**Log Loss:**

This technique of performance measuring works on actual probability score, no technique we read so far works on actual prob-score.

So how **it works**,It’s formula is given in below image.

Let’s say for Binary Classification, we have dataset along with actual output for each data point, now our model gives the probability of getting 1 for each point(denoted as pi).

Now we apply log-loss on each data point output.

**How log loss formula work?**

See 1/n over the sum of all the values, is basically saying the average.

Now for binary classification we have two parts, **log(pi) \* yi** and **( 1-** **yi ) \* ( 1-log(pi) ) .**

* When actual output is 1, **log(pi) \* yi** will be non-zero and **( 1-** **yi ) \* ( 1-log(pi) )** will be zero, because of (1-yi) since yi is 1.

So It is calculating the log of probability of getting output as 1, when actual output is also 1.

* When actual output is 0, **( 1-** **yi ) \* ( 1-log(pi) )** will be non-zero and **log(pi) \* yi** will be zero, because of yi since yi is 0.

So It is calculating the log of probability of getting output as 0(since there are two classes only if prob of getting one class is p then prob of getting other class is 1-p), when actual output is also 0.

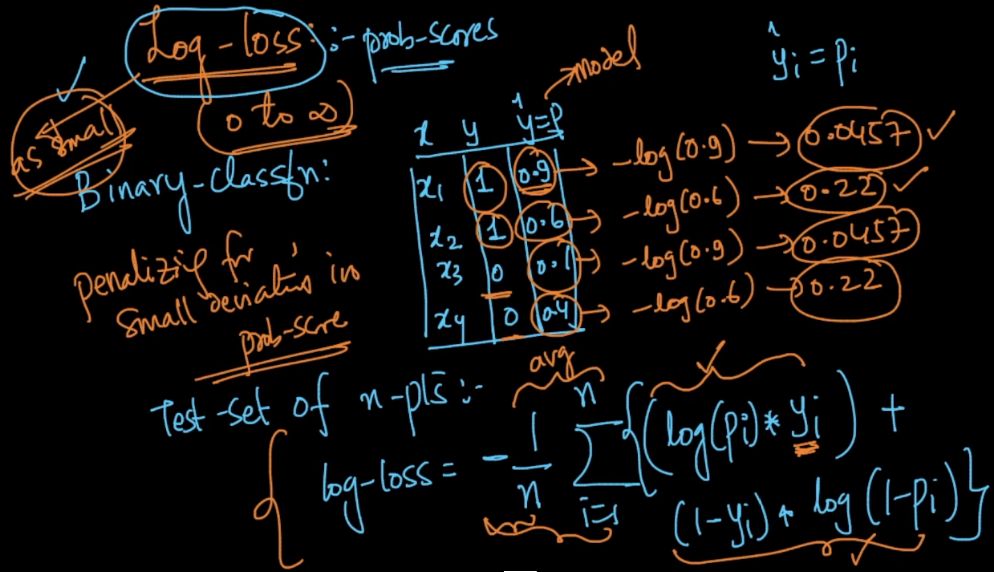
Now we know the lower the value is provided in negative log the higher will be it’s result and

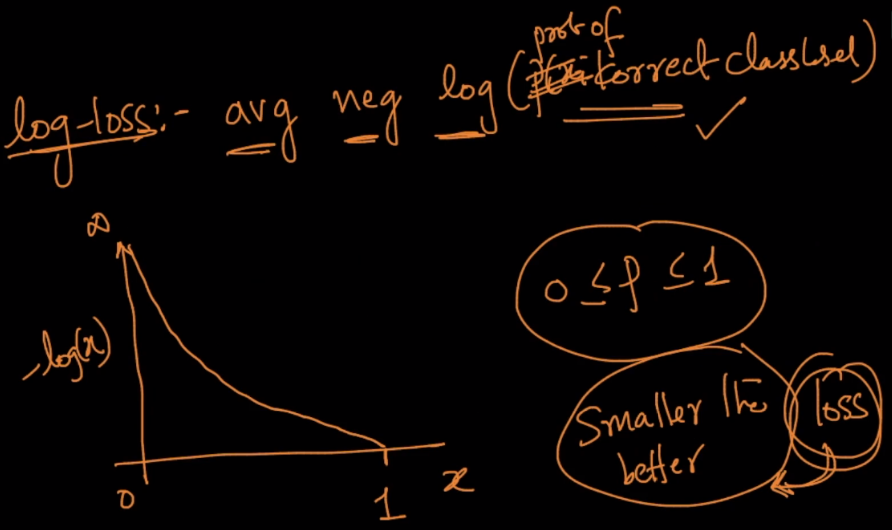
Higher the value is provided in log the lower the value will be it’s result.

Therefore if the probability of getting prediction as actual output is high, then log-log or error would be less.

And if the probability of getting prediction as actual output is low, then log-loss or error would be very high, that means for small less proabability log-loss penalize it by increasing the log-loss or error by high value.

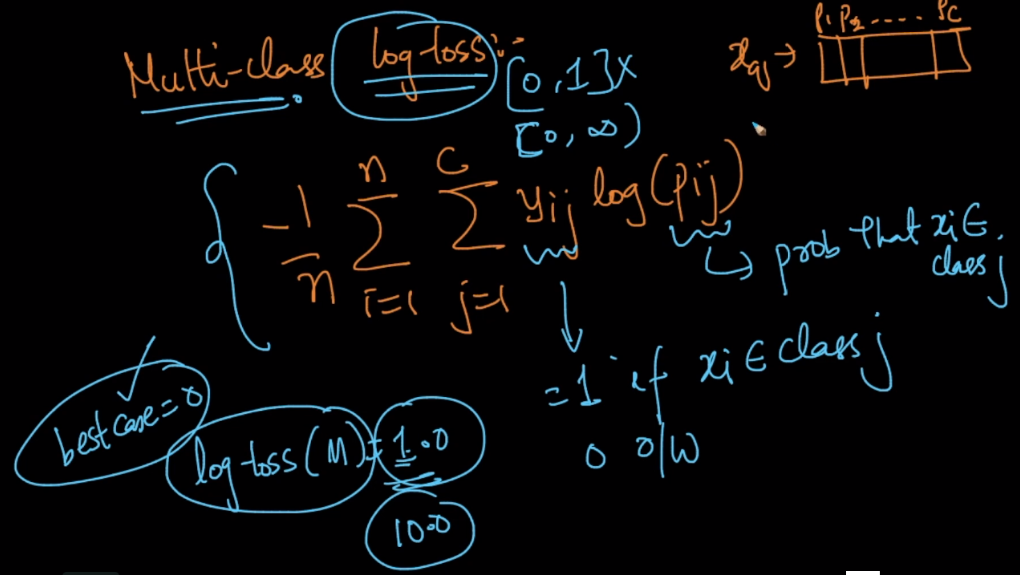
In English we can say that log loss is the **negative average of the log of the probability of correct result.**

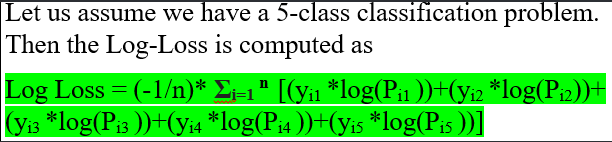




**Log Loss for Mutliclass:**

Multiclass log loss is just extension of binary log loss. In multiclass log loss or categorical cross entropy, we just put the value of yi=1 for that class only to which the datapoint belongs. For rest of all the classes it will be zero. So effectively for one data point we are looking at the log of probability of the data point belonging to the actual class.





**Comments:**

* Given any such metric like AUC of ROC or Log Loss how do we go back and tune our model (say for KNN) to give us a better model?

Divide your data into train, cv and test sets. For different values of k (in KNN), train the model on training dataset using k, and calculate the metric score on cv data. Choose the value of k, which gives the best metric score on cv data.

* Log loss also not work well for imbalanced data so what technique we can use for imbalanced data?

**Yes for the unbalanced data set we usually use AUC , F1 score  Jaccard index metrics**