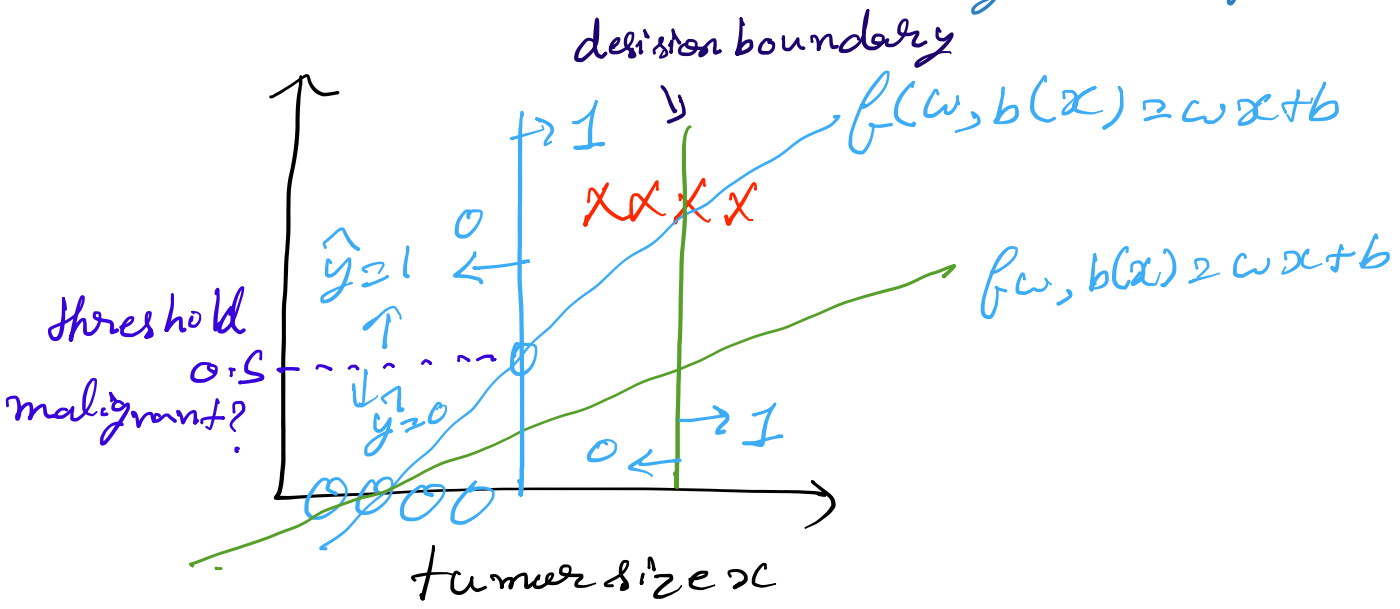
# Classification

In classification y can be one of two values 0 or 1 this is called binary classification

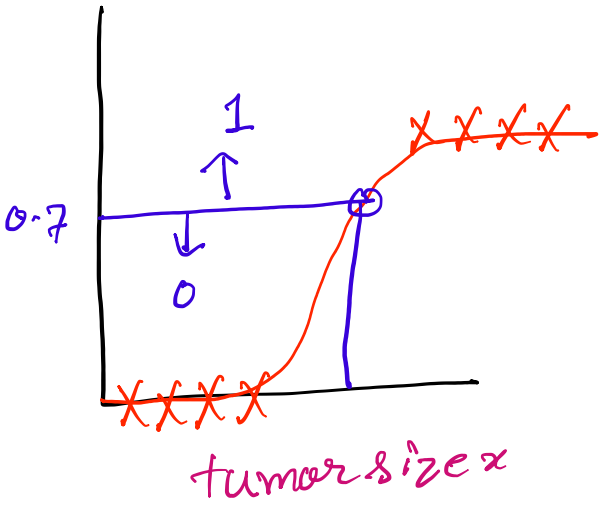


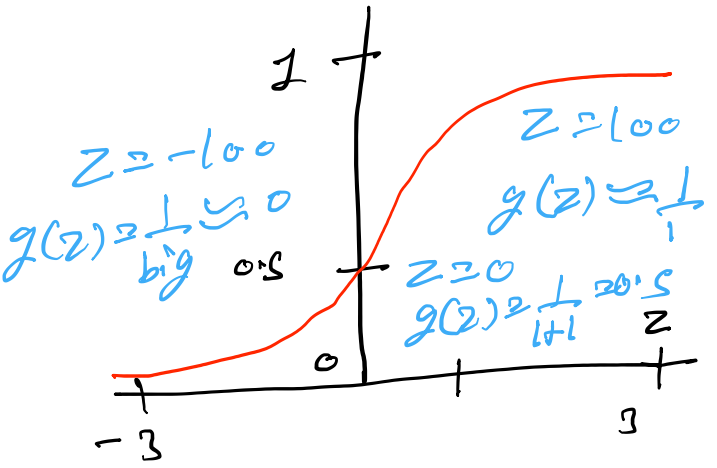

here we assume threshold value is 0.5 if value below the 0.5 so that is benign and if value is above 0.5 that is malignant

if $f_{w,b}(x) < 0.5 \rightarrow \hat{y}=0$

if $f_{w,b}(x) \geq 0.5 \rightarrow \hat{y}=1$

linear regression is not a good algorithm for this




sigmoid function

logistic function

this give output b/w 0 & 1

# Sigmoid function

$$g(z) = \frac{1}{1+e^{-z}} \qquad 0 < g(z) < 1 \qquad z = wx + b$$

$$f_{\vec{w},b}(\vec{x}) = g(\vec{w} \cdot x + b) = \underbrace{\frac{1}{1+e^{-(\vec{w}\vec{x}+b)}}}$$

this is logistic regression

$f_{\vec{w},b}(\vec{x}) = P(y=1 \mid \vec{x}; \vec{w}, b)$

Probability that $y$ is 1, given input $\vec{x}$, parameters $\vec{w}$, $b$

$P(y=0) + P(y=1) = 1$

when $f_{\vec{w},b}(\vec{x}) \geq 0.5$ ?

$g(z) \geq 0.5$

$z \geq 0$ when $\vec{w}\vec{x} + b \geq 0$

or when $\underline{\vec{w} \cdot \vec{x} + b \geq 0}$ then $\hat{y} = 1$

$\underline{\vec{w} \cdot \vec{x} + b < 0}$ then $\hat{y} = 0$

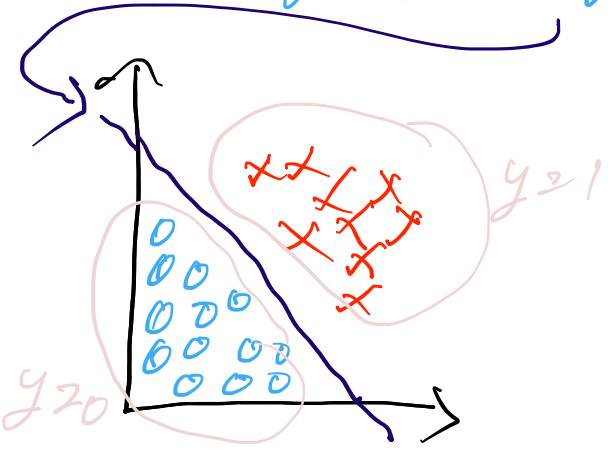
## Decision boundary

$f_{\vec{w},b}(\vec{x}) = g(z) = g(w_1 x_1 + w_2 x_2 + b)$

$z = \vec{w} \cdot \vec{x} + b = 0$

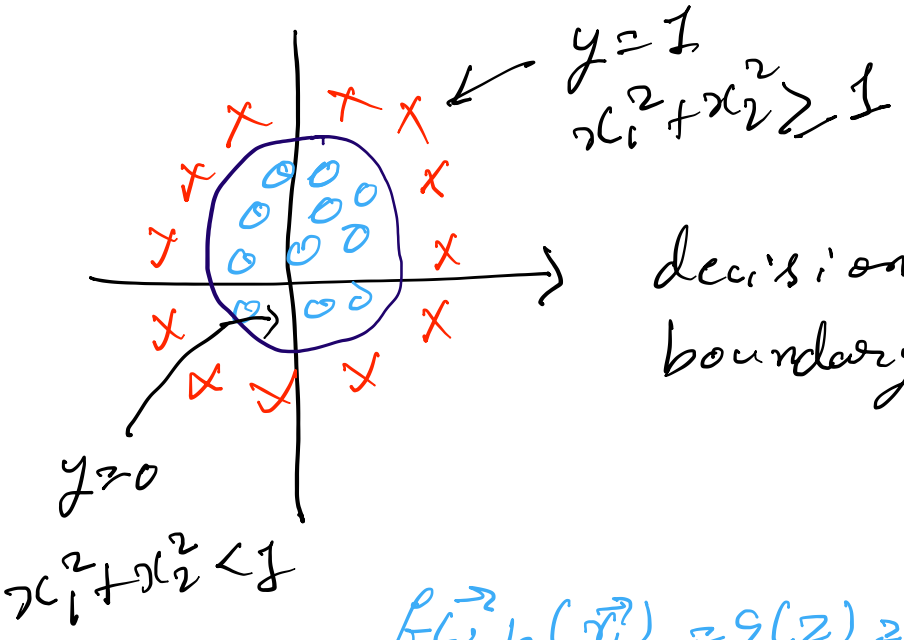
$z = x_1 + x_2 - 3 = 0$

$z$ tabhi 0 hoga jab $x_1 + x_2 = 3$ ho jaae



when $z=0$ we have decision boundarie

## Non linear Decision boundaries




decision boundary

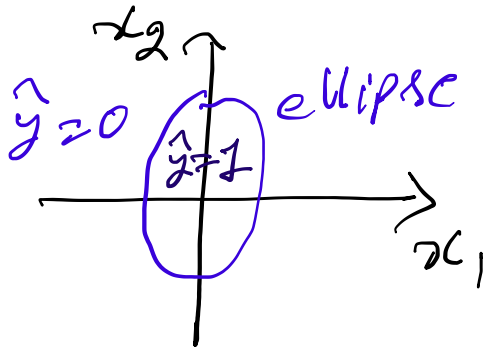
$z = x_1^2 + x_2^2 - 1 = 0$

$x_1^2 + x_2^2 = 1$

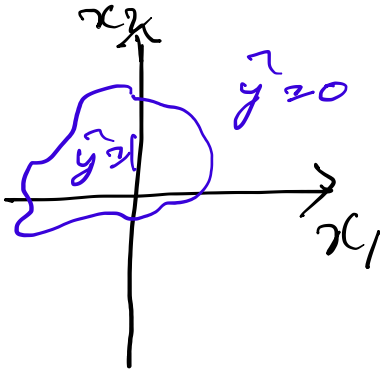
$f_{\vec{w},b}(\vec{x}) = g(z) = g(\underbrace{w_1 x_1^2 + w_2 x_2^2}_{z} + b)$

$f_{\vec{w},b}(\vec{x}) = g(z) = g(w_1 x_1 + w_2 x_2 + w_3 x_1^2 + w_4 x_1 x_2 + w_5 x_2^2$

So we have this curve



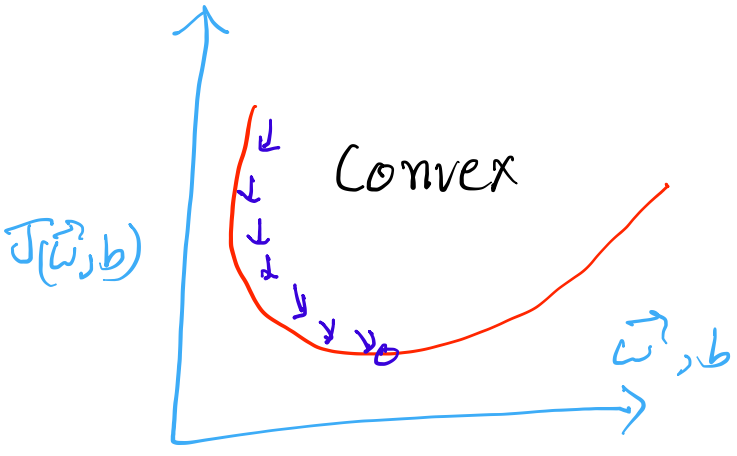
when we have eqn have contain more higher order polynomial than our decision boundary is more complex




$$f_{\vec{w},b}(\vec{x}) = g(w_1x_1 + w_2x_2 + w_3x_1^2 + w_4x_1x_2 + w_5x_2^2 + w_6x_1^3 + \cdots + b$$

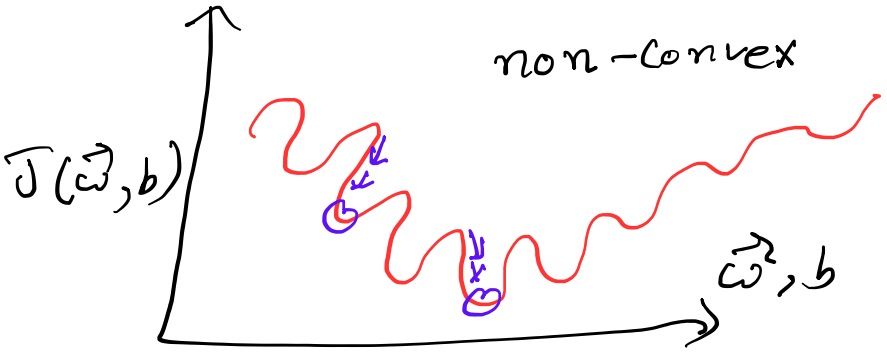
<u>Note</u> :- If we have polynomial like $x_1, x_2 \ldots$ not higher order polynomial than our decision boundary is straight but when the higher order terms are more than the decision boundary is more complex

In logistic regression $f_{\vec{w},b}(\vec{x}) = \vec{w}\cdot\vec{x} + b$




here we have only one minimum by when

we have more than one minimum like this




then we use logistic regression

$$f_{\vec{w},b}(\vec{x}) = \frac{1}{1+e^{-(\vec{w}\cdot\vec{x}+b)}}$$

so to make this <u>convex</u> we use new

**Squared Error cost function**

$$J(\vec{w}, b) = \frac{1}{m}\sum_{i=1}^{m} \frac{1}{2}\left(f_{\vec{w},b}(\vec{x}^{(i)}) - y^{(i)}\right)^2$$

from this cost function we don't get convex curve so here we have another cost function

$$L(f_{\vec{w},b}(\vec{x}^{(i)}), y^{(i)}) = \begin{cases} -\log(f_{\vec{w},b}(\vec{x}^{(i)})) & \text{if } y^{(i)} = 1 \\ -\log(1-f(\vec{w}\, b(\vec{x}^{(i)})) & \text{if } y^{(i)} = 0 \end{cases}$$

this 1 and 0 are the class that are given in the data set

from this cost function we can reach a global minimum

we simply write this cost function as

$$\boxed{\begin{aligned} & L(f_{\vec{w},b}(\vec{x}^{(i)}), y^{(i)}) \\ &= -y^{(i)} \log(f_{\vec{w},b}(\vec{x}^{(i)})) - (1-y^{(i)}) \log(1-f_{\vec{w},b}(\vec{x}^{(i)})) \end{aligned}}$$

is function ko use karne par hame ye nahi dekna ke 0 k liye konsa fun^n h 1 k liyen konsa function h

**Simplified Cost function**

$$\boxed{J(\vec{w},b) = -\frac{1}{m}\sum_{i=1}^{m}\left[y^{(i)} \log(f_{\vec{w},b}(\vec{x}^{(i)})) + (1-y^{(i)}) \log(1-f_{\vec{w},b}(\vec{x}^{(i)}))\right]}$$

# Gradient descent for logistic regression

$$temp_w = w_j - \alpha\left[\frac{1}{m}\sum_{i=1}^{m}\left(f_{\vec{w},b}(\vec{x}^{(i)}) - y^{(i)}\right)x_j^{(i)}\right]$$

$x_j^{(i)}$ = is the jth feature of training example i

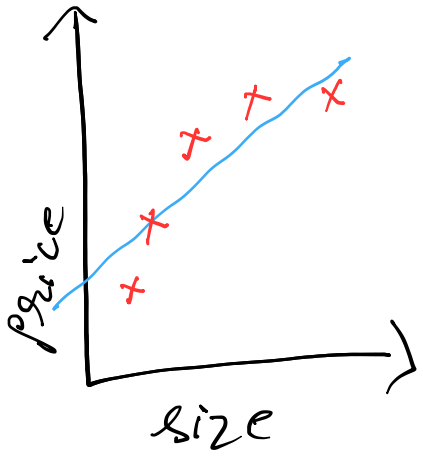
$$temp_b = b - \alpha\left[\frac{1}{m}\sum_{i=1}^{m}\left(f_{\vec{w},b}(\vec{x}^{(i)}) - y^{(i)}\right)\right]$$

$w = temp_w$

$b = temp_b$

Note:- here we also use feature scaling to take on similar ranges of values can also speed up gradient descent for logistic regression.

1

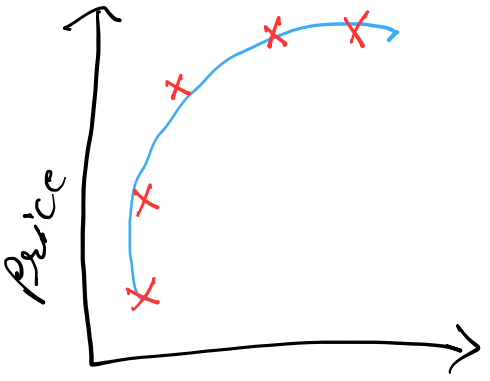



$w_1 x + b$

underfit

high bias

2

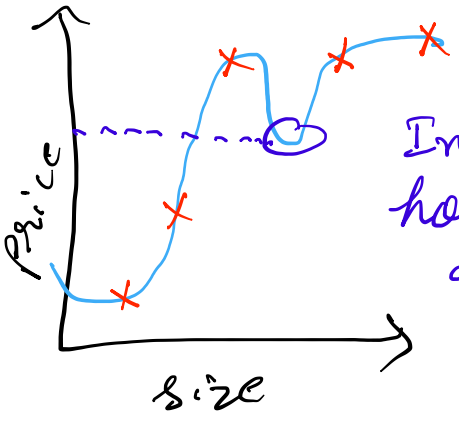



$w_1 x + w_2 x^2 + b$

just right

generalization

3




In this model would predict that this house is cheaper than houses that are smaller than it

so this is not a good model to predict values

$w_1 x + w_2 x^2 + w_3 x^3 + w_4 x^4 + b$

overfit

high variance

In these example 2nd example is right for predication because new feature is come it also predict right for that

In 3rd example our model is overfit that is not good

Note:- when your model is overfit so to make it right model we have so tips

1. Collect more training data
2. If we have more feature insted of using all feature we were to pick just a subset of most useful ones than your model no longer overfit as badly (feature selection)

disadvantage of that is by using only a subset of the features the algorithm is throwing away some of the information that you have about the houses. If these all features are important to predict price you can't throw some of the features

3. Regularization

In this when in our eqn we have more higher order polynomials so can reduce the size of parameters of those polynomials $\approx 0$

# Cost function with regularization

$$\min_{\vec{w},b} J(\vec{w},b) = \min_{\vec{w},b} \frac{1}{2m}\sum_{i=1}^{m}\left(f_{\vec{w},b}(\vec{x}^{(i)}) - y^{(i)}\right)^2 + \frac{\lambda}{2m}\sum_{j=1}^{n} w_j^2$$

Note:- here when we use $\lambda = 0$ then it is underfit

when we use $\lambda$ more large like $10^{10}$ is goes to overfit

so we have choose the value of $\lambda$ that are not very large and not very small

# Regularized logistic regression

$$\min_{\vec{w},b} J(\vec{w},b) = -\frac{1}{m}\sum_{i=1}^{m}\left[y^{(i)}\log\left(f_{w,b}(x^{(i)})\right) + (1-y^{(i)})\log\left(1-f_{\vec{w},b}(x^{(i)})\right)\right] + \frac{\lambda}{2m}\sum_{j=1}^{n} w_j^2$$

# Implementating gradient descent

$$temp_w = w_j - \alpha\left[\frac{1}{m}\sum_{i=1}^{m}\left[\left(f_{\vec{w},b}(x^{(i)}) - y^{(i)}\right)x_j^{(i)}\right] + \frac{\lambda}{m}w_j\right.$$

$$temp_b = b - \alpha\frac{1}{m}\sum_{i=1}^{m}\left(f_{\vec{w},b}(\vec{x}^{(i)}) - y^{(i)}\right)$$

$w = temp_w$

$b = temp_b$