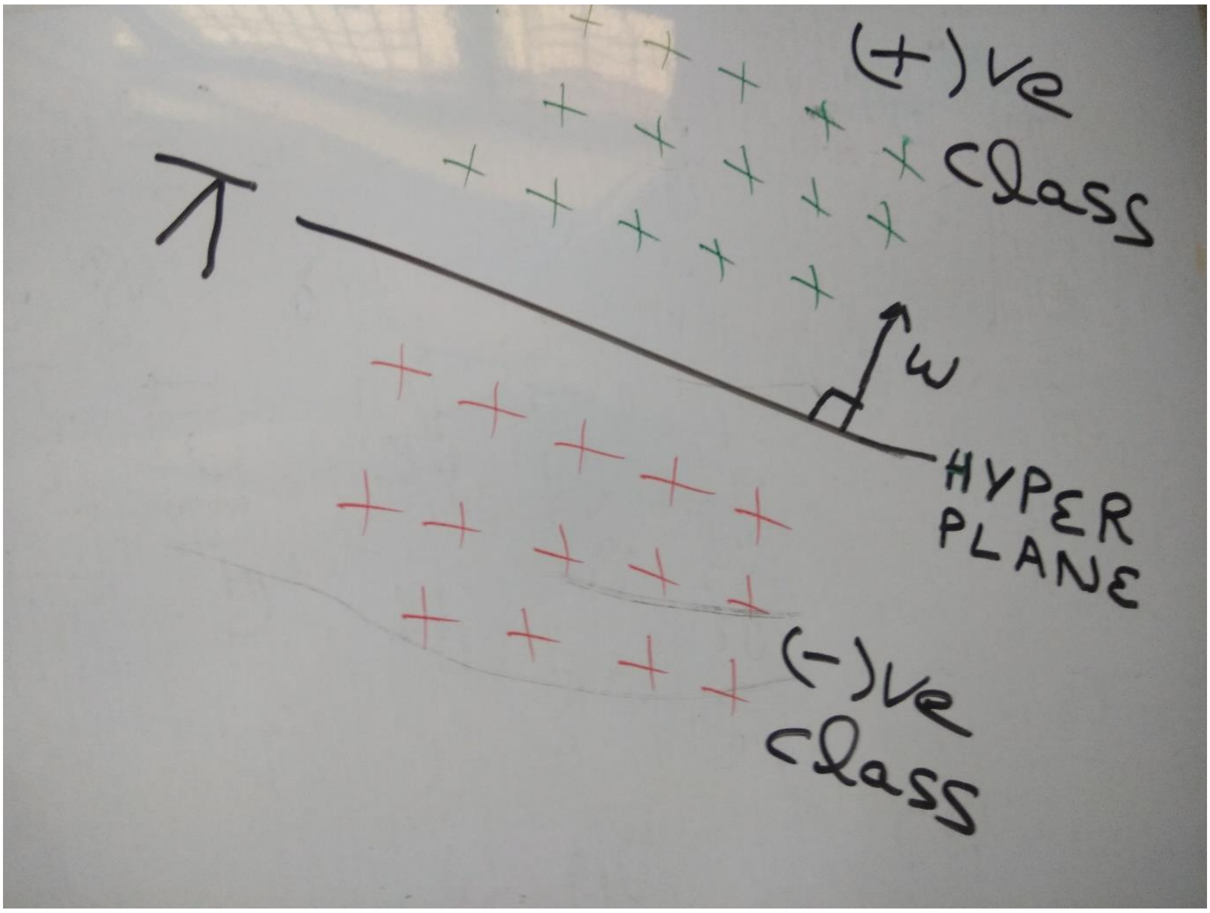
GEOMETRIC INTUITION OF LOGISTIC REGRESSION

What is Logistic Regression:

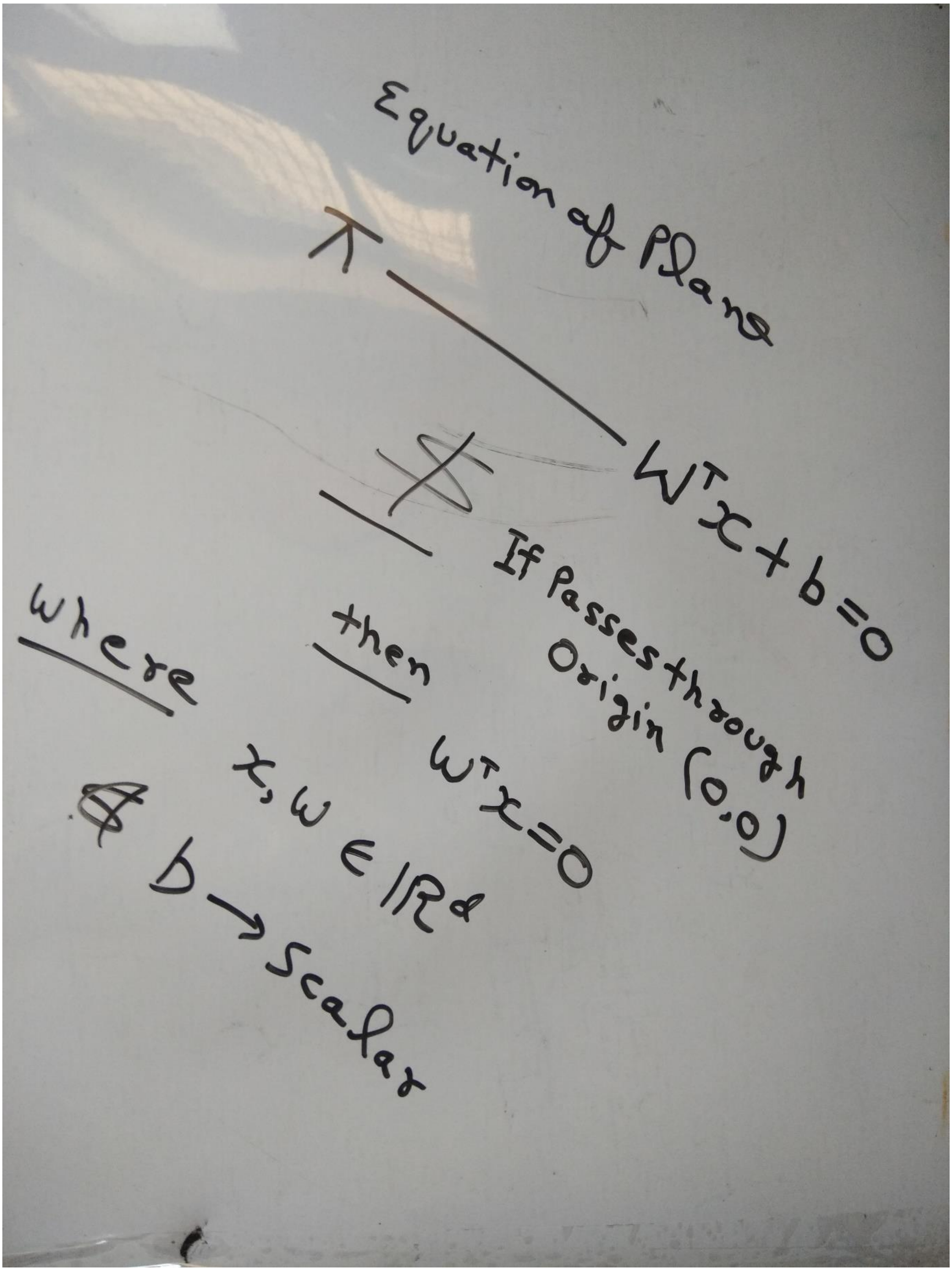
It is the multi-class classification technique used in Machine Learning to predict the class label i.e $\widetilde{y_i}$,s where $\tilde{y}_i$ belong to some class but not a real value. Let,s understand it geometrically:

ASSUMPTION : The biggest assumption before solving it is our data is linearly separable or almost linearly separable.

⇒ Let we have two classes of points i.e positive and negative.



In above picture we have: W⇒Normal to plane, Pi(π)⇒ Plane,So the equation of plane will be : $w^T x + b$=0 and If we pass through origin : $w^T x + b$=0 Where x, w ∈ $\mathbb{R}^d$ and b is scalar .



Equation of Plane

π

$W^T x + b = 0$

& If Passes through Origin (0,0)

then

$W^T x = 0$

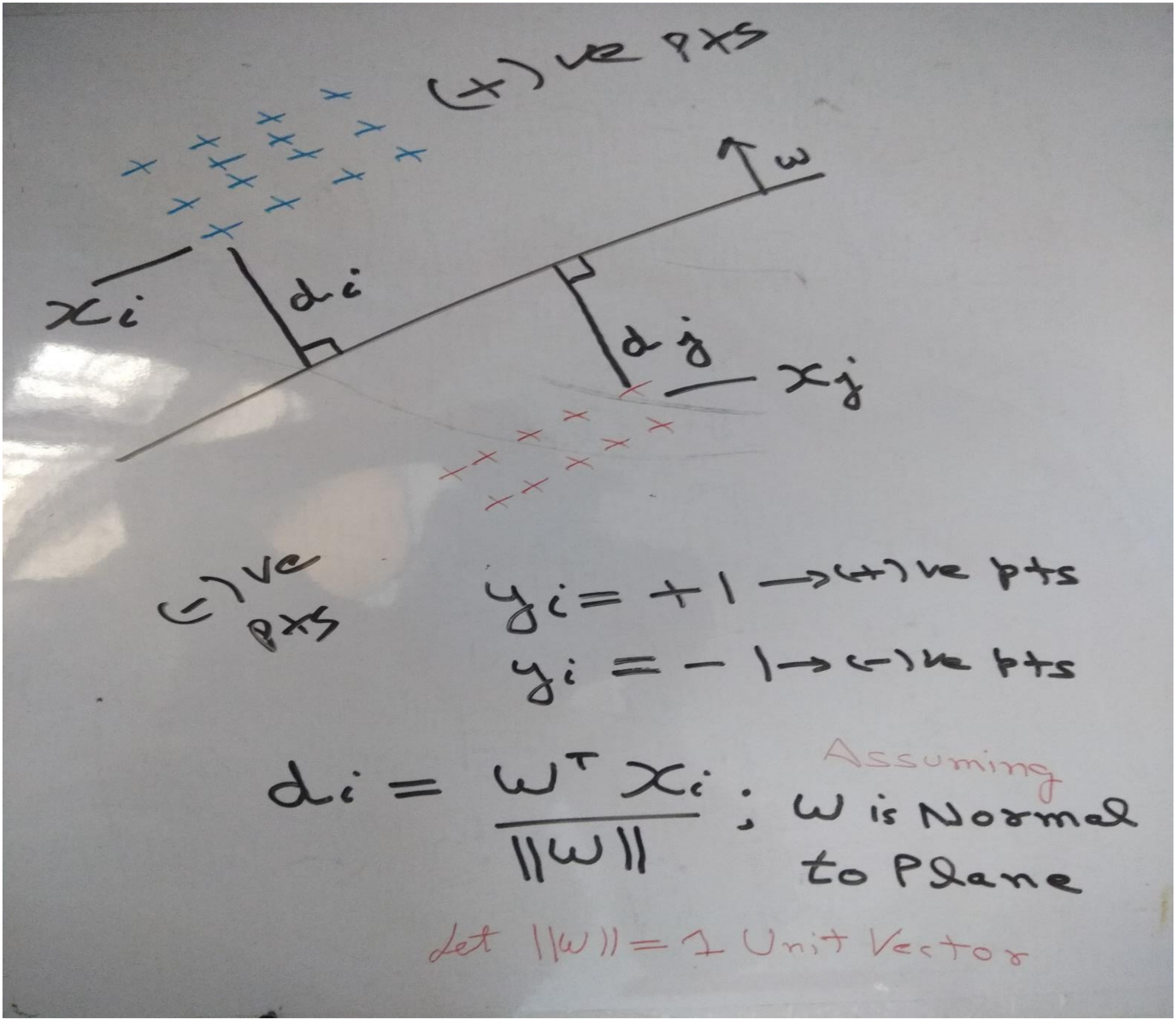
Where

$x, w \in \mathbb{R}^d$

& $b \rightarrow$ Scalar

So now we have are given DATASET D_n having positive and negative points.

TASK IN LOGISTIC REGRESSION IS : To find W and b to discover a plane such that it separates positive and negative point i.e we have to find (π).



$d_i = Distance\ from\ plane\ \pi\ to\ point\ x_i$ where W is normal to plane π

$$so \qquad d_i = \frac{\mathrm{W}^T x_i}{\|W\|} \quad \text{and} \quad d_j = \frac{\mathrm{W}^T x_j}{\|W\|}$$

Here, in above image we can find out the distance from any point to plane Pi(π). Also we assumed W is a unit vector and normal to plane.

So now comes to interesting part via seeing diagram in above picture i.e if we calculate:

1> Distance from positive point to plane (π) it will be positive .i.e d_i = $\mathrm{W}^T x_i$ > 0 because (W and x_i are on same side).

2> Distance from negative point to plane (π) it will be negative.i.e d_j= $\mathrm{W}^T x_j$ < 0 because (W and x_j are on opposite side).

NOW THE LOGICAL PART COMES: So our classifier says ⇒
if ($W^T x_i$) > 0 then y_i = +1 and if ($W^T x_i$) < 0 then y_i = -1

Now let,s take different cases:

case1: For positive points

let Pi(π) passes through origin

if x_i is positive i.e (+1) and ($W^T x_i$) > 0 our classifier saying

Now if π > 0 and y_i is positive

then W is correctly classifying the positive point.

case2: For negative points, let Pi(π) passes through origina and if y_i is negaitive i.e (-1) and $\mathrm{W}^T x_i$) < 0 our classifier saying

Now if (y_i* $W^T x_i$) > 0 and y_i is negative, because (-) * (-) = (+)

then W is correctly classifying the negative points.

OBSERVATION from above logical part is for both positive and negative points if (y_i* $W^T x_i$)>0 then L.R model is correctly classifying the points x_i ,s

NOW in the same fashion for negative points:

case3: If y_i is positive i.e (+1) and $W^T x_i$) < 0 then L.R is saying it is negative class then (y_i* $W^T X_i$) < 0 i.e miss classified point.

case4: If Yi is negative i.e (-1)and ($W^T x_i$) > 0 then L.R is saying it is positive class then (y_i* $W^T x_i$)) < 0 i.e miss classified point.

OBSERVATION from above logical part is:

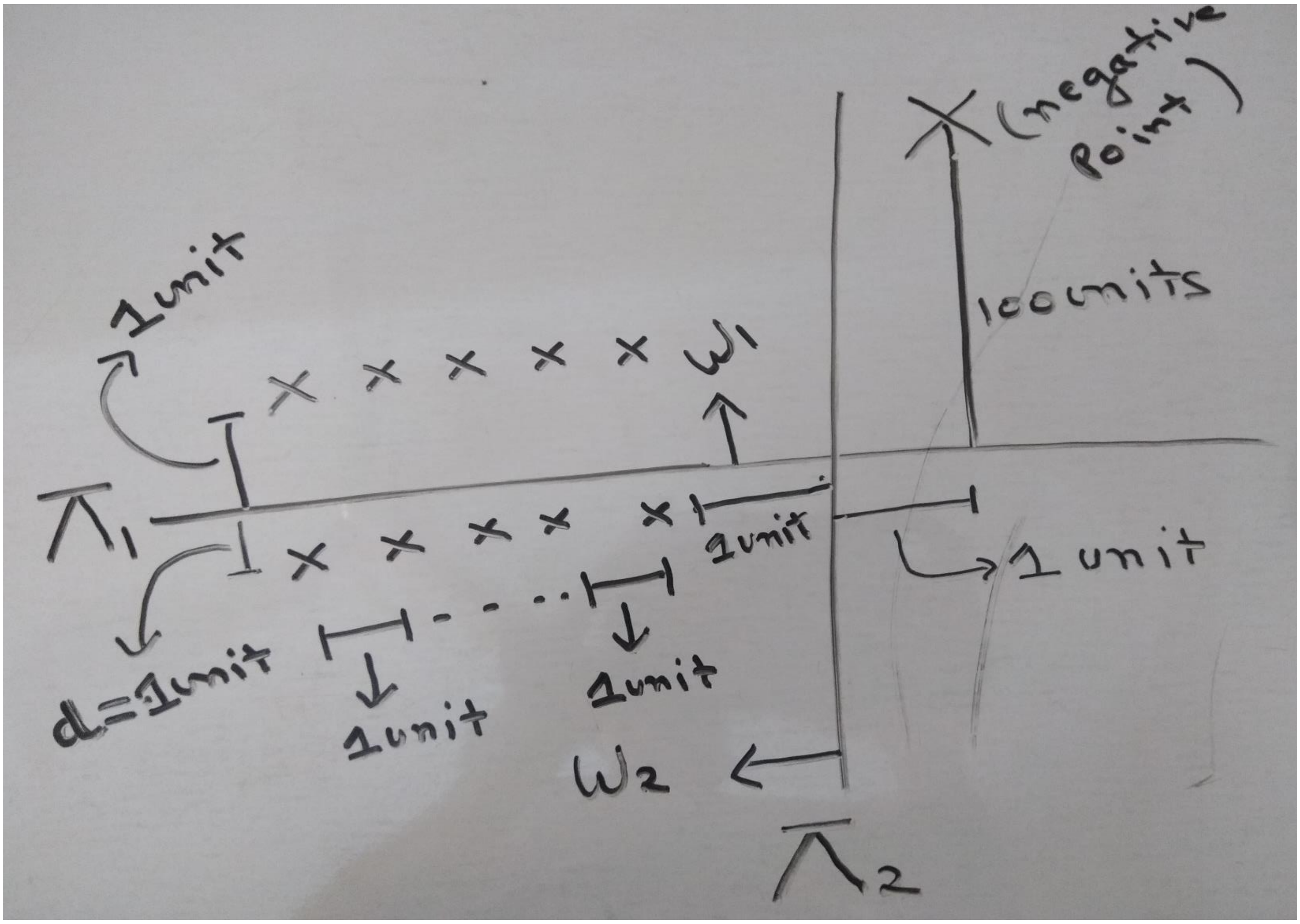
For both positive and negative points if (y_i* $W^T X_i$))<0 then L.R model is incorrectly classifying the points x_i,s.

At the end of the day we want our classifier V.good i.e minimize the no. of miss classification or maximize the no. of correctly classified point. i.e we want as many points possible to have (y_i* $W^T X_i$))>0

$$W^* = argmax(w) \sum_{i=1}^{n} (y_i W^T X_i)$$

Here in above equation X_i and y_i are fixed comes from (Dtrain),now we have to compute W variable here.

SO our optimization problem here is to find W* (optimal w) to maximize:
Now by understanding function we have (y_i* $W^T X_i$) is signed distance.



NOW BY OBSERVING FROM ABOVE IMAGE:

let we have 10 points equal distance from plane Pi-one(π_1) and X 100 unit distance from plane Pi-one(π_1):

CASE 1: Now if we choose W1 and Pi-one(π_1) as our separator,then ($\sum_1^n i$):(y_i* ${W_1}^T X_i$)=5+5–100=-90, here we are trying to maximize the signed distance.

CASE2: Now if we choose W2 and Pi-two(π_2) as our separator,then ($\sum_1^n i$):(y_i* ${W_2}^T X_i$)=1+2+3+4+5-1-2-3-4–5+1=1.Now as per our objective we will choose Pi-two(π_2) as our classifier. But if we think intuitively Pi-two(π_2) is terrible classifier i.e only one point is correctly classified.

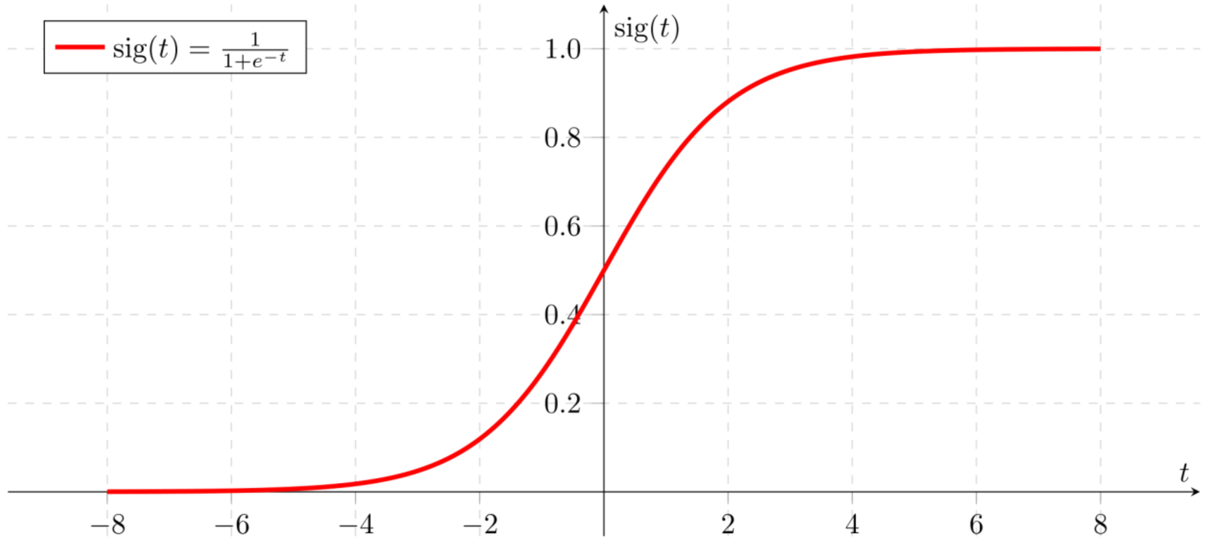
Now ,Accuracy of Pi-one(π_1)=(10/11) AND Accuracy of Pi-two(π_2)=(6/11)

Here the accuracy to maximum sum of signed distance Pi-two(π_2) is better then Pi-one(π_1) as a classifier.This is happening because of extreme point which is 100 units distance from Pi-one(π_1) in the direction of W1,i.e single extreme/outlier point is changing our model.So max-sum of signed distance can be impacted by outlier.

Now we have to modify W*=argmax(w)($\sum_1^n i$):(y_i* $W^T X_i$) by Technique called Squshing.

SQUASHING →

So instead of using simple signed- distance we will use :If signed- distance is small: Use as it is and If signed- distance is large: Make it a small value.So we want a function :When its value is small : increasing linearly.When its value becomes large : Tapper it off. Now, there are many such functions but One such function we have is SIGMOID FUNCTION because of its probabilistic interpretation.



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

Now if we apply sigmoid function to our W* we will have:

W*= (argmax(w) ($\sum_1^n i$): ($\frac{1}{1+exp\,(-(\,\boldsymbol{y_i}*\,W^T\boldsymbol{X_i})}$))

So this above function will be less impacted by outlier. Now we know the property of monotonic function: If x increases then G(x) increases and If x1>x2 then G(x1)>G(x2),Now if G(x) is monotonic then ::: 1>argmin F(x)=argmin G(F(x))::: 2>argmaxF(x)=argmax G(F(x)) NOW ⇒ We know that log is monotonic function and **log(1/x)=-log(x),**Now if we imply log on W* then W* will be:

W*=(arg max(w) ($\sum_1^n i$): log(($\frac{1}{1+exp\ (-(\ y_i * \ W^T X_i)}$)))))

W*=(arg max(w) ($\sum_1^n i$): -log($1 + exp\ (-(\ y_i\ *\ W^T X_i)$))))

W*=(arg min(w) ($\boldsymbol{\sum_1^n i}$): log($\mathbf{1} + \boldsymbol{exp}\ (-(\ y_i\ *\ W^T X_i)$…….equation 1

Now if there will be no 1 in the above equation then:

W* = (argmin(w) ($\sum_1^n i$): log($exp\ (-(\ y_i\ *\ W^T X_i)$

W* = (argmin(w) ($\sum_1^n i$): $(-(\ y_i\ *\ W^T X_i)$

W* = (argmin(w) ($\boldsymbol{\sum_1^n i}$): $(-(\ \boldsymbol{y_i}\ *\ W^T \boldsymbol{X_i})$))…….equation 2

i.e SUM of signed distances.

So in optimization problem of Logistic Regression with a small change using exp and log we have sum of signed distance which have huge problem of outlier so that,s why we will not use …(equation 2) for optimization problem.

Other formulation of W* using probabilistic method is as follows:

W*=argmin(w)($\boldsymbol{\sum_1^n i}$): σ -{Yi*log(P(i))+(1-Yi)*log(1-P(i))}

If we correlate this equation with……(equation 1),then P(i) — -> σ (${W_1}^T X_i$). So from ……(equation 1) we get W* optimal W is also called Weight Vector i.e W belong to $\mathbb{R}^d$ =<w1,w2,w3……..wd> WHERE d=no of features/dimentions. In L.R we have weight associated with every featuresIf w_i is positive , x_qi high ⇒ $w_i * x_q i$ will high ⇒ (i:1 to d ($w_i * x_q i$)) will high thenσ (Wi*$x_q i$) also high i.e P(y_q=Positive) will high.

In similar fashion : If Wi is nagative , $x_q i \ \ will$ high ⇒Wi*$x_q i$ will low ⇒ (i:1 to d (Wi*$x_q i$)) will low then σ (Wi*$x_q i$) also low i.e P(y_q=Positive) will low ⇒ P(y_q=Negative) will high.