},\boldsymbol{\alpha})\}\right\}$. Solving the inner problem by applying first-order optimality (since it is an unconstrained problem) gives us $\frac{\partial\mathcal{L}}{\partial\mathbf{w}}=\mathbf{0}\Rightarrow X^\top(X\mathbf{w}-\mathbf{y})-\boldsymbol{\alpha}=\mathbf{0}$. Putting this in the Lagrangian and neglecting constant terms gives us

$$\min_{\boldsymbol{\alpha}\geq\mathbf{0}}\left\{\frac{1}{2}\boldsymbol{\alpha}^\top C\boldsymbol{\alpha}+\boldsymbol{\alpha}^\top\mathbf{s}\right\}$$

where $C=[c_{ij}]\overset{\text{def}}{=}(X^\top X)^{-1}\in\mathbb{R}^{D\times D}$ and $\mathbf{s}=[s_i]\overset{\text{def}}{=}CX^\top\mathbf{y}\in\mathbb{R}^D$.

Problem 3 (10 marks)

3. (a) (6 marks)

(Optimal DT) Melbo has a multiclass problem with three classes $+,\times,\square$. There are 16 datapoints in total, each with a 2D feature vector (x,y). x,y can take value 0 or 1. The table below describes each data point. All 16 points are at the root of a decision tree. Melbo wishes to learn a decision stump based on the entropy reduction principle to split this node into two children. Help Melbo finish this task. *Hint: take logs to base 2 so no need for calculator* 😊.

SNo	Class	(x,y)	SNo	Class	(x,y)	SNo	Class	(x,y)	SNo	Class	(x,y)
1	$+$	(0,1)	5	$+$	(0,1)	9	$\times$	(1,0)	13	$\square$	(1,0)
2	$+$	(1,1)	6	$+$	(0,1)	10	$\times$	(1,0)	14	$\square$	(0,0)
3	$+$	(0,1)	7	$+$	(1,1)	11	$\times$	(0,0)	15	$\square$	(1,0)
4	$+$	(1,1)	8	$+$	(1,1)	12	$\times$	(0,0)	16	$\square$	(0,0)

1. What is the entropy of the root node?

2. What is the entropy of the two child nodes (give answers for the two nodes separately) if the split is done using the x feature ($x=0$ becomes left child, $x=1$ becomes right child)?

3. What is the reduction in entropy (i.e., $H_{\text{root}}-H_{\text{children}}$) if the split is done using the x feature as described above?

4. What is the entropy of the two child nodes (give answers for the two nodes separately) if the split is done using the y feature ($y=0$ becomes left child, $y=1$ becomes right child)?

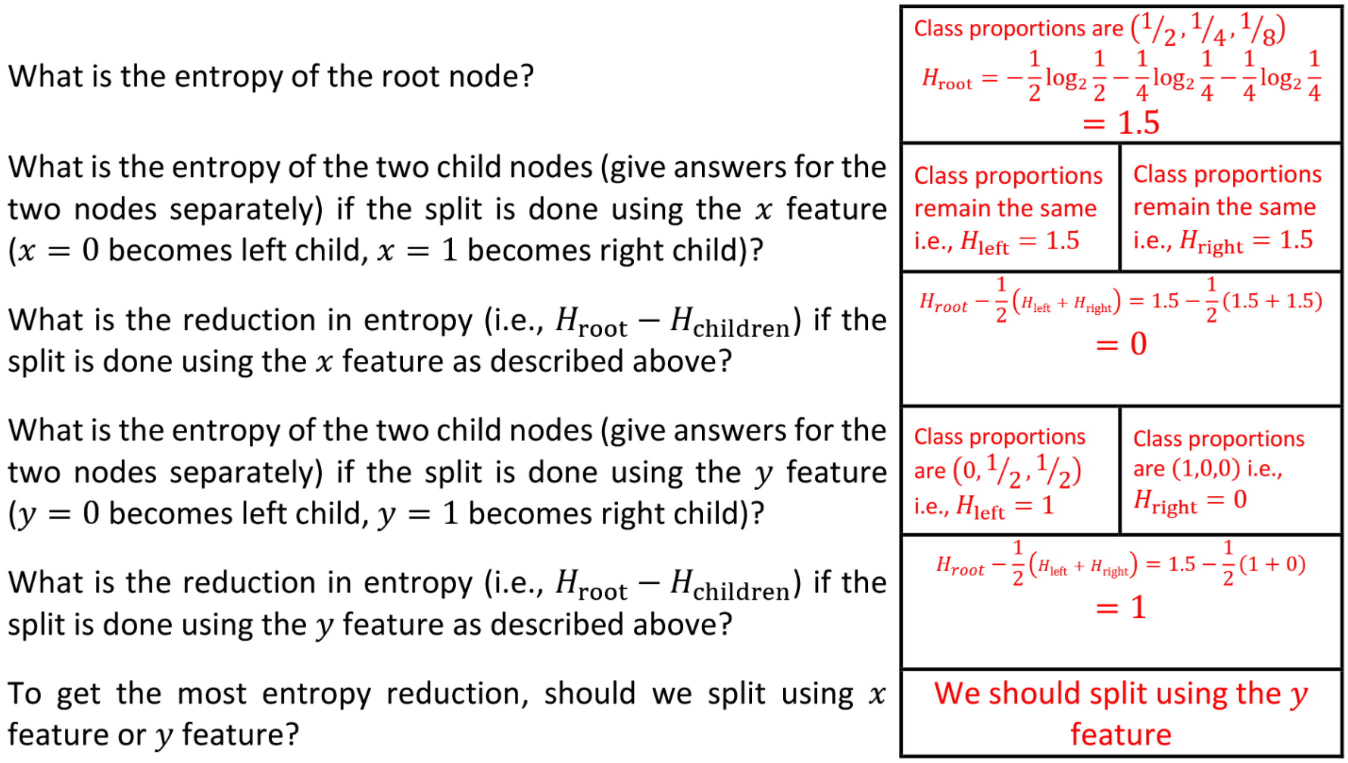
5. What is the reduction in entropy (i.e., $H_{\text{root}}-H_{\text{children}}$) if the split is done using the y feature as described above?

6. To get the most entropy reduction, should we split using x feature or y feature?

3. (b) (4 marks)

What is the role of the learning rate in gradient descent? What can go wrong if it is too high or too low?

Answer:



What is the entropy of the root node?	Class proportions are $({}^1\!/_2, {}^1\!/_4, {}^1\!/_8)$ $H_{\text{root}} = -\frac{1}{2}\log_2\frac{1}{2} - \frac{1}{4}\log_2\frac{1}{4} - \frac{1}{4}\log_2\frac{1}{4}$ $= 1.5$
What is the entropy of the two child nodes (give answers for the two nodes separately) if the split is done using the x feature ($x = 0$ becomes left child, $x = 1$ becomes right child)?	Class proportions remain the same i.e., $H_{\text{left}} = 1.5$ Class proportions remain the same i.e., $H_{\text{right}} = 1.5$
What is the reduction in entropy (i.e., $H_{\text{root}} - H_{\text{children}}$) if the split is done using the x feature as described above?	$H_{root} - \frac{1}{2}(H_{\text{left}} + H_{\text{right}}) = 1.5 - \frac{1}{2}(1.5 + 1.5)$ $= 0$
What is the entropy of the two child nodes (give answers for the two nodes separately) if the split is done using the y feature ($y = 0$ becomes left child, $y = 1$ becomes right child)?	Class proportions are $(0, {}^1\!/_2, {}^1\!/_2)$ i.e., $H_{\text{left}} = 1$ Class proportions are (1,0,0) i.e., $H_{\text{right}} = 0$
What is the reduction in entropy (i.e., $H_{\text{root}} - H_{\text{children}}$) if the split is done using the y feature as described above?	$H_{root} - \frac{1}{2}(H_{\text{left}} + H_{\text{right}}) = 1.5 - \frac{1}{2}(1 + 0)$ $= 1$
To get the most entropy reduction, should we split using x feature or y feature?	We should split using the y feature

Solution:

The learning rate (α) determines the step size in each iteration of gradient descent.

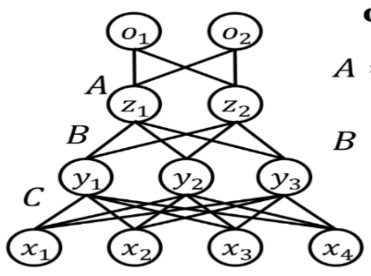
- If **too low**, the algorithm will converge very slowly, taking a long time to reach the minimum.
- If **too high**, the algorithm might overshoot the minimum or diverge, oscillating around or moving away from the optimal point.

An optimal learning rate balances convergence speed and stability.

Problem 4 (10 marks)

4. (a) (5 marks)

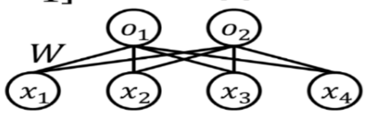
Consider the NN with 2 hidden layers – all nodes use the identity activation function. This NN is clearly equivalent to a network with no hidden layers since all activation functions are linear. Find the weights of this new network



$$\mathbf{o} = A\mathbf{z}, \mathbf{z} = B\mathbf{y}, \mathbf{y} = C\mathbf{x}, \qquad \mathbf{o} = W\mathbf{x}$$

$$A = \begin{bmatrix} 2 & 0 \\ 2 & 2 \end{bmatrix}, C = \begin{bmatrix} 1 & 3 & 0 & 1 \\ 4 & 1 & 4 & 0 \\ 0 & 2 & 0 & 2 \end{bmatrix}$$

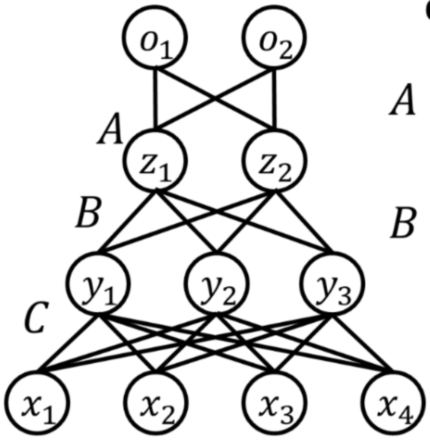
$$B = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 3 & 1 \end{bmatrix} \qquad W^{\top} = \begin{bmatrix} \\ ? \\ \\ \end{bmatrix}$$



4. (b) (5 marks)

Explain the structure and function of an artificial neural network (ANN). Describe the roles of weights, biases, and activation functions.

Answer:

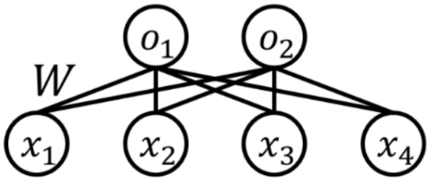


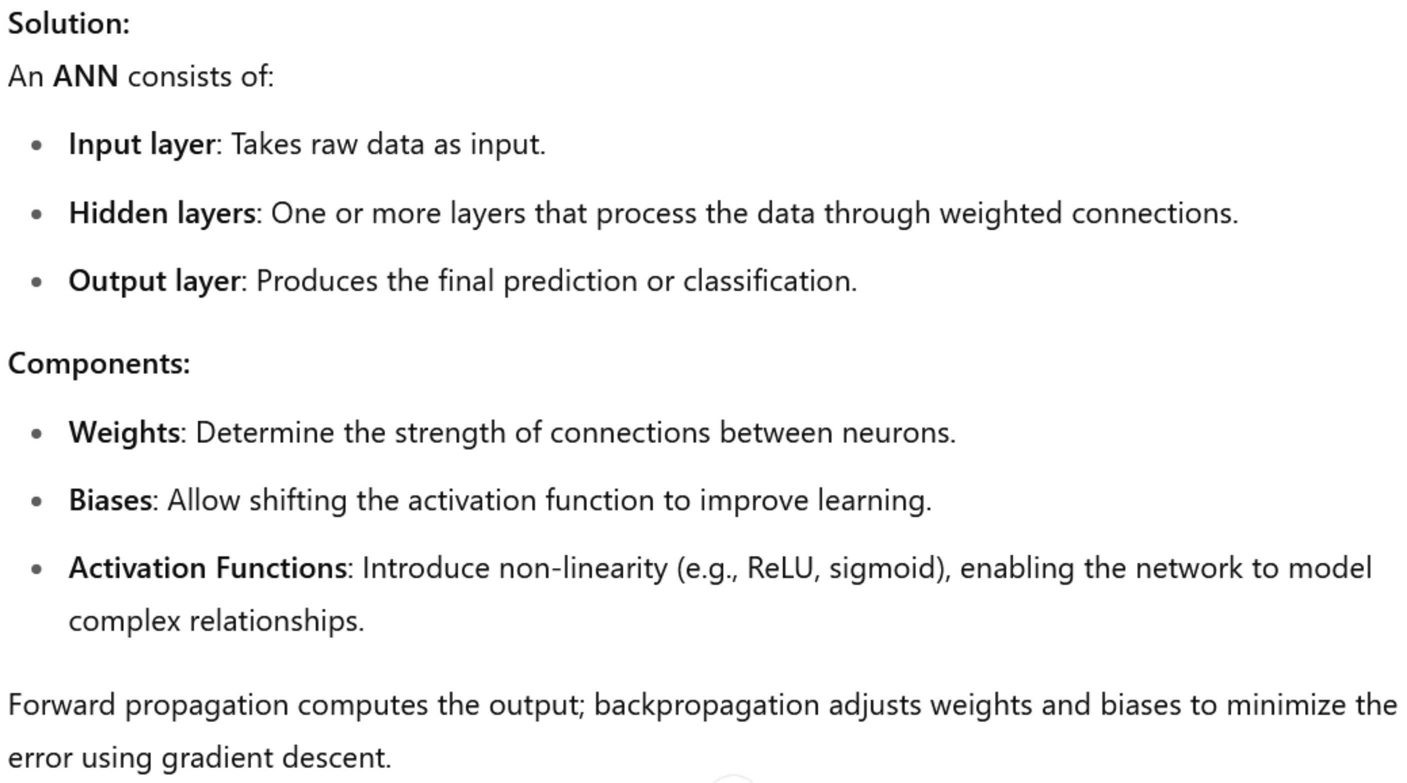
$$\mathbf{o} = A\mathbf{z}, \mathbf{z} = B\mathbf{y}, \mathbf{y} = C\mathbf{x}, \qquad \mathbf{o} = W\mathbf{x}$$

$$A = \begin{bmatrix} 2 & 0 \\ 2 & 2 \end{bmatrix}, C = \begin{bmatrix} 1 & 3 & 0 & 1 \\ 4 & 1 & 4 & 0 \\ 0 & 2 & 0 & 2 \end{bmatrix}$$

$$B = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 3 & 1 \end{bmatrix}$$

$$W^\top = \begin{bmatrix} 18 & 50 \\ 22 & 56 \\ 16 & 40 \\ 14 & 26 \end{bmatrix}$$





Solution:

An **ANN** consists of:

- **Input layer**: Takes raw data as input.
- **Hidden layers**: One or more layers that process the data through weighted connections.
- **Output layer**: Produces the final prediction or classification.

Components:

- **Weights**: Determine the strength of connections between neurons.
- **Biases**: Allow shifting the activation function to improve learning.
- **Activation Functions**: Introduce non-linearity (e.g., ReLU, sigmoid), enabling the network to model complex relationships.

Forward propagation computes the output; backpropagation adjusts weights and biases to minimize the error using gradient descent.

Problem 5 (10 marks) : Maximum likelihood

Consider the following probability distribution:

$$P_\theta(x) = 2\theta x e^{-\theta x^2}$$

where θ is a parameter and x is a positive real number. Suppose you get m i.i.d. samples x_i drawn from this distribution. Show how one can compute the maximum likelihood estimator for θ based on these samples.

Answer:

Solution: We with down the likelihood under the iid assumption:

$$L(D, \theta) = \prod_{i=1}^{m} P_\theta(x_i)$$

Taking the log, we geT:

$$\log L(D, \theta) = \sum_{i=1}^{m} \log P_\theta(x_i) = \sum_{i=1}^{m} (\log 2 + \log \theta + \log x_i - \theta x_i^2)$$

Taking the derivative wrt θ, we get:

$$\frac{\partial \log L(D, \theta)}{\partial \theta} = \sum_{i=1}^{m} \left(\frac{1}{\theta} - x_i^2 \right) = \frac{m}{\theta} - \sum_{i=1}^{m} x_i^2$$

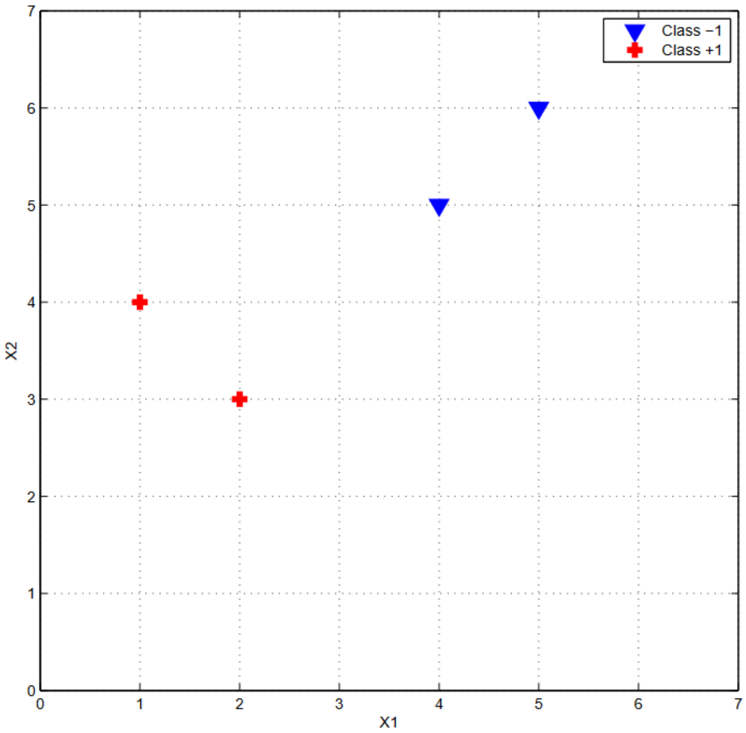
Setting this to 0 and solving for θ we get:

$$\theta = \frac{m}{\sum_{i=1}^{m} x_i^2}$$

Problem 6 (10 marks)

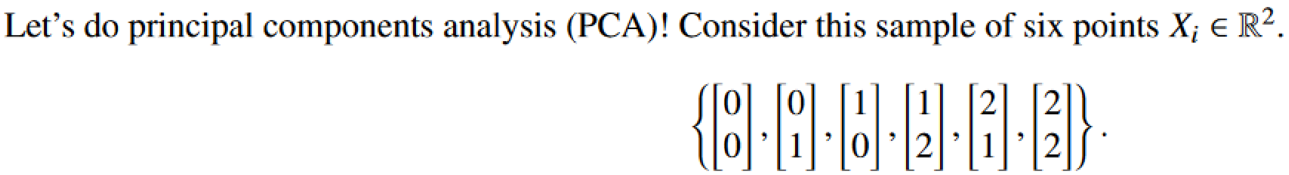
6. (a) (5 marks)

Support vector machines learn a decision boundary leading to the largest margin from both classes. You are training SVM on a tiny dataset with 4 points shown in the Figure. This dataset consists of two examples with class label +1 (denoted with plus), and two examples with class label -1 (denoted with triangles)



What's the equation corresponding to the decision boundary?

6. (b) (5 marks)

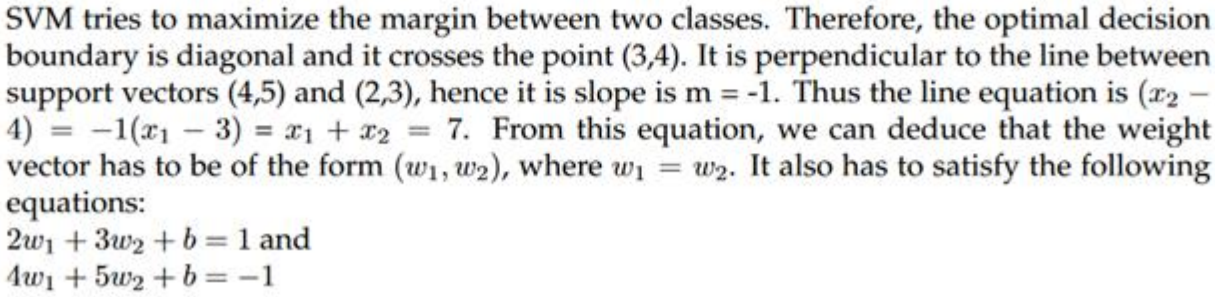

Let's do principal components analysis (PCA)! Consider this sample of six points $X_i \in \mathbb{R}^2$.

$$\left\{\begin{bmatrix}0\\0\end{bmatrix},\begin{bmatrix}0\\1\end{bmatrix},\begin{bmatrix}1\\0\end{bmatrix},\begin{bmatrix}1\\2\end{bmatrix},\begin{bmatrix}2\\1\end{bmatrix},\begin{bmatrix}2\\2\end{bmatrix}\right\}.$$

6.(b).1. [2 Marks] Compute the mean of the sample points and write the centered design matrix (By subtracting the mean from each sample).

6.(b).2. [3 Marks] Find all the principal components of this sample. Write them as unit vectors.

Answer:


SVM tries to maximize the margin between two classes. Therefore, the optimal decision boundary is diagonal and it crosses the point (3,4). It is perpendicular to the line between support vectors (4,5) and (2,3), hence it is slope is m = -1. Thus the line equation is $(x_2 - 4) = -1(x_1 - 3) = x_1 + x_2 = 7$. From this equation, we can deduce that the weight vector has to be of the form (w_1, w_2), where $w_1 = w_2$. It also has to satisfy the following equations:

$2w_1 + 3w_2 + b = 1$ and

$4w_1 + 5w_2 + b = -1$

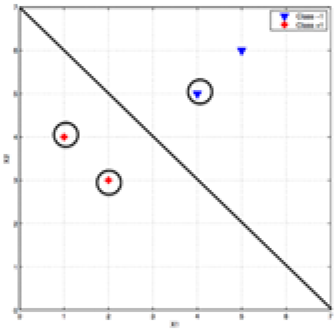
Hence $w_1 = w_2 = -1/2$ and $b = 7/2$

Decision boundary $\boldsymbol{w}^\top\boldsymbol{x}+b=0$

x1 + x2 = 7

Circle the support vectors and draw the decision boundary.

Solution:



Solution:

The sample mean is

$$\mu = \frac{1}{6}\sum_{i=1}^{6} X_i = \begin{bmatrix} 1 \\ 1 \end{bmatrix}.$$

By subtracting the mean from each sample, we form the centered design matrix

$$\dot{X} = \begin{bmatrix} -1 & -1 \\ -1 & 0 \\ 0 & -1 \\ 0 & 1 \\ 1 & 0 \\ 1 & 1 \end{bmatrix}.$$

The principal components of our dataset are the eigenvectors of the matrix

$$\dot{X}^\top \dot{X} = \begin{bmatrix} 4 & 2 \\ 2 & 4 \end{bmatrix}.$$

The characteristic polynomial of this symmetric matrix is

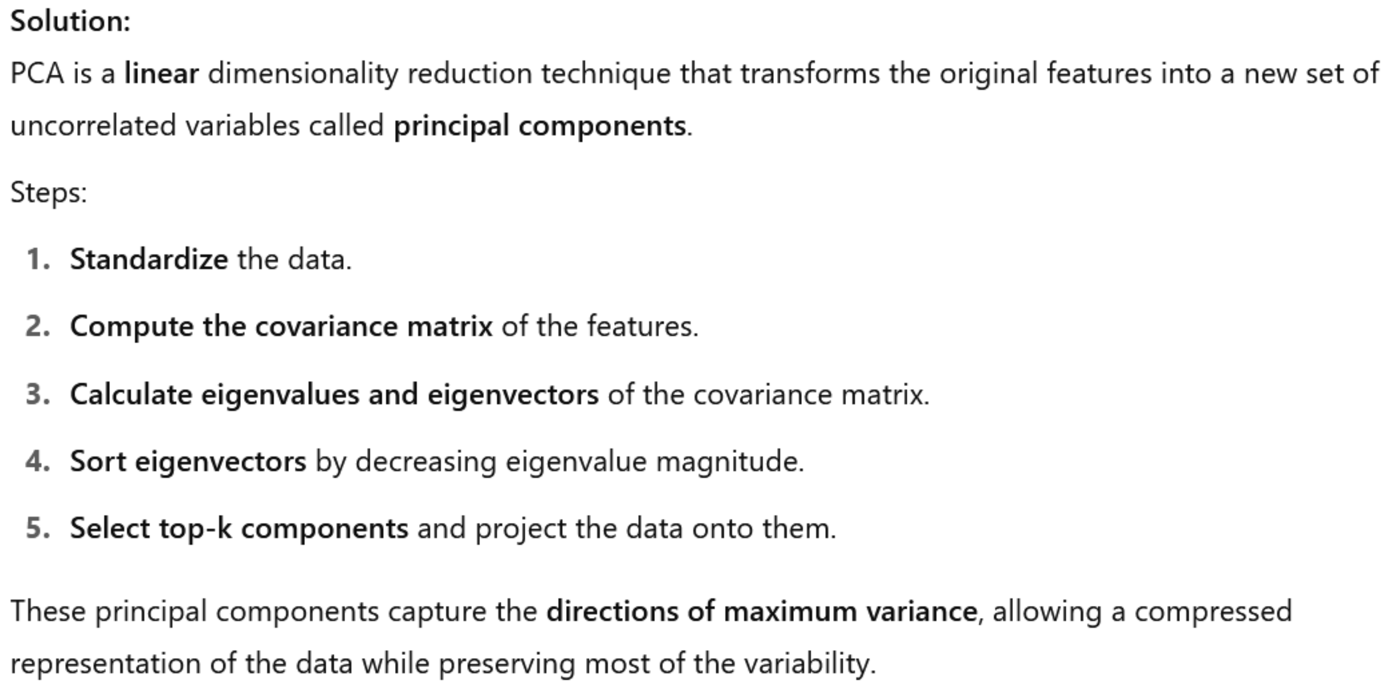
$$\begin{aligned} \det(sI - X^\top X) &= \det \begin{bmatrix} s-4 & -2 \\ -2 & s-4 \end{bmatrix} = (s-4)(s-4) - (-2)(-2) \\ &= s^2 - 8s + 12 = (s-6)(s-2). \end{aligned}$$

Hence the eigenvalues of $\dot{X}^\top \dot{X}$ are $\lambda_1 = 2$ and $\lambda_2 = 6$. With these eigenvalues, we can compute the eigenvectors of this matrix as follows. (Or you could just guess them and verify them.)

$$\begin{bmatrix} \lambda_1 - 4 & -2 \\ -2 & \lambda_1 - 4 \end{bmatrix} v_1 = \begin{bmatrix} -2 & -2 \\ -2 & -2 \end{bmatrix} v_1 = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \Rightarrow v_1 = \begin{bmatrix} 1/\sqrt{2} \\ -1/\sqrt{2} \end{bmatrix}$$

$$\begin{bmatrix} \lambda_2 - 4 & -2 \\ -2 & \lambda_2 - 4 \end{bmatrix} v_2 = \begin{bmatrix} 2 & -2 \\ -2 & 2 \end{bmatrix} v_2 = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \Rightarrow v_2 = \begin{bmatrix} 1/\sqrt{2} \\ 1/\sqrt{2} \end{bmatrix}$$

Problem 7 (10 marks)
7. (a) (5 marks)
Explain the working of Principal Component Analysis (PCA) and how it achieves dimensionality reduction.
Answer:



Solution:

PCA is a **linear** dimensionality reduction technique that transforms the original features into a new set of uncorrelated variables called **principal components**.

Steps:

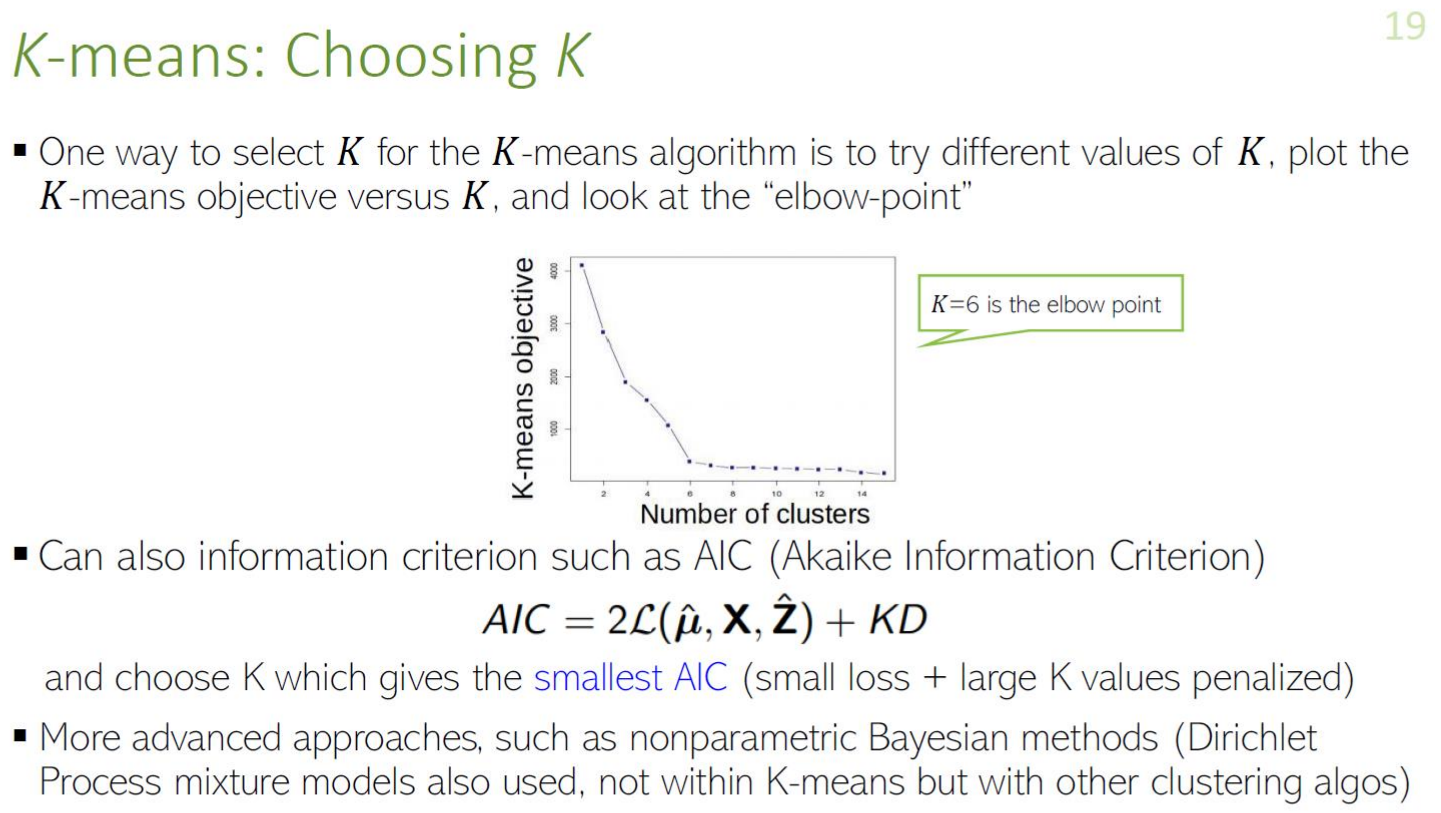
1. **Standardize** the data.
2. **Compute the covariance matrix** of the features.
3. **Calculate eigenvalues and eigenvectors** of the covariance matrix.
4. **Sort eigenvectors** by decreasing eigenvalue magnitude.
5. **Select top-k components** and project the data onto them.

These principal components capture the **directions of maximum variance**, allowing a compressed representation of the data while preserving most of the variability.

7. (b) (5 marks)

How do you determine the optimal number of clusters (K) in K-means?

Answer:



K-means: Choosing *K*

- One way to select K for the K-means algorithm is to try different values of K, plot the K-means objective versus K, and look at the "elbow-point"

K=6 is the elbow point

- Can also information criterion such as AIC (Akaike Information Criterion)

$$AIC = 2\mathcal{L}(\hat{\mu}, \mathbf{X}, \hat{\mathbf{Z}}) + KD$$

and choose K which gives the smallest AIC (small loss + large K values penalized)

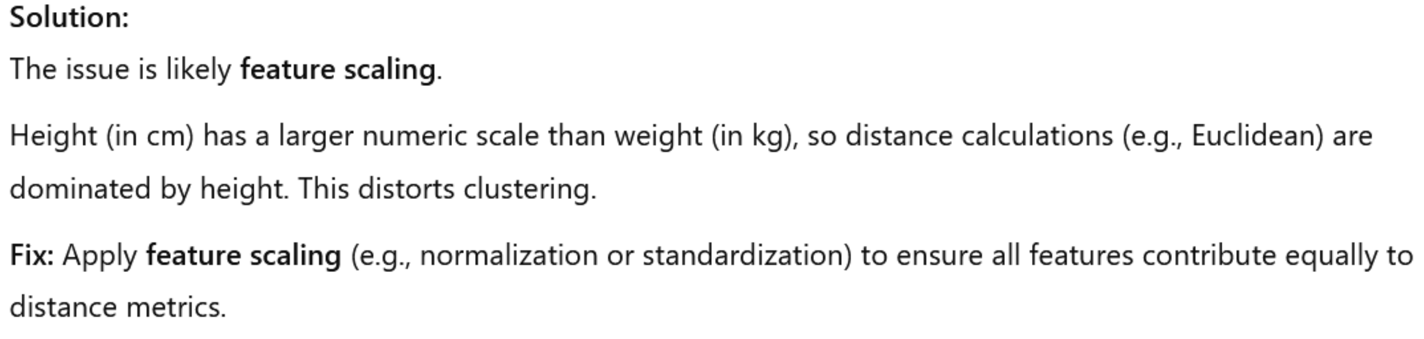
- More advanced approaches, such as nonparametric Bayesian methods (Dirichlet Process mixture models also used, not within K-means but with other clustering algos)

Problem 8 (10 marks)

8. (a) (5 marks)

You applied K-means clustering to a dataset with two features: height (in cm) and weight (in kg). The algorithm formed poor clusters. What might be the issue?

Answer:



Solution:

The issue is likely **feature scaling**.

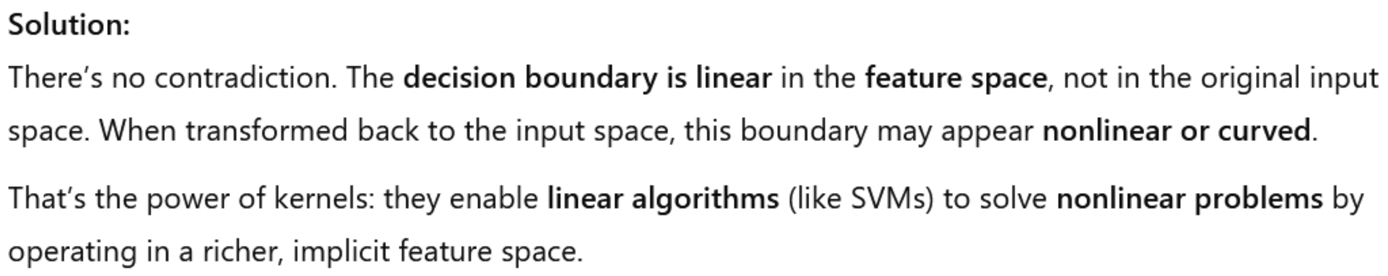
Height (in cm) has a larger numeric scale than weight (in kg), so distance calculations (e.g., Euclidean) are dominated by height. This distorts clustering.

Fix: Apply **feature scaling** (e.g., normalization or standardization) to ensure all features contribute equally to distance metrics.

8. (b) (5 marks)

In a kernelized SVM using a nonlinear kernel (e.g., RBF), the decision boundary in the input space appears curved. Yet, we say SVM finds a linear separator. Isn't this a contradiction?

Answer:



Solution:

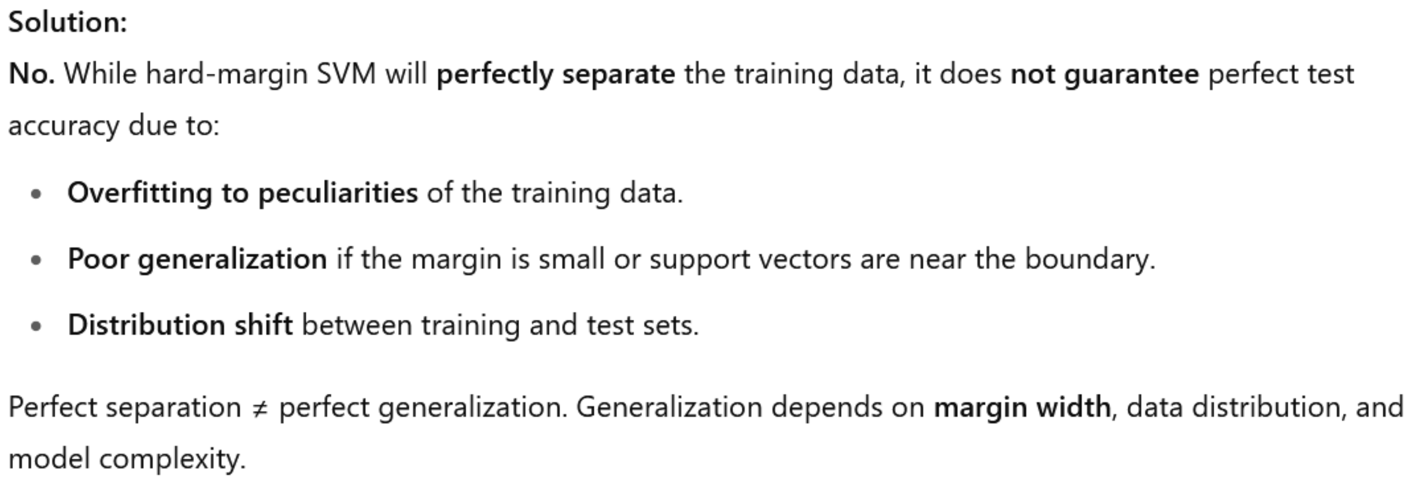
There's no contradiction. The **decision boundary is linear** in the **feature space**, not in the original input space. When transformed back to the input space, this boundary may appear **nonlinear or curved**.

That's the power of kernels: they enable **linear algorithms** (like SVMs) to solve **nonlinear problems** by operating in a richer, implicit feature space.

Problem 9 (10 marks)
9. (a) (5 marks)

Suppose you train a hard-margin SVM on a perfectly linearly separable dataset. Does this guarantee 100% test accuracy? Why or why not?

Answer:



Solution:

No. While hard-margin SVM will **perfectly separate** the training data, it does **not guarantee** perfect test accuracy due to:

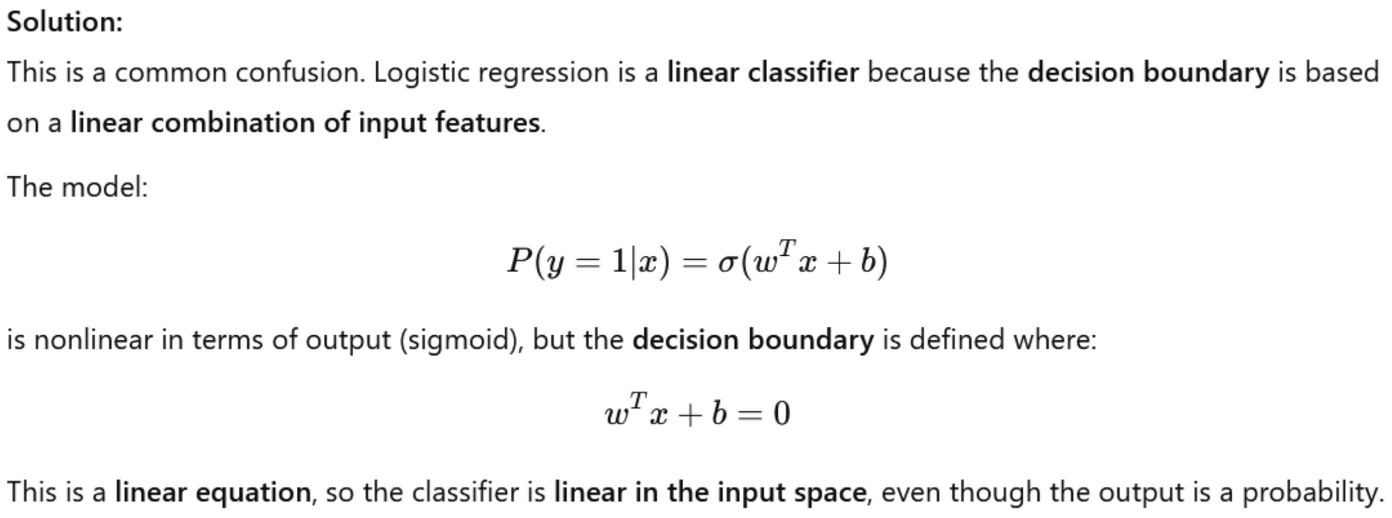
- **Overfitting to peculiarities** of the training data.
- **Poor generalization** if the margin is small or support vectors are near the boundary.
- **Distribution shift** between training and test sets.

Perfect separation ≠ perfect generalization. Generalization depends on **margin width**, data distribution, and model complexity.

9. (b) (5 marks)

Logistic regression outputs probabilities using a sigmoid function, which is nonlinear. So how can it be called a linear classifier?

Answer:



Solution:

This is a common confusion. Logistic regression is a **linear classifier** because the **decision boundary** is based on a **linear combination of input features**.

The model:

$$P(y=1|x) = \sigma(w^T x + b)$$

is nonlinear in terms of output (sigmoid), but the **decision boundary** is defined where:

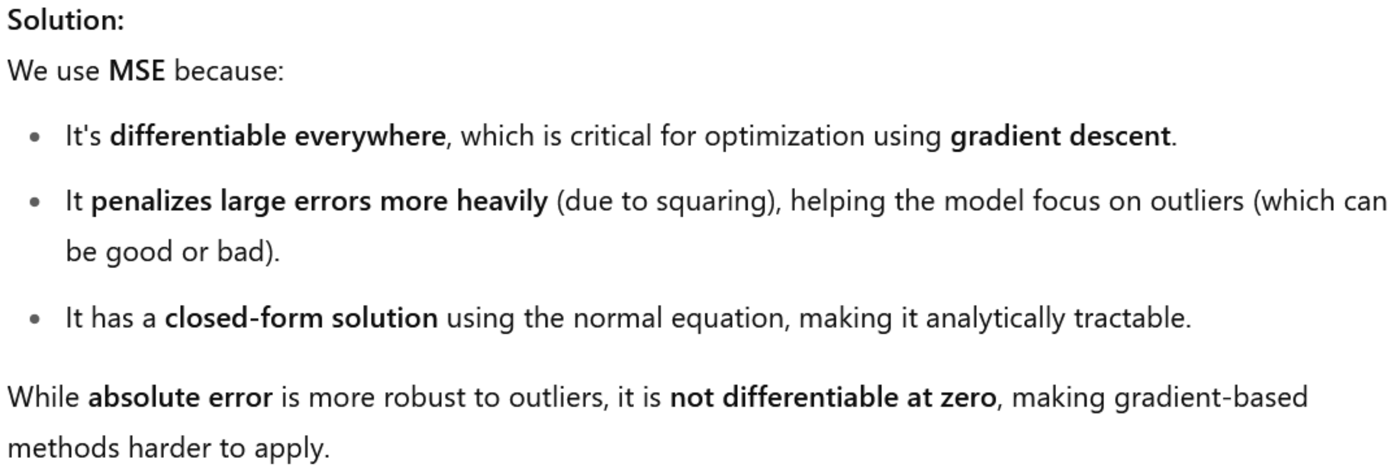
$$w^T x + b = 0$$

This is a **linear equation**, so the classifier is **linear in the input space**, even though the output is a probability.

Problem 10 (10 marks)
10. (a) (5 marks)

Why do we use the mean squared error (MSE) as the cost function in linear regression, and not just absolute error?

Answer:



Solution:

We use **MSE** because:

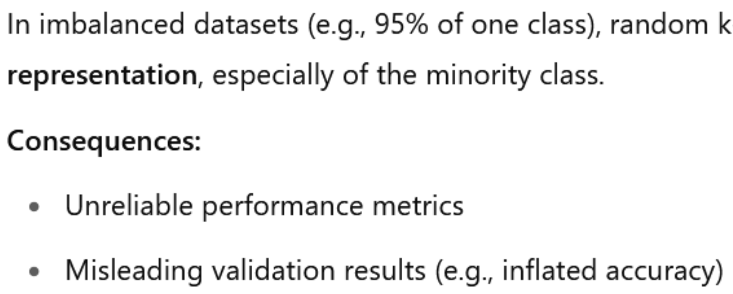
- It's **differentiable everywhere**, which is critical for optimization using **gradient descent**.
- It **penalizes large errors more heavily** (due to squaring), helping the model focus on outliers (which can be good or bad).
- It has a **closed-form solution** using the normal equation, making it analytically tractable.

While **absolute error** is more robust to outliers, it is **not differentiable at zero**, making gradient-based methods harder to apply.

10. (b) (5 marks)

What problem might arise when applying standard k-fold cross-validation to imbalanced datasets? How can you address it?

Answer:



In imbalanced datasets (e.g., 95% of one class), random k-fold CV may produce **folds without sufficient class representation**, especially of the minority class.

Consequences:

- Unreliable performance metrics
- Misleading validation results (e.g., inflated accuracy)

How to Address It:

1. **Stratified k-Fold Cross-Validation:**
    - Instead of randomly splitting the dataset into k folds, **stratified k-fold cross-validation** ensures that each fold maintains the same class distribution as the original dataset.
    - This helps prevent any fold from being overly biased toward the majority class and ensures both training and validation sets have a **representative distribution** of classes.

2 **Resampling Techniques:**

- **Oversampling the minority class** (e.g., using techniques like **SMOTE** — Synthetic Minority Over-sampling Technique) or **undersampling the majority class** can help balance the class distribution in both the training and validation folds.
- However, these methods should be applied **within each fold** to avoid data leakage, ensuring that the model doesn't see the resampled data from the test set during training.

3 **Alternative Evaluation Metrics:**

- Instead of relying on **accuracy**, which can be misleading in imbalanced datasets, use more informative metrics:
    - **Precision, Recall, and F1-Score** for each class, particularly focusing on the minority class.
    - **AUC-ROC Curve (Area Under the Receiver Operating Characteristic Curve)**, which shows how well the model distinguishes between the classes at various thresholds.
    - **Confusion Matrix** to visualize misclassifications and help identify patterns in errors.