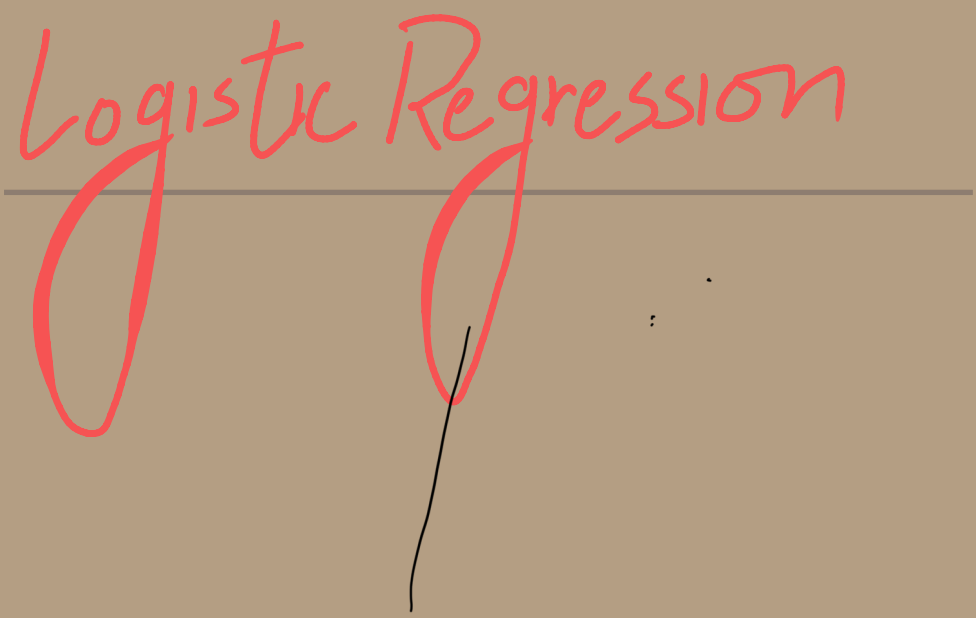
Logistic Regression



DJ

Classification of data

Image data

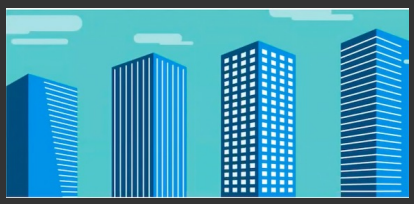
Static

- Doesnot change after it being recorded
- Order/Sequence is not important

Eg: Heights of buildings

Day 1 | 1 | 2 | 3 | 4 |

Day 2 | 1 | 2 | 3 | 4 | .....



Dynamic

- Data is periodically updated, new info is available every time
- Must follow a sequence

Eg Stock Price

| 25 | 50 | 45 | Day 1

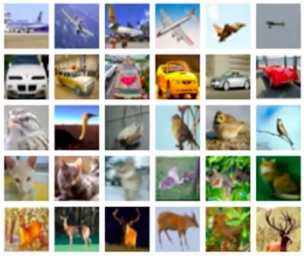
| 25 | 50 | Day 2.

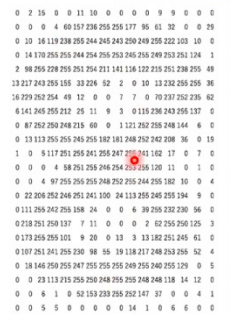
| 50 | Day 3

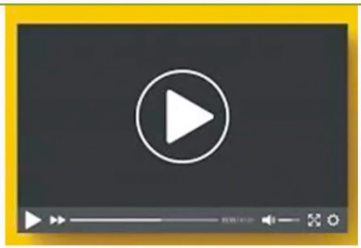
Classification of Data

3) Image data

- Data in the form of images or videos.
- Image is a matrix of pixels.







Videos

(Pixel Intensity:
0 – Black
255 – White)

Classification of Data (based on label)

- **Labeled Data**

Example: Haber Process $2N_2(g) + 3H_2(g) \leftrightarrow 2NH_3(g)$

3 Classes or Labels: Low Medium High

Inputs

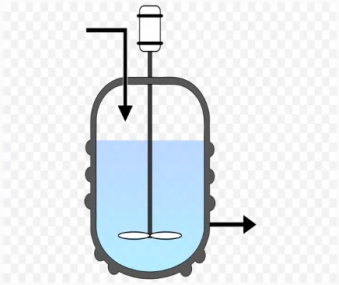
Numerical (continuous) Output

Categorical (discrete) Output

Temperature	Pressure	Ammonia Yield	Operational Risk
T_1	P_1	A_1	High
T_2	P_2	A_2	Low
T_3	P_3	A_3	High
...	...	...	...
T_{200}	P_{200}	A_{200}	Medium

Classification of Data (based on label)

- **Unlabeled Data:** A group of samples (or inputs) that have only the input data (also called as features) and no labels or outputs.



Height	Diameter	Temperature	Concentration
15	20	303	0.1

Classification of Data (based on label)

Unlabelled Data

Dimension 1	Dimension 2	Dimension 3
X_1	Y_1	Z_1
X_2	Y_2	Z_2
X_3	Y_3	Z_3
X_4	Y_4	Z_4
X_5	Y_5	Z_5

3 – Dimensional Inputs – Given data

Corresponding outputs/labels – NOT KNOWN

Label/Output
Class 1
Class 2
Class 2
Class 2
Class 1

Categorical/Numerical Outputs: Labels

↓

Not Known

Types of ML

- Supervised → labelled data
- Unsupervised → Unlabelled data
- Reinforcement

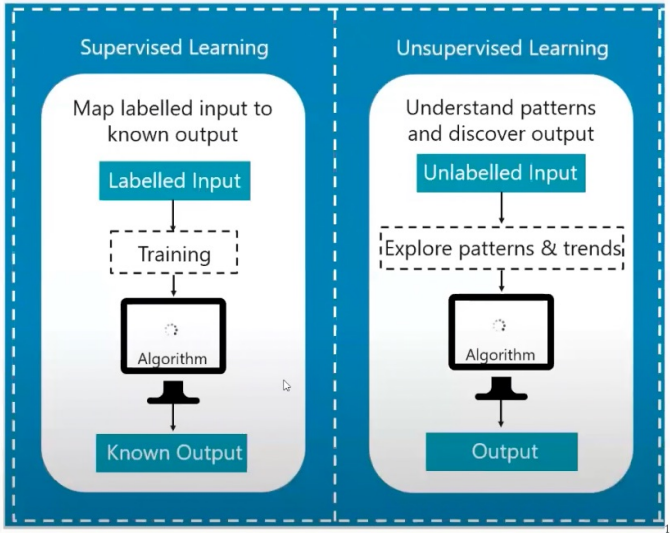
Types of Machine Learning

Supervised learning is a method in which we teach the machine using labelled data

In unsupervised learning the machine is trained on unlabelled data without any guidance

In Reinforcement learning an agent interacts with its environment by producing actions & discovers errors or rewards

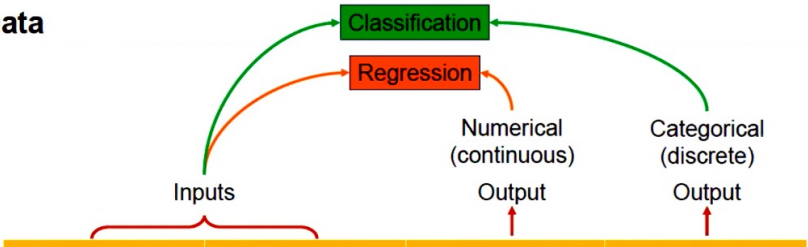
Approach in Supervised & Unsupervised Machine Learning



11

Supervised Learning

❖Data



Temperature	Pressure	Ammonia Yield	Operational Risk
T_1	P_1	A_1	High
T_2	P_2	A_2	Low
T_3	P_3	A_3	High
...	...	...	...
T_{200}	P_{200}	A_{200}	Medium

12

Supervised Learning: Multi-class Classification

	setosa	versicolor	virginica
S.L	[4.3,5.8]	[4.9,7]	[4.9,7.9]
S.W	[2.3,4.4]	[2,3.4]	[2.2,3.8]
P.L	[1,1.9]	[3,5.1]	[4.5,6.9]
P.W	[0.1,0.6]	[1,1.8]	[1.4,2.5]

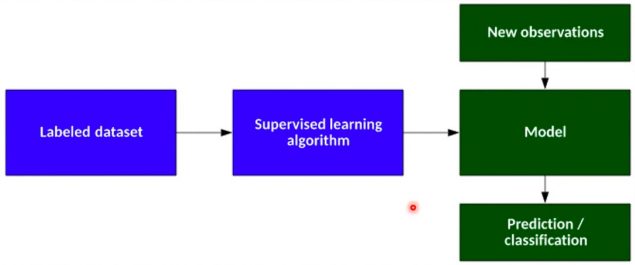
Output Variable – Categorical, Can take any number of values.
Input Variables – Can be any in number, Can be continuous or categorical

14

More eg

- Distinguishing b/w diff products in retail store shelf.
- classifying diff music pieces by Genre.

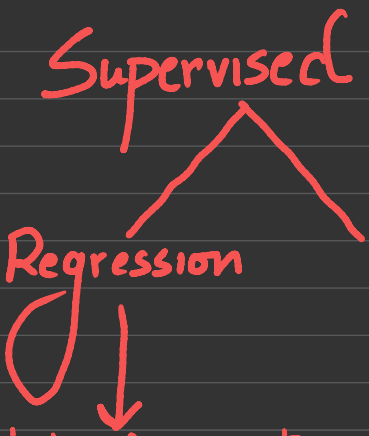
Supervised Learning



16

Few other applications of SL

- Face Detection
- Signature recognition
- Customer discovery
- Spam detection
- Weather forecasting
- Predicting housing prices based on the prevailing market price
- Stock price predictions



It is used to forcast cont. variables such as weather, market trends and so on.

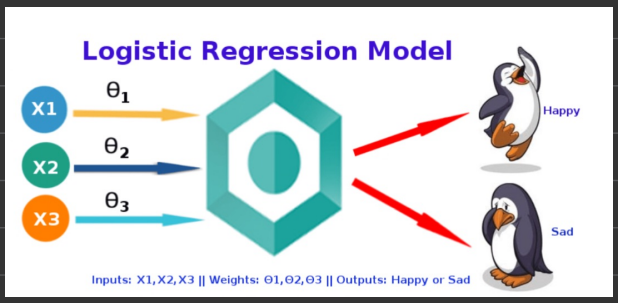
when the output variable is categorical T-F, 1-0, and so on.

Logistic Regression

→ It is a ML method used to solve classification issues.

→ It is a predictive analytic technique that is based upon probability idea!

(It is used to predict likelihood of categorical dependent variable.



Q. Why the name Logistic Regression ??

It is called so, since the technique behind it is quite similar to Linear Regression.

The name 'Logistic' comes from the Logit fxn, which is utilized in this categorization approach.

Q Why we can't use Linear Regression instead of Logistic Regression ??

Lin. Reg ⟷ (siblings) Log. Reg

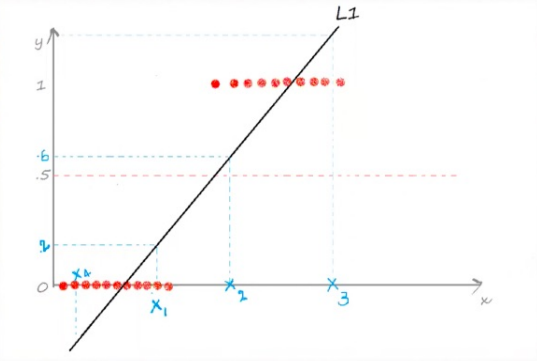
Maths

$y = mx + c$ → intercept-Y

- y → predicted values
- m → slope
- x → input data.

Linear Regression vs Logistic Regression

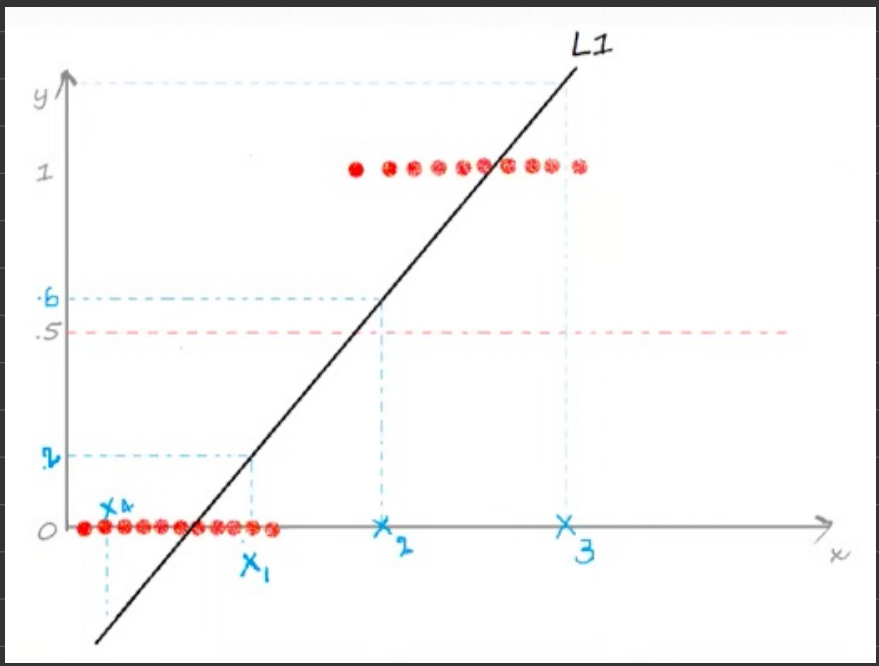
- Let us try if we can use linear regression to solve a binary class classification problem.
- Assume we have a dataset that is linearly separable and has the output that is categorical - two classes (0, 1).



- We define a threshold T = 0.5, above which the output belongs to class 1 and class 0 otherwise.

$$y = mx + c, \text{ Threshold } T = 0.5$$

$$y = \begin{cases} 1, & mx+c \geq 0.5 \\ 0, & mx+c < 0.5 \end{cases}$$

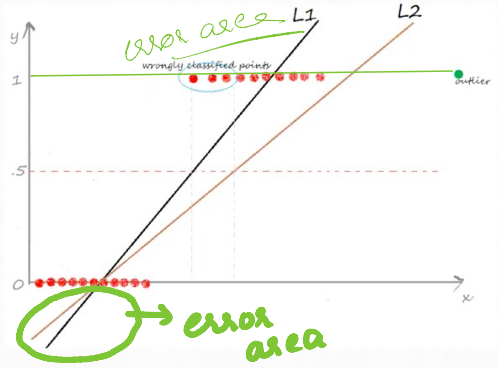


Case 1: $x_1 \approx 0.2 \ll$ threshold (0.5)
x_1 belongs to class 0

Case 2: $x_2 \approx 0.6 > 0.5$
x_2 belongs to class 1

Problems with Linear Regression for Classification

- Now, introduce an outlier and see what happens.
- The regression line gets deviated (L2) to keep the distance of all the data points to the line to be minimal → wrongly classified points → increase in error term.



The two limitations of using a linear regression model for classification problems are:

- **the predicted value may exceed the range (0,1),** (bcz we want only prob. values)
- **error rate increases if the data has outliers.**

Need for Logistic regression

Logistic Regression

- The logistic regression equation is quite similar to the linear regression model.
- Consider we have a model with one predictor (or input) "x" and one Bernoulli response variable (or output) "ŷ" and p is the probability of ŷ. The linear equation can be written as:

$p = b_0 + b_1 x$ --------> eq 1

- The right-hand side of the equation (b_0+b_1x) is a linear equation and can hold values that exceed the range (0,1). But we know probability will always be in the range of (0,1).

To overcome that, we predict odds instead of probability.

- **Odds:** The ratio of the probability of an event occurring to the probability of an event not occurring.
- Odds = p/(1-p)

23

eqn 1 can be rewritten as

$$\frac{p}{1-p} = b_0 + b_1 x \longrightarrow \text{eq 2.}$$

Odds can only be a (+ve) value, to tackle the -ve value, we predict the logarithm of odds

$$\ln\left[\frac{p}{1-p}\right] = b_0 + b_1 x \longrightarrow \text{eq 3}$$

(Note: if we do log both side it will become a linear reg eqn only)

- we need to bring p at one side and x on other side.

$$\exp\left(\ln\left[\frac{p}{1-p}\right]\right) = \exp(b_0 + b_1 x)$$

$$\frac{p}{1-p} = e^{(b_0 + b_1 x)}$$

$$p = (1-p) * e^{(b_0 + b_1 x)}$$

$$p = e^{(b_0+b_1x)} - p * e^{(b_0+b_1x)}$$

$$p + p * e^{(b_0+b_1x)} = e^{(b_0+b_1x)}$$

$$p(1 + e^{(b_0+b_1x)}) = e^{(b_0+b_1x)}$$

$$p = \frac{e^{(b_0+b_1x)}}{(1 + e^{(b_0+b_1x)})}$$

divide by $e^{(b_0+b_1x)}$

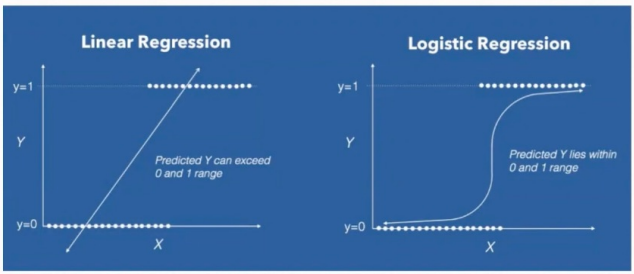
$$p = \frac{1}{1 + \frac{1}{e^{(b_0+b_1x)}}}$$

↳ This is the sigmoid fxn

Logistic Regression

- We started with a linear equation and ended up with a logistic regression model with the help of a sigmoid function.

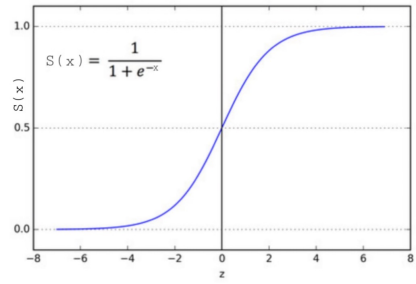
- Linear model: $\hat{y} = b_0 + b_1 x$ → Linear eqn
- Sigmoid function: $\sigma(z) = 1/(1+e^{-z})$
- Logistic regression model: $\hat{y} = \sigma(b_0+b_1x) = 1/(1+e^{-(b0+b1x)})$ → Non linear eqn



=> Sigmoid fxn

1. It always returns a prob. value b/w 0 and 1.
2. Used to convert expected values to probabilities
3. fxn converts any real number into a number b/w 0 and 1.
4. We utilize sigmoid to translate predictions to probabilities in ML.

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



Types

1. Binary → binary output (yes/No, 1/0)
2. Multinomial → 3 or more outcomes
3. Ordinal → 3 or more classes → with order like cust. rating in Supermart from 1 to 5

Similarly, the eqn for logistic model with 'n' predictors is as below

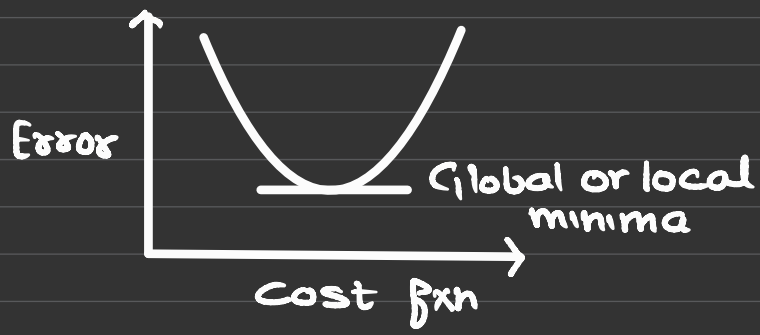
$$P = \frac{1}{1 + e^{-(b_0 + b_1x_1 + b_2x_2 + b_3x_3 \cdots b_nx_n}}$$

- In linear reg, we use the cost fxn as the MSE, which was the diff b/w y_predicted and y_actual

Cost Fxn

Q: Can we use linear reg cost fxn in logistic ??

- The graph of the cost fxn in linear reg is like this



$$J = \frac{\sum_{i=1}^{n} (\hat{y}_i - y_i)^2}{n}$$

Cost fxn

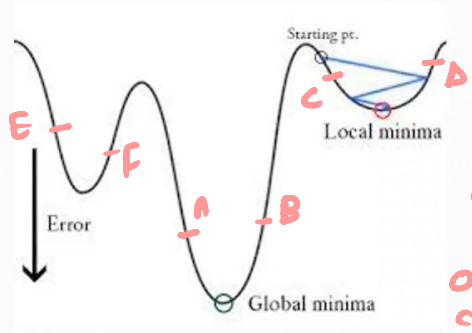
Note: When I consider error on the basis of one single point we consider it as cost fxn and we we consider all points it is loss fxn.

Local Minima → Minima with respect to its neighbours

Global Minima → W.r.t whole fxn

Cost function in Logistic Regression

- In logistic regression, the predicted output is a non-linear function ($\hat{Y}=1/1+e^{-z}$).
- If we use this in the above MSE equation then it will give a non-convex graph with many local minima as shown -



If we start with points C, D, E, F we will get optimization in local minima and we will not reach at best optimization that is global minima.
Only points such as A, B can give me best optimization

28

- The problem here is that the solution may highly get struck in local minima, and thus we may miss out on our global minima and the error will inc.

- To overcome this problem, a diff cost fxn is used, namely cross-entropy loss fxn.

Cost function in Logistic Regression

- The cross-entropy loss function is used to measure the performance of a classification model whose output is a probability value.
- Thus, in Logistic regression, the following loss function is used -

$$\text{Log loss} = \frac{1}{N} \sum_{i=1}^{N} -(y_i * \log(\hat{Y}_i) + (1 - y_i) * \log(1 - \hat{Y}_i))$$

This eqn can be used not only in logistic but other classification algo. as well.

29

which can be minimized using an optimization algo such as steepest descent etc to get the predicted values.

Summary

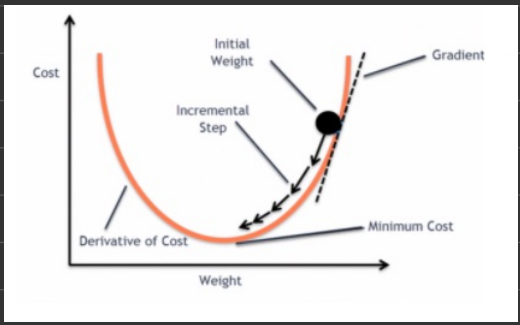
Selecting the right model is not enough. You need a function that measures the performance of a Machine Learning model for given data. Cost Function quantifies the error between predicted values and expected values.

`If you can't measure it, you can't improve it.`

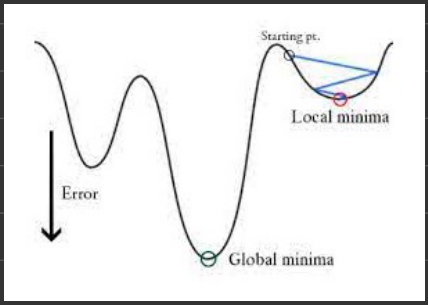
Another thing that will change with this transformation is Cost Function. In Linear Regression, we use `Mean Squared Error` for cost function given by:-

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

and when this error function is plotted with respect to weight parameters of the Linear Regression Model, it forms a convex curve which makes it eligible to apply Gradient Descent Optimization Algorithm to minimize the error by finding global minima and adjust weights.



In Logistic Regression Ŷi is a nonlinear function(Ŷ=1 /1+ e-z), if we put this in the above MSE equation it will give a non-convex function as shown:



When we try to optimize values using gradient descent it will create complications to find global minima.

The cost function used in Logistic Regression is Log Loss.

What is Log Loss?

Log Loss is the most important classification metric based on probabilities.

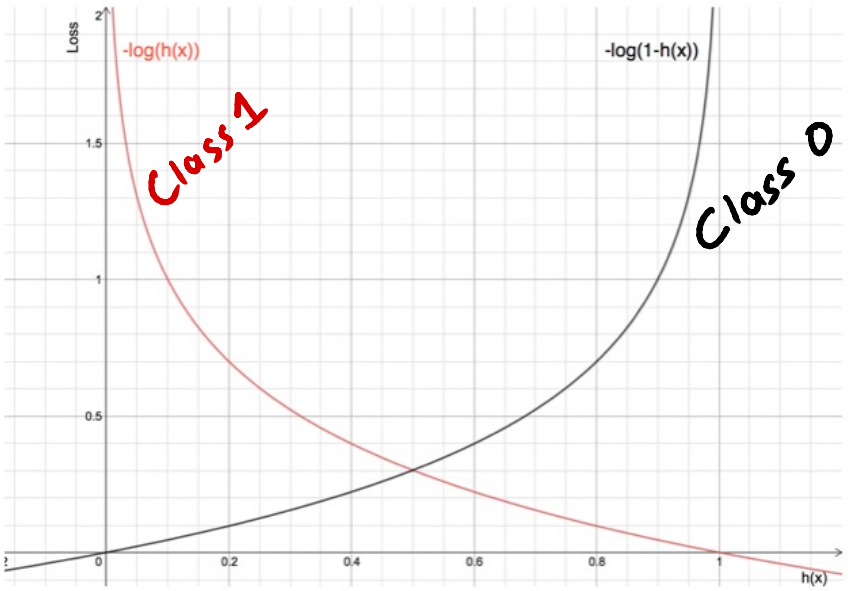
It's hard to interpret raw log-loss values, but log-loss is still a good metric for comparing models.

For any given problem, a lower log loss value means better predictions.

Mathematical interpretation:

Log Loss is the negative average of the log of corrected predicted probabilities for each instance.

The benefits of taking logarithm reveal themselves when you look at the cost function graphs for actual class 1 and 0 :



- The Red line represents 1 class. As we can see, when the predicted probability (x-axis) is close to 1, the loss is less and when the predicted probability is close to 0, loss approaches infinity.

- The Black line represents 0 class. As we can see, when the predicted probability (x-axis) is close to 0, the loss is less and when the predicted probability is close to 1, loss approaches infinity.

log loss function and cross entropy are same or not ???

They are essentially the same; usually, we use the term log loss for binary classification problems, and the more general cross-entropy (loss) for the general case of multi-class classification, but even this distinction is not consistent, and you'll often find the terms used interchangeably as synonyms