Lecture - 02:

→ Ridge and Lasso Regression.

→ Assumption of Linear regression.

→ Logistic Regression.

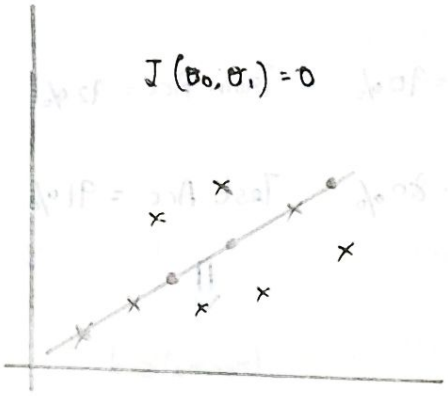
→ Confusion Matrix.

→ Practicals for linear, Ridge, Lasso and Logistics.

Ridge and Lasso Regression:

Let's take,

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



Suppose We trained the model with help of subset of data. And also fit the line.

Then, new data points are come (x). But the difference between new datapoints (testing data) and predicted points are huge differences.

It is categorized into two conditions,

Overfitting:

Model perform well in Training data. [Low Bias]

fails to perform well in Testing data [High Variance]

Underfitting:

Model Accuracy is bad with training data. [High bias]

Model Accuracy is also bad with Test data. [High Variance]

Bias - The error between average model prediction and the ground truth.

Let's Consider Some models,

Model 1	Model 2	Model 3
Training Accuracy = 90%	Train Acc = 92%	Train Acc = 70%
Testing Accuracy = 80%	Test Acc = 91%	Test Acc = 65%
⇓	⇓	⇓
Overfitting	Generalised model.	Underfitting.
[Low Bias, High Var]	[Low Bias, Low Variance]	[High bias, High Var]

Let Consider the One Scenario,

Assumption,

$$J(\theta_1) = 0$$

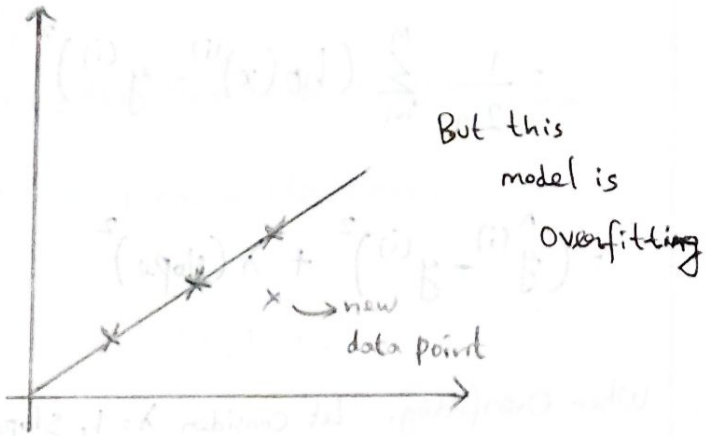
$$\text{Cost function} = \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)}\right)^2$$

$$\boxed{h_\theta(x) = \hat{y}}$$

$$= \left(\hat{y}^{(i)} - y^{(i)}\right)^2 = 0$$

The value comes to the zero. Because the $J(\theta_1) = 0$

With below graph,



Because of overfitting, we conclude this model value does not come to zero. Because, the new data points and costfunction gives different value after applying a best fit line. we need to minimize the cost function. Then only get generalized model.

So, we need to minimize and prevent overfitting conditions. Therefore ridge regression will comes to the picture.

Ridge Regression: [L2 Regularization]

Ridge Regression says that the,

$$\left(\hat{y}^{(i)} - y^{(i)}\right)^2 + \lambda\,(\text{slope})^2$$

The Cost function, $h_\theta(x) = \theta_0 + \theta_1 x$

$\boxed{\theta_0 = 0}$

When, $J(\theta_1) = 0$

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

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

$\boxed{h_\theta(x) = \theta_1 x}$ ↳ slope

$$= \left(\hat{y}^{(i)} - y^{(i)}\right)^2 + \lambda (\text{slope})^2$$

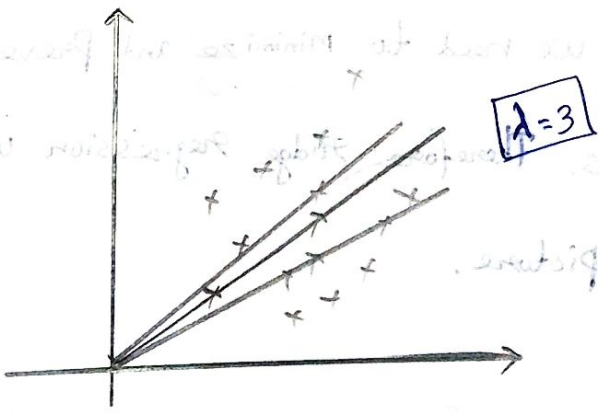
$\boxed{\hat{y}^{(i)} = h_\theta(x)}$

When Overfitting, Let consider $\lambda = 1$, slope = 2

$$= 0 + 1(2)^2$$

$$= 4 \downarrow\downarrow$$

Then minimize the cost function 4, In order to reduce this the θ_1 will be Change.



Try multiple Combination of θ_1 value. To fit the line and avoid Overfitting.

So, take the next fit line (i.e) change θ_1 value,

$$= (\hat{y}^{(i)} - y^{(i)})^2 + \lambda (\text{slope})^2$$

$\Downarrow$

Obivously the value to be small Compare than the previous iteration.

$$= (\text{small value}) + 1(1.36)^2$$

$$\simeq 3 \downarrow\downarrow$$

In this way, The cost function reduce and fit the best line.

After many iteration, the value will small and fit the best line.

λ is nothing but, Hyperparameter

iterations {Hyperparameter} (λ)

λ decides the how the line steeps makes the line better with the help of iterations.

θ_1 value change is based on the Convergence Algorithm.

$\lambda \rightarrow$ Hyperparameter.

R^2, adjusted R^2 is based on the Hyperparameter

Lasso (L_1 Regularization):

Lasso Regression

The L1 regularization says that the,

$$= (\hat{y} - y)^2 + \lambda |\text{slope}|$$

↳ It is also used for feature selection.

So, the equation will be,

$$h_\theta(x) = \hat{y} = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3 + \dots + \theta_n x_n$$

$$= (\hat{y} - y)^2 + \lambda \left| \theta_0 + \theta_1 + \theta_2 + \theta_3 + \theta_4 + \theta_5 \dots \theta_n \right|$$

In this case, we are not squaring the slope. Instead we are using Mod. So, it is neglecting the features those are not important to perform.

The important of the L1 regularization are the,

1. Prevent Overfitting.
2. Feature Selection.

} → L1 Regularization.

The hyperparameter (λ) is assigned Something Called as Cross Validation.

In real time, Most of the cases we are performing the L1 regularisation and L2 regularization then use the performance Metrics we can use that.

Ridge Regression (L2 Regularization) Overview:

It is also Called as L2 Normalization.

$$\text{Cost function} = (h_\theta(x)^{(i)} - y^{(i)})^2 + \lambda (\text{slope})^2$$

Purpose: Prevent Overfitting.

Lasso Regression (L1 Regularization) Overview:

It is also Called as L1 Normalization.

$$\text{Cost function} = (h_\theta(x^{(i)}) - y^{(i)})^2 + \lambda |\text{slope}|$$

Purpose:

→ Prevent Overfitting.

→ feature Selection

Assumption of Linear Regression:

1. Normal / Gaussian distribution → Model will get trained well.

2. Standardization {Scaling data} → Applying Z-Score $\{\mu = 0, \sigma = 1\}$

→ If the distribution is not Gaussian, Make the normal distribution using mathematical and Statistical methods.

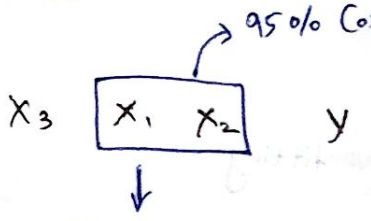
→ Standardization is Compulsory ?

If the Standardization is not used, The value of gradient descent points to be large. So, It takes to long ~~train~~ time to train the algorithm.

3. Datapoints should be linearity.

4. Multi Collinearity.

Let say,



This feature is correlated to(y). Definitely any one of the feature is dropped.

No need to take both the features for the prediction.

5. Homoscedasticity. [It means two group variance is approx Similar].

6. Variance Inflation factor.

Variance Inflation factor:

We need to Understand VIF we know,

1. R^2 2. Simple mathematics $\left[\frac{x}{y}\right]$

if X Constant, Y increases. The value should be an decrease. Example: $\frac{2}{6} = 0.333$

if y constant, x increases is just Vice-Versa.

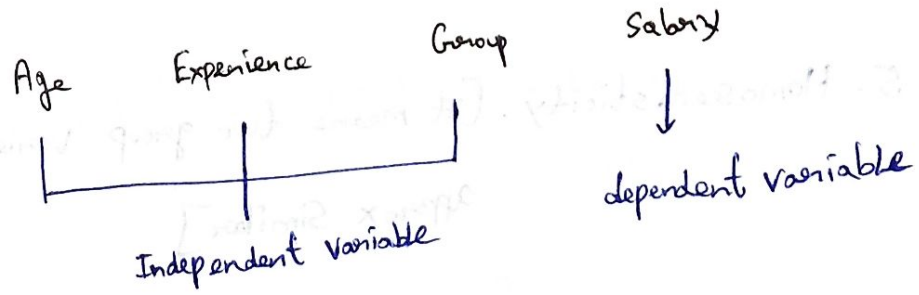
VIF definition,

VIF is a term or measurement through which we can know which variables in the data highly Correlated with other variables.

Example:

Consider the employee dataset.

Features,



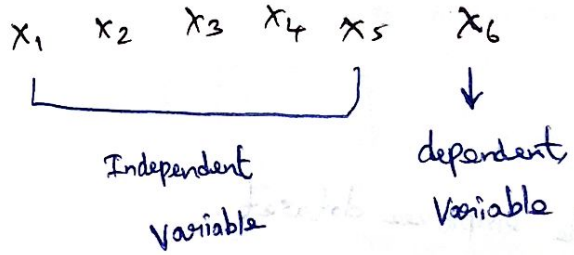
Suppose we know One-one Correlation Coefficient, we use normally matrix.

like,

	x_1	x_2	x_3
x_1	0.0	0.1	0.2
x_2	-	-	-
x_3	-	-	-

We know only two features Correlation. But the VIF is used to one feature alone and Compare all the independent features in the dataset and give single Coefficient value.

Suppose,



We need to Compare X_1 feature to all other independent features.

How?

Take the feature as independent Variables,

X_1 | X_2 X_3 X_4 X_5

Take X_1 feature and Create Seperate model of regression.

After fit linear regression,

$$X_1 = \theta_0 + \theta_1 X_2 + \theta_2 X_3 + \theta_3 X_4 + \theta_4 X_5$$

After fit this model, the R^2 value will be give as Something.

$$\text{VIF formula} = \frac{1}{1 - R^2}$$

If R^2 Something 1.237, The VIF $= \frac{1}{1 - 1.237} = \frac{1}{-0.237}$

$$= -4.2194$$

1.237 R^2 is away from the magnitude of R^2. So, in this situated at very rare Cases.

If R^2 Something 0.9, The VIF $= \frac{1}{1-0.9} = \frac{1}{0.1}$

$= 10$ ..

This way the VIF is Calculated, The final output of all VIF of each feature would be,

X_1	X_2	X_3	X_4	X_5
6	10	2-1	5-6	5-8

The feature X_3 is low VIF, So the feature X_3 will be considered for analysis.

The high VIF will be dropped based on the features.

The X_4 and X_5 have Similar Same VIF, Suppose the X_5 is 5-8. Definitely, drop any one of the feature. It is Some thing Called as the Multicollinearity.

The VIF is play a Vital role in the Case of multicollinearity.

Logistic Regression: [Classification Algorithm]

Logistic Regression Very wells works in Binary Classification.

It also used to solve multiclassification problems.

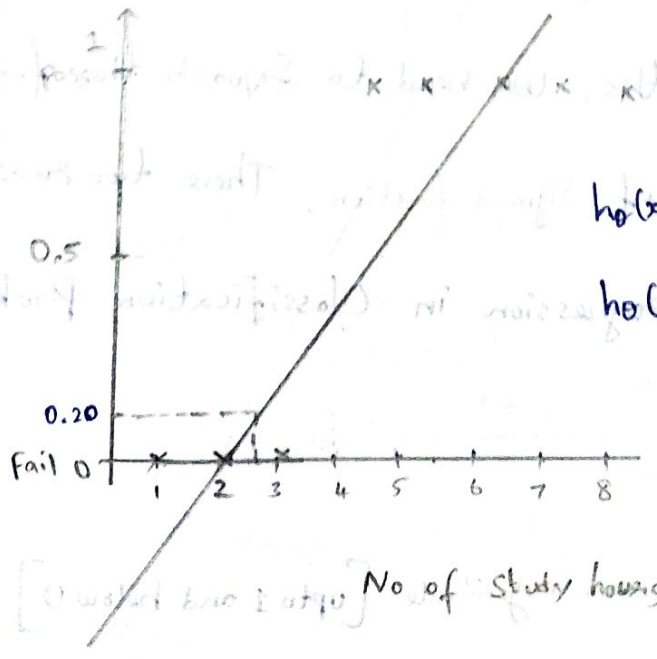
Example,

No of Study	No of play	Pass / fail
-	-	Pass
-	-	Fail

Why linear regression is not used in Classification Problems?

Suppose Used linear regression it would be,

(In dataset says less than 3 hour study leads to fail)



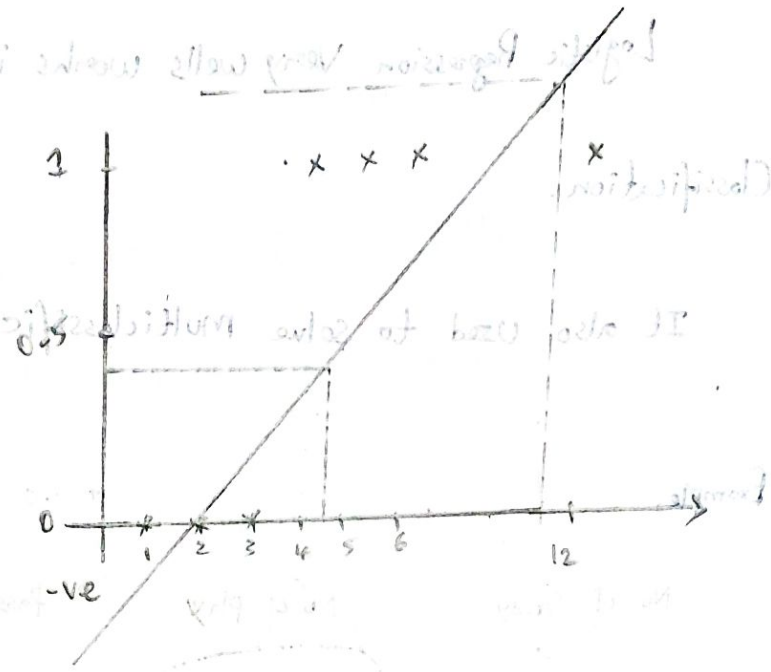
Linear Regression

$h_\theta(x) < 0.5 \Rightarrow 0 \rightarrow$ fail

$h_\theta(x) \geq 0.5 \Rightarrow 1 \rightarrow$ Pass

2.5 hours $\Rightarrow$ fail

Suppose any outliers are there, The graph should be,



The linear line change because of outliers. So, the student study 4.5 hours also fail in this case. But it is not correct. In some times, The outlier or any related values predict the output as high magnitude and above 1 in the graph and same like sometime below in the graph. It leads to causes the problem.

To overcome this, we need to squash the function with the help of Sigmoid function. These two reasons to avoid linear regression in Classification problem are,

→ Outliers.

→ More over Magnitude. [upto 1 and below 0]

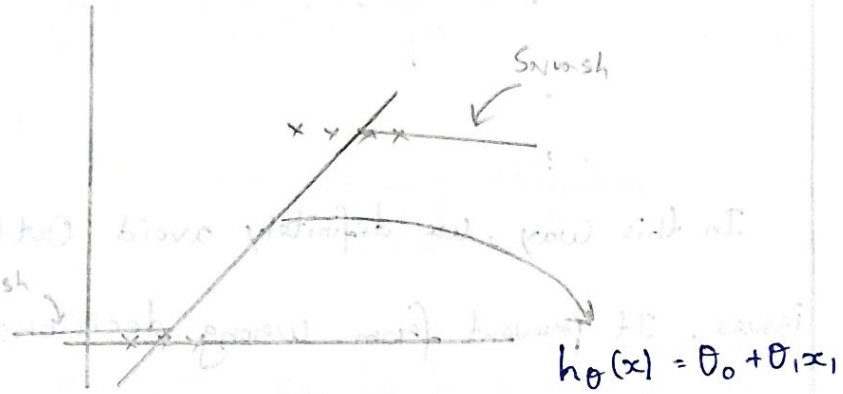
Decision Boundary Logistic function:

Let,

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

Written as,

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



We need to Squash, so applied some function. So,

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

Let $z = \theta_0 + \theta_1 x_1$

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

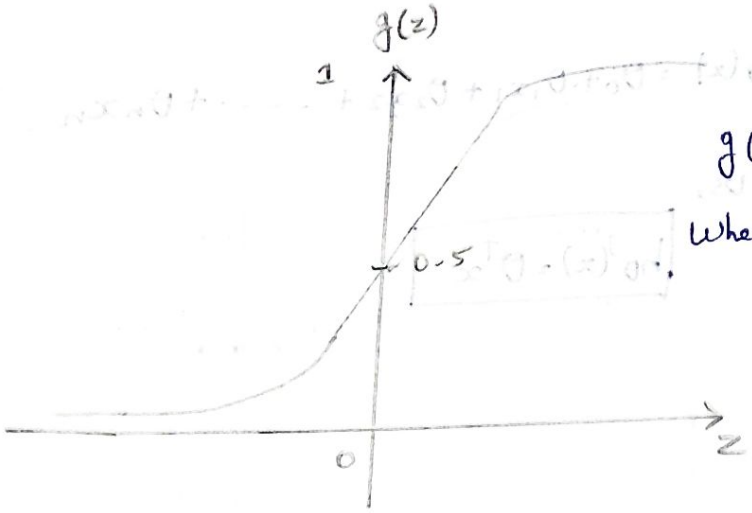
$$h_\theta(x) = \frac{1}{1+e^{-z}}$$ → Sigmoid or logistic function. [This function helps to squash]

Finally, the equation will be

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

This is known as the logistic function (or) regression.

The Sigmoid function look like,



$g(z) \geq 0.5$ when $z \geq 0$ → Pass

$g(z) < 0.5$ when $z < 0$ → Fail

In this way, we definitely avoid Outliers Magnitude issues. It prevent from wrong decisions we are made.

Solve problem statement:

Training set,

$$\{(x^1, y^1), (x^2, y^2), (x^3, y^3) \ldots (x^n, y^n)\}$$

(x^i, y^i) → datapoints.

$y \in \{0, 1\}$ → 2 o/p!

[Binary Classification]

Let's take,

$$h_\theta(z) = \frac{1}{1+e^{-z}} \quad \text{where } z = \theta_0 + \theta_1 x_1 \quad [\because \theta_0 = 0]$$

$$\boxed{z = \theta_1 x_1}$$

Change the parameter θ_1 ?

Yes, Because we need to fit the line.

Cost function,

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

↳ Cost function of linear.

Logistic Regression equation,

$$\boxed{h_\theta(x) = \frac{1}{1+e^{-(\theta_1 x)}}} \rightarrow \text{Equation}$$

We just apply logistic eq to linear cost function,

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

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

This is the cost function of logistic regression. But we cannot use this cost function for logistic regression. Because $\boxed{h_\theta(x) = \frac{1}{1+e^{-(\theta_1 x)}}}$ this equation is a non-convex function..

↳ Non-Convex function.

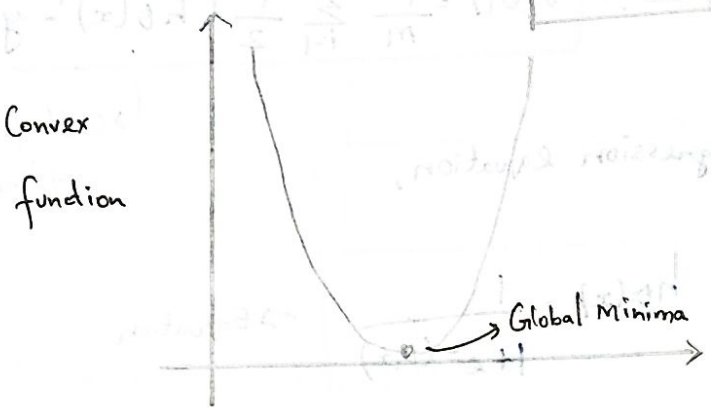
Convex and Non-Convex functions are related to the Gradient descent.

The Convex function mostly occurs in the linear regression, $h\theta(x) = \theta_0 + \theta_1 x$

So, the function looks like,

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

Cost function.



In linear regression Cost function easily to reach the Global minima. So, it is Called as Convex function.

But Logistic Regression,

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

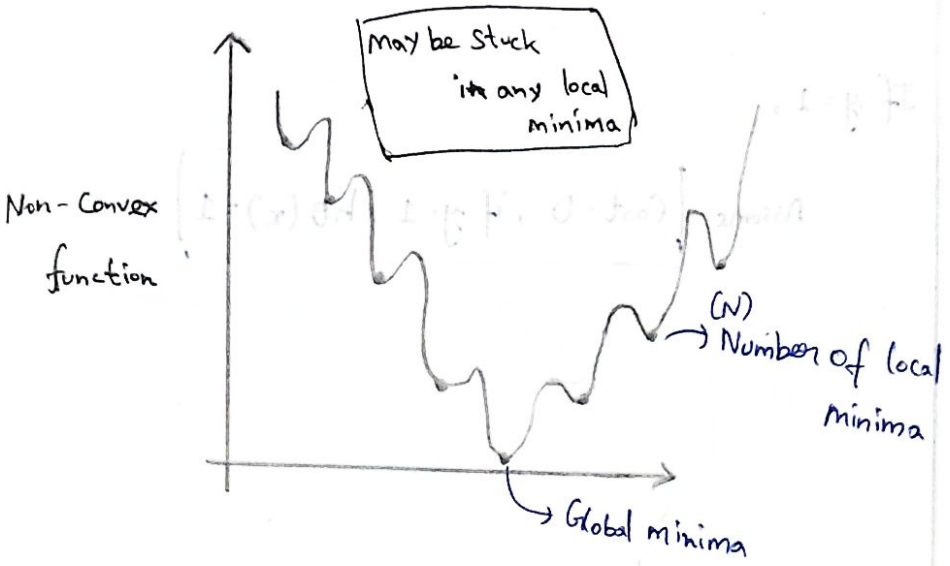
The non-convex function mostly occurs in the logistic (or) Sigmoid.

So, the function look like,

↙ Applied Logistic Cost function (It is not Correct one)

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

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



In this Case, The Global minima never reached. Because, So many number of local minima. So, we Cannot get the Global Minima.

Inorder to avoid the above Secenario,

The actual Cost function of logistic regression is

$$J(\theta_1) = \begin{cases} -\log\left(h_\theta(x)^{(i)}\right) & ; \text{if } y=1 \\ -\log\left(1-h_\theta(x)^{(i)}\right) & ; \text{if } y=0 \end{cases}$$

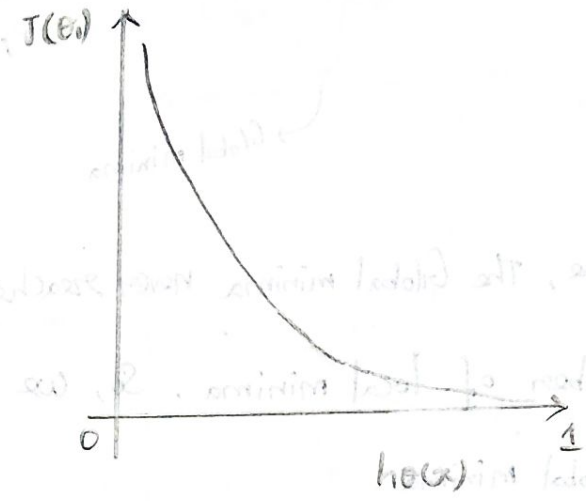
$$\text{where } h_\theta(x) = \frac{1}{1+e^{-(\theta_1 x)}}$$

This is the required Cost function of Logistic regression.

The above Cost function basically mean,

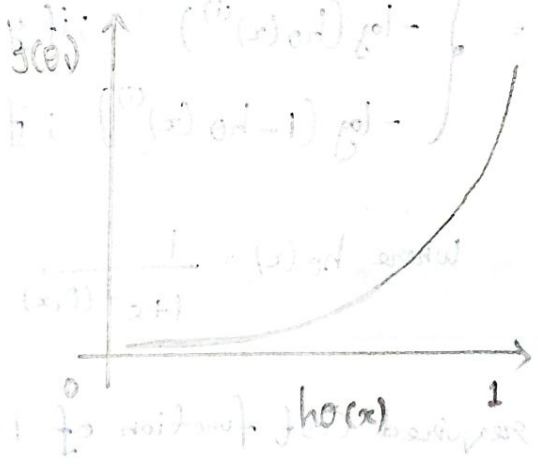
If $y = 1$,

Assume $\boxed{\text{Cost} = 0 \text{ , if } y = 1 \text{ , } h\theta(x) = 1}$



If $y = 0$,

Assume $\boxed{\text{Cost} = 0 \text{ , if } y = 0 \text{ , } h\theta(x) = 0}$



When Combine above two graph, we get the gradient descent. Then, find the Global minima

The final cost function,

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

We combine the above cost function we get,

$$\text{Cost}(h_\theta(x)^{(i)}, y) = -y \log(h_\theta(x)^{(i)}) - (1-y) \log(1 - h_\theta(x))$$

↳ Cost function

if we apply $y = 1$,

$$\text{cost}(h_\theta(x^i), y) = -\log(h_\theta(x)^{(i)})$$

if $y = 0$,

$$\text{cost}(h_\theta(x^i), y) = -\log(1 - h_\theta(x^{(i)}))$$

} Cost function

why y^i?
Because, It is binary classification there are two outputs.

So finally,

$$J(\theta_1) = -\frac{1}{2m} \sum_{i=1}^{m} \left[(y^i \log(h_\theta(x)^i) + (1-y^i) \log(1 - h_\theta(x^i)) \right]$$

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

↳ This is Required final Cost function for logistic regression

Repeat until Convergence {

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} (J(\theta_1))$$

}

↳ This is the Convergence algorithm of Logistic regression

Performance Matrics [Classification problem]

The Consider One problem for binary Classification

let Say,

x_1	x_2	y [Actual o/p]	$\hat{y}$ [Predicted o/p]
-	-	0	1
-	-	1	1
-	-	0	0
-	-	1	1
-	-	1	1
-	-	0	1
-	-	1	0

Based on the y and $\hat{y}$, we Calculate the how good the model is with the help of Confusion Matrix.

The first y and $\hat{y}$ output are different. In this Case our first prediction was wrong.

The accuracy of the Model is Calculated with the help of Confusion Matrix.

Actual o/p (y)

Predicted o/p ($\hat{y}$)	1	0
1	1	1
0		1

y	$\hat{y}$	Put
0	1	1
1	1	1
0	0	1

Actual o/p (y)

Predicted ($\hat{y}$)	1	0
1	2+1	2
0	1	1

y	$\hat{y}$	Put
1	1	Already 1 + 1 = 2
1	1	Already 2 + 1 = 3
0	1	Already 1 + 1 = 2
1	0	1

The final Confusion matrix,

	1	0
1	3	2
0	1	1

The binary Classification Confusion matrix look like,

Predicted o/p \ Actual o/p	1	0
1	TP	FP
0	FN	TN

→ This is called as Confusion matrix.

TP and TN are the right because the actual input equal to the predicted input. But FN and FP are just vice-versa. So, The better accuracy we need to decrease FN and FP.

So, the accuracy formula will be,

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

$$\left[\because \text{Acc} = \frac{\text{No of Correct Predict}}{\text{Total no of Pred}}\right]$$

$$= \frac{3+1}{3+2+1+1} = \frac{4}{7}$$

$$= 0.57 \Rightarrow 57\%.$$

We reduce FP and FN, we get better accuracy.

Suppose, the binary Classification problem dataset look like this,

In a dataset,

0 → 900
1 → 100

} In this Scenario, we called it imbalanced dataset (or) biased dataset.

Suppose,

0 → 600
1 → 400

} In this case, we say Balanced dataset (or) Normal dataset.

So, let's take,

0 → 900
1 → 100
} Biased dataset

The above case, The model was predicted mostly as 0. Because the model trained more 0 because of high count. But the model accuracy is good and high.

{model → 0 = 900/1000 = 90%} ∴ Accuracy = (Correct number of predict) / (Total no of data pred)

↑ Mostly predicted

The accuracy is good and fine. But the prediction is mostly zero.

If the data is balanced, Then we go Accuracy performance metrics. If the data is Unbalanced, we go to the following one,

→ Recall.
→ Precision.
→ F-Score.

Recall, Precision and F-Score will be based on what type of the problem actually Given.

Precision: Out of all Tp and Fp, how many are predict corrected. (TP).

$$\text{Precision} = \frac{TP}{TP+FP}$$

Recall: Out of all TP and FN, how many are predict Corrected. (TP)

$$\text{Recall} = \frac{TP}{TP+FN}$$

The Confusion matrix look like,

Predict Value	Actual value 1	Actual value 0
1	TP	FP
0	FN	TN

Real time example of Confusion Matrix, [Recall]

Classification evaluation metrics.

Primary building blocks - TP, TN, FP, FN.

Our Classification problem about wheather the person is Covid or not.

Covid - POSITIVE CLASS

Not covid - NEGATIVE CLASS.

TP Actual : Covid Model_pred : Covid Outcome : Patient Admitted.	FP Actual : Not Covid Model : Covid. Outcome: Patient Admitted
FN Actual : Covid Model : Not Covid Outcome: Patient is not admitted risk of infecting others.	TN Actual: Not Covid Model: Not Covid Outcome: Patient is not admitted.

Out of above four cases, FN has need to reduce. Because, the model predict 'Not covid' but actual is Covid. It may be infecting others. So, we need to focus of FN. So, we use the performance metrics called as Recall.

$$\text{Recall} = \frac{TP}{TP+FN}$$

Another Example,

has person Cancer (or) Not

⇓

Recall

Consider another one: [precision]

Our classification problem is about the Wheather the mail is Spam (or) Ham. [Spam Classification]

		Actual: Spam	Actual: Ham
Predict	Spam	TP	FP
	Ham	FN	TN

Let's breakdown;

[TP] Actual → Spam (T) but predict → Spam (P) [ok] ✓

[FP] Actual → Ham (F) but predict → Spam (P) (x)

↳ It causes danger because the Spam Message also Consider in the Inbox. [maybe missing some Original msg]

[FN] Actual → Spam (F) but predict → Ham (N) (x)

↳ It also in the inbox and mixing in Original message.

[TN] Actual → Ham but predict → Ham [ok] ✓

The FP and FN has more danger. Compare both

FP is more important.

Suppose message is Spam Ham, but not consider we lose Something. [FP]

And other Side Machine does not make 100% cret prediction. [Actual → Spam, pred → Ham]. we also need Some Safetyness. [FN]

So, Therefore in this case, we use Precision formula to evaluate.

$$\text{Precision} = \frac{TP}{TP+FP}$$

Last Example: [F-Beta]

Our Classification problem is about Tomorrow Share ~~mo~~ Market going to Crash.

Confusion matrix, Actual

Predict	Crash	Not Crash
Crash	TP	FP
Not Crash	FN	TN

The above Confusion Matrix,

TP ↔ ~~Says the bank Crash~~

FP → Actual the market → not Crash

Predict → Crash

So, in this case the peoples are sells the stock because of Crash and the people does not buy any stocks.

FN → Actual the market → Crash

Predict → Not crash.

In this case, the market going to be Crash but prediction is not Crash. So, the market owners are not ready to recover.

In this both cases, we need to focus both FP and FN. Then we are go to the F-Beta.

$$\text{F-Beta} = (1+\beta^2)\frac{\text{Precision} \times \text{Recall}}{\beta^2 \times \text{precision} + \text{Recall}}$$

Let's Consider,

$$\beta = 1 = \frac{(1+1)\ \text{Precision} \times \text{Recall}}{(1)\ \text{Precision} + \text{Recall}}$$

$$= \frac{2\ (\text{Precision} \times \text{Recall})}{\text{Precision} + \text{Recall}}.$$

Harmonic mean

↓ It is Somthing called

Harmonic mean $= \frac{2xy}{x+y}$

Suppose, $\beta = 0.5$,

$$= \frac{(1+(0.5)^2)\ P \times R}{(0.25)\ P + R}$$

Same $\beta = 2$, ...

Suppose Use $\beta = 2$ is called F_2-Score.
Suppose Use $\beta = 0.5$ is called $F_{0.5}$-Score
Suppose Use $\beta = 1$ is called F_1-Score.

} likewise we say F-Score

What is and why Beta Value Used ?

Beta is deciding parameter of F-Score.

If $\beta = 1$, The Same importance for both FP and FN.

If $\beta < 1$ like $\beta = 0.5$, we give more imporitant to FP than FN.

If $\beta > 1$ like $\beta = 2$, we give more important to FN ther FP.