✱ Logistic regression (it not Regression model)

⑧ Logistic Regression

Linear model for classification (supervised learning)

✱ Probability based model

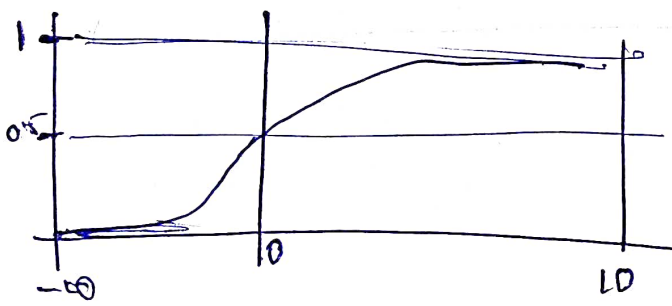
→ Logistic Regression is used for performing "classification problems".

→ It calculates the probability that a given value belongs to a specific class. If the probability is more than 50%, it assigns the value to that particular class,
else if probability is less than 50%, the value is assigned to the other class.

∴ We can say that logistic Regr. acts as a binary classifier.

✱ Here, As we give input x, we get y, where y is the probability or given var x belonging to a certain class.
thus, it is obvious that the value of y should we between 0 & 1.

So, we use Sigmoid function as the underlying function in Logistic Regression.



$$\sigma(x) = y = \frac{1}{1+e^{-x}}$$

Q. Why we sigmoid?

① the sigmoid function's range is bounded between 0 & 1, which is what we need.

② Easy to calculate than other functions which is useful during gradient descent calculation.

③ It is a simple way to introduce non-linearity to model

→

the logistic func is given as,

$$P(x) = \frac{e^{\beta_0 + \beta_1 x}}{1 + e^{\beta_0 + \beta_1 x}}$$

we have $\beta_0 + \beta_1 x$

$$P(x) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x)}} = \frac{e^{\beta_0 + \beta_1 x}}{1 + e^{\beta_0 + \beta_1 x}} \quad \text{--- (1)}$$

$$1 - P(x) = 1 - \frac{e^{h(\theta)}}{1 + e^{h(\theta)}} \quad (h(\theta) = \beta_0 + \beta_1 x)$$

$$1 - P(x) = \frac{1}{1 + e^{h(\theta)}} \quad \text{--- (2)}$$

From divide (1), (2)

$$\frac{(1)}{(2)} \Rightarrow \frac{P(x)}{1 - P(x)} = e^{h(\theta)} = e^{\beta_0 + \beta_1 x}$$

$$\beta_0 + \beta_1 x = \log\left(\frac{P(x)}{1 - P(x)}\right)$$

→ Logarithmic function.

Now we have,

$$\log\left(\frac{p(x)}{1-p(x)}\right) = \beta_0 + \beta_1 x$$

(logit function)

If we input (x), we will get p(x) (probability)

→ And we get p(x), based on the line with β_0, β_1 constants.

This is how logistic regression works

Now, if we have 2 classes (0, 1)

$$y = \begin{cases} 0 & , \text{if } p(x) < 0.5 \\ 1 & , \text{if } p(x) \geq 0.05 \end{cases}$$

0.5 - threshold

we can change thresholds

→ Based on the scenario, we should change threshold.

⚠ Cost function / loss function

For a single training instance,

$$\text{cost}(p(x), y) = \begin{cases} -\log(p(x)) & , \text{if } y = 1 \\ -\log(1-p(x)) & , \text{if } y = 0 \end{cases}$$

If $p(x) = 1$ & $y = 1$, cost / loss $= 0$

If $p(x) = 0$ & $y = 0$, cost / loss $= 0$

But,

If $p(x) = 0$ & $y = 1$, cost / loss $\to \infty$

If $p(x) = 1$ & $y = 0$, cost / loss $\to \infty$

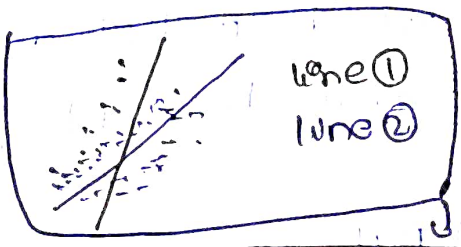
→ cost func for whole training set is given as

$$J(\theta) = \frac{-1}{m} \sum_{i=1}^{m} \left[y_i \cdot \log(\hat{p}_i) + (1-y_i) \cdot \log(1-\hat{p}_i) \right]$$

where, m - rows

y_i - expected y

$\hat{p}_i$ - predicted y

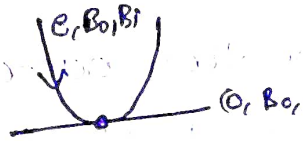


→ In logit func, RHS we have a line, And we will keep on adjusting the line w.r.t error so that we'll get correct probability // for train data

→ the values of parameter (θ) for which the cost func is (min) is calculated using the gradient decent algorithm (as showed in LR).

the partial derivative for above cost func is,

$$\frac{\partial}{\partial \theta_j} J(\theta) = \frac{1}{m} \sum_{i=1}^{m} \left[\sigma(\theta^T \cdot x_i) - y_i \right] \cdot x_j$$



multinomial logistic Regression

→ we can extend Logistic Regression for multi-class classification. the logic is, we train our model for each class and calculate the probability that a specific feature belongs to that class. once we have trained the model for all the classes, we predict a new values class by choosing that class for which prob (hoc(x)) is maximum.

* we rarely use it this way, because we have many other classification models for these scenarios.