Classification Algorithm.

1) Logistic Regression.

Logistic regression is used when the dependent feature is categorical.

Ex: Pass or fail in Exam.

Dataset

study hours	UPSC. o/p P/F	
1	P	← outlier
2	F	
3	F	
4	F	
5	F	
6	P	
7	P	
8	P	
9	F	← outlier
9	P	

⊗ Train data

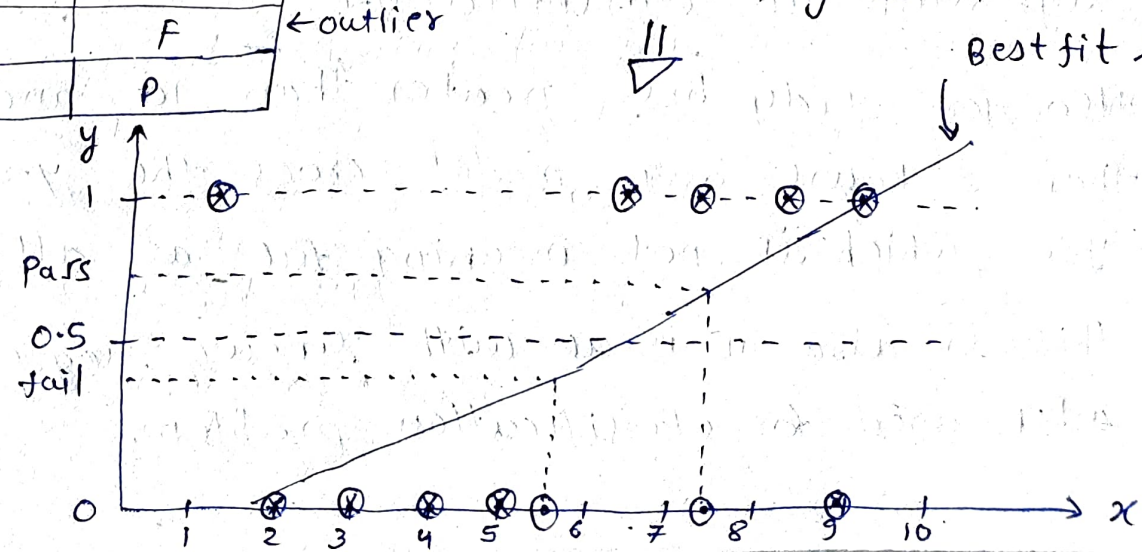
⊗ Test data

$y \leq 0.5 \Rightarrow$ fail

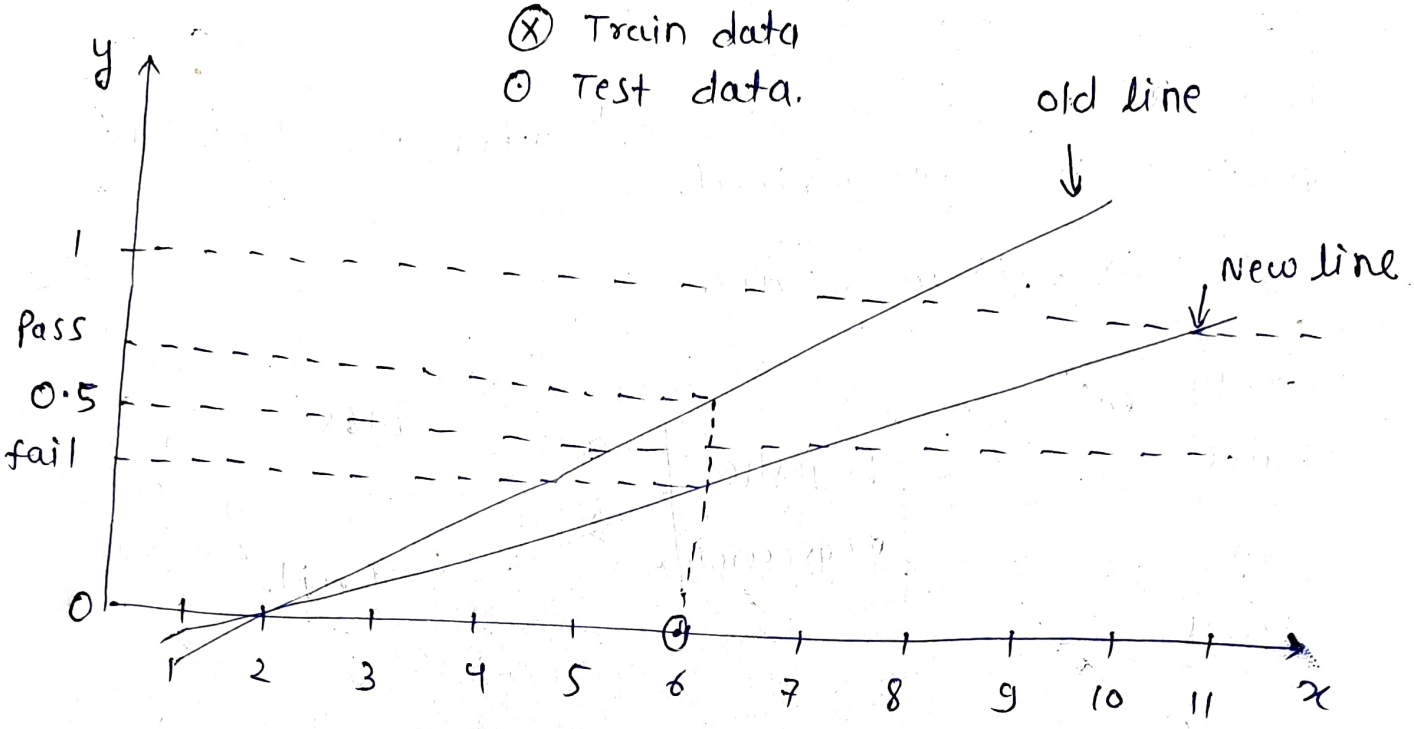
$y \geq 0.5 \Rightarrow$ pass

0.5 is threshold here.

Linear regression.



Due to outliers the above best fit line will change lets observe the changes.



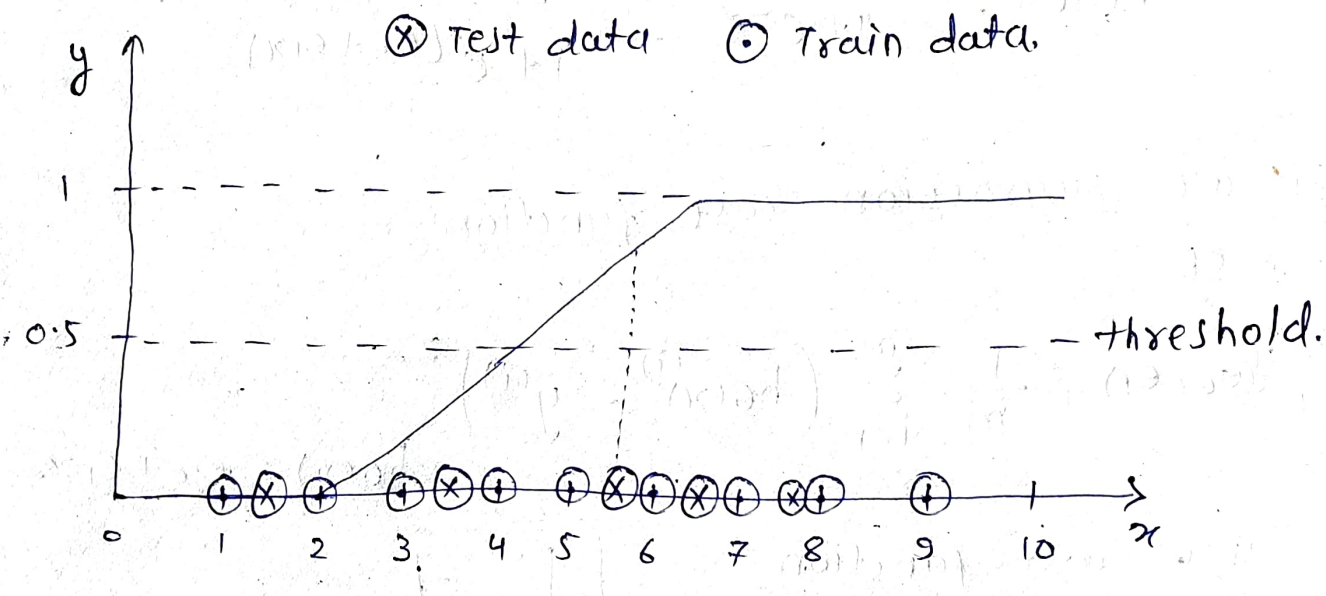
from above diagram it is clearly visible that with old best fit line at 6 study hours the candidate was passing the exam, but due to outliers now with new best fit line the same candidate is failing the exam as per prediction of our model.

This is the reason we cannot use linear regression in classification.

Also for study hrs greater than 10 and less than 2 hours our model cross the $y=1$ and $y=0$ which is not meaning ful at all.

This is also an issue with linear regression when used for classification problem.

So to avoid this what we can do is make best fit line parallel to x axis when it tries to cross. $y=1$ or $y=0$ lines. This is the Logistic Regression model.



Test data	model prediction.
1.5	F
3.6	F
5.4	P
6.7	P
7.9	P

To get this type of output that ranges from 0 to 1 we use sigmoid activation function. This function will squash the line parallel to x axis when it tries to cross. $y=1$ and $y=0$ line.

Step 1 Best fit line

$$h_\theta(x) = \theta_0 + \theta_1 x$$

② Apply sigmoid activation function to Best fit line.

lets say $h_\theta(x) = z = \theta_0 + \theta_1 x$

Sigmoid function $= \frac{1}{1+e^{-z}} = \frac{1}{1+e^{-(\theta_0+\theta_1 x)}}$

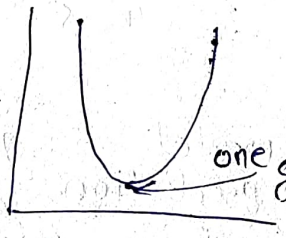
Linear regression cost function

MSE

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

$h_\theta(x) = \theta_0 + \theta_1 x$

This cost function has gradient descent as convex function



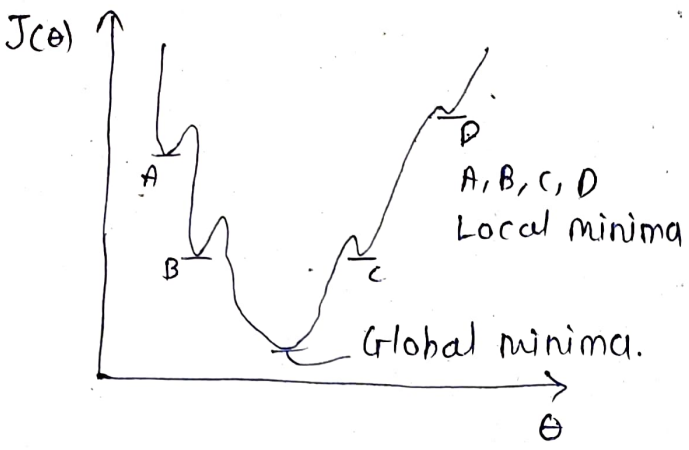
Now lets create logistic regression cost function using above cost function.

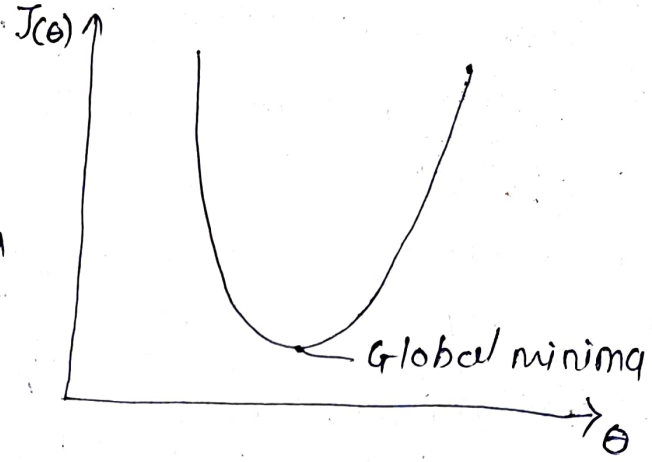
$h_\theta(x) = z = \theta_0 + \theta_1 x.$

$$\text{sigmoid}(z) = \frac{1}{1+e^{-z}}$$

$$= \frac{1}{1+e^{-(\theta_0+\theta_1 x)}}$$

But there is one very big issue with this is that it creates a non-convex gradient descent function. which has lot of local minima. where our gradient descent algorithm is stuck on the first local minima and never reaches global minima.





To fix this we must change the cost function.
we use log loss cost function.

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

Truth value.

where $h_\theta(x) = \dfrac{1}{1+e^{-(\theta_0+\theta_1 x)}}$

Simplified version of cost function

$$\boxed{cost(h_\theta(x), y) = -y \log(h_\theta(x)) - (1-y)\log(1-h_\theta(x))}$$

↳ This will create a convex gradient descent.

Aim is to minimise the cost function $J(\theta_0, \theta_1)$ by changing θ_0, θ_1 using convergence algorithm.

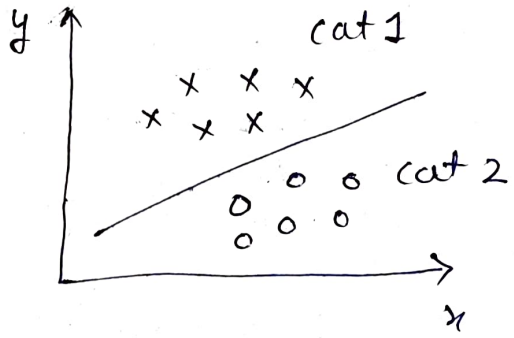
Repeat convergence

$$\{ \quad \theta_j = \theta_j - \alpha \frac{\partial J(\theta_j)}{\partial \theta_j} \quad \text{at } j=0,1$$

$$\}$$

Performance matrix

1) confusion matrix
2) Accuracy
3) Precision
4) Recall
5) F-Beta score.

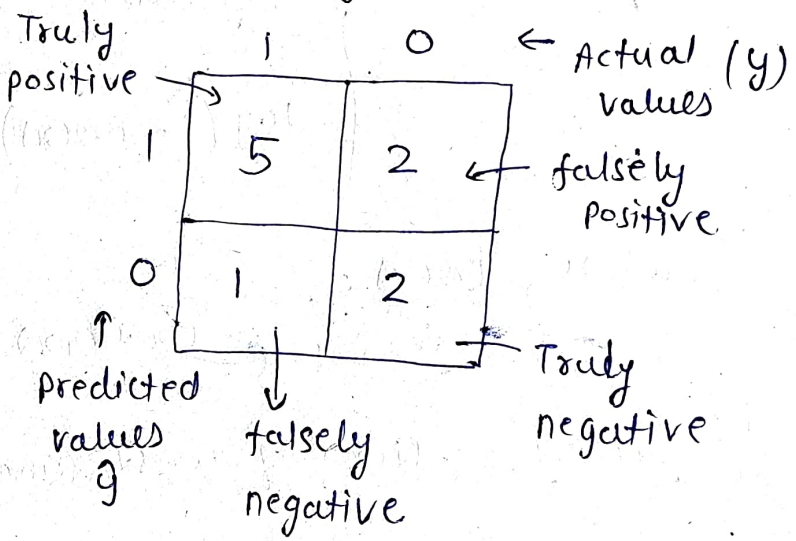


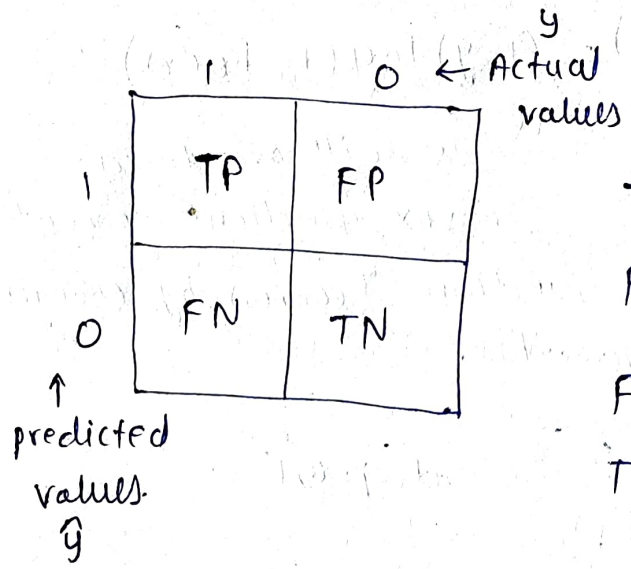
① confusion matrix

model prediction

f_1	f_2	o/p	$\hat{y}$
-	-	0	1
-	-	1	1
-	-	0	0
-	-	1	1
-	-	1	1
-	-	0	1
-	-	1	1
-	-	1	0
-	-	0	0
-	-	1	1

for binary classification





	Prediction
TP – Truly positive	correct
FP → Falsely positive	Incorrect
FN → Falsely Negative	Incorrect
TN → Truly Negative	correct

② Accuracy

It is defined as the ratio of correct predictions to total No. of observations / predictions

$$\text{Accuracy} = \frac{\text{correct predictions}}{\text{Total predictions}}$$

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

* for our confusion matrix

	1	0
1	5	2
0	1	2

$$\text{Accuracy} = \frac{5+2}{5+2+2+1} = \frac{7}{10} = 0.7$$

ie - our model is 70% accurate.

③ * Limitation of accuracy

consider a binary classification data set with

⇒ 1000 observations

900 Pass 100 fail. Imbalance dataset.

if we create a dumb model which predicts only pass. So for our data set it will predict correctly for 900 observations.

So accuracy = $\frac{900}{1000} = 0.9$ or 90%

conclusion: we cannot decide whether a model is good or bad by just considering accuracy.

③ Precision

precision is defined for pass or fail. ie precision of pass or precision of fail.

$$\text{precision} = \frac{\text{correctly predicted pass or fail}}{\text{Actual pass or fail}}$$

$$(\text{Precision})_{\text{pass}} = \frac{TP}{TP+FP}$$

Ex: $(\text{precision})_p = \frac{5}{7}$

$$(\text{Precision})_{\text{fail}} = \frac{TN}{TN+FN}$$

$(\text{precision})_f = \frac{2}{3}$

* The main aim in precision is to reduce wrong predictions. (ie False positive and False negative)

Example 1) Spam prediction.

if our mail is ham but our model predicts it as spam, we may miss out on important mails, we must reduce this, false negatives.

2) Diabetes prediction

if we have diabetes, but our model predicts that we don't have diabetes, we will be in quite danger of getting seriously ill.
we must reduce this, False negatives.

④ Recall

Recall is also defined for pass or fail.

$$\text{Recall} = \frac{\text{correctly predicted pass or fail}}{\text{Total prediction of pass or fail}}$$

$$(\text{Recall})_{pass} = \frac{TP}{TP+FN}$$

$$(\text{Recall})_{fail} = \frac{TN}{TN+FP}$$

	1	0
1	TP	FP
0	FN	TN

Example

	1	0
1	5	2
0	1	2

$$(\text{Recall})_p = \frac{5}{5+1} = \frac{5}{6}$$

$$(\text{Recall})_f = \frac{2}{2+2} = \frac{2}{4} = \frac{1}{2}.$$

Example

predict if Tomorrow is going to crash.

model for consumer

	1	0
1	TP	FP
0	FN	TN

model for companies.

Aim: Aim will be to reduce False negative, so that people can remove money in time.

FN ↓↓

Aim: Aim is to reduce False positive, so that companies can sell their stocks to reduce losses.

FP ↓↓

⑤ F-Beta score F-β ⇒ 0 to 1 - optimal (0 ← worse)

It is weighted harmonic mean of precision and recall. β parameter determines the weight of recall in the combined score.

$\beta < 1$ ⇒ more weight to precision

$\beta > 1$ ⇒ more weight to Recall.

$\beta = 0$ ⇒ only precision.

$\beta = \infty$ ⇒ only Recall.

$$F\text{-}\beta \text{ score} = \frac{(1+\beta^2)\,(\text{Precision} \times \text{Recall})}{\beta^2(\text{precision}) + \text{Recall}}$$

1) if we need to give equal importante to FP and FN reduction. ($\beta = 1$)

$$\text{f-1 score} = \frac{2 \times P \times R}{P + R}$$

2) if FP is more important than FN $\beta = 0.5$

$$\text{F-0.5 score} = \frac{1.25\ P \times R}{0.25\ P + R}$$

3) if FN is more important than FP ⇒ $\beta = 2$

$$\text{F-2 score} = \frac{5\ P \times R}{4P + R}$$