24) a)

→ In logistic Regression, we're trying to figure out some important things using complicated (sigmoid function).

$$P(Y=1/x) = 1/(1+e^{-z}))$$

z: is a lineal combination of predictor Variables and model parameter

* NON-linearity: The logistic sigmoid function is non-lineal because of the exponential term (e^{-z}), which makes it impossible to find a closed-form solution for the model parameters. Instead, an iterative optimization technique is required to find the optimal values of 'w' that minimize the error function (usually negative log-likelihood)

→ To find the optimal 'w' in logistic Regression, you typically use optimization algorithms like gradient-descent, Newton-Raphson numerical optimization Techniques. These techniques adjust the parameters to minimize the error function until Convergence is reached.

→ Now, minimizing the error function by an efficient iterative technique, based on the Newton - Raphson iterative optimization scheme. It's like slowly decreasing the error rate in an iterative way one piece at a time, by taking small steps. [By adjusting our variables. At each step, it looks at the slope of the error function (how steep it is) and moves in the direction that makes the error smaller.

In other words, we have a way to make our mistakes as small as possible by using Newton - Raphson method. We start with some guesses for the Right Answers, and then we keep making our guesses better and better until we're really close to the best solution.

Newton-Raphson Algorithm for Logistic Regression

- * Initializing the parameter vector 'w' to some initial values. (eg small Random values)

- * Repeat until convergence

- * Compute gradient of the error function with respect to the parameter vector 'w'

$$\underbrace{\nabla E(w)}_{\text{gradient vector}} = \underbrace{\Phi^T}_{\text{transpose of the design matrix} \rightarrow \boxed{N \times m \text{ Design matrix}}}(\underbrace{y}_{\text{predicted probabilities}} - \underbrace{t}_{\text{target values}})$$

- * Compute Hessian matrix (H) then its inverse matrix.

- € And update the parameter vector 'w' using the Newton Raphson update equation. then check the convergence criteria. If the criteria are met, stop the iteration.

$$H = \Phi^T R \Phi$$

$$H = \nabla\nabla E(w) = \sum_{n=1}^{N} y_n(1-y_n)\phi_n \phi_n^T$$

$$= \Phi^T R \Phi$$

$$\nabla E(w) = \sum_{n=1}^{N} (w^T \phi_n - t_n)\phi_n$$

$$= \Phi^T \Phi_w - \Phi^T t$$

$$H = \nabla\nabla E(w) = \sum_{n=1}^{N} \phi_n \cdot \Phi_n^T = \Phi^T \Phi$$

* The Newton - Raphson update then takes the form.

$$w^{(new)} = w^{(old)} - (\Phi^T \Phi)^{-1} \{ \Phi^T \Phi_w^{(old)} - \Phi^T t \}$$

$$= (\Phi^T \Phi)^{-1} \Phi^T t$$

The Newton Raphson update formulae for the logistic Regression model then becomes

$$w^{(new)} = w^{(old)} - (\Phi^T R \Phi)^{-1} \Phi^T (y-t)$$

$$= (\Phi^T R \Phi)^{-1} \{ \Phi^T R \Phi w^{(old)} - \Phi^T (y-t) \}$$

$$= (\Phi^T R \Phi)^{-1} \Phi^T R z.$$

→ This iterative process continues until parameters converge to values that minimize the error function, effectively finding the best-fitting logistic regression model for the given data.

⊛ Gradient (First Derivative) ⊛ Hessian (Second Derivative)

$\nabla E(w) = \Phi^T (y-t)$ $H = \Phi^T R \Phi$

⊛ Updated Equation (Newton - Raphson)

$$\therefore w(new) = w(old) - \underset{\uparrow H}{(\Phi^T T R \Phi)^{-1}} \Phi^T (y-t)$$

⊛ Gradient (First Derivative)

$$\nabla E(w) = \phi^T (y-t)$$

$\downarrow$ gradient vector

$\rightarrow$ transpose of design matrix ϕ

y: vector of predicted probabilities obtained using the logistic sigmoid function.

t: is the vector of actual target values (0 or 1)

⊛ Hessian (Second Derivative)

$$H = \theta^T R \phi$$

$\downarrow$ Hessian matrix

$\rightarrow$ diagonal matrix with elements R_{nn}.

⊛ Updated Equation (Newton-Raphson)

$$\therefore \; w(new) = w(old) - (\phi^T T R \phi)^{-1} \phi^T (y-t)$$

$\uparrow$ updated parameter

$\downarrow$ current parameter vector

$\uparrow$ H

$\downarrow$ gradient vector.

Q.4

b) we have seen in Q.3(c) that Solution for w is given by

$$[X^T R X]^{-1} X^T R Y$$

R is a diagonal matrix of weights

Y is the vector of target values.

X is the design matrix.

we also know that,

Newton-Raphson Update Equation for Logistic Regressionis given by

$$W_{new} = W_{old} - (X^T W X)^{-1} X^T (Y - P)$$

Now we apply the Newton-Raphson update to the cross-entropy error function for the logistic regression model.

we see that the gradient and Hessian of this error function are given by

$$\nabla E(w) = \sum_{n=1}^{N} (y_n - t_n)\phi_n$$

$$= \phi^T (y - t)$$

$$H = \nabla\nabla E(w) = \sum_{n=1}^{N} y_n(1-y_n)\,\phi_n \phi_n^T$$

$$= \phi^T R \phi$$

where we know that $\frac{d\sigma}{da} = \sigma(1-\sigma)$

R is N×N diagonal matrix

The Newton-Raphson update formula for the logistic regression model then

$$w^{(new)} = w^{(old)} - (\phi^T R \phi)^{-1} \phi^T (y-t)$$

$$= (\phi^T R \phi)^{-1} (\phi^T R \phi w^{(old)} - \phi^T (y-t)\}$$

$$= (\phi^T R \phi)^{-1} \phi R z. \rightarrow (a)$$

where z is an N-dimesional vector with elements

$$z = \phi w^{(old)} - R^{-1}(y-t)$$

⇒ we see that the update formala (a) ~~t~~is in WLS where R depends on the w.

at each iteration we use new weights. vector w to compute a revised weighting matrix for this reason, the algorithm is known as iterative reweighted least square.

Q4)

c) To show that error function of logistic regression is a Convex function of the parameter vector 'w'. and has a unique minimum solution with help of Hessian matrix, if its error function is positive semidefinite. To show double derivative always positive (≥ 0). So, from we prove curve is always (w).

* Error function in logistic Regression,

$$\Rightarrow \underset{\uparrow \text{ error function}}{E(w)} = -\sum \left(\underset{\uparrow \text{ target value}}{t_n} * \log(y_n) + (1-t_n) * \log(1-\underset{\uparrow \text{ predicted probability for } n^{th} \text{ data}}{y_n}) \right]$$

$$\Rightarrow H = \underline{\nabla\nabla E(w)}.$$

=> To show H is +ve semi-definite, which means all its eigen values are non-negative. If all eigen values are non-negative, implies error function is convex.

* Calculate the Gradient ($\nabla E(w)$).

$$\nabla E(w) = \Phi^T(y-t)$$

* Calculate Hessian Matrix ($\nabla\nabla E(w)$)

To compute this consider the derivation of logistic sigmoid function.

$$\nabla y / \nabla w = y(1-y)\Phi$$

where y is the predicted probability from the logistic sigmoid function.

3) Hessian matrix.

$\nabla\nabla E(w) = \Phi^{T}T(y\cdot(1-y)\Phi)$

* For Convexity:

The its Eigenvalues are non-negative, when all eigenvalues are non-negative. it confirms that the error function is convex.

→ matrix $\Phi^{T}\Phi$ is +ve Semidefinite matrix.

$y(1-y)$ is always +ve

→ multiplying the +ve Semi-definite matrix ($\Phi^{T}\Phi$) by a positive term, the resulting Hessian matrix will also be positive semidefinite

→ As a result, the error function in logistic regression is indeed convex, it has a unique minimum. The Unique minimum represents the best parameters (w that make the logistic regression model perform well