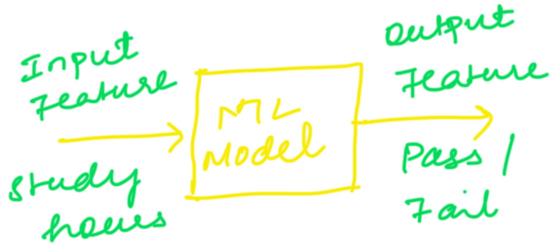
Logistic Regression (Binary classification)

Dataset

Input Feature	output Feature
study hours	Pass / Fail
2	Fail
3	Fail
4	Fail
5	pass
6	Pass
7	Pass

(Binary classification)

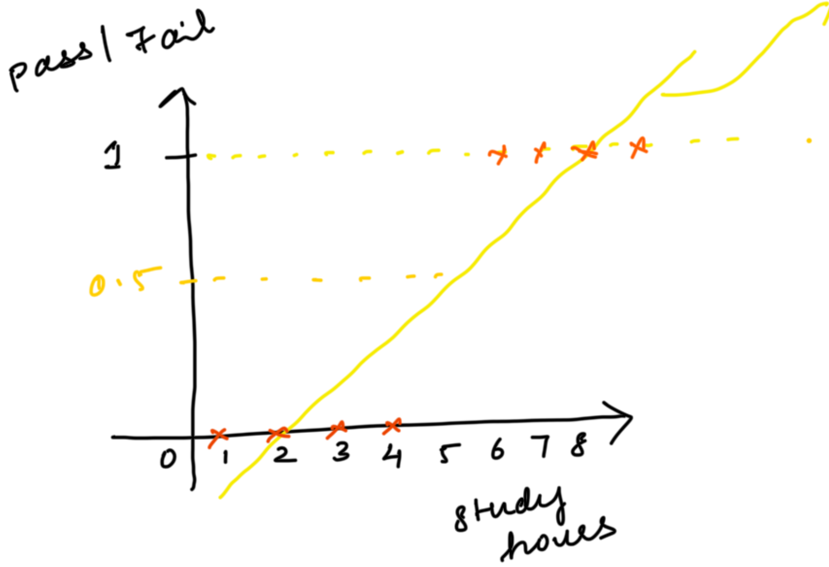


Why we cannot use Linear regression for classification?

1. outlier → Best fit line changes
2. >1 and <0 → squashing the line is not possible.

How Logistic Regression solves classification problem ?

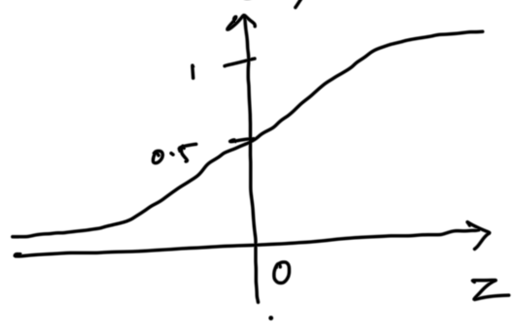
$$\boxed{h_\theta(x) = \theta_0 + \theta_1 x_1}$$



Sigmoid Activation

$$\frac{1}{1+e^{-x}}$$

$\sigma(z)$



$\Downarrow$

$z \geq 0$

$\sigma(z) \geq 0.5$

In Logistic regression also we will create a best fit line, and top of the best fit line we will apply the activation function

$$h_\theta(x) = \sigma(\theta_0 + \theta_1 x) \qquad \sigma = \frac{1}{1+e^{-z}}$$

$\Downarrow$

$$\boxed{h_\theta(x) = \frac{1}{1+e^{-z}}}$$

$z = \theta_0 + \theta_1 x_1$

Logistic regression hypothesis

Logistic regression cost function

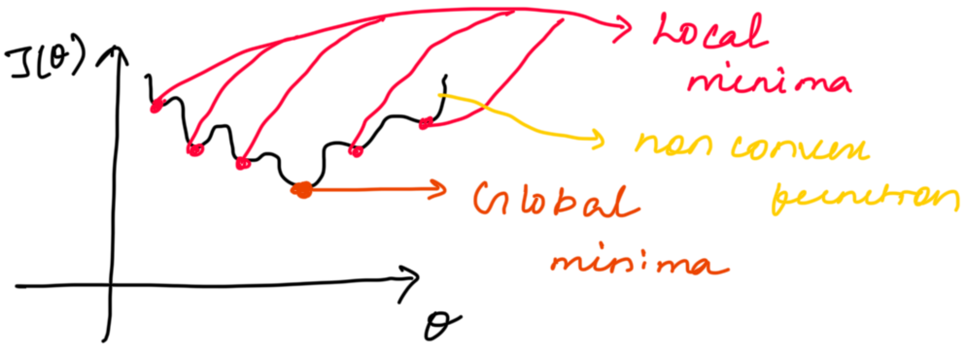
$$J(\theta_0, \theta_1) = \frac{1}{\ } \sum^{m} \left(h_\theta(x)^i - y^i \right)^2$$

$$J(\theta_0,\theta_1) = \frac{1}{2m}\sum_{i=1} \Big(\qquad \Big)$$

$$h_\theta(x) = \frac{1}{1+e^{-z}} \quad \Big| \quad z = \theta_0 + \theta_1 x_1$$

This will not provide gradient descent

This function provides non convex function



In the above example, my θ will stangent at one position.

Inorder to prevent this what we can do is that to bring the convex function

$$J(\theta_0,\theta_1) = \frac{1}{2m}\sum_{i=1}^{m}\left(h_\theta(x)^i - y^i\right)^2$$

we know that

$$h_\theta(x)^i = \frac{1}{1+e^{-z}} \quad \Big| \quad z = \theta_0 + \theta_1 x$$

Lets denote $\left(h_\theta(x)^i - y^i\right)^2$ as

$$\text{cost}\left(h_\theta(x)^i,\ y^i\right)$$

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

$\Downarrow$

we basically call this as log loss

this log loss help us to come up with a convex function. — so that we can achieve a global minima.

therefore,

$$cost(h_\theta(x)^i, y^i) = \underbrace{-y\log(h_\theta(x))}_{\Downarrow \; y=1} - \underbrace{(1-y)\log(1-h_\theta(x))}_{\Downarrow \; y=0}$$

therefore, the cost function can be denoted as.

$$J(\theta_0,\theta_1) = -\frac{1}{2m}\sum_{i=1}^{m}\left(y^i - \log(h_\theta(x)^i) - (1-y^i)\log(1-h_\theta(x)^i)\right)$$

⇓

this cost function will give me the concern function, probably we can call it as global minima.

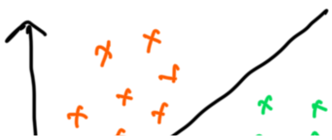
Minimize cost function $J(\theta_0, \theta_1)$ by changing θ_0 and θ_1

<u>convergence</u> <u>Algorithm</u>

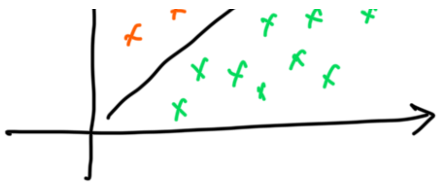
Repeat

$$\Big\{ \quad \theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1) \quad \Big| \; j = 0 \text{ and } 1 \quad \Big\}$$

Performance Metrics, Accuracy, Precision, Recall and F - Beta score



<u>Data set</u>



x_1	x_2	y	y
–	–	0	1
–	–	1	1
–	–	0	0
–	–	1	1
–	–	1	1
–	–	0	1
–	–	1	0

x_1 & x_2 → I/P feature

y → output feature

$\hat{y}$ → predicted feature.

derived from above data set.

① confusion Matrix

Basically its 2 × 2

Actual values

Predicted values	1	0
1	3	2
0	1	1

- wrong prediction
- correct prediction

Actual values

Predicted values	1	0
1	TP	FP
0	FN	TN

TP - True positive, when actual is 1 and predicted is 1

FP - False positive, when actual is 0 and Predicted is 1

FN - False negative, when actual is 1, and predicted is 0

TN - True negative, when

here, TP and TN

are the correct results, and FP and FN are the wrong results.

actual is 0, and predicted is 0

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + FN + TN}$$

Lets compute the accuracy for our dataset

$$\text{Accuracy} = \frac{3 + 1}{3 + 2 + 1 + 1}$$

$$= \frac{4}{7}$$

$$= 0.5714$$

Dataset — Binary classification

→ 1000 Data set
- → 900 → 1
- → 100 → 0

The above classification is imbalanced data set, and we cannot directly use accuracy.

$\Downarrow$

In order, to prevent this we can use precision and recall.

Precision

$$\text{Precision} = \frac{TP}{TP + FP}$$

$\Downarrow$

out of all the actual values, how many are correctly predicted.

Recall

$$\text{Recall} = \frac{TP}{TP + FN}$$

$\Downarrow$

out of all the predicted values, how many are correctly predicted.

1. When should we use precision and

when should we use recall ?

Use case ①

Spam classification

Mail → spam
Model → spam } Good

Mail → Not spam
Model → spam } Blender

Predicted \ Actual	1	0
1	TP	FP
0	FN	TN

FP → should be reduced

1 - spam
0 - Not spam

For this case we can use Precision

$$\text{Precision} = \frac{TP}{TP + FP}$$

Use case 2

To predict whether person has

diabities or not

Truth → Diabities
Model → Dees'nt Diabities
} → Blender.

Truth → Diabities
Model → Diabities
} Good

Truth → Not diabities
Model → Diabities
} retrain

		Actual Diabities	Actual No diabities
Predicted	diabities	TP	FP
Predicted	no diabities	FN	TN

should be reduced in this usecase.

$$\text{Recall} = \frac{TP}{TP + FN}$$

F - Beta score

Precission × Recall

$$\text{F- Beta Score} = (1+\beta^2)\frac{}{\text{Precission} + \text{Recall}}$$

① If FP and FN are both important

$\beta = 1$

therefore,

$$\text{F1 score} = (1+1^2)\frac{\text{Precission} \times \text{Recall}}{\text{Precission} + \text{Recall}}.$$

This is reffered as Harmonic Mean.

② If FP is more important than FN

$\beta = 0.5$

therefore,

$$\text{F0.5 score} = (1+0.5^2)\frac{\text{Precission} \times \text{Recall}}{\text{Precission} + \text{Recall}}$$

③ If FN is more important than FP

$\beta = 2$

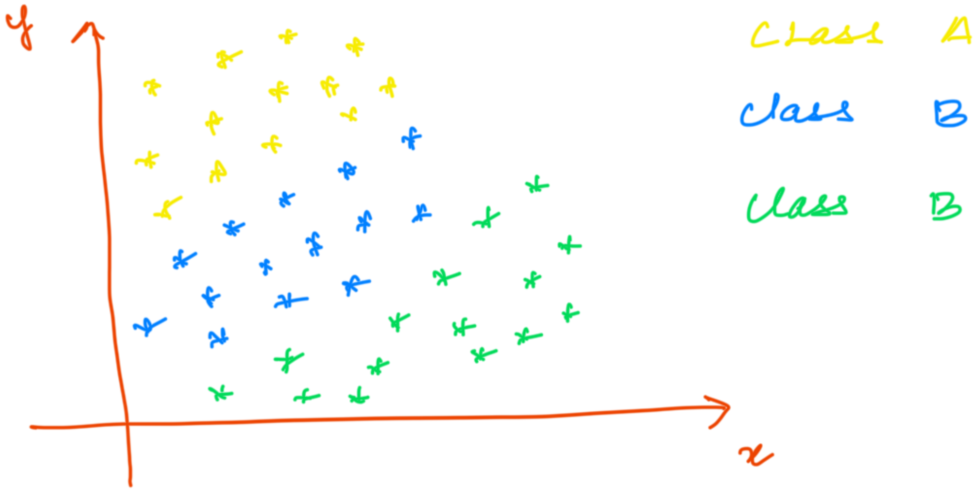
therefore,

$$\text{F2 score} = (1+2^2)\frac{\text{Precission} \times \text{Recall}}{\text{Precission} + \text{Recall}}.$$

Logistic Regression (one versus rest)

In Binary classification,

we can deal with only two output classification, what if i have more than two output classification.



The above representation is the multiclass classification problem

Multiclass classification problem can be solved with the help of one versus

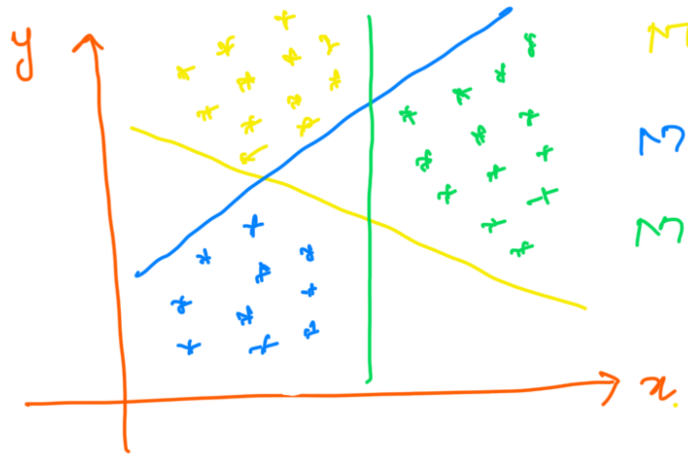
Rest (OVR)

<u>Dataset</u>   we have 3 classification O_1 O_2 O_3

f_1	f_2	f_3	o/p
-	-	-	O_1
-	-	-	O_2
-	-	-	O_3
-	-	-	O_1
-	-	-	O_3
-	-	-	O_2

with the help of one versus Rest what we do is that, we try to create internally multiple model, and each model act as a binary classification

here we have 3 classes, hence OVR create 3 internal model



$M_1 \rightarrow$ Binary classification

$M_2 \rightarrow$ Binary classification

$M_3 \rightarrow$ Binary classification

Basically,

$M_1 \rightarrow$ class A + class B and class C

$M_2 \rightarrow$ class B + class A and class C

$M_3 \rightarrow$ class C + class A and class C

$$\Downarrow$$

Each model, M_1, M_2, M_3 will perform binary classification

We form the results using one hot encoding.

O/P	O_1	O_2	O_3
O_1	1	0	0
O_2	0	1	0
O_3	0	0	1
O_1	1	0	0
O_3	0	0	1
O_2	0	1	0

Model

$M_1 \rightarrow$ Input $\rightarrow O_1$

$M_2 \rightarrow$ Input $\rightarrow O_2$

$M_3 \rightarrow$ Input $\rightarrow O_3$

New Test Data

$\rightarrow M_1 \rightarrow 0.25 \quad O_1$

$\rightarrow M_2 \rightarrow 0.20 \quad O_2$

$\rightarrow M_3 \rightarrow 0.55 \quad O_3$