

Logistic Regression

BCSE0105: MACHINE LEARNING

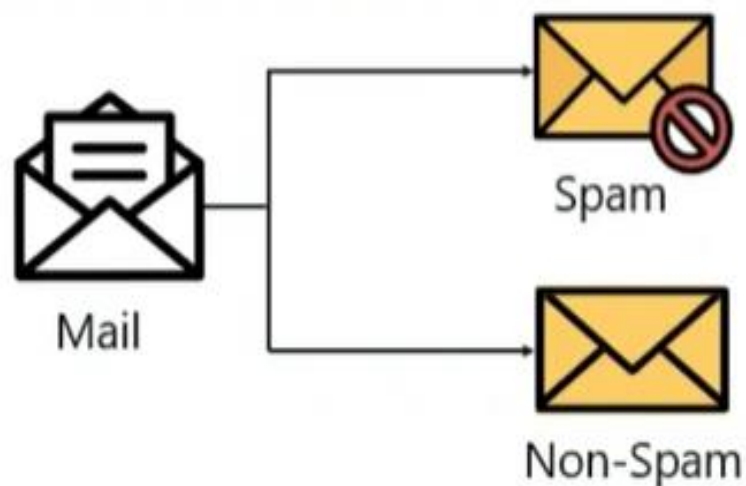
Ref: [#Edureka](#) [#EdurekaMachineLearning](#) [#logisticregression](#)

Regression Vs Classification

Classification

Classification is the task of predicting a discrete class label

- In a classification problem data is classified into one of two or more classes
- A classification problem with two classes is called binary, more than two classes is called a multi-class classification



Regression

Regression is the task of predicting a continuous quantity

- A regression problem requires the prediction of a quantity
- A regression problem with multiple input variables is called a multivariate regression problem



Introduction

- **Logistic regression** is a Machine Learning algorithm which is used for the **classification problems**.
- It extends the ideas of linear regression to the situation where the dependent variable, Y , is categorical.
- Logistic regression is designed as a binary classifier (output say $\{0,1\}$) but actually outputs the probability that the input instance is in the “1” class.

Logistic Regression

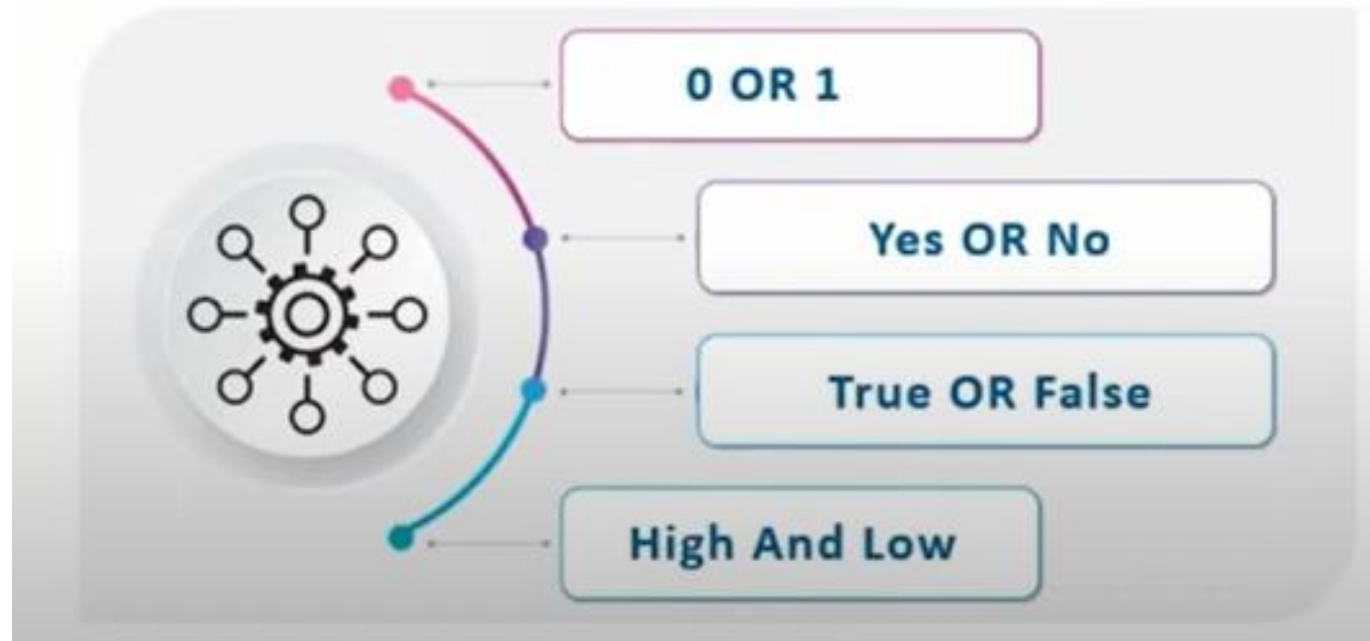
- Primarily used for binary classification .
- It is a predictive analysis algorithm and based on the concept of **probability**.
- Predicts the probability (therefore always exists between 0 and 1), and based on this prediction **perform classification**.

Logistic Regression

- Logistic Regression uses a more **complex cost function**.
- This cost function is defined as the '**Sigmoid function**' or '**logistic function**' instead of a linear function.
- The Sigmoid function **maps** any **real value** into another value **between 0 and 1**.

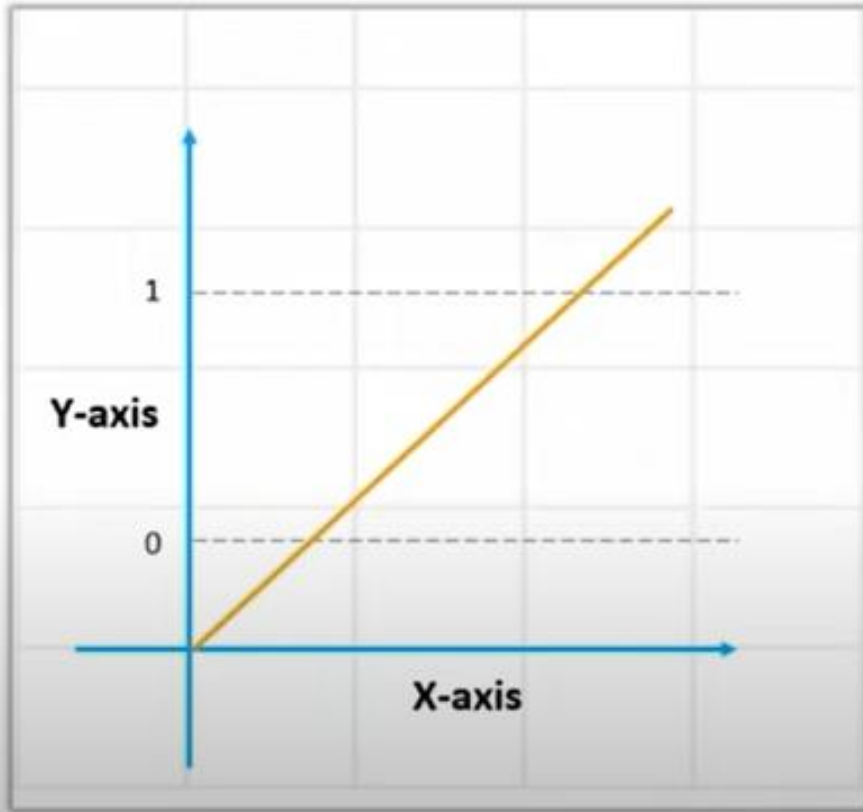
Logistic Regression

Logistic Regression produces results in a **binary format** which is used to predict the outcome of a categorical dependent variable. So the outcome should be **discrete/ categorical** such as:



Why not linear regression?

We don't need the value which is below 0 and above 1 .



Since our value of Y will be between 0 and 1, the linear line has to be clipped at 0 and 1.

Why not linear regression?

The line has to be clipped at 0 and 1



Why not linear regression?

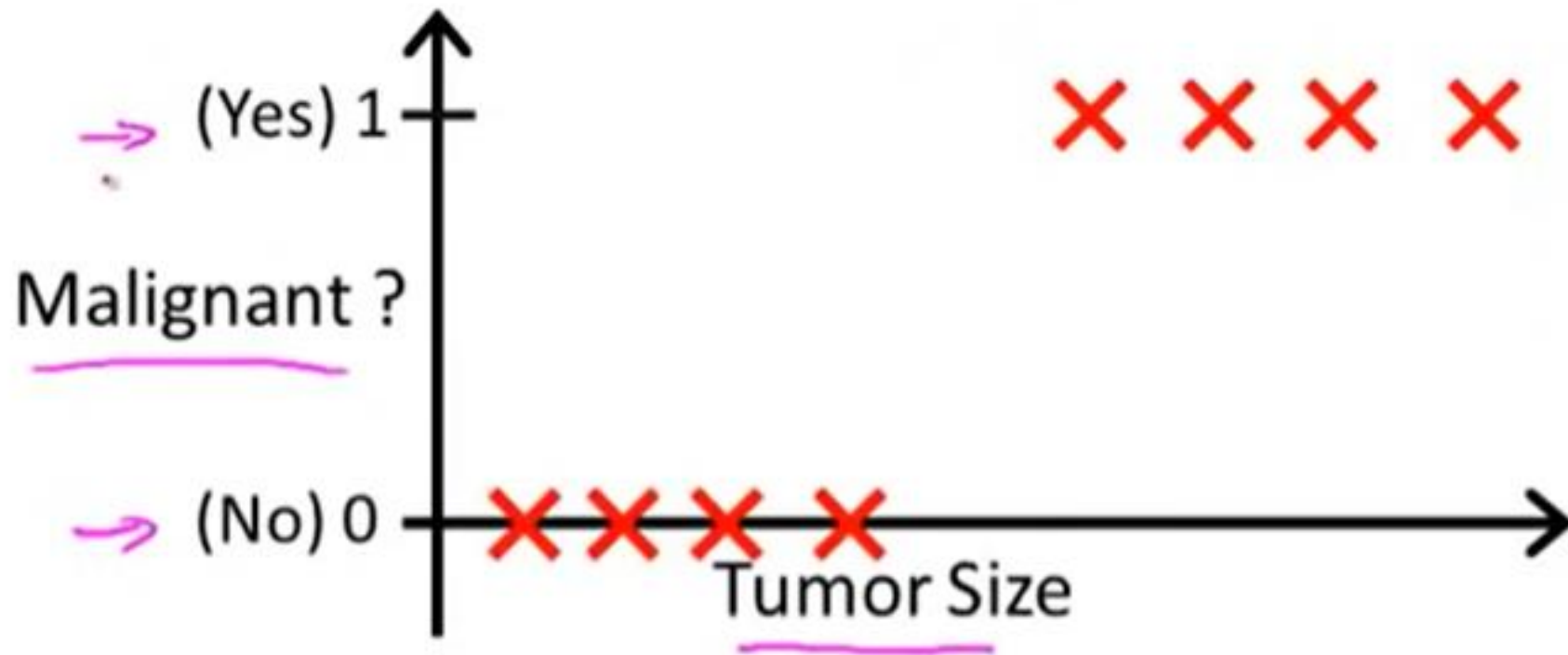
- This is converted into 3 different straight lines.



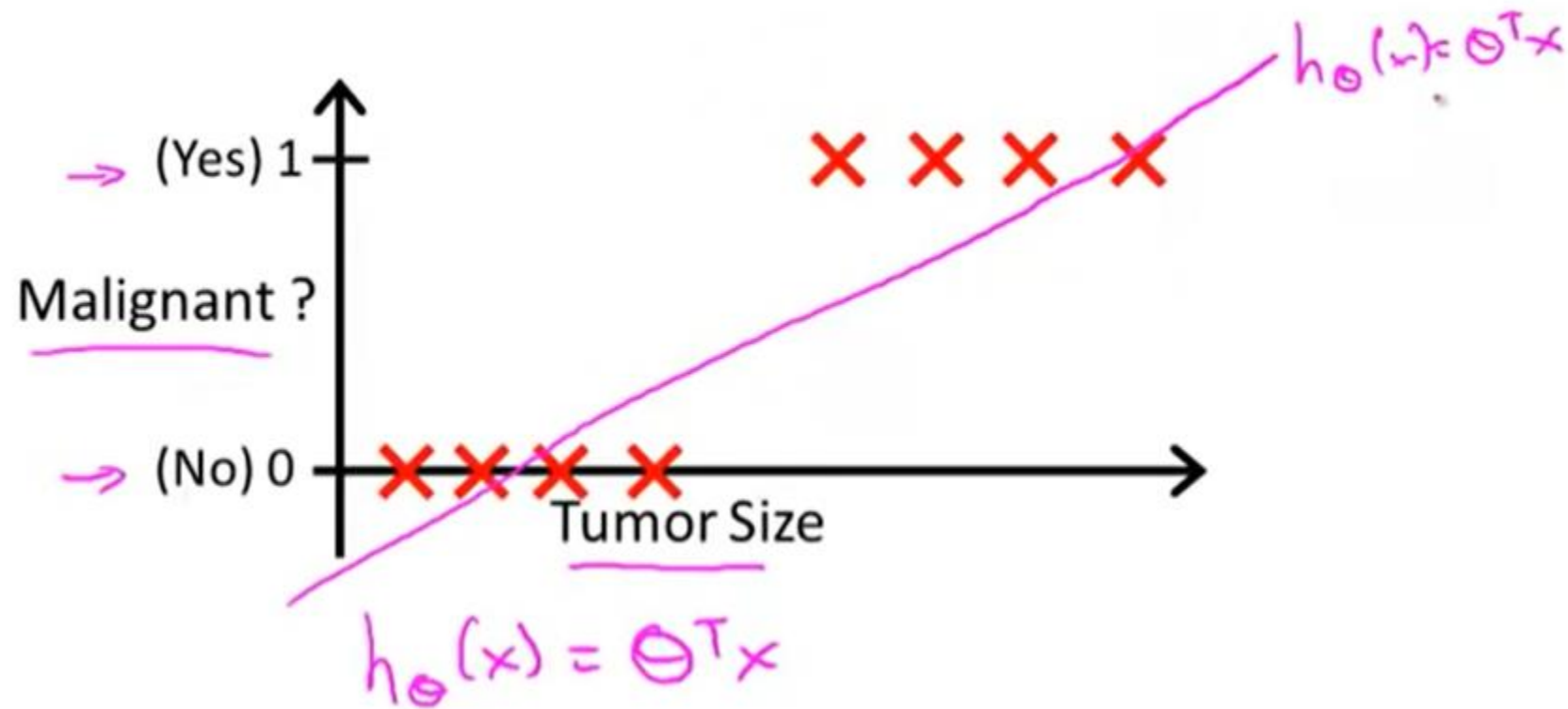
- Therefore, we need a different type of solution to solve this problem, so that it is formulated into an equation
- Hence we come up with the **logistic regression**, so here the outcome is either 0 or 1 (which is the main task of logistic regression).

Example

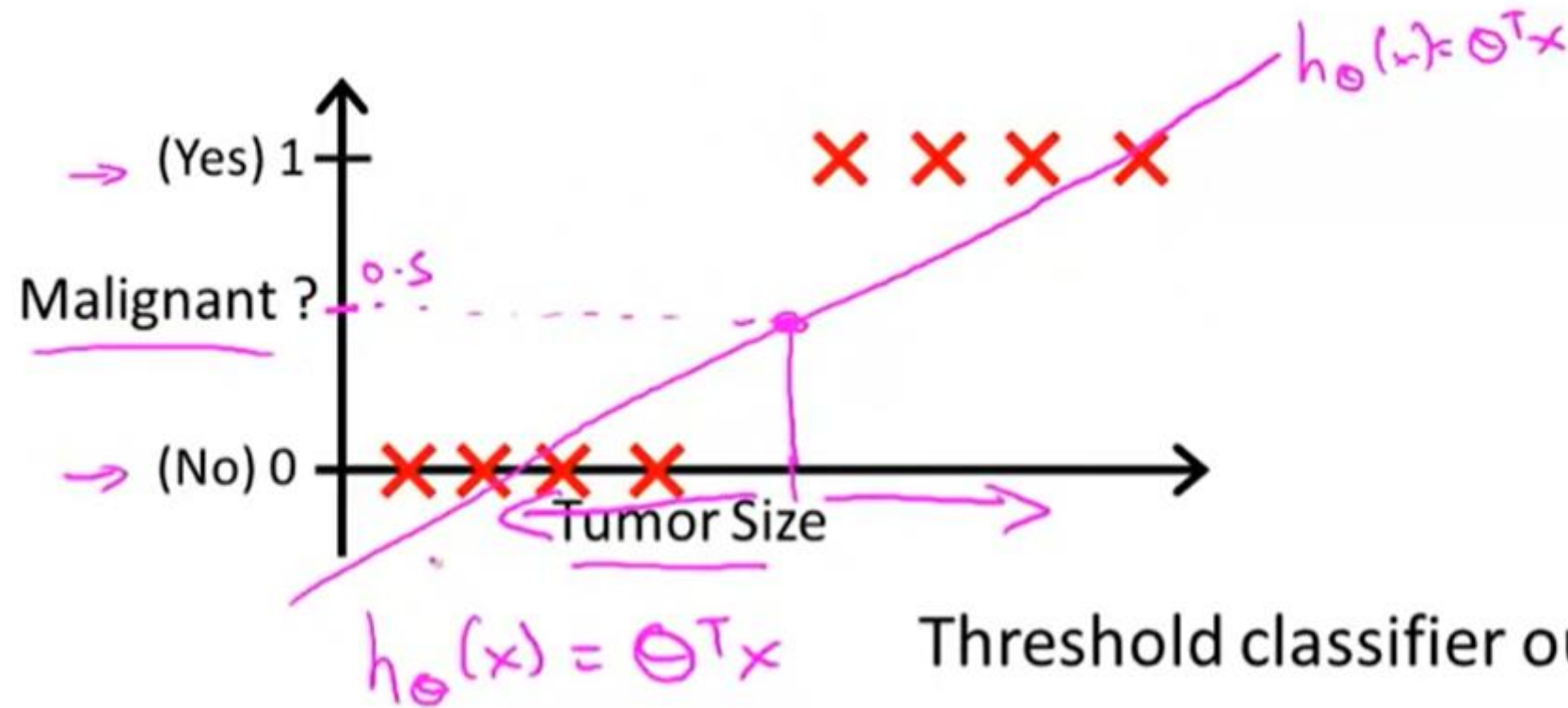
If we apply **linear regression** to classification problem



If we apply **linear regression** to classification problem



If we apply **linear regression** to classification problem

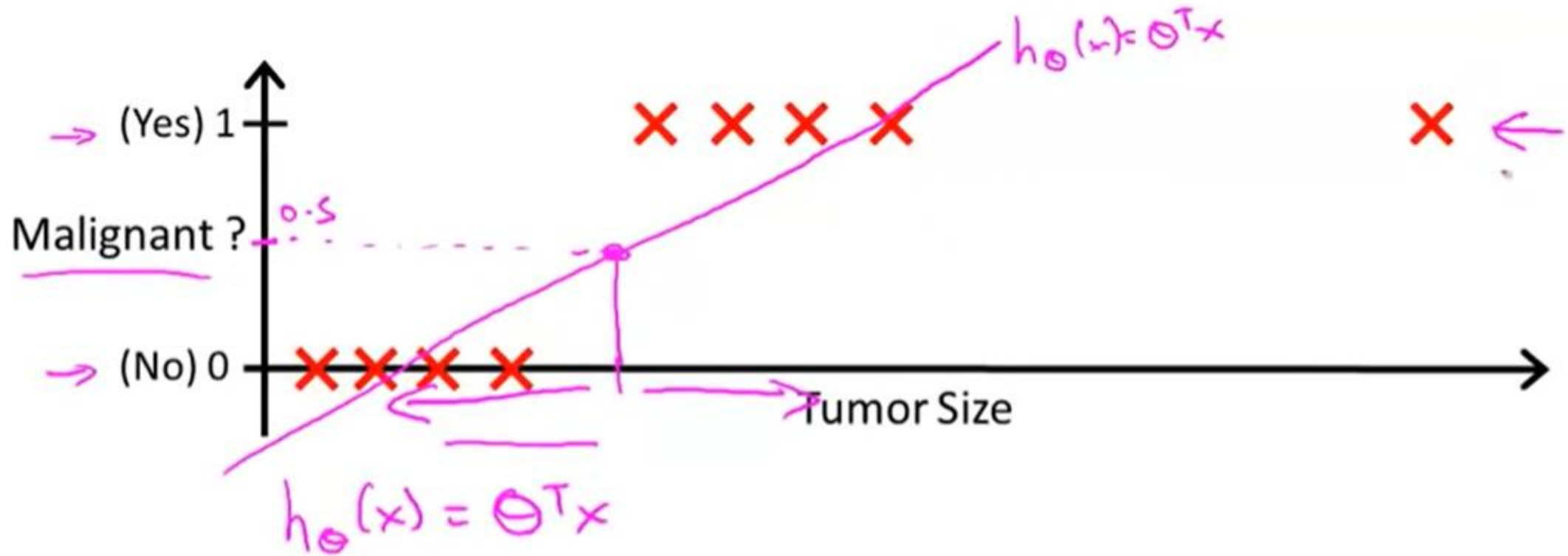


Threshold classifier output $h_{\theta}(x)$ at 0.5:

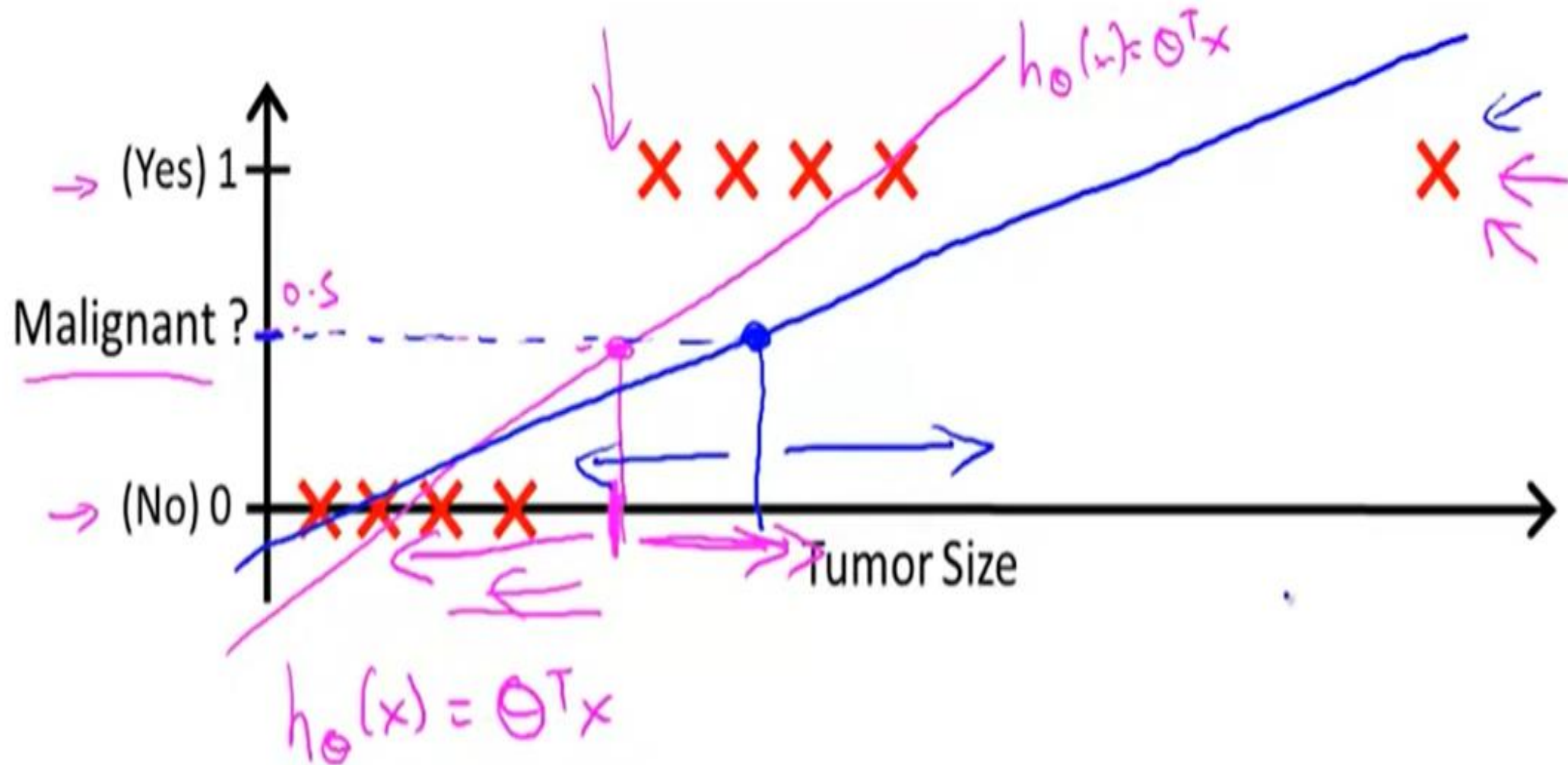
If $h_{\theta}(x) \geq 0.5$, predict "y = 1"

If $h_{\theta}(x) < 0.5$, predict "y = 0"

If we add one more training example
then no issues with this regression line



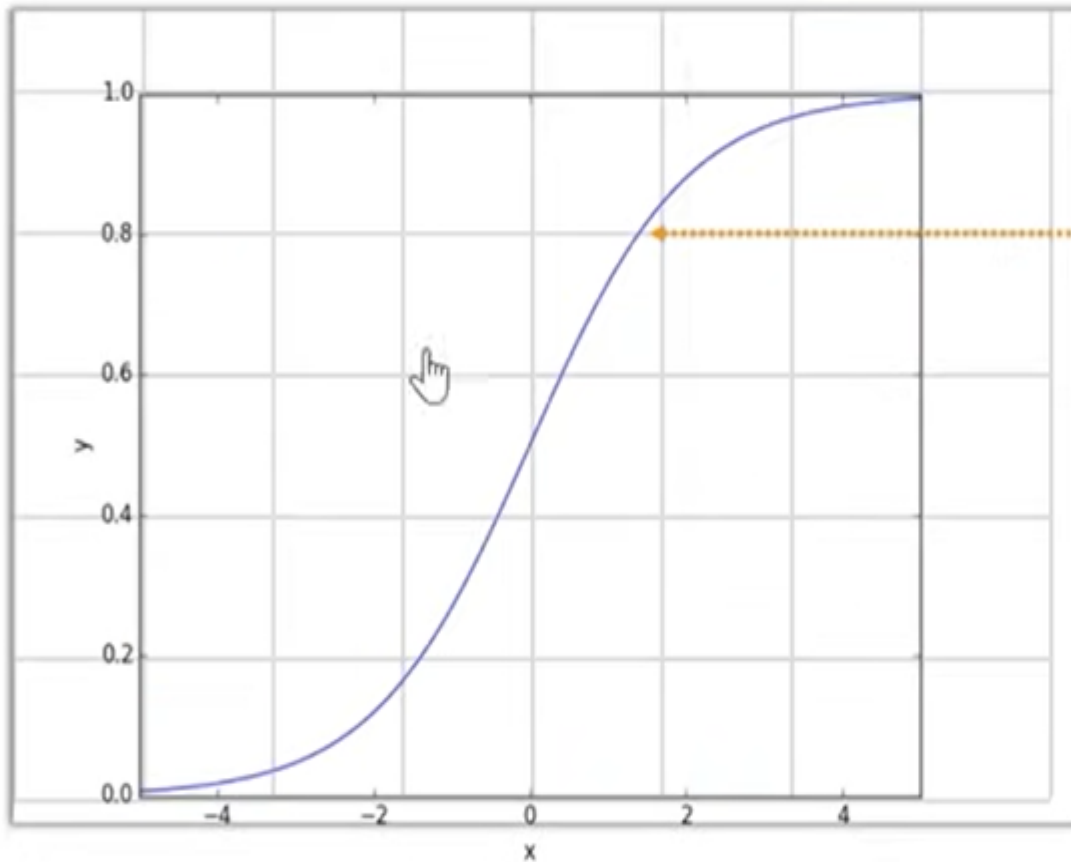
But If **regression line changes** then we can see that linear regression is not suitable for classification problem



- Therefore linear regression is not good for classification problem due to 2 problems:
- Outliers
- Output may also be >1 or <0 which is not preferred here (as we are predicting the probability which lies between 0 and 1)

Once it get formulated into equation it looks like this

Logistic Regression Curve



The Sigmoid "S"
Curve

The formula of a sigmoid function /logistic function

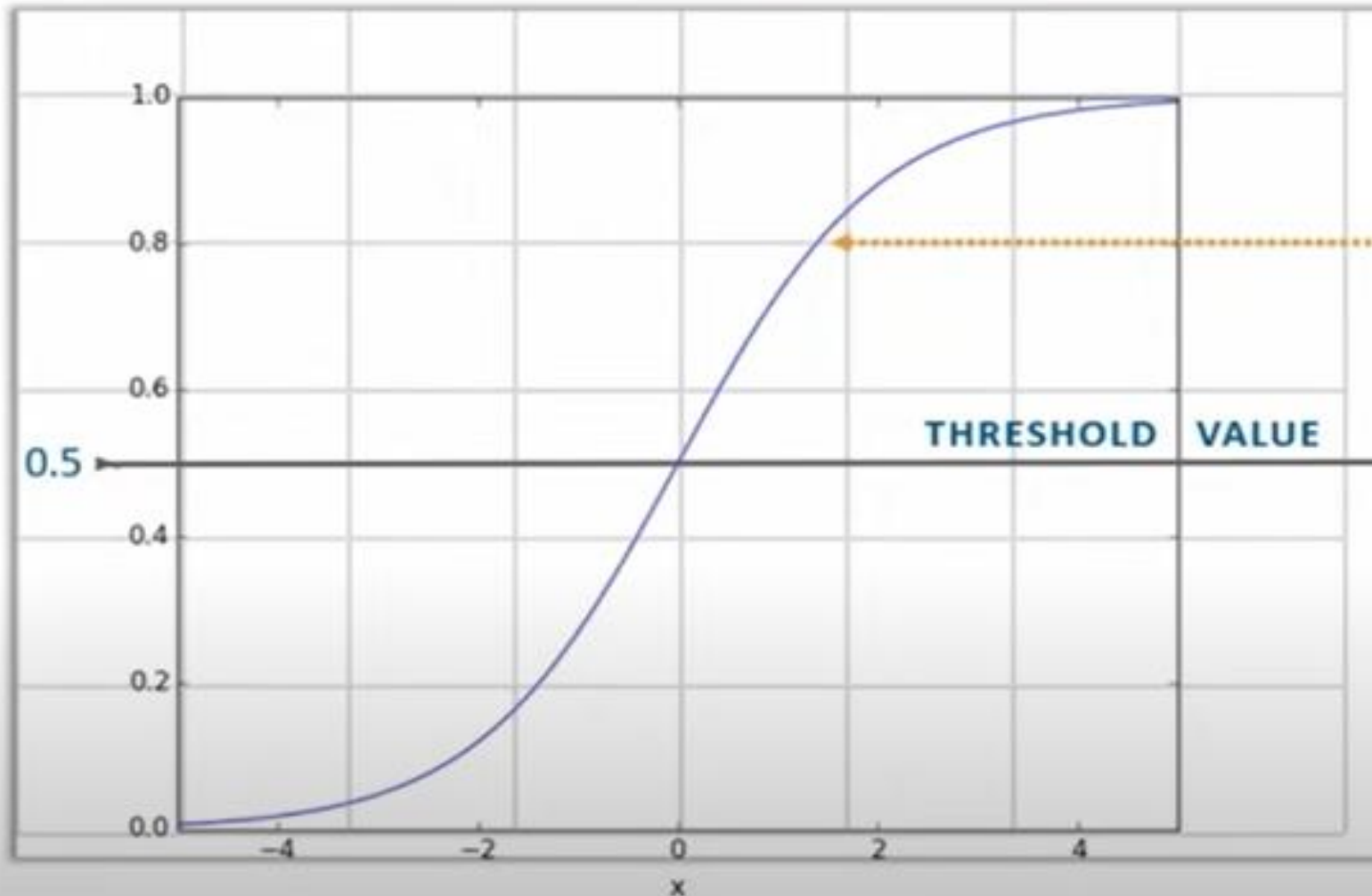
$$y = f(x) = \frac{1}{1 + e^{-(x)}}$$

At $x = \infty, y = 1$

At $x = -\infty, y = 0$

At $x = 0, y = 1/2$

Sigmoid function converts any value from $-\infty$ to $+\infty$ into discrete value either 0 or 1 using threshold



The Sigmoid "S"
Curve

With this, the
threshold value
indicates the
probability of winning
or losing

Classification

Email: Spam / Not Spam?

Online Transactions: Fraudulent (Yes / No)?

Tumor: Malignant / Benign ?

If we want to predict the variable y

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

0: "Negative Class" (e.g., benign tumor)

1: "Positive Class" (e.g., malignant tumor)

Hypothesis Representation

- When using **linear regression**, we used a formula of the hypothesis as follows:

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

For multiple variables, in linear regression

$$X = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ \cdot \\ \cdot \\ \cdot \\ x_n \end{bmatrix}$$

$$\theta = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \\ \cdot \\ \cdot \\ \cdot \\ \theta_n \end{bmatrix}$$

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

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

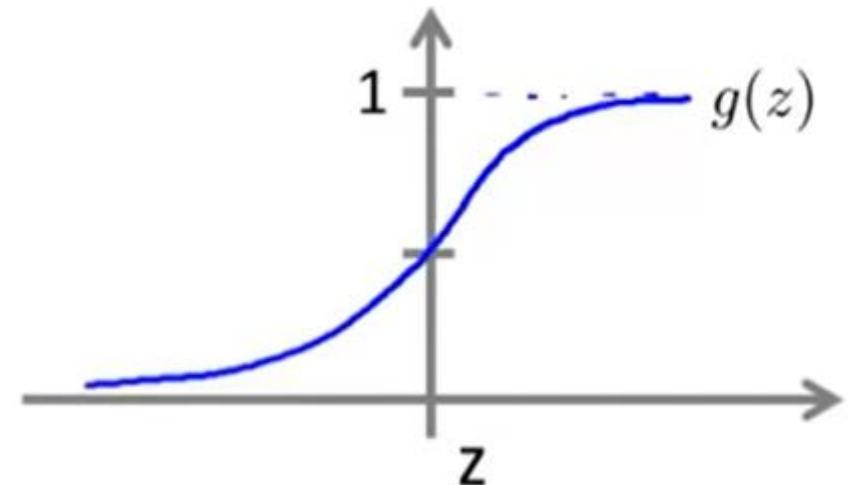
For **logistic regression**, hypothesis is:

$$h_{\theta}(x) = g(\theta^T x)$$

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

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

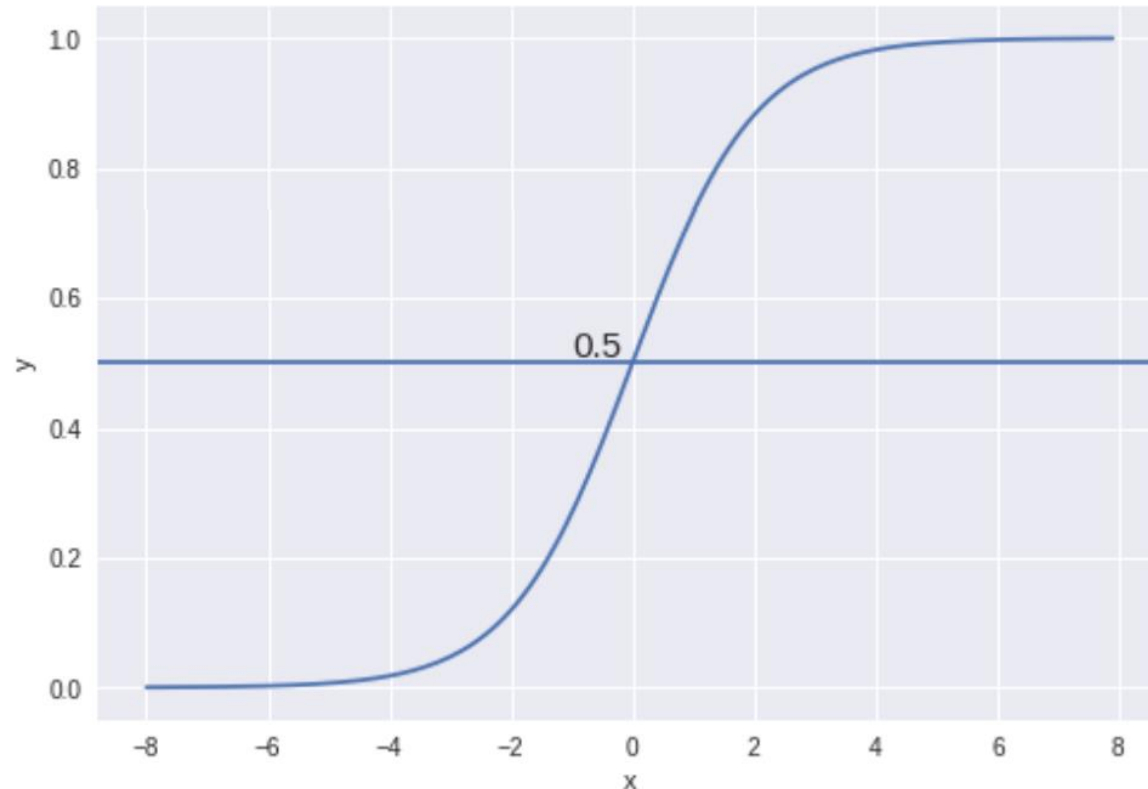
and $0 \leq h_{\theta}(x) \leq 1$



Decision Boundary

- We expect our classifier to give us a set of outputs or classes based on probability when we pass the inputs through a prediction function and returns a probability score between 0 and 1.
- For Example, We have 2 classes, let's take them like cats and dogs (1 — dog , 0 — cats).
- We basically decide with a threshold value above which we classify values into Class 1 and as the value goes below the threshold then we classify it in Class 2.

As shown in the below graph we have chosen the threshold as 0.5, if the prediction function returned a value of 0.7 then we would classify this observation as Class 1(DOG). If our prediction returned a value of 0.2 then we would classify the observation as Class 2(CAT).

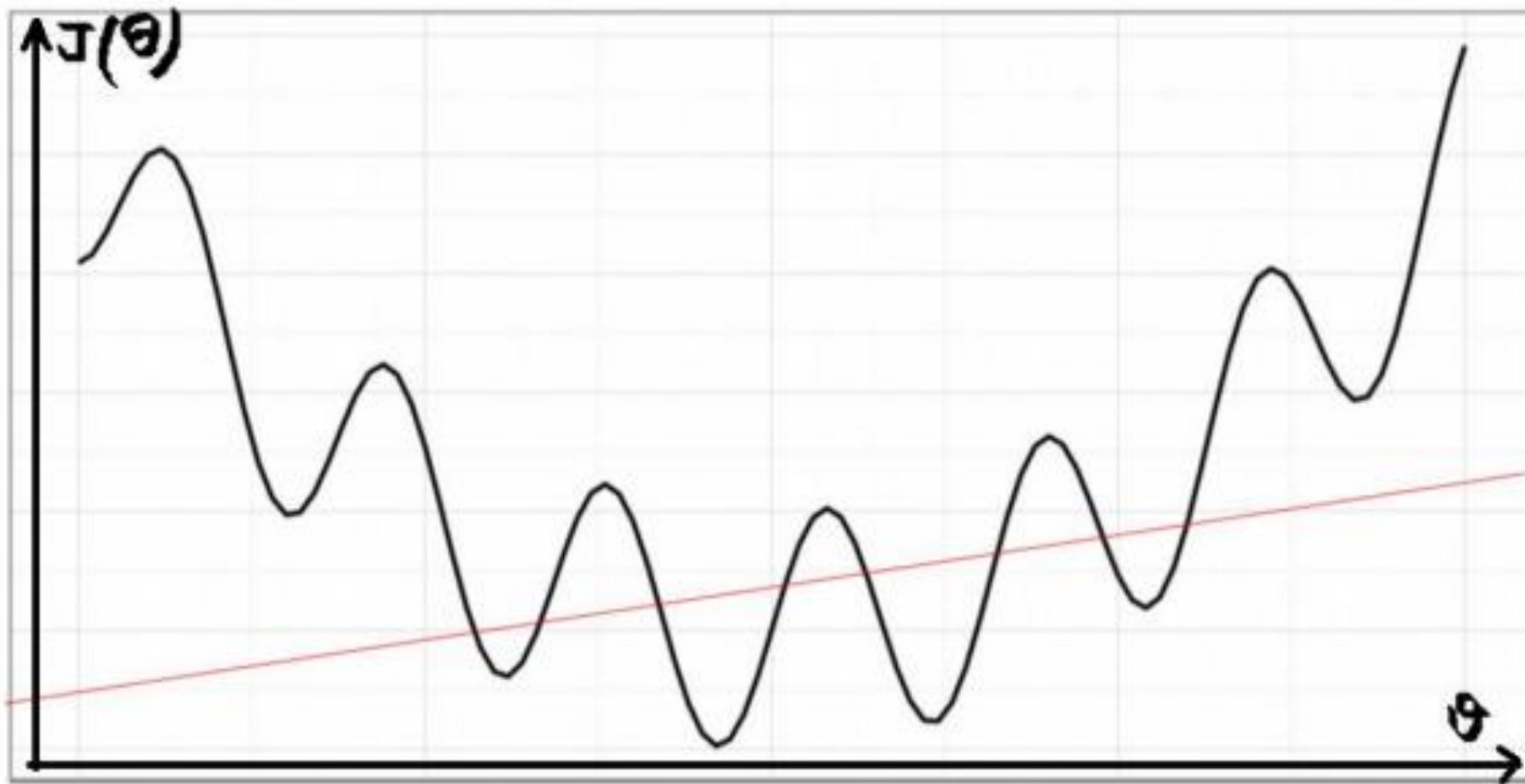


Cost function

- The cost function represents optimization objective.
- The cost function in the Linear regression is:

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

- With the modified hypothesis function, taking a **square error function** won't work as it **no longer convex** in nature and tedious to minimize. We take up a new form of cost function.



In this case, finding an optimal solution with the gradient descent method is not possible

To find an optimal solution with the gradient descent method

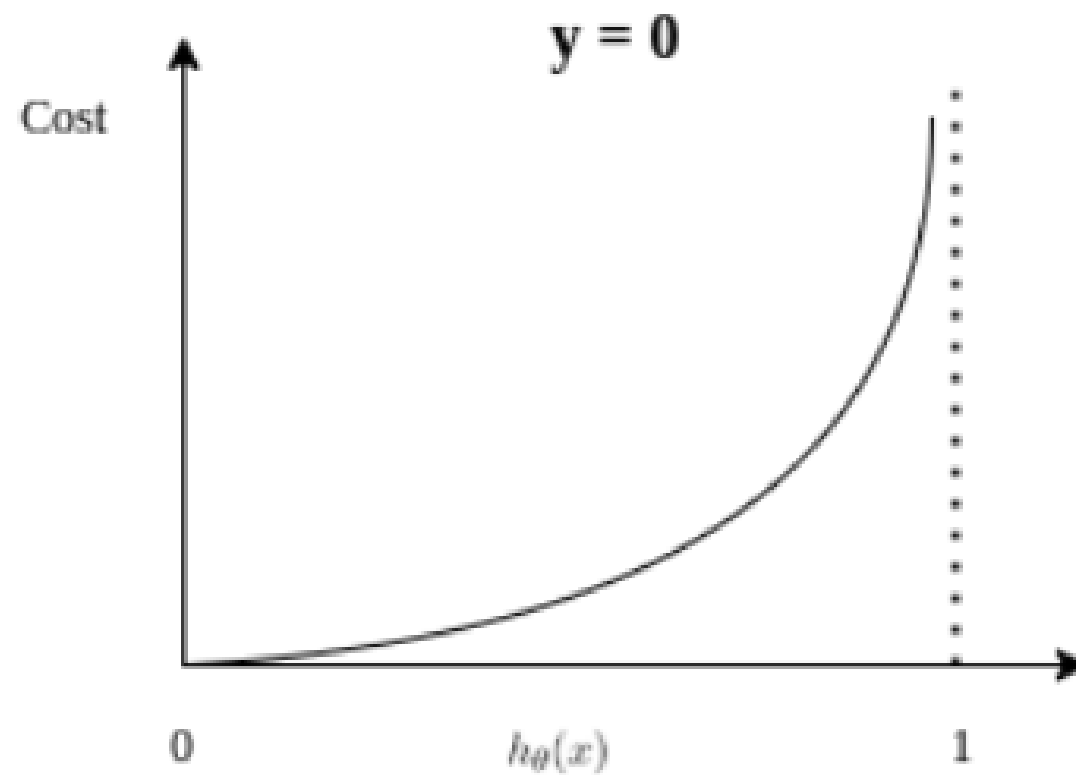
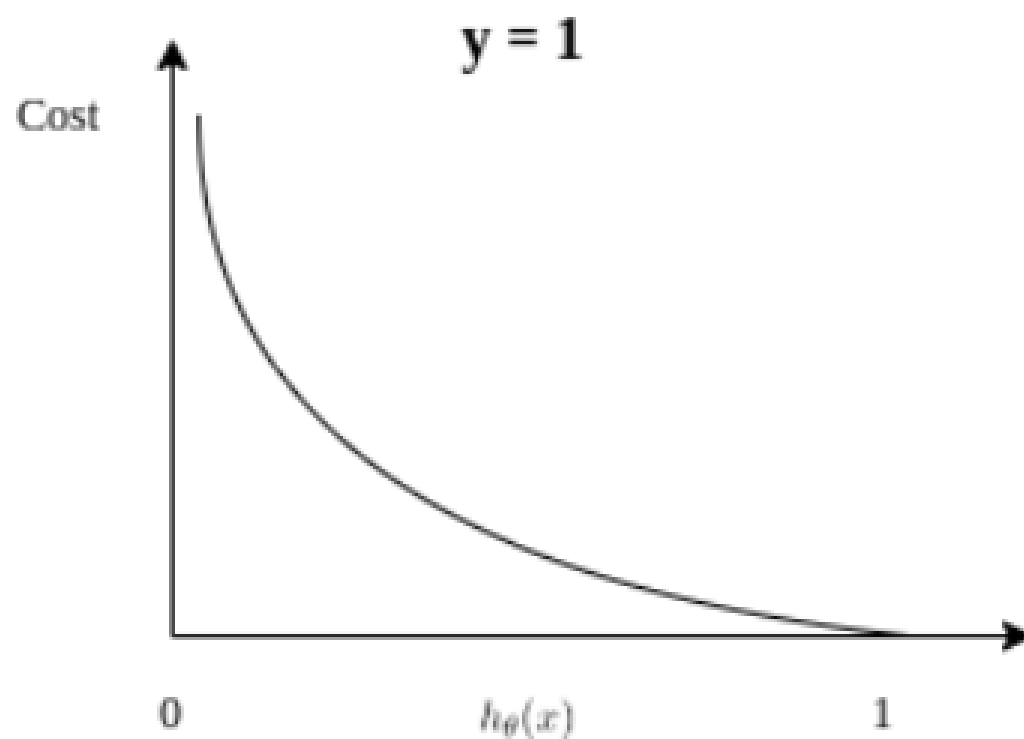
- we need to use a **logarithmic function** to represent the cost of logistic regression.
- It is guaranteed to be **convex** for all input values, containing only one minimum, allowing us to run the gradient descent algorithm.

For logistic regression, the Cost function is defined as:

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

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

Which results in:



Observations from Graphs

- when the actual outcome $y = 1$, the cost is 0 for $h_{\theta}(x) = 1$ and takes the maximum value for $h_{\theta}(x) = 0$.
- Similarly, if $y = 0$, the cost is 0 for $h_{\theta}(x) = 0$ and takes the maximum value for $h_{\theta}(x) = 1$.

Cost function of Logistic Regression

As the output can either be 0 or 1, the cost function for both the cases can be combined into a single function as:

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

To fit parameters θ :

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

To make a prediction given new x :

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

Gradient Descent

Repeat {

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

}

(simultaneously update all θ_j)

Some useful differentiations

$$\frac{\partial x^n}{\partial x} = n x^{n-1}$$

$$\frac{\partial}{\partial x} \left(\frac{1}{x} \right) = \frac{\partial}{\partial x} x^{-1} = -\frac{1}{x^2}$$

$$\frac{\partial \log x}{\partial x} = \frac{1}{x}$$

$$\frac{\partial e^x}{\partial x} = e^x$$

$$\frac{\partial e^{-x}}{\partial x} = -e^{-x}$$

$$\frac{\partial [f(x)]^n}{\partial x} = n [f(x)]^{n-1} \times \frac{\partial f(x)}{\partial x}$$

Derivative of cost function for Logistic Regression

Cost function for **all training** examples:

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

Cost function for **one training** example:

$$C = -y \log h_{\theta}(x) - (1-y) \log (1-h_{\theta}(x))$$

where, $h_{\theta}(x) = g(\theta^T \cdot x)$

Now cost function for **one observation**:

$$E = -y \log h_0(x) - (1-y) \log (1-h_0(x))$$

Let $a = h_0(x)$, where $h_0(x) = g(\theta x)$

Rewrite E ,

$$E = -y \log a - (1-y) \log (1-a)$$

$$\text{where, } a = h_0(x) = g(z) = \frac{1}{1 + e^{-z}}$$

$$\text{and } z = \theta x$$

$$\frac{\partial \bar{E}}{\partial \theta} = \frac{\partial E}{\partial a} \times \frac{\partial a}{\partial z} \times \frac{\partial z}{\partial \theta} \text{ --- ① (By Chain Rule)}$$

$$\frac{\partial E}{\partial a} = \frac{-y}{a} - \frac{(1-y)}{(1-a)} \times -1$$

$$= \frac{-y}{a} + \frac{(1-y)}{(1-a)}$$

$$= \frac{-y + ay + a - ay}{a(1-a)}$$

$$\boxed{\frac{\partial E}{\partial a} = \frac{a-y}{a(1-a)}}$$

$$\frac{\partial a}{\partial z} = \frac{\partial}{\partial z} \left(\frac{1}{1+e^{-z}} \right) = \frac{\partial}{\partial z} (1+e^{-z})^{-1}$$

$$\frac{\partial a}{\partial z} = -1 \times (1+e^{-z})^{-2} \times -e^{-z}$$

$$\frac{\partial a}{\partial z} = \frac{e^{-z}}{(1+e^{-z})^2}$$

As we know, $a = \frac{1}{1+e^{-z}} \Rightarrow a^2 = \frac{1}{(1+e^{-z})^2}$

Also, $1+e^{-z} = \frac{1}{a}$, $e^{-z} = \frac{1}{a} - 1 \Rightarrow e^{-z} = \frac{1-a}{a}$

Putting these values we get,

$$\frac{\partial a}{\partial z} = \frac{(1-a)}{a} \times a^2 = a(1-a) \Rightarrow \boxed{\frac{\partial a}{\partial z} = a(1-a)}$$

$$\frac{\partial Z}{\partial \theta} = \frac{\partial \theta x}{\partial \theta} = x$$

$$\boxed{\frac{\partial Z}{\partial \theta} = x}$$

Put these derivatives in $\frac{\partial E}{\partial \theta}$, we get

$$\frac{\partial E}{\partial \theta} = \frac{(a-y)}{a(1-a)} \times a(1-a) \times x$$

$$\boxed{\frac{\partial E}{\partial \theta} = (a-y)x}$$

$$\therefore \frac{\partial E}{\partial \theta} = (h_{\theta}(x) - y)x \quad (h_{\theta}(x) = g(\theta x))$$

This is the derivative of cost function for *one observation*,

Now, for *one training example*, we can write

$$\frac{\partial C}{\partial \theta_j} = (h_{\theta}(x) - y)x_j \quad (h_{\theta}(x) = g(\theta^T \cdot x))$$

For all **training examples**, we can write

$$\frac{\partial J(\theta)}{\partial \theta_j} = \frac{1}{m} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)}) x_j^{(i)}$$

Gradient Descent

Repeat {

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

}

(simultaneously update all θ_j)

Putting the value of derivative, we get

Repeat {

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

}

Gradient Descent for Logistic Regression

Repeat {

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

}

Algorithm looks identical to linear regression!

But there is a difference in hypothesis

Gradient Descent

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

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

$$\Theta = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \\ \vdots \\ \theta_n \end{bmatrix}$$

Repeat {

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

}

(simultaneously update all θ_j)

$$h_{\theta}(x) = \Theta^T x$$

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

Linear vs logistic regression

Linear Regression is a method to predict dependent variable (Y) based on values of independent variables (X). It can be used for the cases where we want to predict some continuous quantity.

Logistic Regression is a method used to predict a dependent variable, given a set of independent variables, such that the dependent variable is categorical.

Linear Vs Logistic Regression



Linear Regression

1

Continuous variables

2

Solves Regression Problems

3

Straight line



Logistic Regression

1

Categorical variables

2

Solves Classification Problems

3

S-Curve

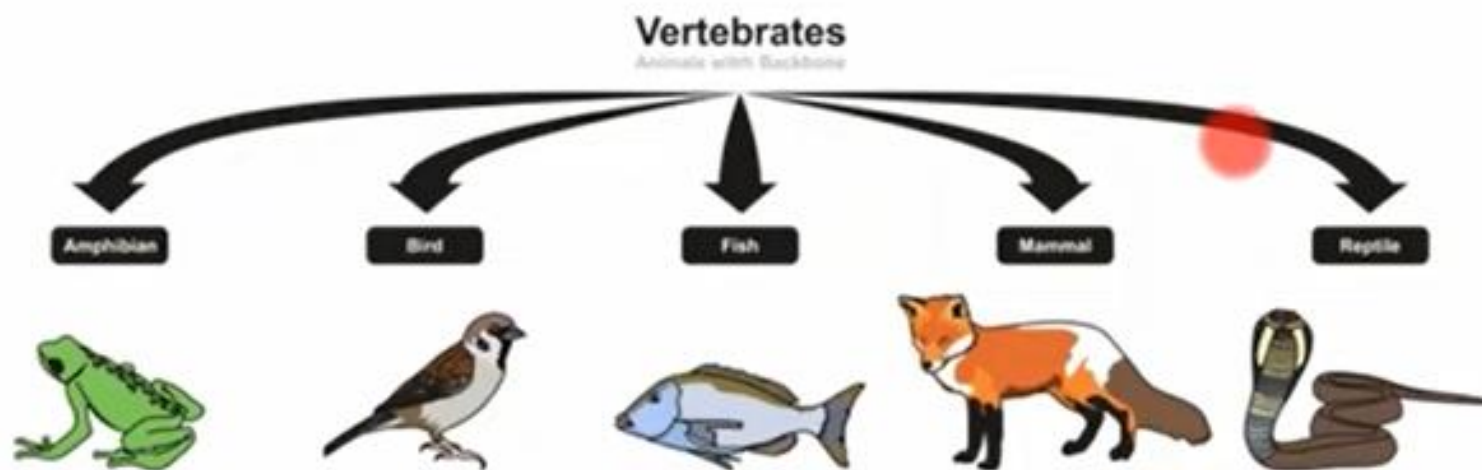
Logistic Regression: Use - Cases



Weather
Predictions

- Logistic regression tells : whether rainy or not rainy, cloudy or not cloudy (classification)
- Linear regression tells: what will be the temperature tomorrow or after 2 days (regression)

Logistic Regression: Use - Cases



Multiclass classification

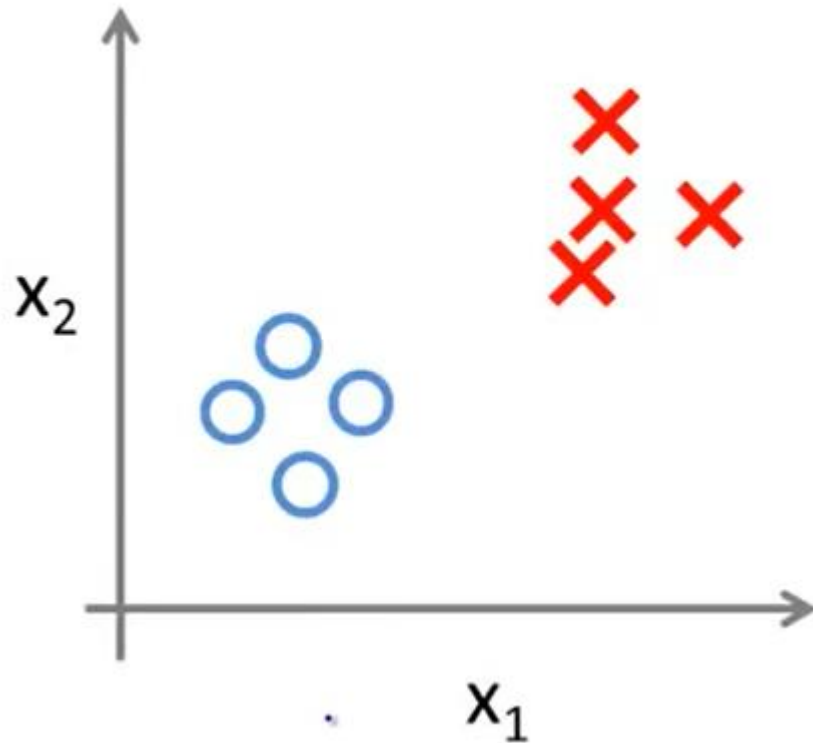
Email foldering/tagging: Work, Friends, Family, Hobby

Medical diagrams: Not ill, Cold, Flu

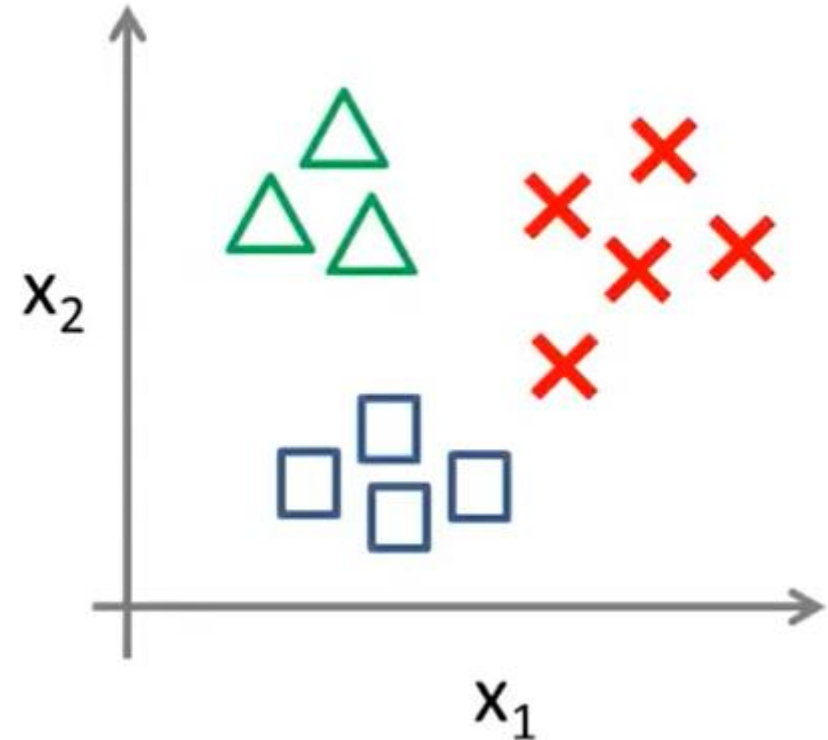
Weather: Sunny, Cloudy, Rain, Snow

Datasets may look like this in different type of classifications

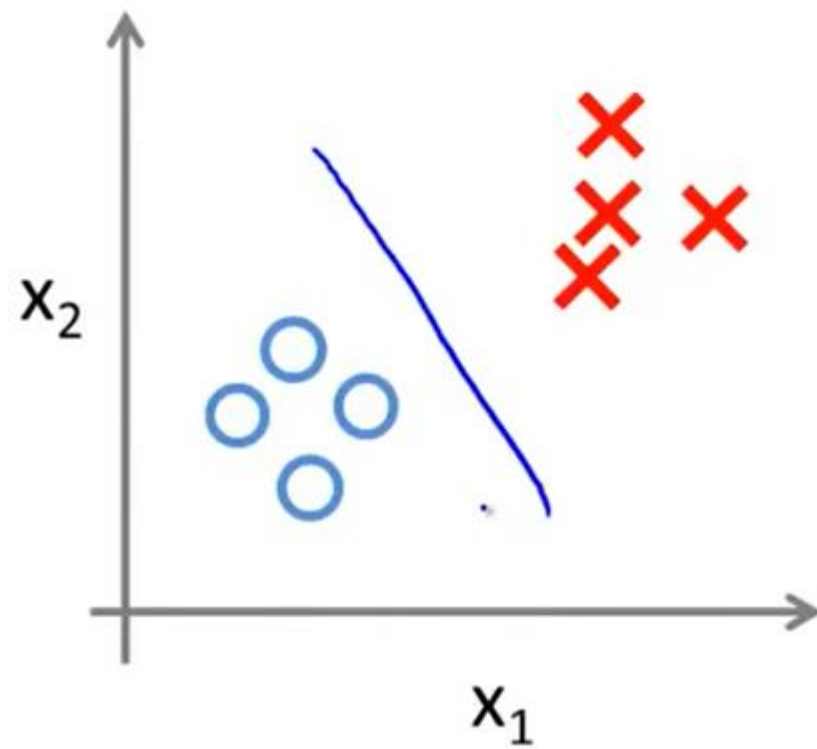
Binary classification:



Multi-class classification:

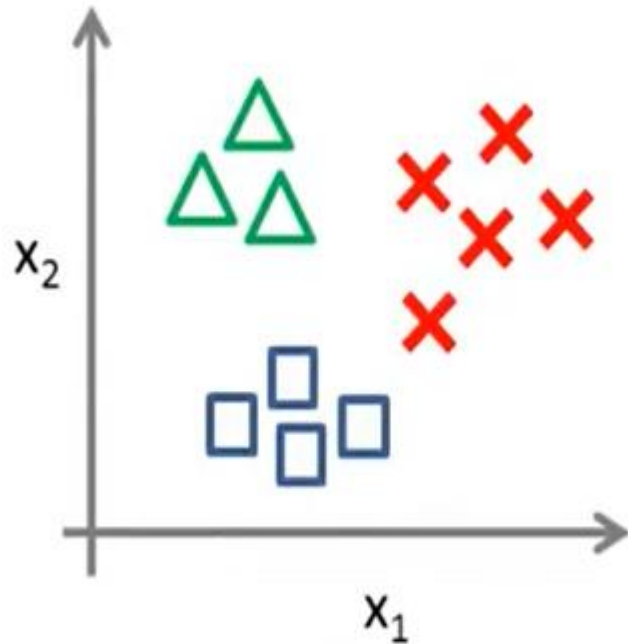





Binary classification:



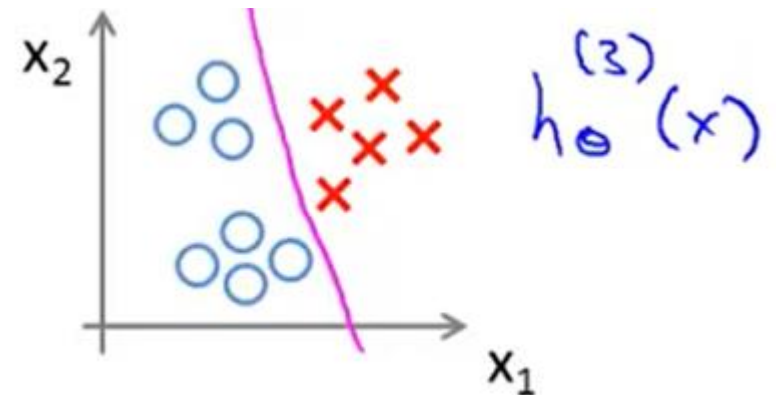
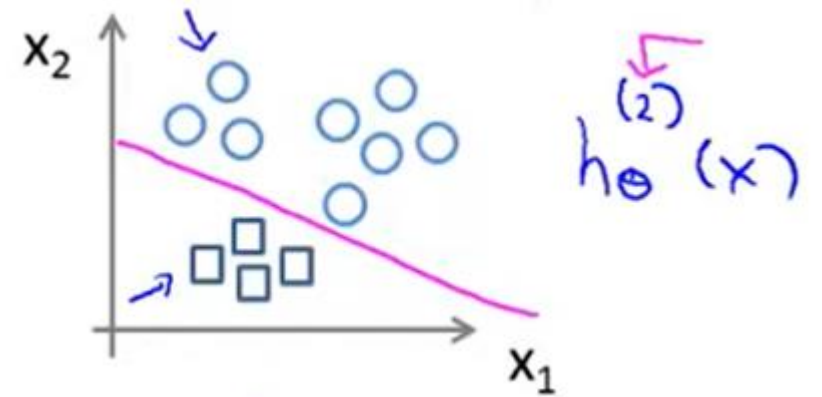
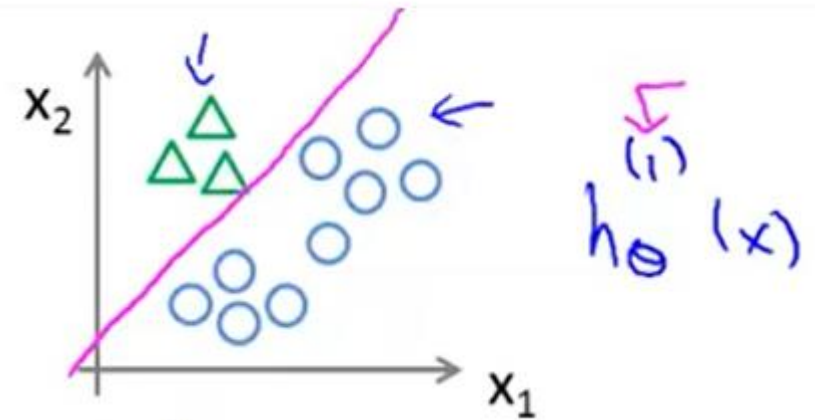
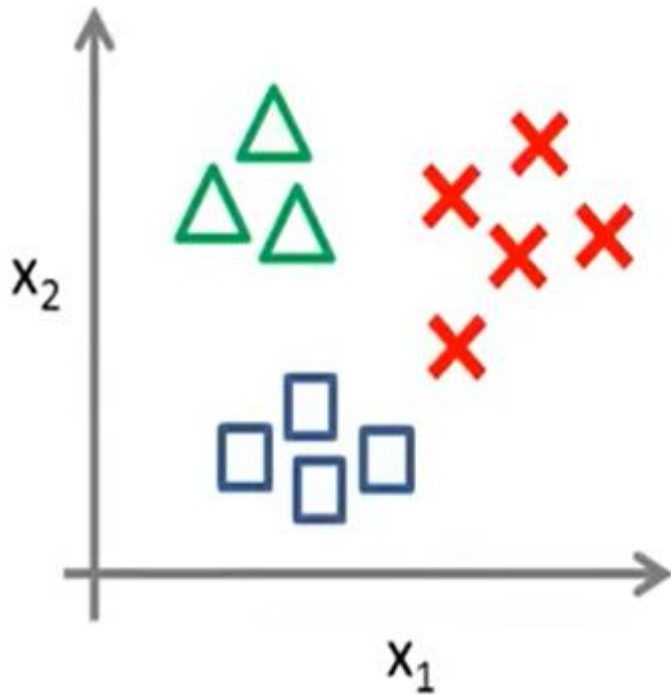
One vs all (one vs rest) approach

It converts the problem into three binary classification problems



Class 1: 
Class 2: 
Class 3: 

One vs all (one vs rest) approach
It converts the problem into three
binary classification problems



Implement Logistic Regression

