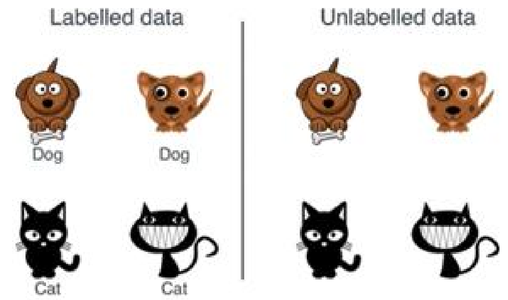
Siddhardhan

# Logistic Regression - intuition

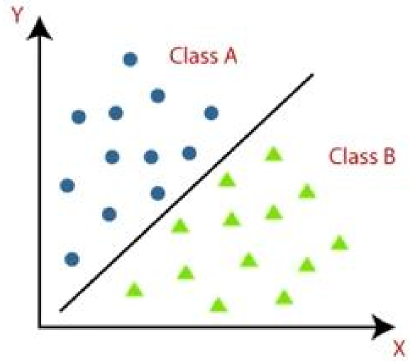
# Logistic Regression

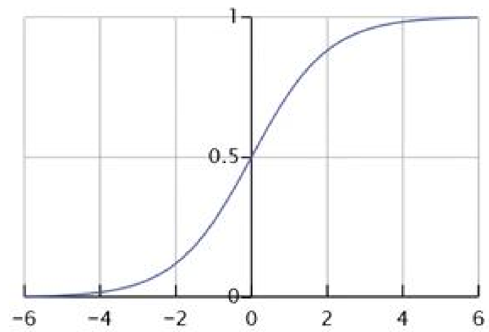
## About Logistic Regression:

1. Supervised Learning Model
2. Classification model
3. Best for Binary Classification Problem
4. Uses Sigmoid function

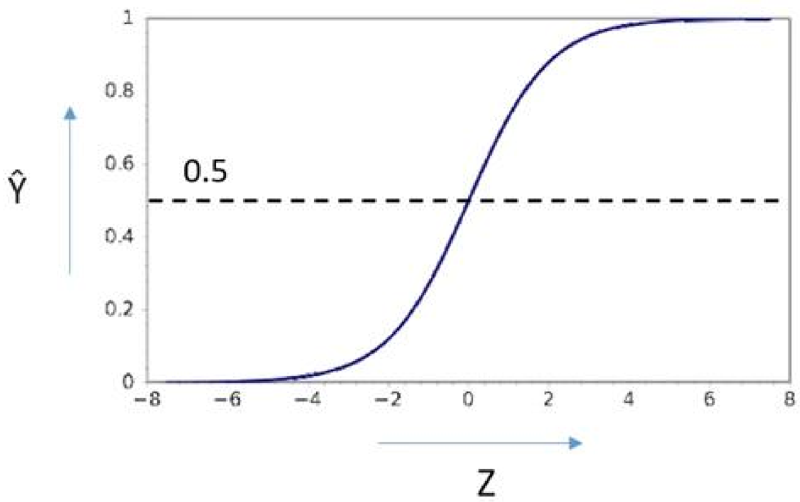




# *Logistic Regression*




$$\hat{Y} = \frac{1}{1 + e^{-Z}} \qquad Z = w.X + b$$

*Sigmoid Function*

$\hat{Y}$ - Probability that (y = 1)

$\hat{Y}$ = P( Y=1 | X )



X - input features

w – weights
( number of weights is equal to the number of input features in a dataset)

b - bias

$\hat{Y}$ = σ (Z)

## *Linear Regression*

### *Advantages:*

1. Easy to implement
2. Performs well on data with linear relationship
3. Less prone to over-fitting for low dimensional dataset



### *Disadvantages:*

1. High dimensional dataset causes over-fitting
2. Difficult to capture complex relationships in a dataset
3. Sensitive to Outliers
4. Needs a larger dataset

# Logistic Regression

| X | -9 | -8 | 0 | 8 | 9 |
|---|---|---|---|---|---|
| Ŷ | | | | | |

$$Z = 5X + 10$$

$$\hat{Y} = \frac{1}{1+e^{-Z}}$$



X = -9

$Z = 5(-9) + 10$

$Z = -35$

$$\hat{Y} = \frac{1}{1+e^{35}}$$

$\hat{Y} = 0$

X = -8

$Z = 5(-8) + 10$

$Z = -30$

$$\hat{Y} = \frac{1}{1+e^{30}}$$

$\hat{Y} = 0$

X = 0

$Z = 5(0) + 10$

$Z = 10$

$$\hat{Y} = \frac{1}{1+e^{-10}}$$

$\hat{Y} = 1$

X = 8

$Z = 5(8) + 10$

$Z = 50$

$$\hat{Y} = \frac{1}{1+e^{-50}}$$

$\hat{Y} = 1$

X = 9

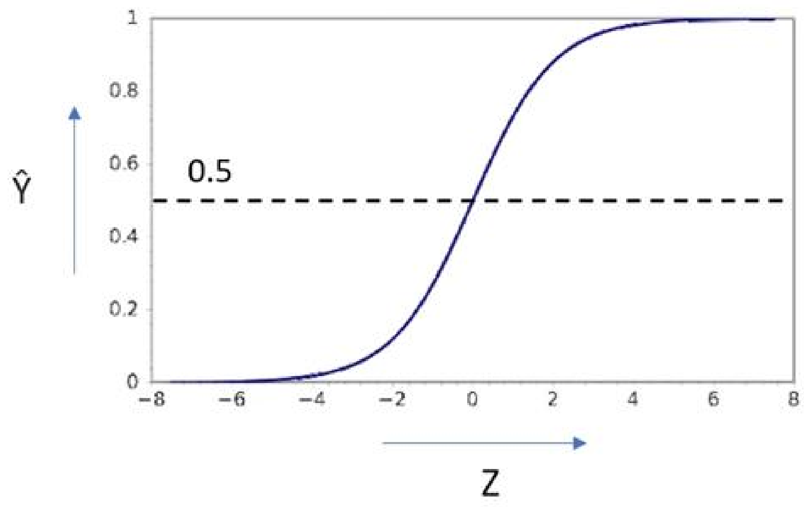
$Z = 5(9) + 10$

$Z = 55$

$$\hat{Y} = \frac{1}{1+e^{-55}}$$

$\hat{Y} = 1$

# Logistic Regression




$$\hat{Y} = \frac{1}{1 + e^{-Z}} \qquad Z = w.X + b$$

*Sigmoid Function*

## Inference:

If Z value is a large positive number,

$$\hat{Y} = \frac{1}{1+0}$$



$$\hat{Y} = 1$$

If Z value is a large negative number,

$$\hat{Y} = \frac{1}{1 + (large\ positive\ number)}$$

$$\hat{Y} = 0$$

Siddhardhan

# Loss Function & Cost Function for Logistic Regression

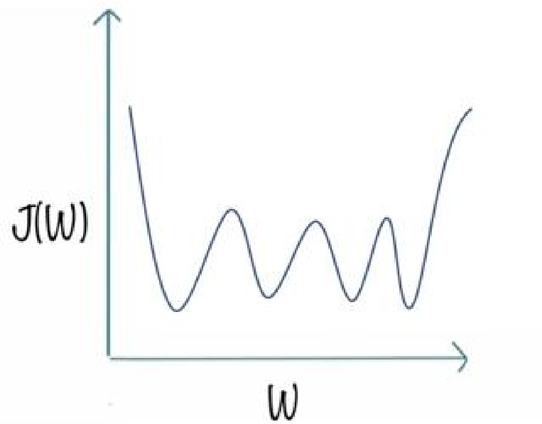


$$J\,(w,b)$$

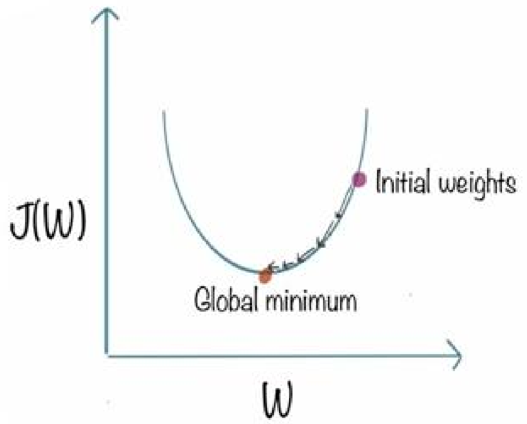
## Loss Function

Loss function measures how far an estimated value is from its true value.

$$\textbf{Loss} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2$$




Gradient Descent
With Local minima



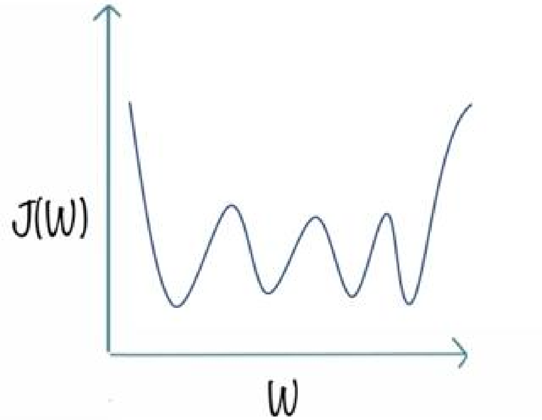

Gradient Descent
With Global minima

# Loss Function

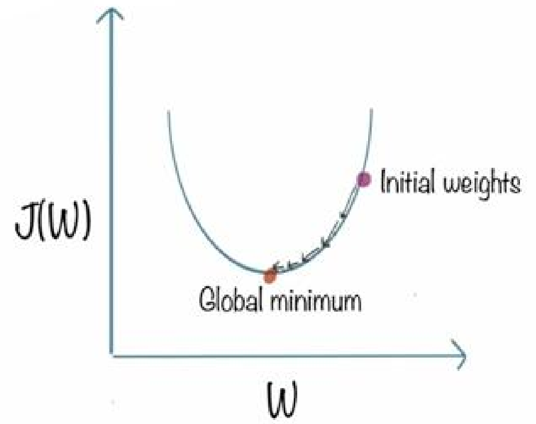
Loss function measures how far an estimated value is from its true value.

$$\textbf{Loss} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2$$

Siddhardhan




Gradient Descent
With Local minima




Gradient Descent
With Global minima

## Loss Function for Logistic Regression

Binary Cross Entropy Loss Function (or) Log Loss :

$$L(y, \hat{y}) = -(y \log \hat{y} + (1 - y) \log (1 - \hat{y}))$$



When $y = 1$, $\Rightarrow$ $L(1, \hat{y}) = -(1 \log \hat{y} + (1 - 1) \log (1 - \hat{y}))$ $\Rightarrow$ $L(1, \hat{y}) = -\log \hat{y}$

We always want a smaller Loss Function value, hence, $\hat{y}$ should be very large,
so that ( $-\log \hat{y}$) will be a large negative number.

When $y = 0$, $\Rightarrow$ $L(0, \hat{y}) = -(0 \log \hat{y} + (1 - 0) \log (1 - \hat{y}))$ $\Rightarrow$ $L(0, \hat{y}) = -\log (1 - \hat{y})$

We always want a smaller Loss Function value, hence, $\hat{y}$ should be very small,
so that $-\log (1 - \hat{y})$ will be a large negative number.

## *Cost Function for Logistic Regression*

Loss function ( L ) mainly applies for a single training set as compared to the cost function (J) which deals with a penalty for a number of training sets or the complete batch.

$$L(y, \hat{y}) = -(y \log \hat{y} + (1 - y) \log (1 - \hat{y}))$$

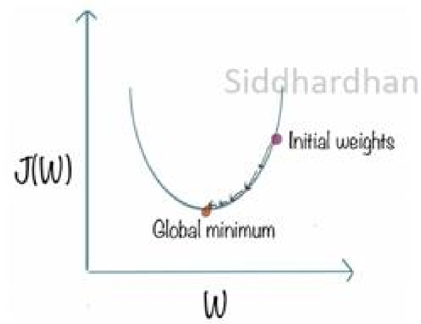


$$J(w, b) = \frac{1}{m} \sum_{i=1}^{m} (L(y^{(i)}, \hat{y}^{(i)})) = -\frac{1}{m} \sum_{i=1}^{m} (y^{(i)} \log \hat{y}^{(i)} + (1 - y^{(i)}) \log (1 - \hat{y}^{(i)}))$$

( ‘m’ denotes the number of data points in the training set)

Siddhardhan

# Gradient Descent for Logistic Regression



# *Logistic Regression*

## *About Logistic Regression:*



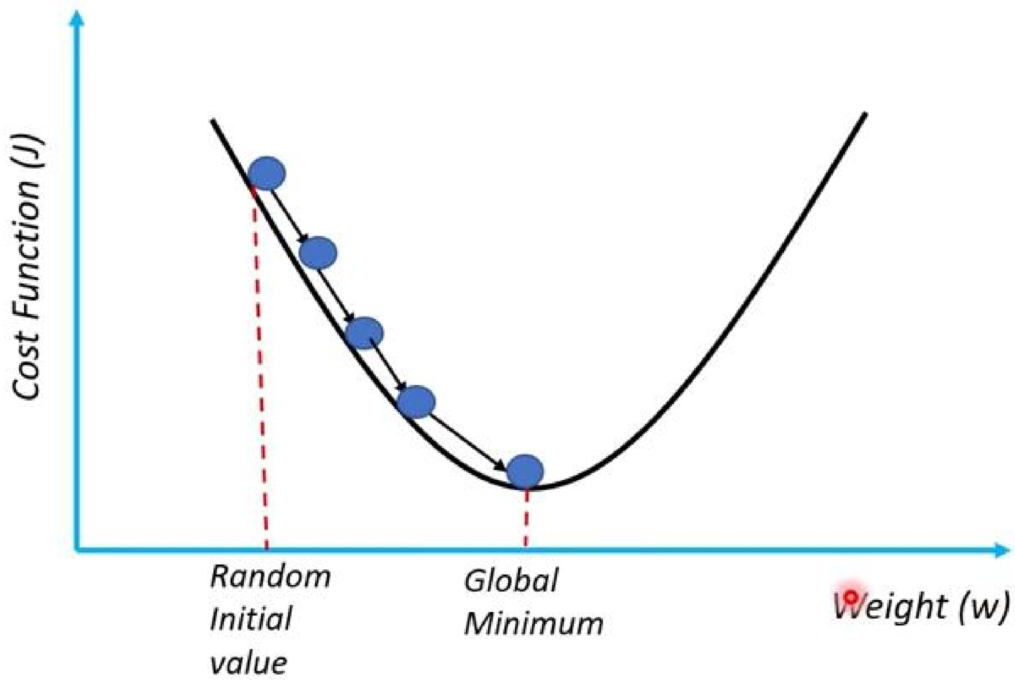
1. Supervised Learning Model
2. Classification model
3. Best for Binary Classification Problem
4. Uses Sigmoid function
5. Binary Cross Entropy Loss Function (or) Log Loss

$$\hat{Y} = \frac{1}{1 + e^{-Z}} \qquad Z = w.X + b$$

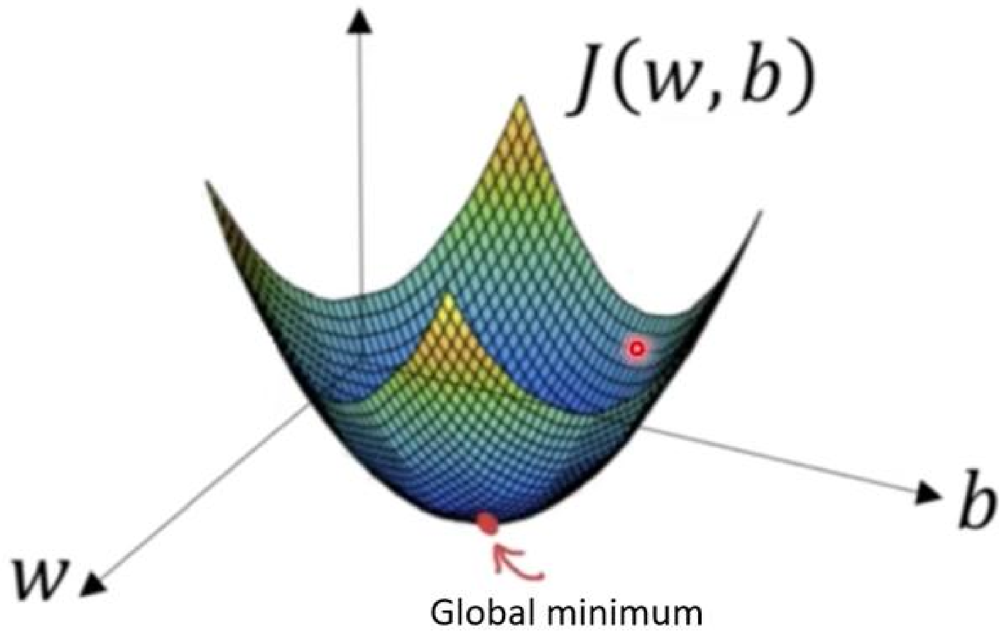
*Sigmoid Function*

$$J(w, b) = \frac{1}{m}\Sigma(L(y^{(i)}, \hat{y}^{(i)})) = -\frac{1}{m}\Sigma(y^{(i)} \log \hat{y}^{(i)} + (1 - y^{(i)}) \log(1 - \hat{y}^{(i)}))$$

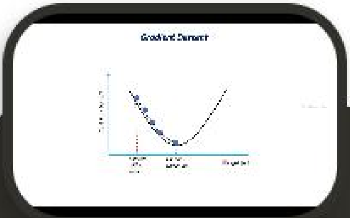
# Gradient Descent



# Gradient Descent in 3 Dimension





# *Gradient Descent*

Gradient Descent is an optimization algorithm used for minimizing the cost function in various machine learning algorithms. It is used for updating the parameters of the learning model.

$$w_2 = w_1 - L*dw$$

$$b_2 = b_1 - L*db$$



w --> weight
b --> bias
L --> Learning Rate

dw --> Partial Derivative of cost function with respect to w

db --> Partial Derivative of cost function with respect to b

$$dw = \frac{1}{m} * (\hat{Y} - Y).X$$

$$db = \frac{1}{m} * (\hat{Y} - Y)$$

# Logistic Regression

## Logistic Regression model:

- Sigmoid Function

$$\hat{Y} = \frac{1}{1 + e^{-Z}} \qquad Z = w.X + b$$

Siddhardhan

- Updating weights through Gradient Descent

$$w_2 = w_1 - L*dw$$

$$b_2 = b_1 - L*db$$

- Derivatives

$$dw = \frac{1}{m} * (\hat{Y} - Y).X$$

$$db = \frac{1}{m} * (\hat{Y} - Y)$$