CLASSIFICATION

★ Logistic Regression :

Logistic regression is a classification algorithm for categorical variables.

★ When is logistic regression suitable ?

- If your data is binary.-
  0/1 , yes/No , True/False
- If you need probabilistic results
- When you need a linear decision boundary
- If you need to understand the impact of a feature.

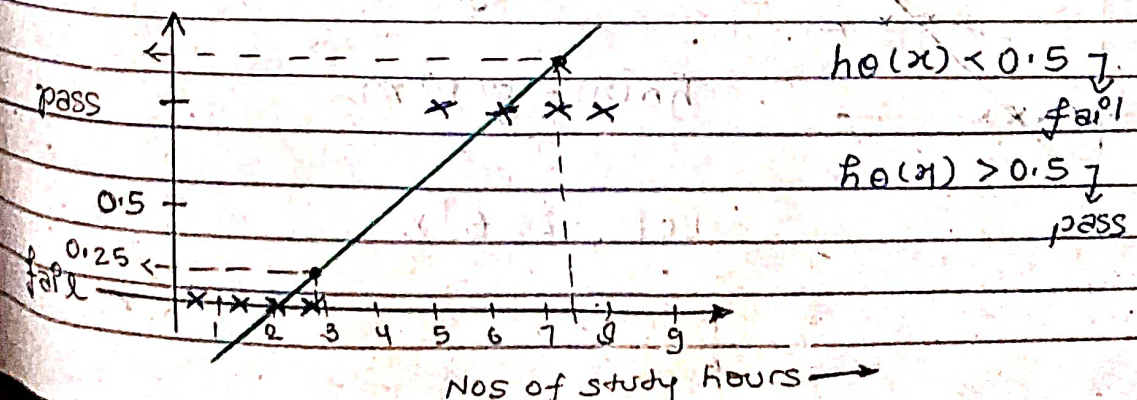
ie Email : spam/not spam ?
Online transactions : Fraudulent (Yes/No) ?
Tumor : Malignant / Belign ?

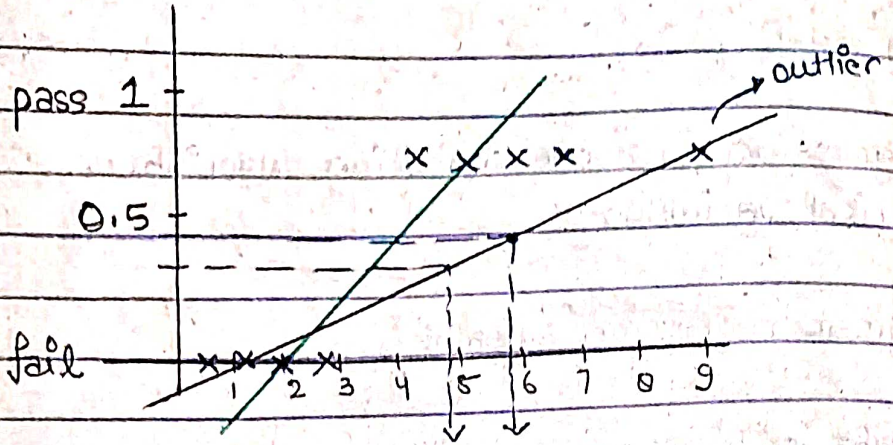
$y \in \{0, 1\}$

0 : 'negative class' (eg. Benign tumor)

1 : 'positive class' (eg. malignant tumor)

★ Why not linear regression ?





$h_\theta(x) \geq 0.5 \Rightarrow 1 \Rightarrow$ pass

(but here fails)

- Threshold classifier output $h_\theta(x)$ at 0.5:
  If $h_\theta(x) \geq 0.5$, predict "$y = 1$"
  If $h_\theta(x) < 0.5$, predict "$y = 0$".

- Classification: $y = 0$ or 1

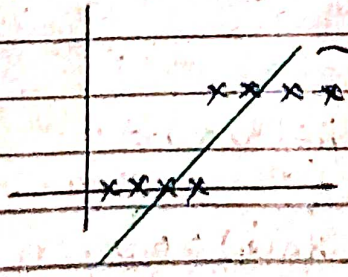
  $h_\theta(x)$ can be > 1 or < 0

- Logistic regression: $0 \leq h_\theta(x) \leq 1$

Hypothesis Representation:

$$h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \dots + \theta_n x_n$$

$$h_\theta(x) = \theta^T X$$



$h_\theta(x) = \theta^T X$ (hyperplane)

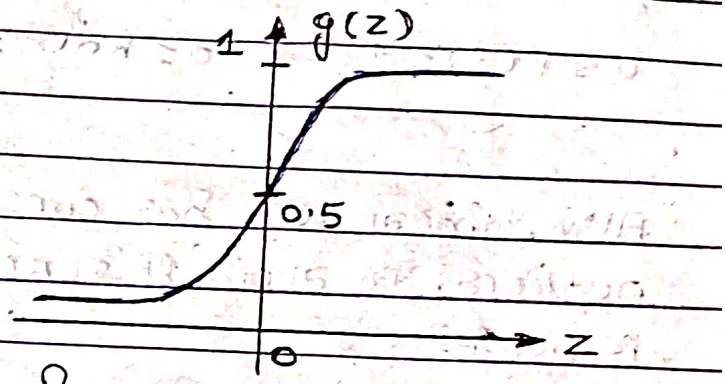
$$h_\theta(x) = g(\theta^T X)$$

Let, $z = \theta^T x$

$h_\theta(x) = g(z)$

$h_\theta(x) = \frac{1}{1+e^{-z}}$ ← [sigmoid or logistic func]

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



$\{ g(z) \geq 0.5 \text{ when } z \geq 0 \}$

- The idea of logistic regression is to find a relationship between features & probability of particular outcomes. This type of problem is referred to as Binomial Logistic Regression.

- Multinomial Logistic Regression deals with situation -ship where the response variable can have three or more possible values.

- With binary classification, Let 'x' be some feature & 'y' be the output which can be either 0 or 1.

- The propability that 'y' can be 1 given it's input can be repre-sented as —

$$P(y = 1 \mid x)$$

- If we predict the probability via linear regression we can state it as —

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

where, $h_\theta(x) = P(y = 1 \mid x)$

Linear regression model can generate the predicted probability as any number ranging from negative to positive infinity, whereas probo of an outcome can only lie b/w

$$0 \le P(X) \le 1 \rightarrow 0 \le h_\theta(x) \le 1$$

Also, linear reg. has considerable effects on outliers. To avoid this problem log function is used

$$\log\left(\frac{P(X)}{1-P(X)}\right) = \theta_0 + \theta_1 x$$

($\frac{P(X)}{1-P(X)}$ → odds)

(log(odds) → logit or log odds function)

The odd signifies probo of success to probo of failure.

Therefore, in logistic regression, linear combination of input are ~~add~~ mapped to log-odds - the output being equal to 1.

↓ inverse of above func°

↓

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

sigmond func°

Logistic regression model want, $0 \leq h_\theta(x) \leq 1$

$h_\theta(x) = \theta^T x$

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

Interpretation of Hypothesis Output:

$h_\theta(x)$ = estimated prob. that $y=1$ on input x

eg: If $x = \begin{bmatrix} x_0 \\ x_1 \end{bmatrix} = \begin{bmatrix} 1 \\ \text{tumor size} \end{bmatrix}$

$h_\theta(x) = 0.7$

tell patient that 70% chance of tumor being malignant

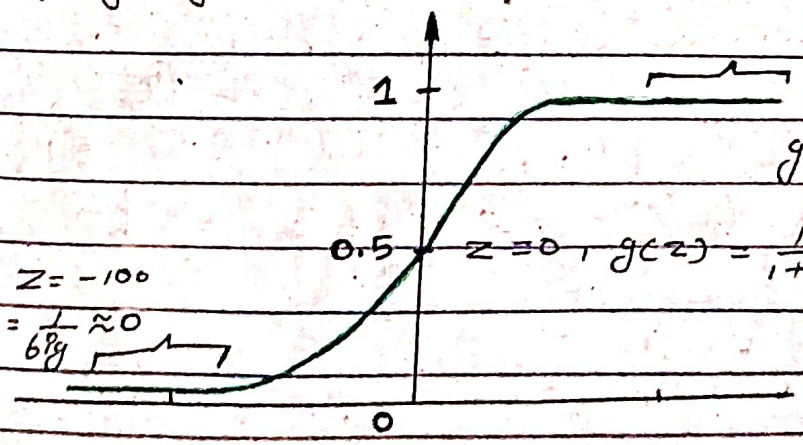
"probability that $y=1$, given x parameterized by θ"

$P(y=0 \mid x;\theta) + P(y=1 \mid x;\theta) = 1$

$P(y=0 \mid x;\theta) = 1 - P(y=1 \mid x;\theta)$

The equn essentially says that given input x & parameters θ, the prob. of either $y=0$ or $y=1$ happening is 100%.

Benefitting by sigmoid function



$g(z) = \frac{1}{1+e^{-z}}$; $0 \leq g(z) < 1$

Decision Boundary :

$h_\theta(x) = g(z)$

$z = \theta_0 + \theta_1 x$

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

z

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

Yes : $h_\theta(x) = 1$ (prediction True)

$h_\theta(x) \geq 0.5$

$g(z) \geq 0.5$

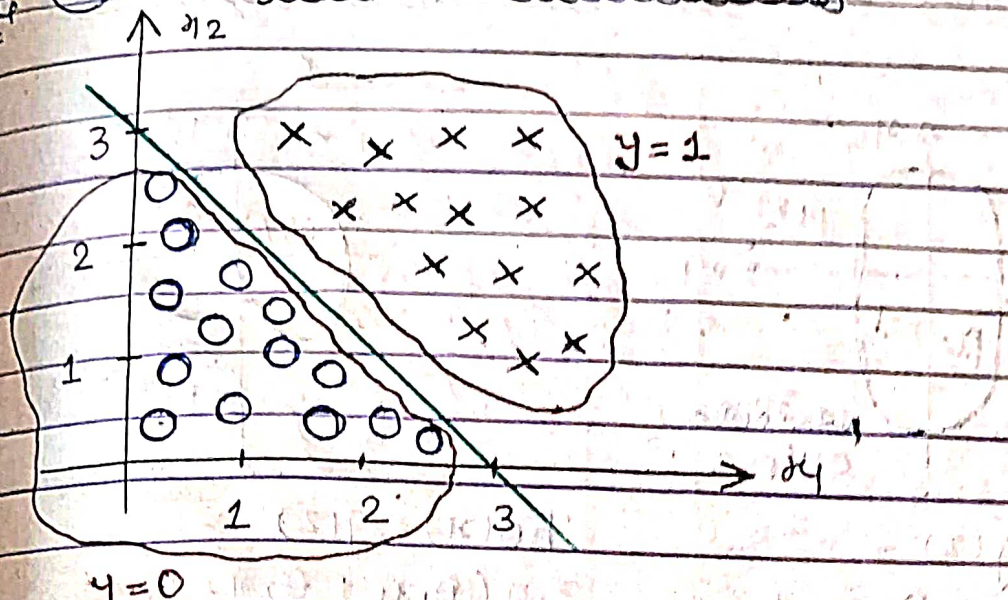
$z \geq 0$ ($z = -\infty$ to $+\infty$)

$\theta_0 + \theta_1 x \geq 0$ (the model predicts 1)

when, $\theta_0 + \theta_1 x \leq 0$

$h_\theta(x) = 0$: false

Eg (1) Linear decision Boundary



$$h_\theta(x) = g(z) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_2)$$

$\theta_0 = -3$, $\theta_1 = 1$, $\theta_2 = 1$

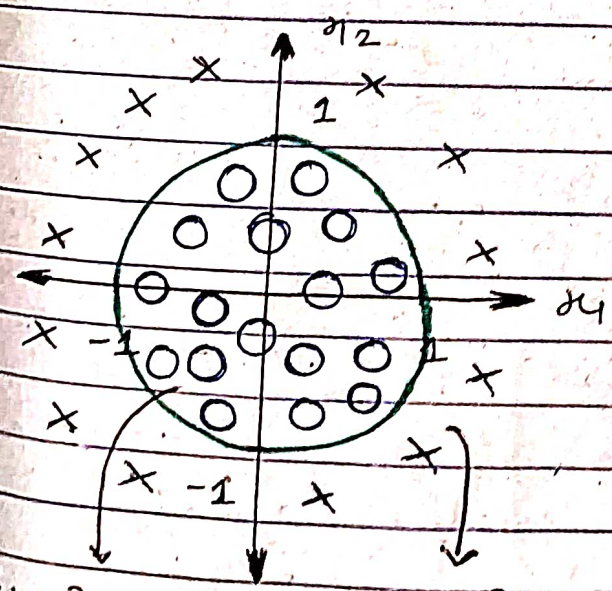
decision boundary :

$$\boxed{z = \theta_0 + \theta_1 x = 0} \text{ if linear}$$

$$z = x_1 + x_2 - 3 = 0$$

$$\boxed{x_1 + x_2 = 3}$$

Eg (2) Non-Linear decision boundary



$$h_\theta(x) = g(z)$$
$$= g(\theta_0 + \theta_1 x_1^2 + \theta_2 x_2^2)$$

$\theta_0 = -1$, $\theta_1 = 1$, $\theta_2 = 1$

$$z = x_1^2 + x_2^2 - 1 = 0$$
$$x_1^2 + x_2^2 = 1$$

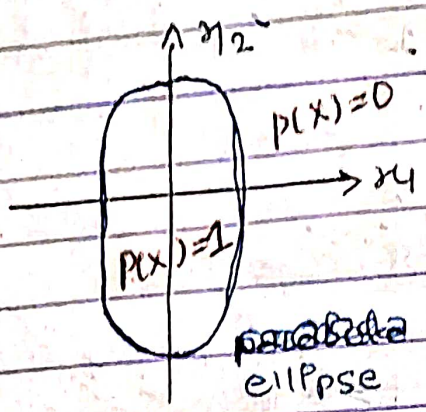
[decision-boundary].

$x_1^2 + x_2^2 < 1$: $h_\theta(x) = 0$, $P(x) = 0$

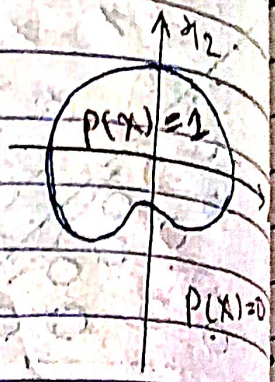
$x_1^2 + x_2^2 \geq 1$: $h_\theta(x) = 1$, $P(x) = 1$

$x^2 + y^2 = r^2$ (equn of circle) centre $(0,0)$ & radius $= r$

i.e. ③ Higher-order Non-Linear decision boundaries



$h_\theta(x) = g(z) =$
$g(\theta_0 + \theta_1 x_1 + \theta_2 x_2 +$
$\theta_3 x_1^2 + \theta_4 x_1 x_2 +$
$\theta_5 x_2^2)$



$h_\theta(x) = g(z)$
$= g(\theta_1 x_1 + \theta_2 x_2 +$
$\theta_3 x_1^2 + \theta_4 x_1 x_2 +$
$\theta_5 x_2^2 + \theta_6 x_1^3 + \dots + \theta_n)$

Cost function for logistic regression :

Training Set

	tumor size (in cm) (x_1)	...	patient's age (x_n)	malignant ? (Y)
$i=1$	10		52	1
	2		73	0
	5		55	0
	12		49	1
$i=m$	...		...	...

$i = 1, \ldots, m$
[training examples].

target Y is 0 or 1

$j = 1, \ldots, n$
[features]

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

or

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

How to choose ?

parameter = θ^T

$[\theta_0\ \theta_1\ \theta_2 \ldots \theta_n]$

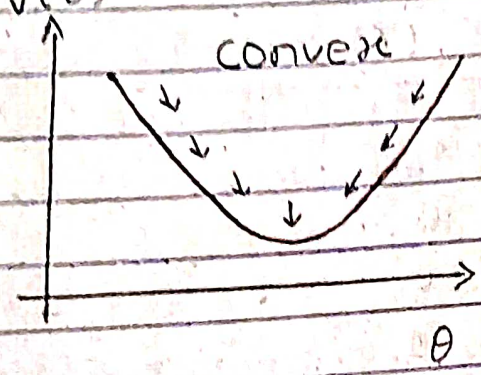
Cost function :

Linear reg : $J(\theta) = \frac{1}{m}\sum_{i=1}^{m} \frac{1}{2}\left(h_\theta(x^{(i)}) - y^{(i)}\right)^2$

$\text{Cost}\left(h_\theta(x^{(i)}), y^{(i)}\right) = \frac{1}{2}\left(h_\theta(x^{(i)}) - y^{(i)}\right)^2$

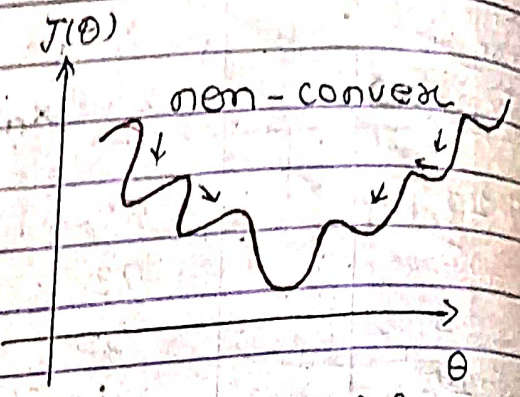
*linear regression

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



*logistic regression

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



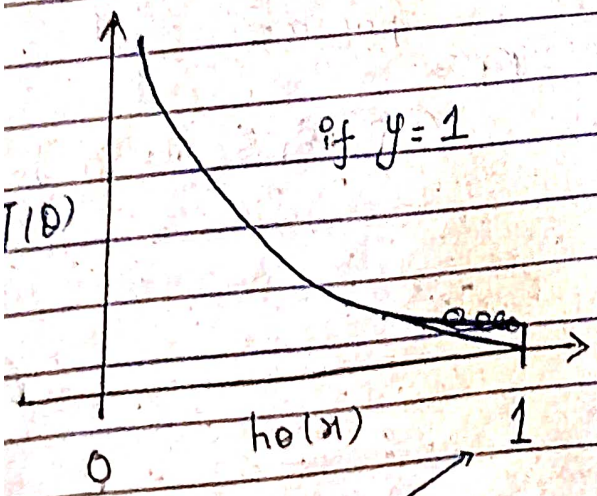
multiple local minimas-

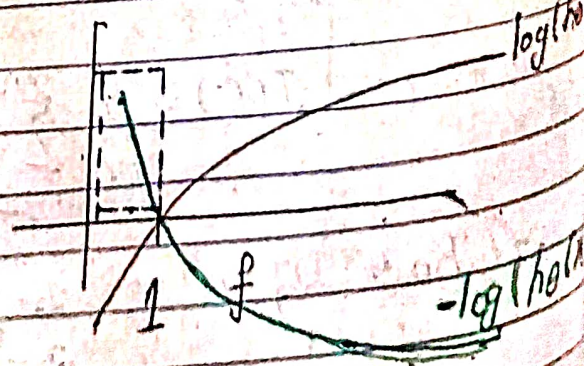
*logistic regression cost function →

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

"or"

$L(h_\theta(x^{(i)}), y^{(i)})$ Loss function





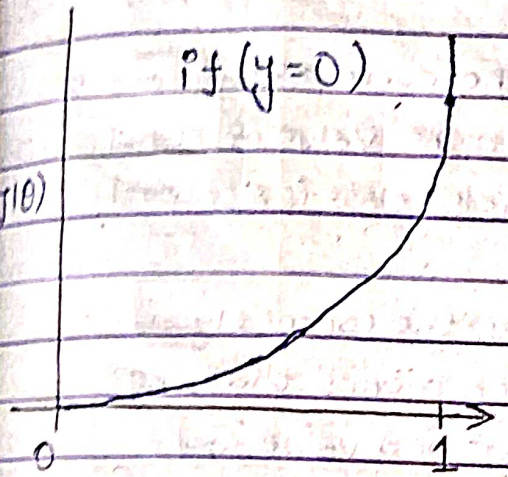
If $h_\theta(x) \to 1$ then loss = 0 (very close to right answer)

cost = 0 if $y = 1$, $h_\theta(x) = 1$

But as $h_\theta(x) \to 0$

$cost \to \infty$

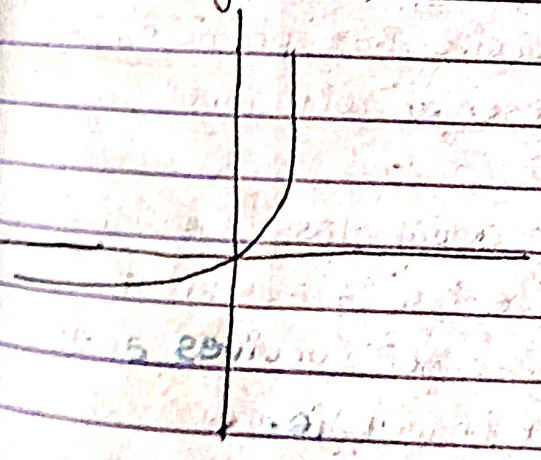
captures intuition that if $h_\theta(x) = 0$, (predict $P(y=1|x;\theta) = 0$), but $y = 1$ ~~with~~ we'll penalize learning algorithm by a very large cost.



As $h_\theta(x) \to 0$
then loss $\to 0$
true lable $\to 0$ &
model prediction close to 0

Larger $h_\theta(x)$ Bigger loss
As $h_\theta(x) \to 1$ then
loss $\to \infty$

* $-\log((1-x))$



★ Performance Evaluation :

(Confusion Matrix)

		Predicted	
		Class = positive	Class = negative
Actual	Class = positive	True Positive	False Negative
	Class = negative	False Positive	True Negative

- True Positives (TP) : These are the correctly predicted positive values which means that the value of actual class is positive & value of predicted class is also positive

- True Negatives (TN) : These are the correctly predicted negative values which means that the value of actual class is negative and value of predicted class is also negative

- False Positives (FP) : when actual class is negative and predicted class is positive i.e if actual class says this passenger didn't survive but predicted class tells you that this passenger will survive.

- False Negatives (FN) : when actual class is positive but predicted class is negative i.e if actual class value indicates that this passenger survives and predicted class tell you that it will die.

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

- Accuracy is the most intutive performance measure & it's simply a ratio of correctly predicted observations to total observations.

- One may think, if we have high accuracy then our model is best. Yes, accuracy is great measure but only when you have symmetric datasets where values of false positives & false negative are almost same.

- $\text{Precision} \Rightarrow \frac{TP}{TP + FP}$ precision is rate of correctly predicted positive observations to the total predicted positive observations.

High precision relates to the low false positive rate.

precision greater than 0.5 is good for model.

- $\text{Recall} \Rightarrow \frac{TP}{TP + FN}$ Recall is ratio of correctly predicted positive observations to the all observations in actual class. generally, recall greater than 0.5 is good for model.

- $\text{F1 Score} \Rightarrow \frac{2(\text{Recall} * \text{Precision})}{\text{Recall} + \text{Precision}}$

- F1 score is the weighted average of precision and recall. Therefore, this score takes both false positives and false negatives into account.

- F1 is usually more useful than accuracy, especially if you have an uneven class distribution. Accuracy works best if false positives & false negatives have similar cost.

- If the cost of false positives & fake negat are very different, it's better to look at Both precision & recall.

Student-ID	Actual Class	Predicted Class	
1	P	P	TP
2	P	P	TP
3	P	P	TP
4	N	N	TN
5	N	P	FP
6	P	P	TP
7	P	P	TP
8	N	N	TN
9	P	P	TP
10	P	P	TP
11	N	P	FP
12	P	N	FN
13	N	N	TN
14	P	P	TP
15	P	P	TP
16	N	N	TN
17	P	P	TP
18	P	P	TP
19	N	N	TN
20	P	P	TP

performance evaluation of two class algorithm

		Predicted	
		Class Positive	class negative
Actual	Class Positive	TP (12)	FN (1)
	Class Negative	FP (2)	TN (5)

Accuracy → (12+5)/(12+5+2+1)
→ 17/20 → 0.85

Precision → TP/TP+FP
→ (12)/(12+2) → 12/14 → 0.857

Recall → TP/TP+FN
→ 12/(12+1) → 12/13 → 0.92

Simplified cost function and gradient descent:

Logistic regression cost function

$$J(\theta) = \frac{1}{m} \sum_{i=1}^{m} \text{Cost}(h_\theta(x^{(i)}), y^{(i)})$$

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

Note: y = 0 or 1 always.

$$J(\theta) = -\frac{1}{m} \left[\sum_{i=1}^{m} y^{(i)} \log h_\theta(x^{(i)}) + (1 - y^{(i)}) \log (1 - h_\theta(x^{(i)})) \right]$$

To fit parameters θ: $\min_\theta J(\theta)$

To make a prediction new x:

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

Want $\min_\theta J(\theta)$:

Repeat {

$$\theta_j := \theta_j - \alpha \frac{d}{d\theta_j} J(\theta)$$

} simultaneously update all θ_j

$\frac{d}{d\theta_0}$

repeat {

$$\theta_j := \theta_j - \alpha \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x_j^{(i)}$$

} (simultaneously update all θ_j)

Algorithm looks identical to Linear Regression.

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