Birla Institute of Technology & Science, Pilani
Work-Integrated Learning Programmes Division
First Semester 2024-2025

Comprehensive Exam
(EC-3 Make-up)

Course No.	: DSECLZG565/ AIMLCLZ565
Course Title	: MACHINE LEARNING
Nature of Exam	: Open Book
Weightage	: 40%
Duration	:
Date of Exam	:

No. of Pages = 3

Note:

1. Please follow all the *Instructions to Candidates* given on the cover page of the answer book.
2. All parts of a question should be answered consecutively. Each answer should start from a fresh page.
3. Assumptions made if any, should be stated clearly at the beginning of your answer.

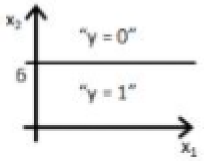
Question 1: [5 marks]

a) "Gini Impurity of a node is generally greater that its parent's"? Is the statement true or False? Justify your answer **[2 marks]**
False. [0.5 marks]
[1.5 marks for justification]

b) Suppose you are asked to build a classifier using logistic regression. You have come up with the following hypothesis function H which is

$$h_\theta(x) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_2)$$

where $\theta_0 = 6, \theta_1 = 0, \theta_2 = -1$
Draw a graph to show the decision boundary for the above classifier. Clearly indicate the class labels.
[3 marks]



[1 mark for the calculations]
[1 mark for the plot]
[1 mark for correctly mentioning class labels]

c) You are using a logistic regression model to detect fraudulent transactions. The model outputs probability scores for each transaction, and a threshold of 0.5 is typically used to classify a transaction as fraud or not. Suppose you increase this threshold to 0.9. Explain how this change is expected to affect the model's ability to detect fraud cases and its handling of false positives. Additionally, discuss how precision and recall are likely to be impacted. Your answer must be justified with reasoning—answers without justification will not be awarded marks. [4]

Increasing the threshold from 0.5 to 0.9 in a logistic regression model means the model will only classify a transaction as fraud if its predicted probability is very high (above 0.9). This makes the model more conservative in labeling something as fraudulent.

As a result:

- **The model will likely miss more actual fraud cases**,
- This will also lead to a **decrease in recall**, which measures the model's ability to correctly identify positive cases (frauds). [1.5 marks for correct explanation]
- On the other hand, **false positives will be reduced**. Fewer legitimate transactions will be incorrectly flagged as fraud, which typically **increases precision**, as precision measures how many of the predicted frauds were actually fraud. [1.5 marks for correct explanation]

In summary, increasing the threshold to 0.9 is likely to reduce the number of false positives, but at the cost of missing more actual frauds. This improves precision but lowers recall. [1 mark]

Question 3: [3 marks]

Describe the bias-variance tradeoff in the context of k-NN. How does the value of k influence both bias and variance? Use examples to illustrate what happens when k is too low or too high.

🔹 *When k is small (e.g., k=1): [1.5 marks for correct explanation]*

- **Low Bias**
- **High Variance**
- **Overfitting risk**

🔹 *When k is large (e.g., k=n, # data points): [1.5 marks for correct explanation]*

- **High Bias and Low Variance**
-

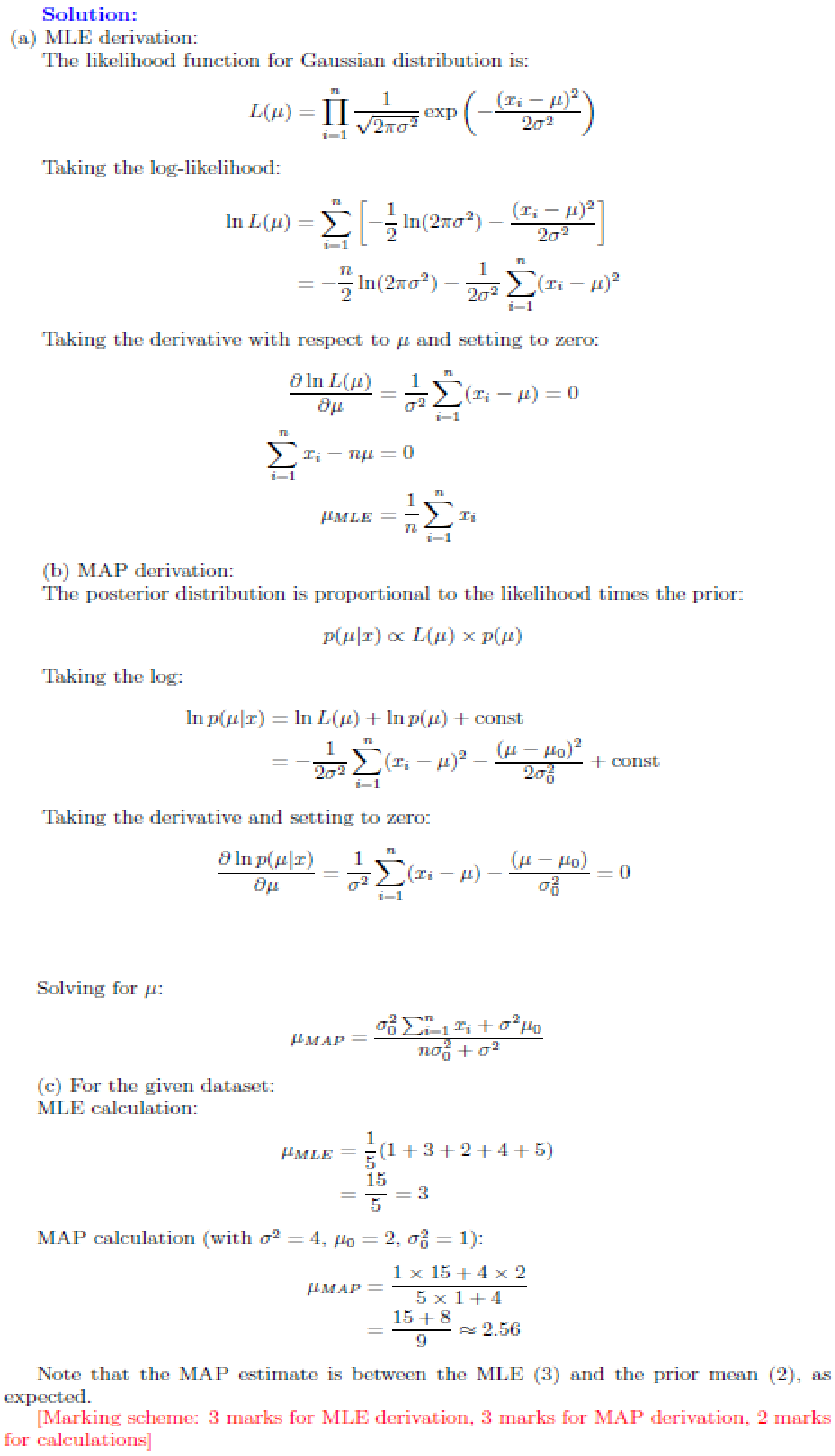
Question 4: [8 marks]

Consider a dataset of n independent observations $\{x_1, x_2, \ldots x_n\}$, from a Gaussian distribution with unknown mean μ and known variance $\sigma^2 = 4$.

(a) Derive the Maximum Likelihood Estimate (MLE) for μ using the log-likelihood function. Show all steps including the derivative. (3 marks)

(b) If we assume a Gaussian prior on μ with mean $\mu_0 = 2$ and variance $\sigma_0^2 = 1$, derive Maximum A Posteriori (MAP) estimate for μ. (3 marks)

(c) For the dataset $\{1, 3, 2, 4, 5\}$, calculate both the MLE and MAP estimates for μ. (2 marks) (2)



Solution:

(a) MLE derivation:

The likelihood function for Gaussian distribution is:

$$L(\mu) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x_i-\mu)^2}{2\sigma^2}\right)$$

Taking the log-likelihood:

$$\begin{aligned}\ln L(\mu) &= \sum_{i=1}^{n}\left[-\frac{1}{2}\ln(2\pi\sigma^2) - \frac{(x_i-\mu)^2}{2\sigma^2}\right] \\ &= -\frac{n}{2}\ln(2\pi\sigma^2) - \frac{1}{2\sigma^2}\sum_{i=1}^{n}(x_i-\mu)^2\end{aligned}$$

Taking the derivative with respect to μ and setting to zero:

$$\frac{\partial \ln L(\mu)}{\partial \mu} = \frac{1}{\sigma^2}\sum_{i=1}^{n}(x_i-\mu) = 0$$

$$\sum_{i=1}^{n} x_i - n\mu = 0$$

$$\mu_{MLE} = \frac{1}{n}\sum_{i=1}^{n} x_i$$

(b) MAP derivation:

The posterior distribution is proportional to the likelihood times the prior:

$$p(\mu|x) \propto L(\mu) \times p(\mu)$$

Taking the log:

$$\begin{aligned}\ln p(\mu|x) &= \ln L(\mu) + \ln p(\mu) + \text{const} \\ &= -\frac{1}{2\sigma^2}\sum_{i=1}^{n}(x_i-\mu)^2 - \frac{(\mu-\mu_0)^2}{2\sigma_0^2} + \text{const}\end{aligned}$$

Taking the derivative and setting to zero:

$$\frac{\partial \ln p(\mu|x)}{\partial \mu} = \frac{1}{\sigma^2}\sum_{i=1}^{n}(x_i-\mu) - \frac{(\mu-\mu_0)}{\sigma_0^2} = 0$$

Solving for μ:

$$\mu_{MAP} = \frac{\sigma_0^2\sum_{i=1}^{n} x_i + \sigma^2\mu_0}{n\sigma_0^2 + \sigma^2}$$

(c) For the given dataset:

MLE calculation:

$$\begin{aligned}\mu_{MLE} &= \frac{1}{5}(1+3+2+4+5) \\ &= \frac{15}{5} = 3\end{aligned}$$

MAP calculation (with $\sigma^2 = 4$, $\mu_0 = 2$, $\sigma_0^2 = 1$):

$$\begin{aligned}\mu_{MAP} &= \frac{1\times 15 + 4\times 2}{5\times 1 + 4} \\ &= \frac{15+8}{9} \approx 2.56\end{aligned}$$

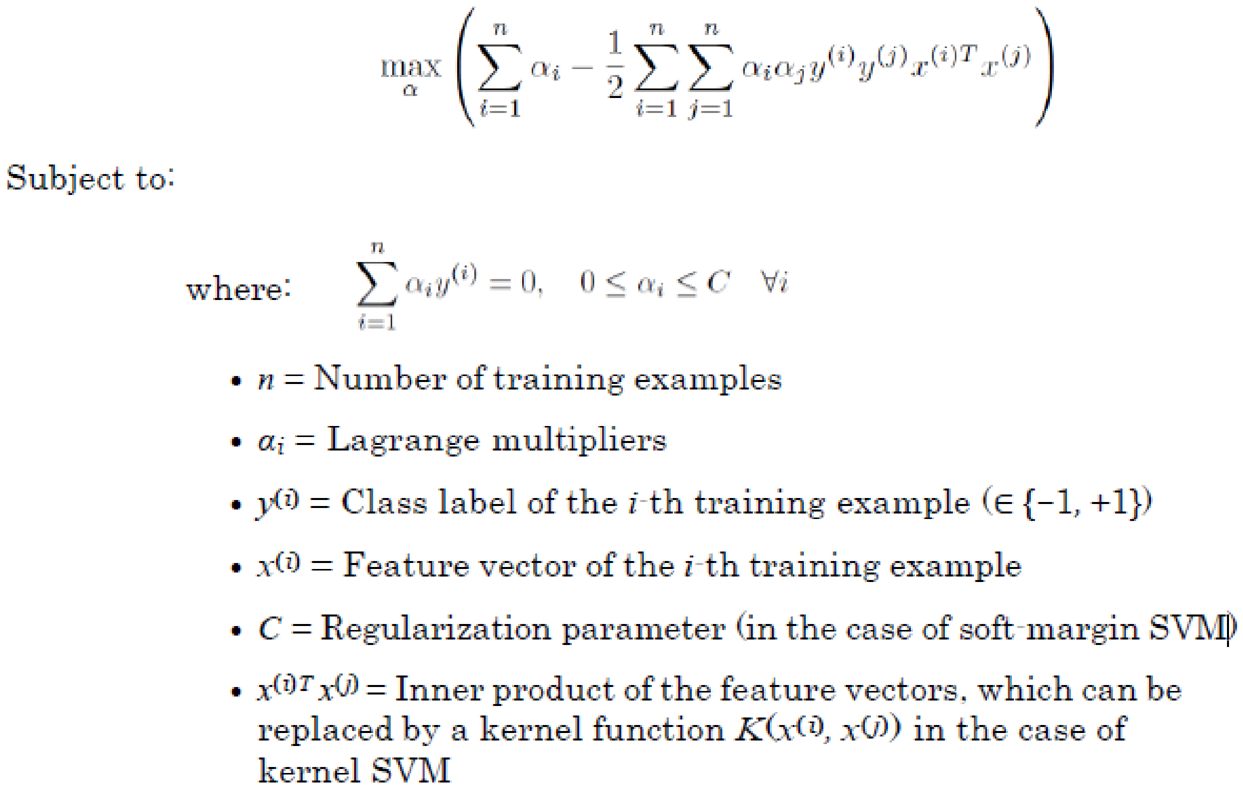
Note that the MAP estimate is between the MLE (3) and the prior mean (2), as expected.

[Marking scheme: 3 marks for MLE derivation, 3 marks for MAP derivation, 2 marks for calculations]

Marks should be awarded if all the steps are properly shown, otherwise deduct marks accordingly

Question 5: [7 marks]

a) Given below is the SVM dual optimization equation: [4 marks]



$$\max_{\alpha}\left(\sum_{i=1}^{n}\alpha_i-\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y^{(i)}y^{(j)}x^{(i)T}x^{(j)}\right)$$

Subject to:

where: $\sum_{i=1}^{n}\alpha_i y^{(i)}=0, \quad 0\le\alpha_i\le C \quad \forall i$

- n = Number of training examples
- α_i = Lagrange multipliers
- $y^{(i)}$ = Class label of the i-th training example ($\in\{-1,+1\}$)
- $x^{(i)}$ = Feature vector of the i-th training example
- C = Regularization parameter (in the case of soft-margin SVM)
- $x^{(i)T}x^{(j)}$ = Inner product of the feature vectors, which can be replaced by a kernel function $K(x^{(i)},x^{(j)})$ in the case of kernel SVM

We pass the *x* values and *y* values to a quadratic programming package to solve the SVM dual problem. The package returns a vector of Lagrange multipliers *α*. We observe that many of the *αi* values are zero.
Is there any issue with the quadratic programming package that might cause it to return zero values for some *αi*? If not, justify your answer.

Note: Marks will be awarded only if justification is provided to support your answer.

Support vectors: Points for which αi>0. These directly affect the position of the decision boundary. **Non-support vectors are p**oints for which αi=0. These do **not** influence the model after training. **[1.5 marks for mentioning that alpha in case of support vector is non-zero, and zero in case of non SV]**

🔍 Why is this not an issue?

- It's mathematically correct and optimal to assign **zero values** to many αi, as per
- **complementary slackness condition**, - part of the KKT conditions. According to complementary slackness, every constraint, gi(x∗)≤0 in the primal corresponds to a dual variable μi (Lagrange multiplier). The condition state that
  - gi(x∗)μi=0
- For points that are not support vectors, we have gi(x∗)<0, hence we must have μi=0

[2.5 marks if student has mentioned the " complementary slackness condition" and its explanation for zero values for most of the alpha parameters]
[1 mark if only complementary slackness condition is mentioned without explanation]

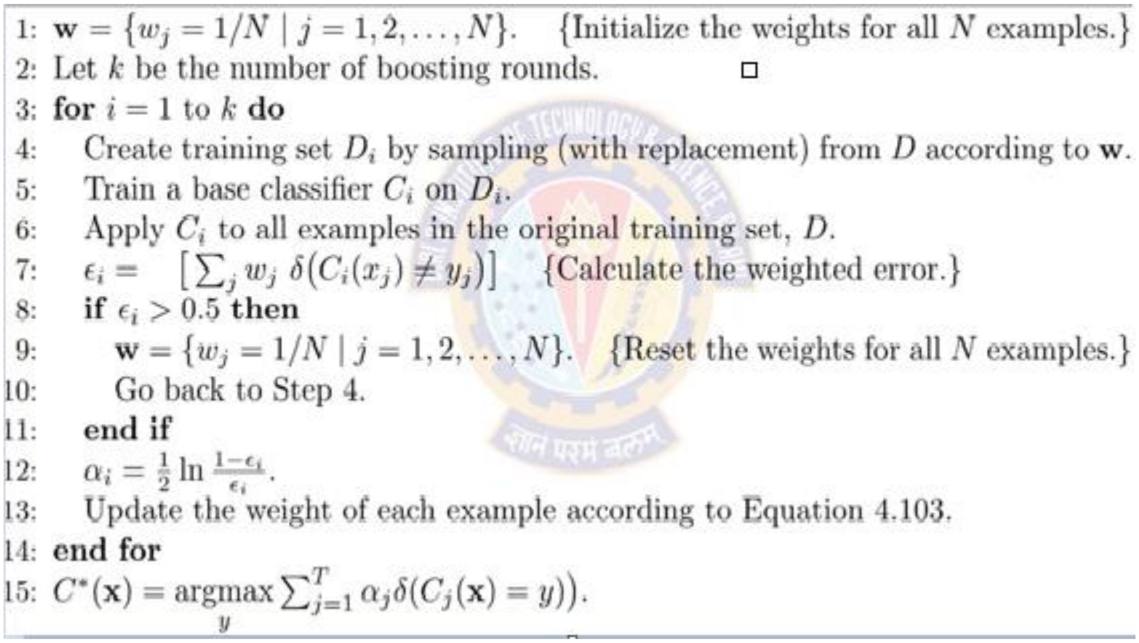
b) Explain the relationship between the regularization parameter C and model performance in Support Vector Machines (SVMs), considering the trade-off between bias and variance. How do lower values of C affect the model, and what are the potential consequences of setting a large value for C? [3]

In Support Vector Machines (SVMs), the regularization parameter **C** controls the trade-off between maximizing the margin and minimizing classification errors. A **low value of C** allows the model to have a wider margin by tolerating some misclassifications, which leads to **higher bias but lower variance**—this can help improve generalization, especially on noisy data.**[1.5 marks if correct explanation is provided].**

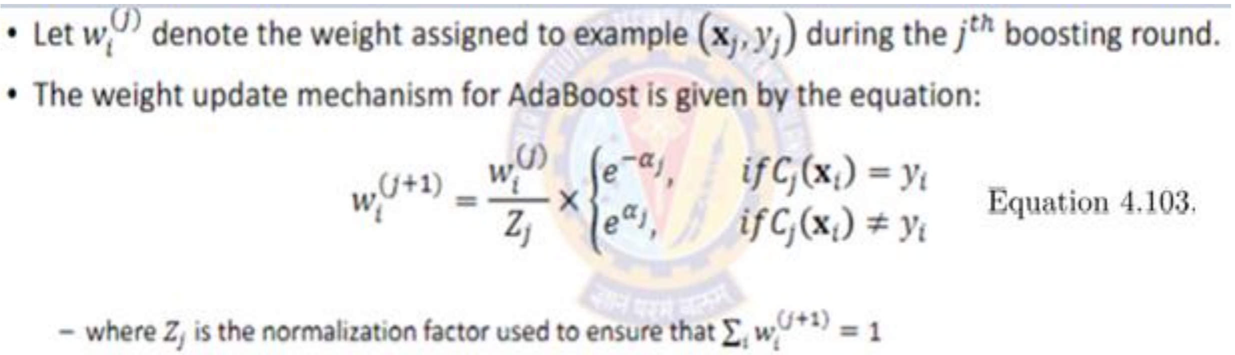
On the other hand, a **high value of C** tries to classify all training points correctly by penalizing misclassifications more heavily, resulting in a **lower bias but higher variance**. While this can improve training accuracy, it increases the risk of **overfitting**, potentially harming the model's performance on unseen data. **.[1.5 marks if correct explanation is provided].**

Question 6: [8 marks]

Given the pseudocode for the AdaBoost algorithm, and a dataset of 5 instances along with their corresponding weights (after applying the first weak classifier h1), use this information to answer the following questions. [1+1+3+2=7]



```
1:  $\mathbf{w} = \{w_j = 1/N \mid j = 1, 2, \ldots, N\}$.   {Initialize the weights for all $N$ examples.}
2:  Let $k$ be the number of boosting rounds.   □
3:  for $i = 1$ to $k$ do
4:      Create training set $D_i$ by sampling (with replacement) from $D$ according to $\mathbf{w}$.
5:      Train a base classifier $C_i$ on $D_i$.
6:      Apply $C_i$ to all examples in the original training set, $D$.
7:      $\epsilon_i = \left[\sum_j w_j \, \delta(C_i(x_j) \neq y_j)\right]$   {Calculate the weighted error.}
8:      if $\epsilon_i > 0.5$ then
9:          $\mathbf{w} = \{w_j = 1/N \mid j = 1, 2, \ldots, N\}$.   {Reset the weights for all $N$ examples.}
10:         Go back to Step 4.
11:     end if
12:     $\alpha_i = \frac{1}{2} \ln \frac{1-\epsilon_i}{\epsilon_i}$.
13:     Update the weight of each example according to Equation 4.103.
14: end for
15: $C^*(\mathbf{x}) = \underset{y}{\operatorname{argmax}} \sum_{j=1}^{T} \alpha_j \delta(C_j(\mathbf{x}) = y))$.
```



- Let $w_i^{(j)}$ denote the weight assigned to example $(\mathbf{x}_j, y_j)$ during the j^{th} boosting round.
- The weight update mechanism for AdaBoost is given by the equation:

$$w_i^{(j+1)} = \frac{w_i^{(j)}}{Z_j} \times \begin{cases} e^{-\alpha_j}, & if\, C_j(\mathbf{x}_i) = y_i \\ e^{\alpha_j}, & if\, C_j(\mathbf{x}_i) \neq y_i \end{cases} \quad \text{Equation 4.103.}$$

– where Z_j is the normalization factor used to ensure that $\sum_i w_i^{(j+1)} = 1$

Instance	x_1	x_2	y (true label)	Weight (w)
1	2	3	+1	0.3
2	3	2	+1	0.3
3	4	5	-1	0.05
4	5	4	-1	0.05
5	1	1	+1	0.3

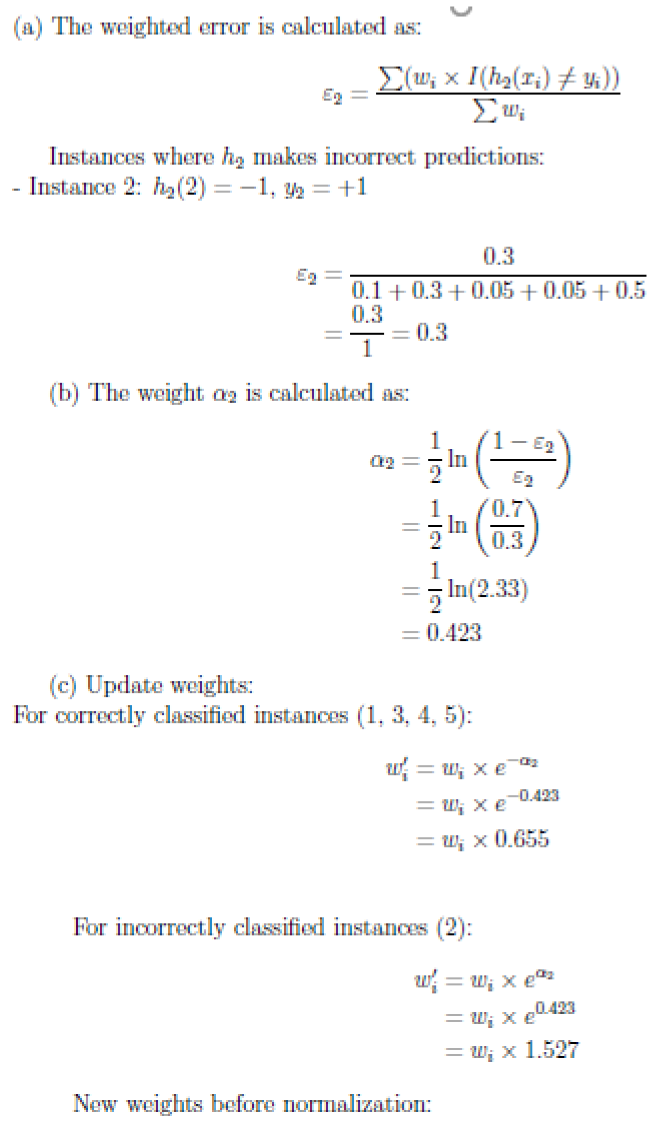
The second weak classifier h_2 predicts:
$h_2(1) = +1,\ h_2(2) = -1,\ h_2(3) = -1,\ h_2(4) = -1,\ h_2(5) = +1$

a) Calculate the weighted error of h_2.

b) Calculate the weight α_2 for h_2 in the ensemble.

c) Update the weights for all instances for the next iteration. Show the normalized values.

d) What would be the final prediction of the AdaBoost ensemble (h_1 and h_2) for instance 2, if $\alpha_1 = 0.4$?



(a) The weighted error is calculated as:

$$\varepsilon_2 = \frac{\sum(w_i \times I(h_2(x_i) \neq y_i))}{\sum w_i}$$

Instances where h_2 makes incorrect predictions:
- Instance 2: $h_2(2) = -1$, $y_2 = +1$

$$\begin{aligned}\varepsilon_2 &= \frac{0.3}{0.1 + 0.3 + 0.05 + 0.05 + 0.5} \\ &= \frac{0.3}{1} = 0.3\end{aligned}$$

(b) The weight α_2 is calculated as:

$$\begin{aligned}\alpha_2 &= \frac{1}{2}\ln\left(\frac{1-\varepsilon_2}{\varepsilon_2}\right) \\ &= \frac{1}{2}\ln\left(\frac{0.7}{0.3}\right) \\ &= \frac{1}{2}\ln(2.33) \\ &= 0.423\end{aligned}$$

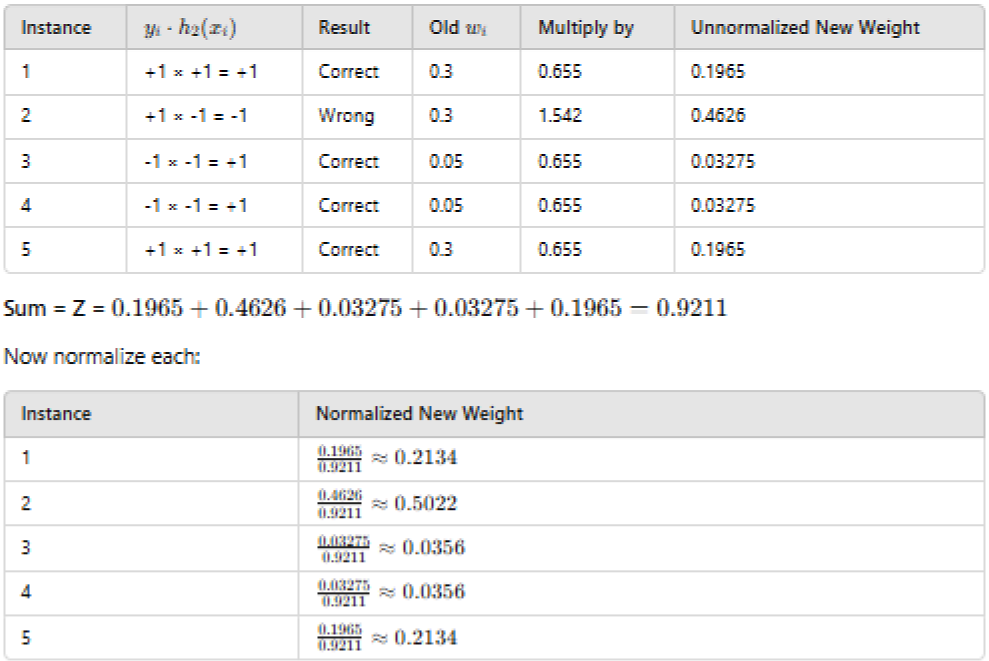
(c) Update weights:
For correctly classified instances (1, 3, 4, 5):

$$\begin{aligned}w_i' &= w_i \times e^{-\alpha_2} \\ &= w_i \times e^{-0.423} \\ &= w_i \times 0.655\end{aligned}$$

For incorrectly classified instances (2):

$$\begin{aligned}w_i' &= w_i \times e^{\alpha_2} \\ &= w_i \times e^{0.423} \\ &= w_i \times 1.527\end{aligned}$$

New weights before normalization:



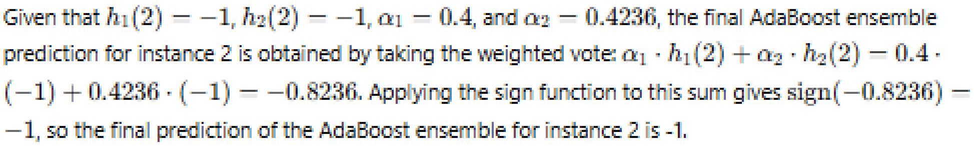
Instance	$y_i \cdot h_2(x_i)$	Result	Old w_i	Multiply by	Unnormalized New Weight
1	+1 × +1 = +1	Correct	0.3	0.655	0.1965
2	+1 × -1 = -1	Wrong	0.3	1.542	0.4626
3	-1 × -1 = +1	Correct	0.05	0.655	0.03275
4	-1 × -1 = +1	Correct	0.05	0.655	0.03275
5	+1 × +1 = +1	Correct	0.3	0.655	0.1965

Sum = Z = $0.1965 + 0.4626 + 0.03275 + 0.03275 + 0.1965 = 0.9211$

Now normalize each:

Instance	Normalized New Weight
1	$\frac{0.1965}{0.9211} \approx 0.2134$
2	$\frac{0.4626}{0.9211} \approx 0.5022$
3	$\frac{0.03275}{0.9211} \approx 0.0356$
4	$\frac{0.03275}{0.9211} \approx 0.0356$
5	$\frac{0.1965}{0.9211} \approx 0.2134$

4) H1(2) have been incorrectly classified since it's weight was increased after round 1.



Given that $h_1(2) = -1$, $h_2(2) = -1$, $\alpha_1 = 0.4$, and $\alpha_2 = 0.4236$, the final AdaBoost ensemble prediction for instance 2 is obtained by taking the weighted vote: $\alpha_1 \cdot h_1(2) + \alpha_2 \cdot h_2(2) = 0.4 \cdot (-1) + 0.4236 \cdot (-1) = -0.8236$. Applying the sign function to this sum gives $\text{sign}(-0.8236) = -1$, so the final prediction of the AdaBoost ensemble for instance 2 is -1.

Part 1) [1 marks],

2) 1 mark

3) 3 marks

4) 2 marks. (full marks should be awarded only if student has correctly mentioned, h1(2) value (=-1 since it's weight was increased after round 1, hence the instance was misclassified), otherwise 0 marks should be awarded.)

Question 6: .[6 Marks]

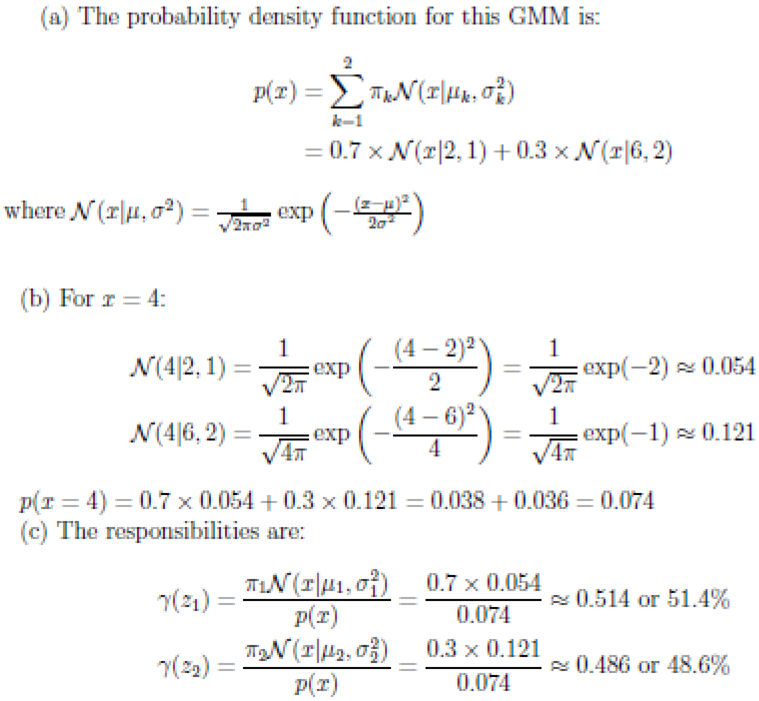
Consider a dataset with two 1-dimensional Gaussian components. The first component has mean $\mu_1 = 2$ and variance $\sigma^2{}_1 = 1$, while the second component has mean $\mu_2 = 6$ and variance $\sigma^2{}_2 = 2$. The mixing coefficients are $\pi_1 = 0.7$ and $\pi_2 = 0.3$. [1.5*4 = 6]

(a) Write the probability density function for this Gaussian mixture model.

(b) Calculate the likelihood of observing the data point $x = 4$.

(c) Calculate the responsibilities (posterior probabilities) for the data point $x = 4$.

(d) Which Gaussian component is more likely to have generated this point, $x = 4$? Justify your answer.



(a) The probability density function for this GMM is:

$$\begin{aligned} p(x) &= \sum_{k=1}^{2} \pi_k \mathcal{N}(x|\mu_k, \sigma_k^2) \\ &= 0.7 \times \mathcal{N}(x|2,1) + 0.3 \times \mathcal{N}(x|6,2) \end{aligned}$$

where $\mathcal{N}(x|\mu,\sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$

(b) For $x = 4$:

$$\mathcal{N}(4|2,1) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{(4-2)^2}{2}\right) = \frac{1}{\sqrt{2\pi}} \exp(-2) \approx 0.054$$

$$\mathcal{N}(4|6,2) = \frac{1}{\sqrt{4\pi}} \exp\left(-\frac{(4-6)^2}{4}\right) = \frac{1}{\sqrt{4\pi}} \exp(-1) \approx 0.121$$

$p(x=4) = 0.7 \times 0.054 + 0.3 \times 0.121 = 0.038 + 0.036 = 0.074$

(c) The responsibilities are:

$$\gamma(z_1) = \frac{\pi_1 \mathcal{N}(x|\mu_1, \sigma_1^2)}{p(x)} = \frac{0.7 \times 0.054}{0.074} \approx 0.514 \text{ or } 51.4\%$$

$$\gamma(z_2) = \frac{\pi_2 \mathcal{N}(x|\mu_2, \sigma_2^2)}{p(x)} = \frac{0.3 \times 0.121}{0.074} \approx 0.486 \text{ or } 48.6\%$$

d) Component 1 is much more likely to have generated the point $x = 4$, with a posterior probability of approximately 51.4%.

[Marking scheme: 1.5 marks for each]