SECTION A

Part c)

Perceptron Algorithm

- Initialize the weight vector w with the first example's feature vector w = x1
- For each example (xi, yi), predict its label using the current weight vector w. The prediction made would look like this :-
  - Yi = sign(w.xi)

  If Yi != yi, then a classification mistake has occurred.
- Even if the prediction is correct, the margin condition can be checked as :-
  - yi(w . xi) >= (gamma)/2

  If this condition is not satisfied, then it would mean that a margin mistake has occurred.
- If a classification mistake or margin mistake is found, then the weight vector is also updated as :-
  - W = w + yi.xi

This is the same as the update rule in the standard Perceptron algorithm, where yixi is added to the weight vector to adjust the separating hyperplane.

- Repeat the algorithm and continue to iterate through the examples, updating the weight vector whenever a classification or margin mistake occurs, until convergence is achieved.

This algorithm guarantees to converge for linearly separable data, as it enforces a more stricter condition by requiring that the margin must be at least (gamma)/2..

PART D)

c) The problem with the zero probability is that it causes the entire posterior probability for **Spam** to become 0, regardless of other features' probabilities. This prevents the classifier from making proper predictions, as the algorithm ignores other potentially relevant information.
The most straightforward and widely used solution is **Laplace Smoothing**, where you add 1 to all feature counts to avoid zero probabilities. By doing so, the classifier can better handle unseen features during training, resulting in more reliable predictions.

a)

4a) $P(\text{spam}) \Rightarrow \frac{2}{4} = 0.5$

$P(\text{Non-spam}) \Rightarrow \frac{2}{4} = 0.5$

$P(\text{buy}=1 \mid \text{spam}) \Rightarrow \frac{\text{no. of spam with "buy"}}{\text{Total spam}} \Rightarrow \frac{2}{2} = 1$

$P(\text{buy}=0 \mid \text{spam}) \Rightarrow 1 - P(\text{buy}=1 \mid \text{spam}) \Rightarrow 1-1 = 0$

$P(\text{cheap}=1 \mid \text{spam}) \Rightarrow \frac{\text{no. of spam with "cheap"}}{\text{Total spam}} \Rightarrow \frac{1}{2} = 0.5$

$P(\text{cheap}=0 \mid \text{spam}) \Rightarrow 1 - 0.5 \Rightarrow 0.5$

Now for Non spam

$P(\text{buy}=1 \mid \text{non spam}) \Rightarrow \frac{\text{no. of non-spam with buy}}{\text{total non-spam}} \Rightarrow \frac{1}{2} = 0.5$

$P(\text{buy}=0 \mid \text{non spam}) \Rightarrow 1-0.5 = 0.5$

$P(\text{cheap}=1 \mid \text{non spam}) \Rightarrow \frac{1}{2} = 0.5$

$P(\text{cheap}=0 \mid \text{non spam}) \Rightarrow 1-0.5 = 0.5$

b)

Z b). new email contains "cheap" (Feature 2 = 1) but not "buy" (Feature 1 = 0) being spam or non spam.

Baye's theorem

$$P(\text{spam} \mid \text{Features}) = \frac{P(\text{Features} \mid \text{spam}) \cdot P(\text{spam})}{P(\text{Features})}$$

$$P(\text{non-spam} \mid \text{Features}) = \frac{P(\text{Features} \mid \text{non spam}) \cdot P(\text{non spam})}{P(\text{Features})}$$

calculating likelihood

* For spam

$P(\text{buy}=0 \mid \text{spam}) = 0$

$P(\text{cheap}=1 \mid \text{spam}) = 0.5$, $P(\text{spam}) = 0.5$

$\therefore P(\text{buy}=0, \text{cheap}=1 \mid \text{spam}) \Rightarrow P(\text{buy}=0 \mid \text{spam}) \cdot P(\text{cheap}=1 \mid \text{spam})$

$\Rightarrow 0 \times 0.5 \Rightarrow 0$

* For Non spam

$P(\text{buy}=0 \mid ns) \Rightarrow 0.5$

$P(\text{cheap}=1 \mid ns) \Rightarrow 0.5$; $P(ns) = 0.5$

$\therefore$ likelihood $\Rightarrow$

$P(\text{buy}=0, \text{cheap}=1 \mid ns) = P(\text{buy}=0 \mid ns) \cdot P(\text{cheap}=1 \mid ns)$

$\Rightarrow 0.5 \times 0.5 \Rightarrow 0.25$

Calculating posterior probabilities

* **For spam**

$$P(S \mid buy=0, cheap=1) \propto P(buy=0, cheap=1 \mid spam) \cdot P(spam)$$

$$\Rightarrow 0 \times 0.5 = 0$$

* **For non spam**

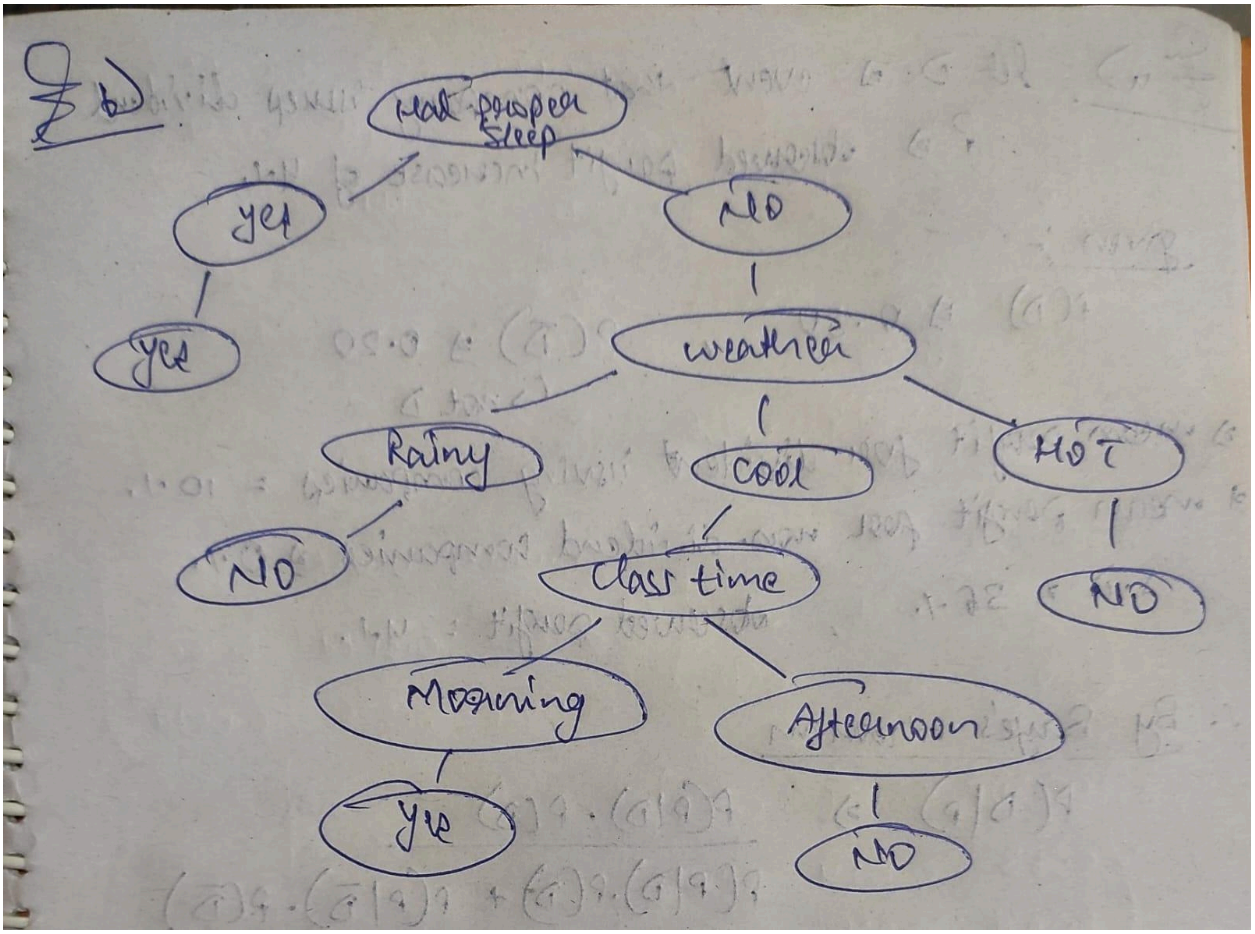
$$P(ns \mid buy=0, cheap=1) \propto P(buy=0, cheap=1 \mid ns) \cdot P(ns)$$

$$\Rightarrow 0.25 \times 0.5 \Rightarrow 0.125$$

Since posterior probability for new email being spam is 0, & for being non spam is 0.125.

$\therefore$ new email is more likely to be non-spam.

PART B)


Q b)
Had proper Sleep
yes
No
yes
weather
Rainy
cool
HOT
NO
class time
NO
Morning
Afternoon
yes
NO

PART A)

8 a) let D ⇒ event that company issues dividend
P ⇒ observed profit increase of 4%.

given :-

$P(D) \Rightarrow 0.80$, $P(\bar{D}) \Rightarrow 0.20$
↳ not D

⇒ mean profit for divident issuing companies = 10%.
⇒ mean profit for non-dividend companies ⇒ 0%.
S.D ⇒ 36% , observed profit = 4%.

∴ By Baye's theorem

$$P(D|P) \Rightarrow \frac{P(P|D) \cdot P(D)}{P(P|D) \cdot P(D) + P(P|\bar{D}) \cdot P(\bar{D})}$$

$$P(P|D) \Rightarrow \frac{1}{36\sqrt{2\pi}} \cdot e^{-\frac{(4-10)^2}{2 \times 36^2}} \Rightarrow 0.0107$$

$$P(P|\bar{D}) \Rightarrow \frac{1}{36\sqrt{2\pi}} \cdot e^{-\frac{(4-0)^2}{2 \times 36^2}} \Rightarrow 0.0109$$

$$\therefore P(D|P) \Rightarrow \frac{0.0107 \times 0.8}{0.0107 \times 0.8 + 0.0109 \times 0.2}$$

$$\Rightarrow \frac{0.0086}{0.0086 + 0.00218} \Rightarrow \frac{0.0086}{0.01078}$$

$$\Rightarrow 79.8\% \text{ or } \boxed{0.798}$$

∴ given a profit increase of 4%, the probability of issuing dividend is 79.8%.

SECTION C)

A)

- After performing the EDA we came to the conclusion that the dataset contains a total of 12600 images which is divided into 15 classes like sleeping, laughing, running etc. Each class has the same number of images of 840.
- With this data we can say that there is no imbalance in the classes of the dataset.

For more insights, you can see the outputs on the code notebook.

B)

General observations :-

- A total of 12,600 images were processed.
- Each image has 8,000 features extracted.

Detailed Observations :-

1. **Mean and Standard Deviation:**
    - The mean values for all features are very close to zero, which is expected after standardization.
    - The standard deviation for all features is approximately one, confirming successful standardization.
2. **Range of Feature Values:**
    - The minimum and maximum values for the features vary significantly, suggesting that the features capture a wide range of information from the images.
    - For example, feature 0 ranges from -2.02 to 3.90, while feature 7999 ranges from -3.73 to 1.81.

Class Distribution :-

- The class distribution is perfectly balanced, with each class representing approximately 6.67% of the total images.

C)

Model Performance Comparison:

Naive Bayes: 0.1627
Decision Tree: 0.1627
Random Forest: 0.3282
Perceptron: 0.2345

Best performing model: Random Forest with accuracy 0.3282