$\text{Average}_{(\text{Rain, True})} = \frac{30+48}{2} = 39.$

$SD_{(\text{Rain, True})} = \sqrt{\frac{(30-39)^2+(48-39)^2}{2}} = 9$

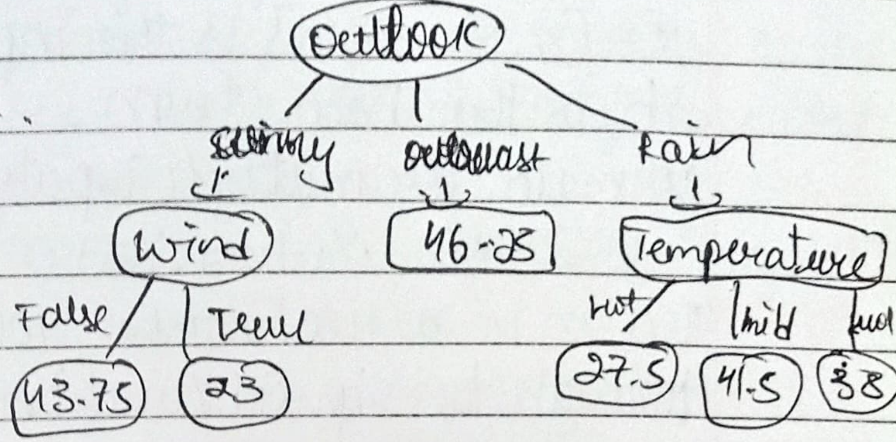
weighted standard deviation $= \frac{3}{5}(5.55) + \frac{2}{5}(9) = 6.93.$

SD for Rain outlook & wind $= 7.78 - 6.93 = 0.85$

$(SD)_{(\text{Rain Temp})} = 4.18$ ✓

$(SD)_{(\text{Rain, humidity})} = 3.32.$

$(SD)_{(\text{Rain, wind})} = 0.85$



Logistic Regression Assignment.

1) write the step by step process used in logistic regression model.

logistic Regression is a popular classification algorithm used to model the probability of a binary outcome (1/0 True/False, yes/no) based on one or more Independent variables.

I Initialization of weights and bias.

* weights ($w = [w_1, w_2 \dots w_n]$) : these parameters are associated with each input feature Initially, they are either randomly initialized or set to zeros.
* Bias (b) : This is the intercept term which helps adjust the decision boundary.

$W = [w_1, w_2, \dots, w_n], b = 0.$

II. compute the linear combination (weighted sum)
for each data point $x = [x_1, x_2, x_n]$ the weighted sum is calculated as follows $z = w^T . x + b$.

where z is the linear combination,
w^T is the transpose of the weight vector,
$x = [x_1, x_2 ... x_n]$ is the input feature vector,
b is the bias term.

formula for multiple inputs:

$$z = w_1 . x_1 + w_2 . x_2 + ... + w_n . x_n + b$$

This z is a real number, which will later be passed through the sigmoid function to transform it into a probability.

III. Apply the sigmoid function.
The sigmoid function transforms the linear combination z into a probability $P(y=1/x)$ b/w 0 & 1

Signmoid formula: $P(y=1/x) = \sigma(z) = \frac{1}{1+e^{-z}}$

- $P(y=1/x)$ is the predicted probability of the positive class
- e is the base of the natural logarithm (Euler's number)

the sigmoid function compresses the input z to lie within the range [0, 1], making it interpretable as a probability.

IV. Prediction of the class: once we have the probability $p(y=1/x)$, we can make predictions logistic regression classifies an instance as positive (1) if the predicted probability is greater than or equal to 0.5, otherwise, it classifies as negative (0)

$$\hat{y} = \begin{cases} 1 & \text{if } P(y=1/x) \geq 0.5 \\ 0 & \text{if } P(y=1/x) < 0.5 \end{cases}$$

PAGE NO.
DATE / /

V calculate the loss function (Binary cross - entropy)
To Evalute how well the model predicts, we can use the binary cross-entropy (or log-loss) as the cost func. The objective is to minimize this cost.
loss function (for one instance):

$$loss = -[y \cdot \log(p(y=1/x)) + (1-y) \cdot \log(1-p(y=1/x))]$$

where: y is the actual label (either 0 or 1)
$\times$ $p(y=1|x)$ is the predicted probability.
The total cost over all m training examples is the average of the individual losses:
Total cost:

$$J(w,b) = -1/m \sum_{i=1}^{m} [y^{(i)} \cdot \log(p^{(i)}) + (1-y^{(i)}) \cdot \log(1-p^{(i)})]$$

where: m is the total number of examples,
$p^{(i)}$ is the predicted probability for the i^{th} example.

VI compute Gradients: To minimize the loss, we compute the gradients of the cost function with respect to the weights and bias. These gradients tell us how much to change the weights to reduce the cost.
Gradient of loss with respect to weights:

$$\frac{\partial J}{\partial w_j} = \frac{1}{m} \sum_{i=1}^{m} (p^{(i)} - y^{(i)}) \cdot x_j^{(i)}$$

where: $x_j^{(i)}$ is the j^{th} feature of the i^{th} example
Gradient of loss with respect to Bias:

$$\frac{\partial J}{\partial b} = \frac{1}{m} \sum_{i=1}^{m} (p^{(i)} - y^{(i)}),$$

VII Update weights and Bias using Gradient Descent
We update the weights and bias using the gradient descent algorithm. This is done by subtracting the gradient of the loss function multiplied by the learning rate (α).
Weight update Rule:
$$w_j = w_j - \alpha \cdot \partial J / \partial w_j$$
Bias update Rule:
$$b = b - \alpha \cdot \partial J / \partial b.$$
where: α is the learning rate, hyperparameter that controls the size of the steps we take toward minimizing the cost function.

VIII Repeat for multiple Epochs: The process of computing the linear combination applying the sigmoid function, calculating the loss, & updating the weights is repeated over multiple iterations (epochs). With each epoch, the weights become more refined and the cost gradually decreases.

IX Make predictions on New Data: After training the learned weights and bias are used to make predictions on new, unseen data for a new input x^{new}, the steps are:
i) Compute $z = W^T \cdot x^{new} + b$, ii) Apply the sigmoid function to get the probability iii) Use the decision rule to predict the class

X Model Evaluation: finally, the model is evaluated on test data using performance metrics such as:
Accuracy, Precision, Recall, F1-Score.

x1	x2	y
2.7810836	2.550537003	0
1.675418651	-0.2420686549	1

PAGE NO.
DATE / /

$x_1 = 2.7810836 \quad x_2 = 2.550537003$

$z = b_0 + b_1 * x_1 + b_2 * x_2$ transformed = $1/(1+e^{\wedge}(-x))$

① Initialize $b_0, b_1, b_2 = 0$ → Initialize coefficients.

② $z = 0 + (0 * 2.7810836) + (0 * 2.550537003) = 0.0$

③ $t = \dfrac{1}{(1+e^{\wedge(-0)})} = 0.5$ compute linear combination (x)

→ Apply sigmoid function to compute transformed probability.

④ update the coefficients → learning rate

$b_{new} = b_{old} + \alpha \cdot (y - t) \cdot t \cdot (1-t) \cdot x_i$

$b_0 = 0 + 0.5 * (0 - 0.5) \cdot 0.5 (1 - 0.5) \cdot 1 = -0.0625$

$b_1 = 0 + 0.5 * (0 - 0.5) * 0.5 (1 - 0.5) * 2.7810836 = -0.1738$

$b_2 = 0 + 0.5 * (0 - 0.5) * 0.5 (1 - 0.5) * 2.550537003 = -0.1594$

ite-2. $z = b_0 + b_1 * x_1 + b_2 * x_2 = -0.9524$.

$= -0.0625 + (-0.1738 * 2.7810836) + (-0.1594 * 2.550537003)$

transform $t = \dfrac{1}{(1+e^{\wedge(-0.9524)})} = 0.2784$

$b_{new} = b(old) + \alpha \cdot (y - t) \cdot t (1 - t) \cdot x_i$

$b_0 = -0.0625 + 0.5 \cdot (0 - 0.2784) \cdot 0.2784 (1 - 0.2784) \cdot 1 = -0.0904$

$b_1 = -0.1738 + 0.5 \cdot (0 - 0.2784) \cdot 0.2784 (1 - 0.2784) \cdot 2.7810836 = -0.2515$

$b_2 = -0.1594 + 0.5 \cdot (0 - 0.2784) \cdot 0.2784 (1 - 0.2784) \cdot 2.550537003 = -0.2307$

$z = -0.0904 + (-0.2515 * 2.7810836) + (-0.2307 * 2.550537003) = -1.1974$

$t = \dfrac{1}{(1+e^{(-1.1974)})} = 0.2312$ the loss is reduced and we can classify to 0 category.

$x_0 = 1, x_1 = 8.675918651, x_2 = -0.242068654 \quad y = 1$

initialize weights $b_0 = b_1 = b_2 = 0$. — ①

epoch-①

② ⇒ $z = b_0 + b_1 x_1 + b_2 x_2 \Rightarrow 0 + 0 \cdot 8.675918651 + 0 \cdot -0.240686549 = 0$

③ ⇒ $t = 1/(1+e^{-(-z)}) = 0.5$

④ ⇒ $b_{new} = b_{old} + \alpha \cdot (y - t) \cdot t(1-t) * x_i$

$b_0 = 0 + 0.5 \cdot (1 - 0.5) \cdot 0.5(1-0.5) \cdot 1 = 0.0625$

$b_1 = 0 + 0.5 \cdot (1-0.5) \cdot 0.5(1-0.5) \cdot 8.675918651 \quad 0.5422$

$b_2 = 0 + 0.5(1-0.5) \cdot 0.5(1-0.5) \cdot -0.242068654 = -0.0151.$

epoch-②

① ⇒ $z = b_0 + b_1 x_1 + b_2 x_2 = 0.0625 + 0.5422 \times 8.6759185 1 + (-0.0151 * (-0.242068654)) = 4.7699$

② ⇒ $t = 1/(1+e^{-(4.7699)}) = 0.9915$

③ ⇒ $b_{new} = b_{old} + \alpha(y-t) * t(1-t) * x_i$

$b_0 = 0.0625 + 0.5(1-0.9915) * 0.9915(1-0.9915) * 1 = 0.0625$

$b_1 = 0.5422 + 0.5(1-0.9915) \times 0.9915(1-0.9915) \cdot 8.675918651 = 0.5422$

$b_2 = -0.0151 + 0.5(1-0.9915) \times 0.9915(1-0.9915) \cdot -0.242068654 = -0.0151$

$z = 0.0625 + (0.5425 * 8.675918651) + (-0.0151 * -0.242068654) = 4.7625$

$t = 1/(1+e^{-(4.7682)}) = 0.9915.$

The transformation is saturated after 2 epochs so we can classify the datapoint into class 1 category