Logistic Regression (IIT Guwahati) Prof. Biplab Bose

It is a type of classification technique.

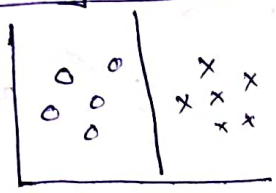
DS: U A_1 A_2 ... A_n D — class label

u_1
u_2
u_m

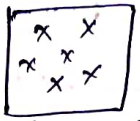
If 2 => binary classification

Consider training data set for classification model Generation

Binary Classification: class label yes / NO } True/false
1 / 0 etc.



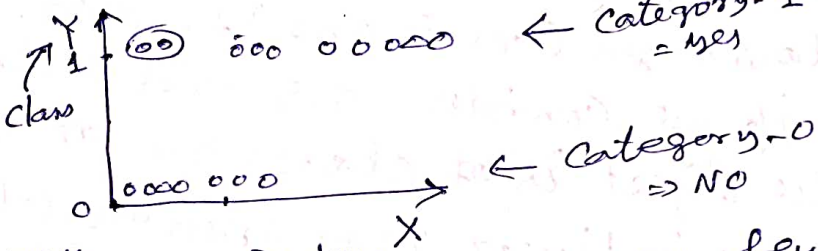
Category-0



Category-1

Plot Dataset

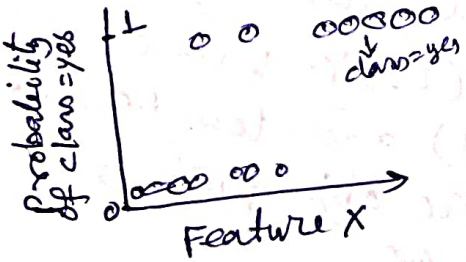
X	Y
x_1	0
x_2	1
x_3	0
...	...



Here, feature (X) value is very high implies the objects are of Category 1 or yes & low => Category 0 or no. But there are few objects (outliers) whose feature values are low yet of Category 1 or yes. For all datasets these type of data outliers exist.

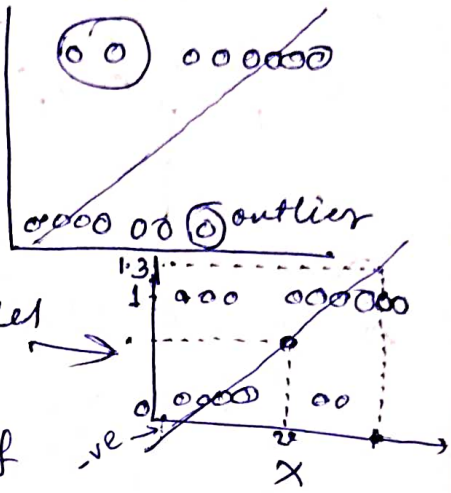
Let we want to build a classifier model which will be a numerical model. For this what we have done is that we have categorized the data into two classes yes & No & visually represented.

But to make the model more quantitative numerical we rearrange the data in different way as follows:



So we just change the vertical axis and scaled it from 0 to 1. In other words, the vertical axis represents the probability of class = yes or category 1. We already know that some objects are class = yes so for them probability = 1. Also some objects of class = no so their probability = 0. Now we will try to fit a model on these data points. So immediately it comes to our mind that we may use regression model (say, linear regression).

We may assume that the feature x have some linear relation with class variable, y. So the model may be $y = a + bx$ as ——→



It is observed that if we leave outliers we can visually see that its quite a good fit.

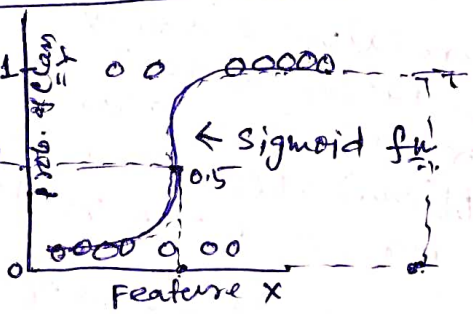
Similarly, if we consider y as probability of having class y = 1 we have the model ——→

Now let a new object is of x-value v, then its y value 0.65. So probability of class = yes (or category 1) is 0.65.

So if we consider 0.5 as a cutoff then we say that the object is of class = yes or category 1.

But this linear regression model has a problem :-

If a new obj of x-value is v, for which say y-value 1·3 which implies that the probability of class = yes is 1·3 which is not possible as prob. is in between 0 and 1. This is the major problem if we fit the linear regression model to the dataset.

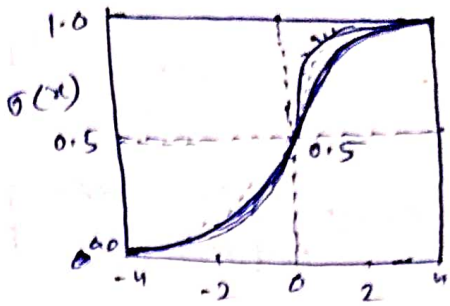


Avoid this problem is possible if we fit a instead of fitting linear eqn., we use equation bounded between 0 & 1. One such function used is sigmoid function. Then for any new obj. whatever may be the feature value the prob. value (i.e., y-value) never goes beyond 1 (i.e., greater than 1) and less than 0. If we use cut off = 0.5. So if the obj. prob. ≥ 0.5 say then class = yes otherwise class = no.

There are many type of sigmoid functions we may use but in logistic regression we use a particular sigmoid function described below.

The sigmoid function is:

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



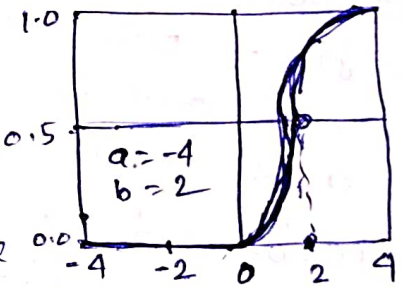
If $x \to \infty$ then $\sigma(x) \to 1$ and it saturates at 1. So this function does not go above 1.

If $x \to -\infty$, then $\sigma(x) \to 0$, so saturates at 0 and does not go below 0.

In between $(-\infty, \infty)$ it centres at $x=0$ where $\sigma(x)=0.5$ So this function gives the value in between 0 & 1 and is centered around 0. But in many real life examples, the x-value is not negative in that case we have to shift this curve on the right (i.e., +ve side). How can we do this? we have to write the function in a different fashion. Rather than using x, we will write $a+bx$.

$$\therefore \sigma(x) = \frac{1}{1+e^{-(a+bx)}}$$

for ex. if $a=-4$ $b=2$ then the curve is shifted to the +ve side and is centered around $x=2$



Thus by choosing the suitable values of a and b, we can easily fit the sigmoid function to our dataset (train data) to create that classifier (i.e., logistic regression).

Something more details about $\sigma(x)$:

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

$$0 \le \sigma(x) \le 1.$$

The probability value is in between 0 & 1. Also $\sigma(x)$ is in between 0 & 1. Thus sigmoid function itself give us the probability.

Thus, we may write, $pr(y=1/x=x_i) = pr(x_i) = \sigma(x_i) = \frac{1}{1+e^{-(a+bx_i)}}$

prob. of class = yes or $y=1$

$\therefore$ for any $x=x_i$, the prob. that it lies in category 1 is

$$pr(y=1/x=x_i) = p(x_i) = \sigma(x_i) = \frac{1}{1+e^{-(a+bx_i)}}$$

i.e., $\frac{p(x_i)}{1} = \frac{1}{1+e^{-(a+bx_i)}} \Rightarrow \frac{p(x_i)}{1-p(x_i)} = \frac{1}{e^{-(a+bx_i)}} = e^{a+bx_i}$

$$\Rightarrow z = a+bx_i = \ln \frac{p(x_i)}{1-p(x_i)}$$

$\therefore \quad a + bx_i = \ln\left[\frac{p(x_i)}{1-p(x_i)}\right]$ → $p(x_i)$ = prob. of being in category 1.

→ $1-p(x_i)$ = prob. of being in category 0.

$\therefore$ $p(x_i)$ ⇒ prob. of yes
$1-p(x_i)$ ⇒ prob. of no } when we ask an obj is in class= yes, we set $p(x_i)$ & $1-p(x_i)$

This ratio $\frac{p(x_i)}{1-p(x_i)}$ is called odds. i.e., odds of being in category 1.

We take log of that odds ⇒ $\ln\left(\frac{p(x_i)}{1-p(x_i)}\right)$ that is why this regression using this function is called logistic regression.

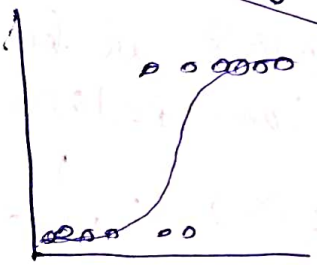
so that fn is called log odds or logit fn.

$\ln \frac{p(x_i)}{1-p(x_i)}$ → log odds or logit

Now we want to fit this function to our data. We have labelled data category-1 & category-0.

To fit ⇒ parameter estimation.

i.e., we have to estimate the value of a & b in $\sigma(x)$.



For this estimation we generally use Maximum Likelihood approach. There are many methods based on maximum likelihood approach. We are briefly explain what we are doing here for estimation.

As $p(x_i)$ probability for being in cate-1.

(1) → $p = pr(y=1/x=x_i) = p(x_i)$

(2) → $\therefore pr(y=0/x=x_i) = 1-p(x_i)$ ⇒ prob. for being in category-0.

$\therefore$ Integrating these two equations we may write

(merge)

(3) → $P(y=y_i/x=x_i) = p(x_i)^{y_i} \cdot [1-p(x_i)]^{1-y_i}$

y_i = 0 or 1 ; x_i = a particular value of x

Looks complicated but very simple.

If $y_i = 1$ (i.e., the sample (x_i, y_i) is in categ-1) then

$p(y=1/x=x_i) = p(x_i)^1 \cdot [1-p(x_i)]^0 = p(x_i)$ which is eqn (1).

Similarly if $y_i = 0$, $p(y=0/x=x_i) = 1-p(x_i)$ which is eqn (2)

31/ Thus we have clubbed these two equations (1) & (2) and get one generalized form (i.e., eq (3)). This eqn (3) helps us to calculate the likelihood and perform the maximum likelihood method.

How do we formulate the likelihood? For a sample (x_i, y_i) we have $p(x_i) = \frac{1}{1+e^{-(a+bx_i)}}$

(3). —— $pr(y=y_i | x=x_i) = [p(x_i)]^{y_i} [1-p(x_i)]^{1-y_i}$ where $y_i = 1$ or 0

Let the training dataset is ——→

X	Y
x_1	y_1
x_2	y_2
⋮	
x_n	y_n

For the algorithm

Initially assume some value of a & b.

then we use (x_1, y_1) and get from eq (3)

$pr(y=y_1 | x=x_1) = p(x_1)^{y_1} [1-p(x_1)]^{1-y_1} = P_1$, say

Similarly, using (x_2, y_2) we get P_2 for same a & b.

..... we get for (x_n, y_n) the ~~value~~ probability P_n.

∴ Each sample has associated prob.

$x_1\ y_1$	$\rightarrow P_1$
$x_2\ y_2$	$\rightarrow P_2$
⋮	
$x_i\ y_i$	$\rightarrow P_i$
⋮	
$x_n\ y_n$	$\rightarrow P_n$

So what is the total probability.

If we assume that these samples are independent then total probability $= P_1 \times P_2 \times \cdots \times P_n$

This total probability is known as likelihood of our model.

Let $L = P_1 \times P_2 \times \cdots \times P_n$

∴ For n-samples, the likelihood function:

$$L = \prod_{i=1}^{n} P_i = \prod_{i=1}^{n} p(x_i)^{y_i} [1-p(x_i)]^{1-y_i}$$

Next time consider another value of a & b and compute maximum likelihood. If it is more than consider this values of a & b and discard the previous values. Thus we have to estimate the values of a & b for which L will be maximum. Thus it is a maximization (i.e; optimization) problem.

So we have to find a & b that max(L).

Maximize $L \cong$ maximize $\log L$

So our optimization algorithm will maximize $\log L$

There are many algorithms like gradient descent algorithm can be used to maximize $\log L$. That will give optimum estimation of a & b.

Thus the model is $p(x_i) = \frac{1}{1+e^{-(a+bx_i)}}$ with estimated values of a & b.

Unlabelled

For ~~test~~ data x_i is given but y_i is not given.

The model give us prob, $p(x_i)$

If it is ≥ 0.5 say, then we say that it is of category-1 class otherwise of category-0 class.

*) Now let there are more than one-predictor.

Instead of x there are say $x_1, x_2, \cdots x_m$ features.

& $Y = 0$ or 1.

x_1 x_2 ... x_m	Y
$x_i \rightarrow x_1\ x_2 \ldots x_m$	

That time we may use multivariate sigmoid function:

$$\sigma(x_i) = \frac{1}{1+e^{-(a+b_1x_1+b_2x_2+\cdots+b_mx_m)}}$$

Similarly, we get, $a+b_1x_1+b_2x_2+\cdots+b_mx_m = \ln\left[\frac{p(x_i)}{1-p(x_i)}\right]$

All the process similar to previous one.

*) Now if many categories (classes) instead of binary categories:

ex. classify tumours as stage 1, stage 2, ... stage 5.

How should we handle? many algorithms.

one simple alg. is that = one versus all or one versus rest.

If there are p-classes then we consider as follows:

1) Category 1 = class 1 (stage 1)
2) Category 0 = rest classes. (Binary classes.)

$\Rightarrow$ Thus we have p-diff. binary regression models.

So we have P number of probability based binary models.

If we consider an unknown sample then calculate the probability of it by all P- classifiers, and finally take the model with max^m probability. If the max^m prob. occurs for model say

$P(Y = \text{stage 4} / X = x_i)$ and the prob. value = 0.7

that means that the class of new sample is stage 4 otherwise not in stage 4.

If not in stage 4 ⇒ Remove stage 4 objects from the training dataset and repeat the same process.

If max^m for stage 3 & prob value = 0.6 ⇒ x_i is in stage 3 category otherwise not in

category 3.

Repeat this process until we predict the category of object (feature vector $X = x_i$).

=