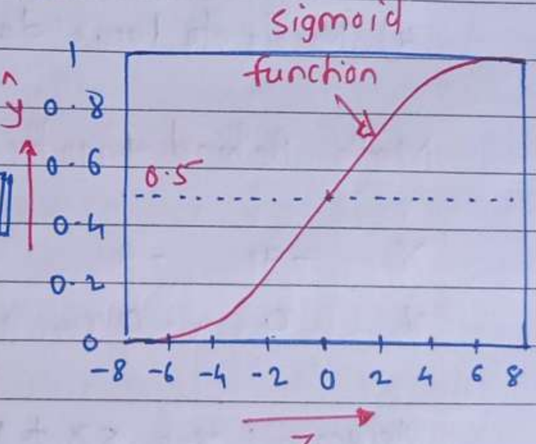
Classification. Algorithm

1) Logistic Regression 2) SVM
3) KNN
4) Decision Tree 5) Random forest

1) Logistic Regression :-
   1. it is supervised learning model.
   2. it is classification model and best for binary classification.
   3. it uses sigmoid function



$$\text{Sigmoid function} = \hat{y} = \frac{1}{1+e^{-z}}$$

where $z = wx + b$.
$(mx + c)$ - eq^n of line

slope $m \longrightarrow$ weight w
intercept $c \longrightarrow$ bias b

- $\hat{y}$ = probability that $(y=1)$
  $\hat{y} = P(y=1|x)$ ... {probability of y being 1 for given value of x}

x = input features
w = weights (it will be in the format)
{number of weight equal to number of feature in the dataset}
b = bias
$\hat{y} = \sigma(z)$.

Date: / /
Page

Advantages: 1) Easy to implement
2) perform well on data with linear relationship
3). less prone to overfitting for low dimensional dataset.

Disadvantages :- 1) High dimensional dataset causes overfitting.
2) difficult to capture complex relationship in dataset
3) sensetive to outlier
4) Needs to large dataset.

Math Behind logistic Regression.

X	-9	-8	0	8	9
y	0	0	1	1	1

Assume · $z = 5x + 10$ $\qquad \hat{y} = \dfrac{1}{1+e^{-z}}$

$x=-9$	$x=-8$	$x=0$	$x=8$	$x=9$
$z=5(-9)+10$ $=-35$	$z=5(-8)+10$ $=-30$	$z=5(0)+10$ $=10$	$z=5(8)+10$ $=50$	$z=5(9)+10$ $=55$
$\hat{y}=\dfrac{1}{1+e^{35}}$	$\hat{y}=\dfrac{1}{1+e^{30}}$	$\hat{y}=\dfrac{1}{1+e^{10}}$	$\hat{y}=\dfrac{1}{1+e^{50}}$	$\hat{y}=\dfrac{1}{1+e^{55}}$
$\hat{y}=0$	$\hat{y}=0$	$\hat{y}=1$	$\hat{y}=1$	$\hat{y}=1.$

Date: / /
Page

Inference : if z value is large positive number.

$$\hat{y} = \frac{1}{1+0} \cong \hat{y} = 1.$$

if z is large negative number.

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

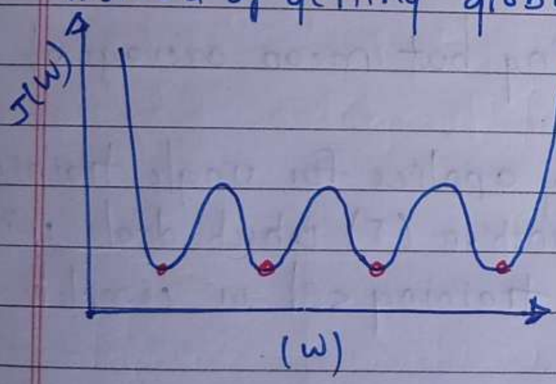
$$\hat{y} = 0.$$

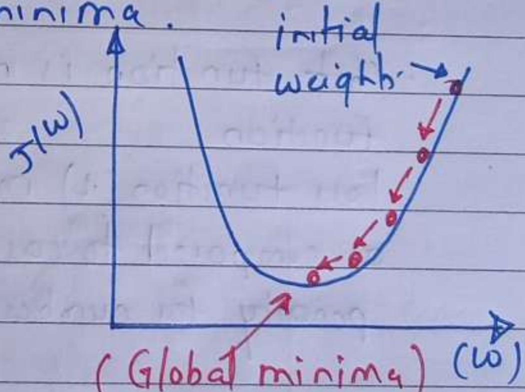
loss function & cost function for Logistic Regression.

- loss function measures how far an estimated value is from true value.

loss function for linear regression $= \frac{1}{n} \sum_{i=1}^{n} (y_i - y_{pred})^2$

if we use this function we will get many local minima instead of getting global minima.





- Binary cross entropy loss function (or) log loss.

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

Here,

$y \longrightarrow 0$ or 1

IIly $\hat{y} \longrightarrow 0$ to 1 (probability could be continuous) when $y=1$

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

- since we always won't smaller loss function value hence $\hat{y}$ should be very large (from 0 to 1) if it is the $-\log \hat{y}$ will be very large negative number or very small number.

when $y=0$

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

- Since we want smaller loss function value, hence $\hat{y}$ should be very small the automatically $(1-\hat{y})$ will be very large thus $-\log(1-\hat{y})$ will be large negative number or very small number.

- Cost function is nothing but mean average of loss function
- loss function (L) mainly applies for single training set as compared to cost function (J) which deals with a penalty for number of training set or complete batch.

loss function:

$$L(y, \hat{y}) = -(y \log \hat{y} + (1-y)\log(1-\hat{y})) \text{ — for single}$$

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

$$-\frac{1}{m}\sum_{i=1}^{m}\left(y^{(i)}\log \hat{y}^{(i)} + (1-y^{(i)})\log(1-\hat{y}^{(i)})\right)$$

{'m' denotes number of data points in the training set}.