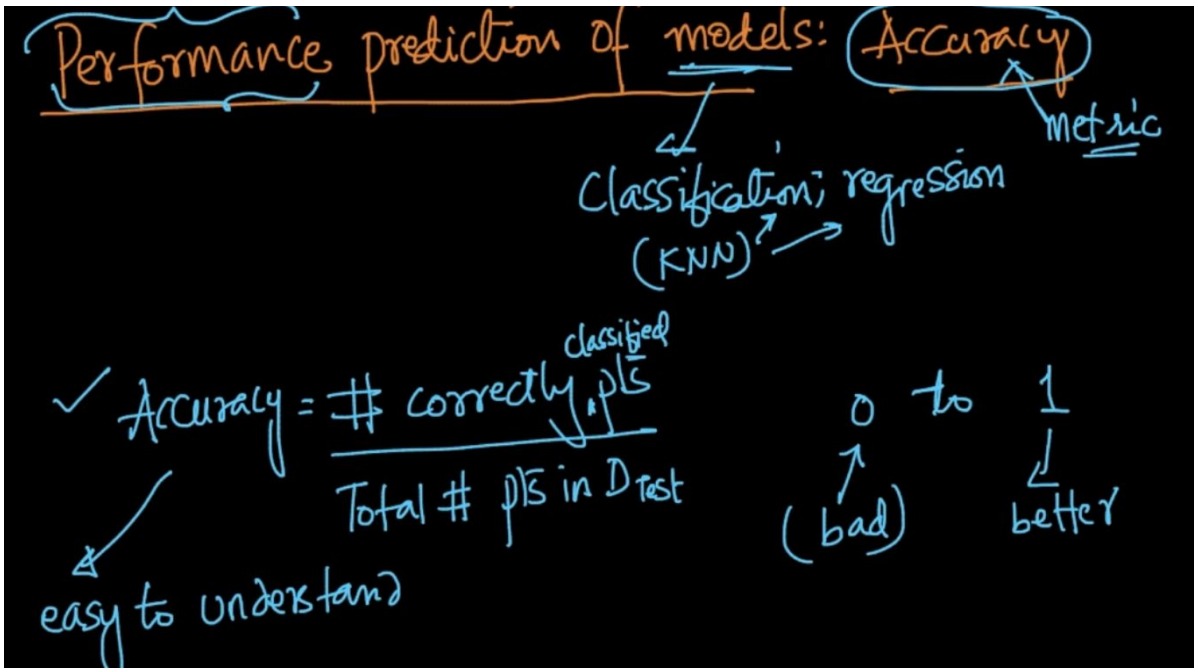
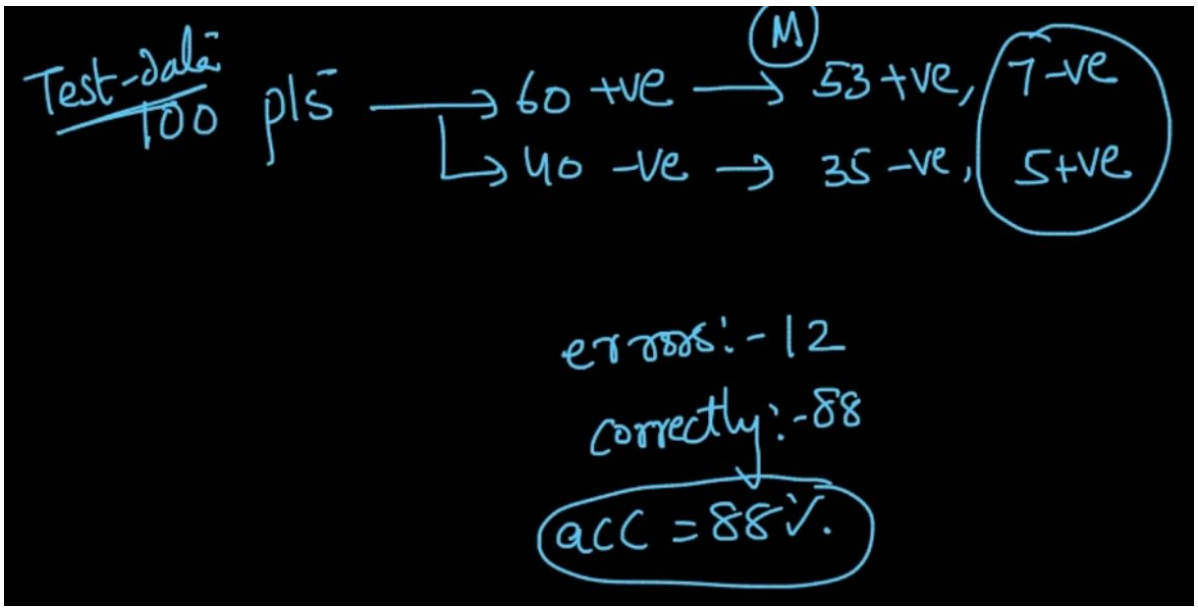
PERFORMANCE MEASUREMENT OF MODELS

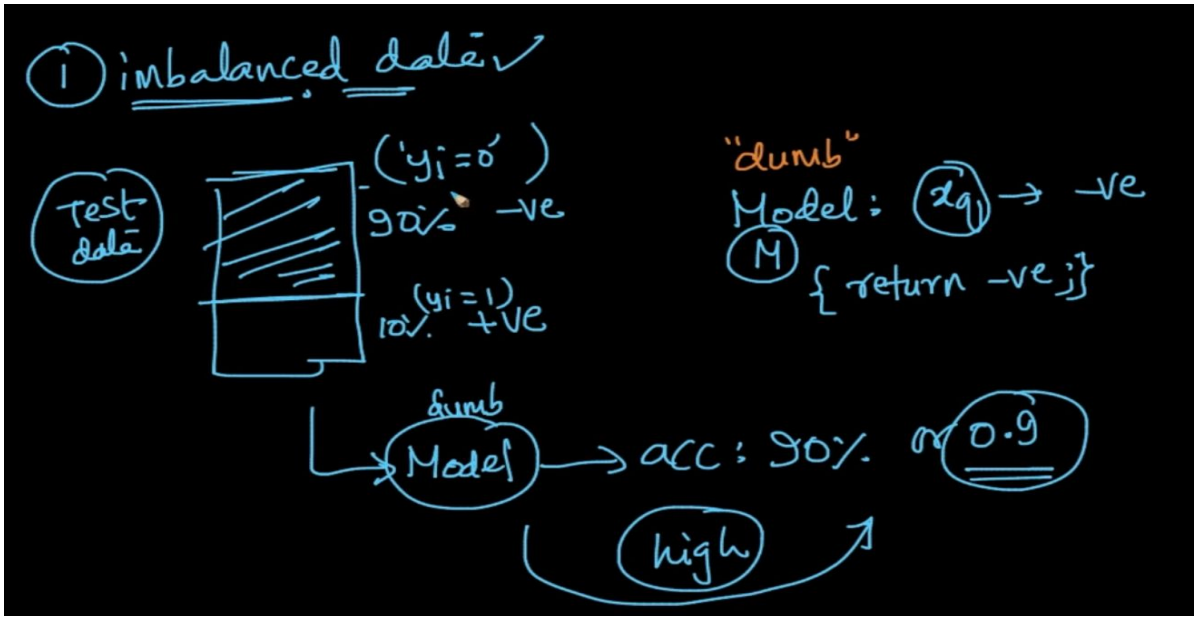
ACCURACY



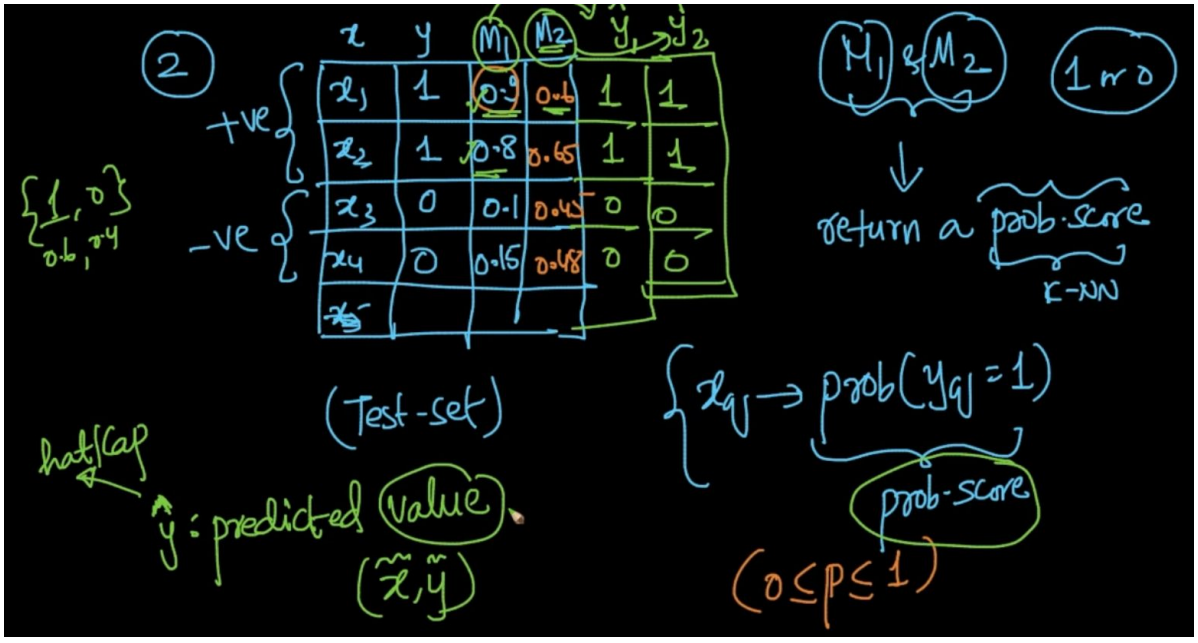
Accuracy calculation



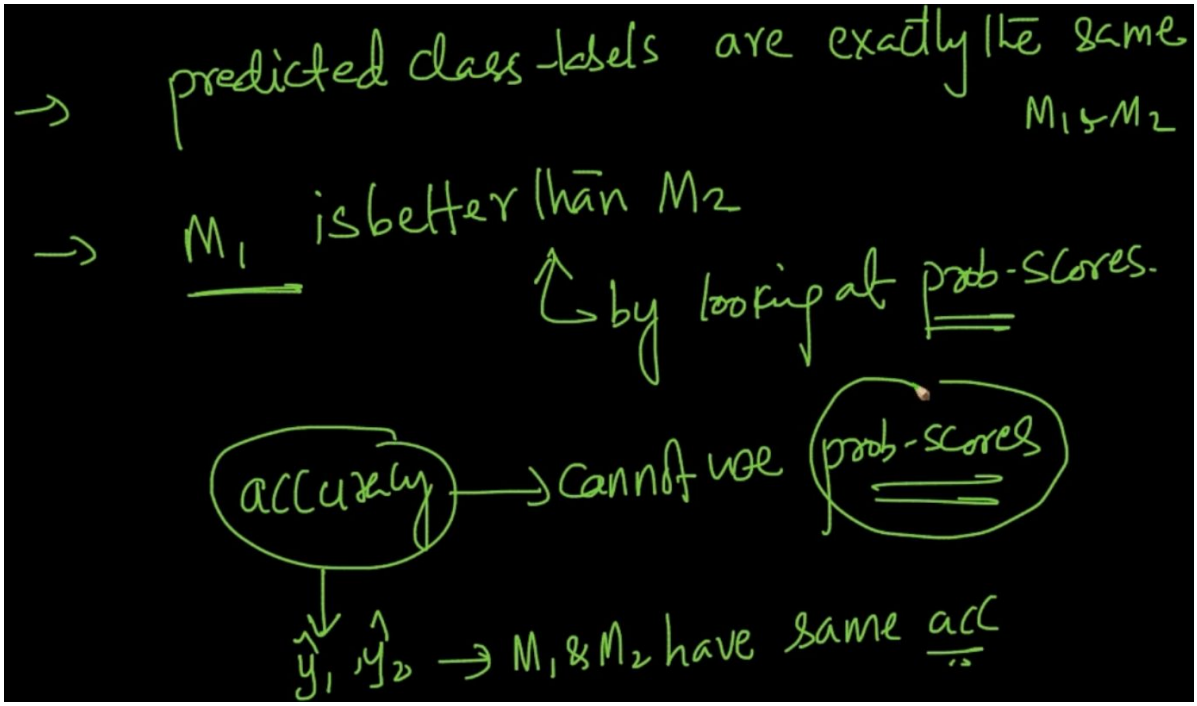
We always do an accuracy calculation on Test Data



Our model is dumb and for every x_q it gives -ve value and suppose if we've an imbalanced dataset where 90% data is -ve then dumb model will give 90% accuracy. So, accuracy is not a good measure on Imbalanced datasets

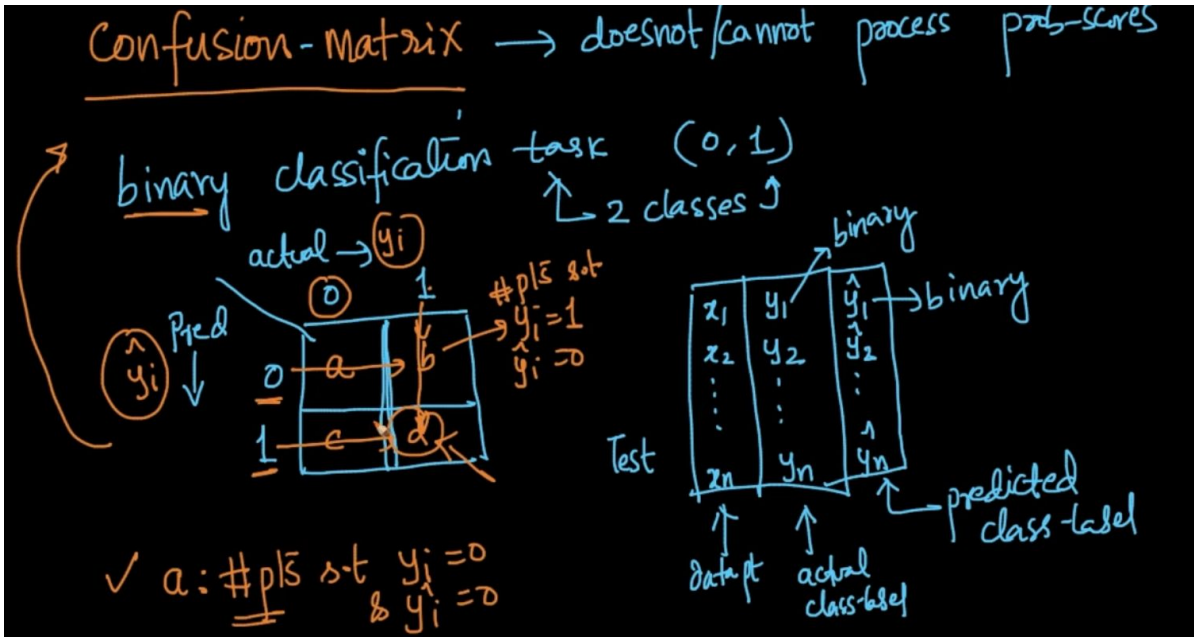


Our models M_1 & M_2 are giving probability scores prob($y_q = 1$) and their corresponding predicted values are $\hat{y}_1$ & $\hat{y}_2$. As we can see M_1 is performing better than M_2 but both are giving the same predicted values

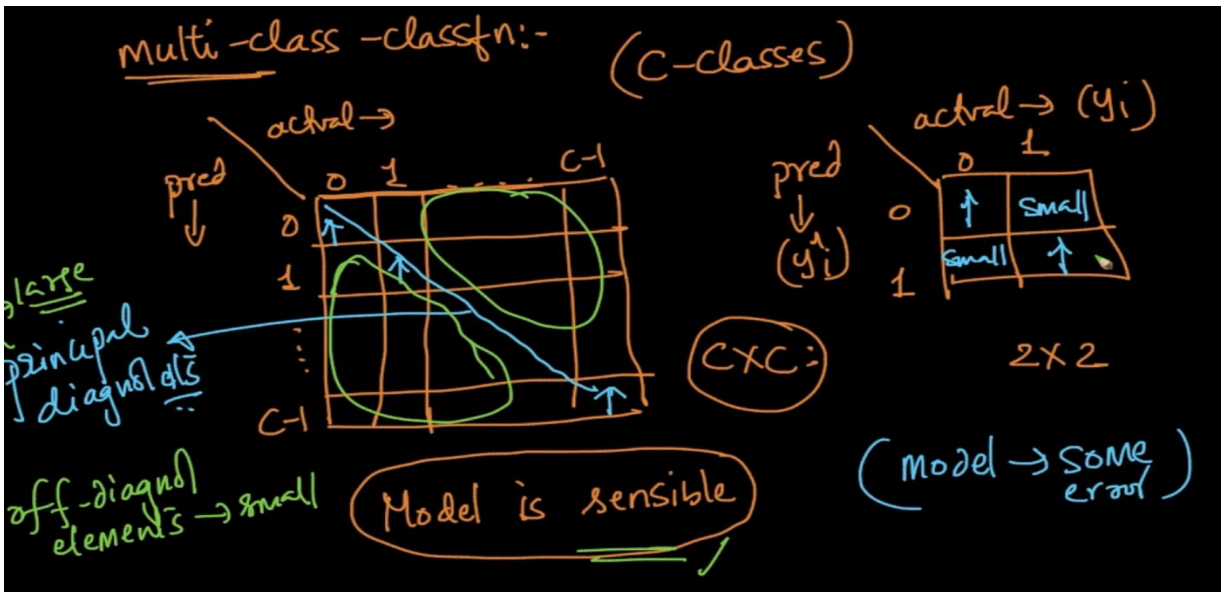


So accuracy isn’t a good measure for all the things

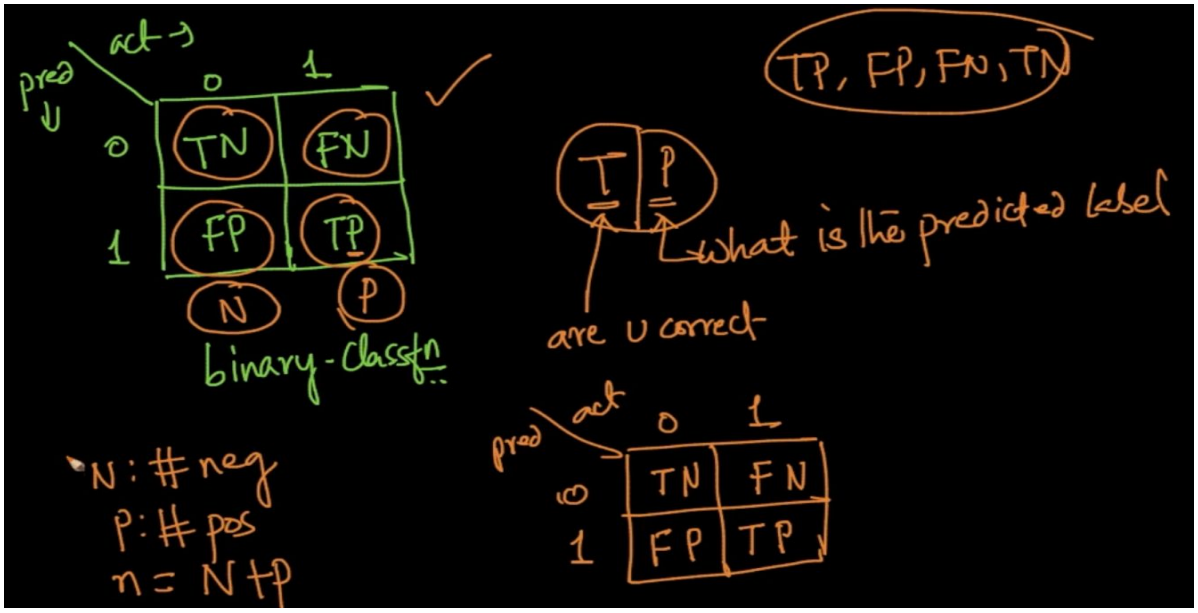
CONFUSION MATRIX



We've actual class label y_i and predicted class-label $\hat{y}_i$. Confusion matrix is table comprises of actual and predicted values. No. of points where the actual class label y = 0 and predicted class label $\hat{y}$ = 0 comes at the 'a' part (0,0) . 'b' is y = 1 & $\hat{y} = 0$



We want the model to work great so (0,0) and (1,1) should be high and (0,1) and (1,0) should be low basically the principal diagonal elements should be large



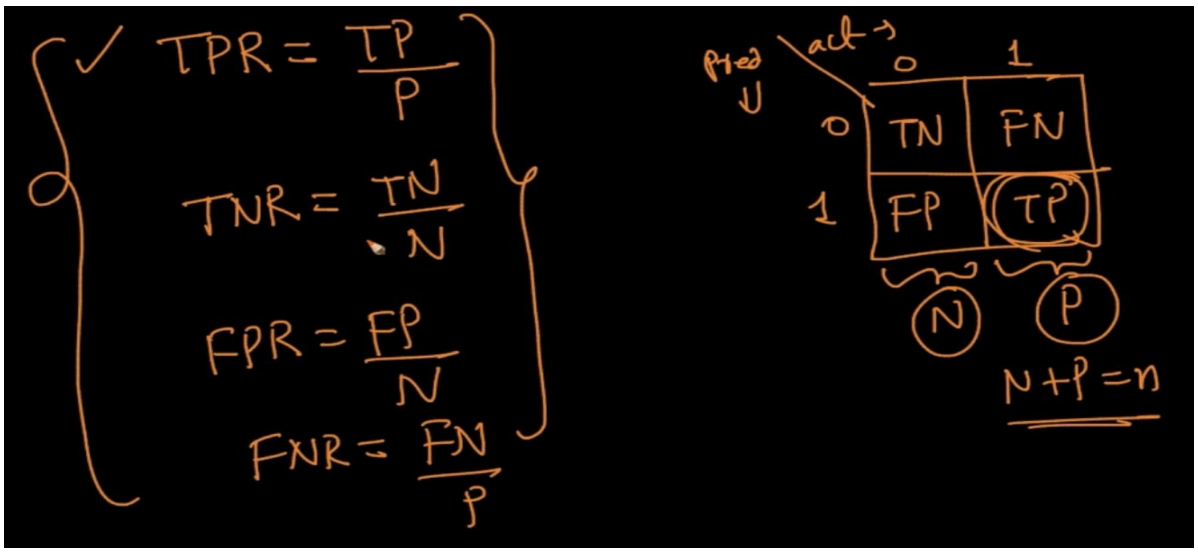
How to know which one is which in Confusion Matrix. We use the above method. For ex:

We want the place of TN. 0 is negative and 1 is positive. What are we predicting here ?

N i.e the second of TN and are we correct i.e T. N is 0 and T is that the predicted and actual class label values are matching.

n = N + P where ‘n’ - Total no. of points, N - Total negative points in actual Class label,

P - Total Positive points in actual class label



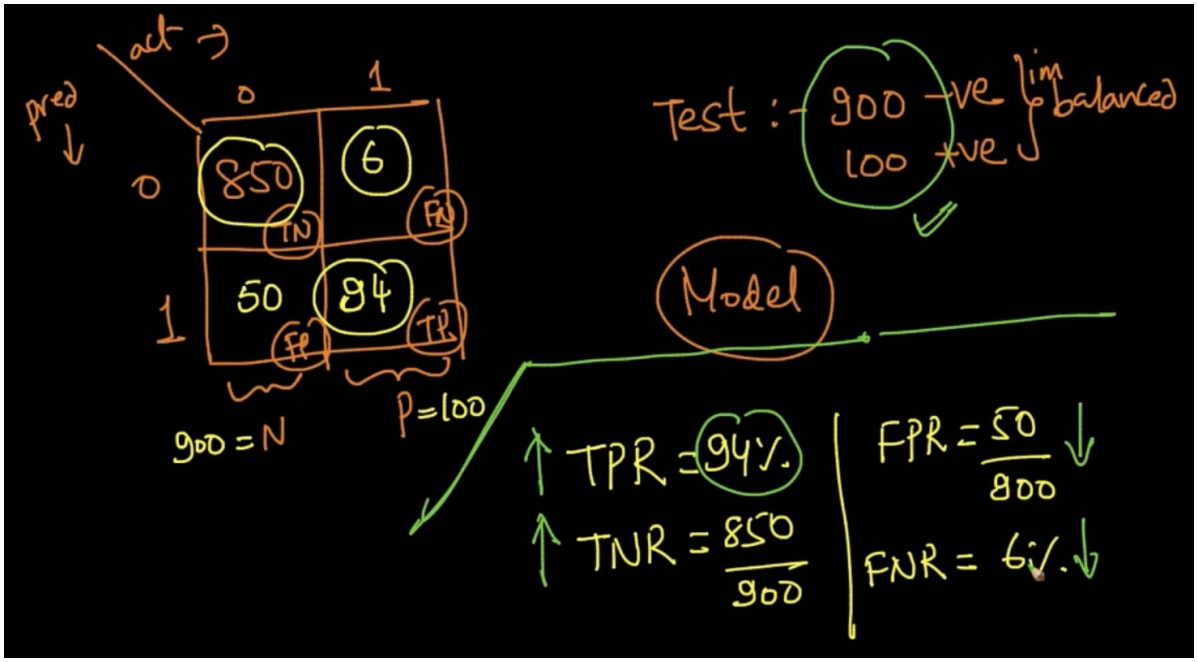
TPR is ratio of True Positive points/ Total positive points

TNR is ratio of True Negative points/ Total Negative points

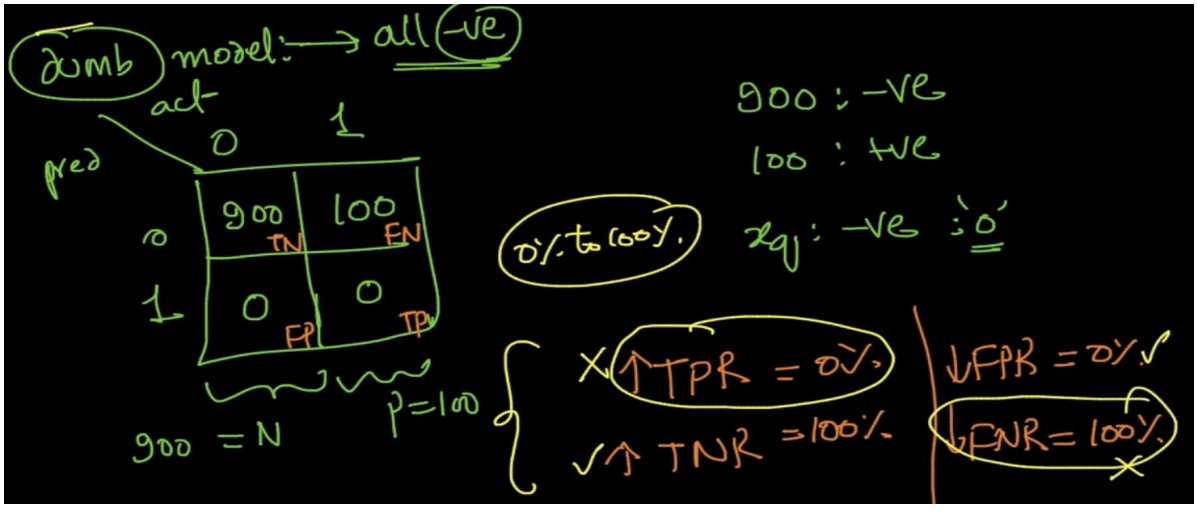
FPR is ratio of False Positive points/ Total positive points

FNR is ratio of False Negative points/ Total Negative points

Basically, draw the confusion matrix and then suppose we want to know FNR. It is FN from F and looking at the column of FN there is N which is total actual no. of negative class labels

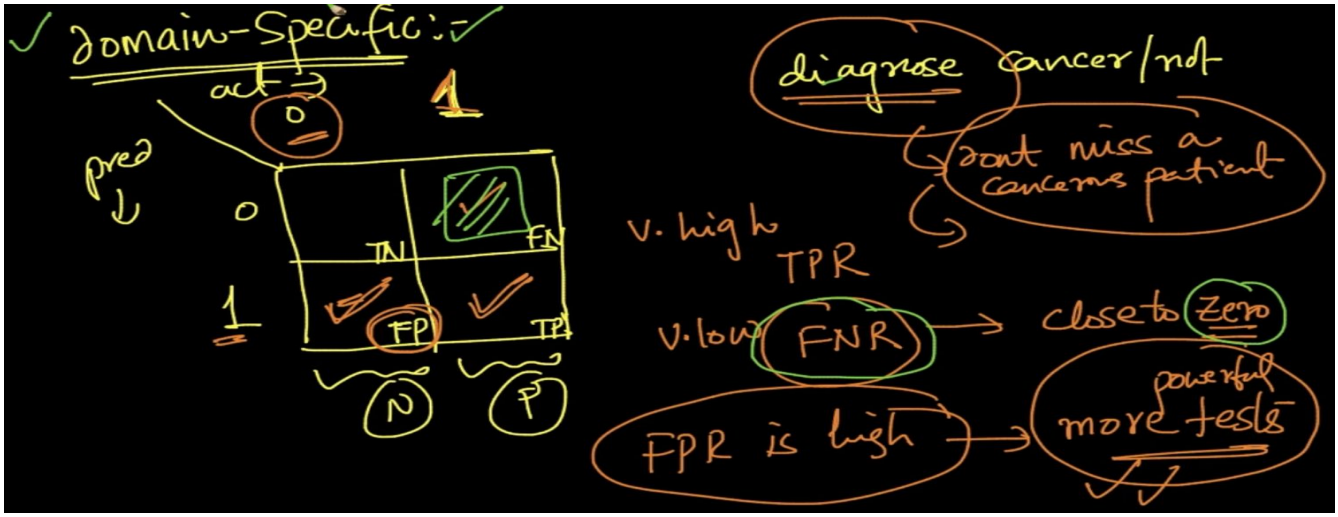


The model is a sensible/good model. Here, TPR and TNR are higher while FPR and FNR are low and that should be the case because we want our prediction to predict the actual values . So if TPR is 94% it means Our model predicted 94 positive class labels correctly



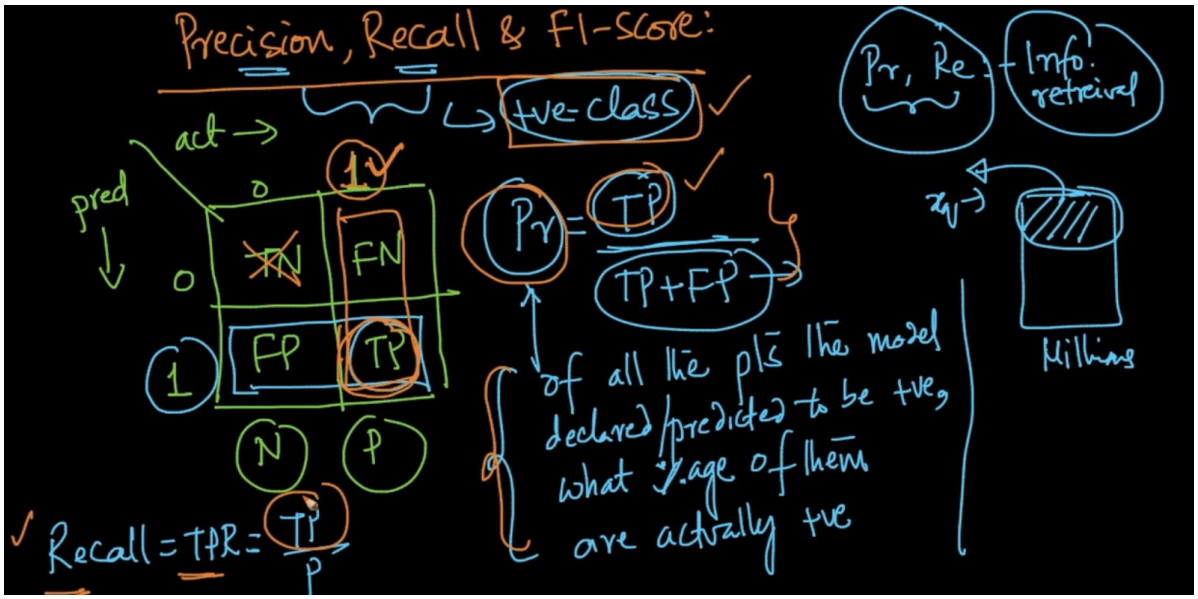
It is a dumb model which is predicting any query x_q as negative: 0. There are 900 negatives so it'll give TNR = 100% as it's predicting everything negative but TPR = 0 which should not be low but high instead for a proper ML model. So our TPR,TNR,FNR,FPR can give a good idea about imbalanced data set as well .

But which one of TPR,FPR,FNR,TNR is important?



The answer is very domain specific. E.g: Cancer diagnosis, here FNR should be very low because we don't wanna predict someone as cancer-free when he's actually diagnosed with cancer. TPR should be very high obviously, we want to diagnose correctly. FPR is high? No Problem if we predict a cancer-free dude diagnosed with cancer. So more powerful tests will be done which will prove that he's cancer-free.

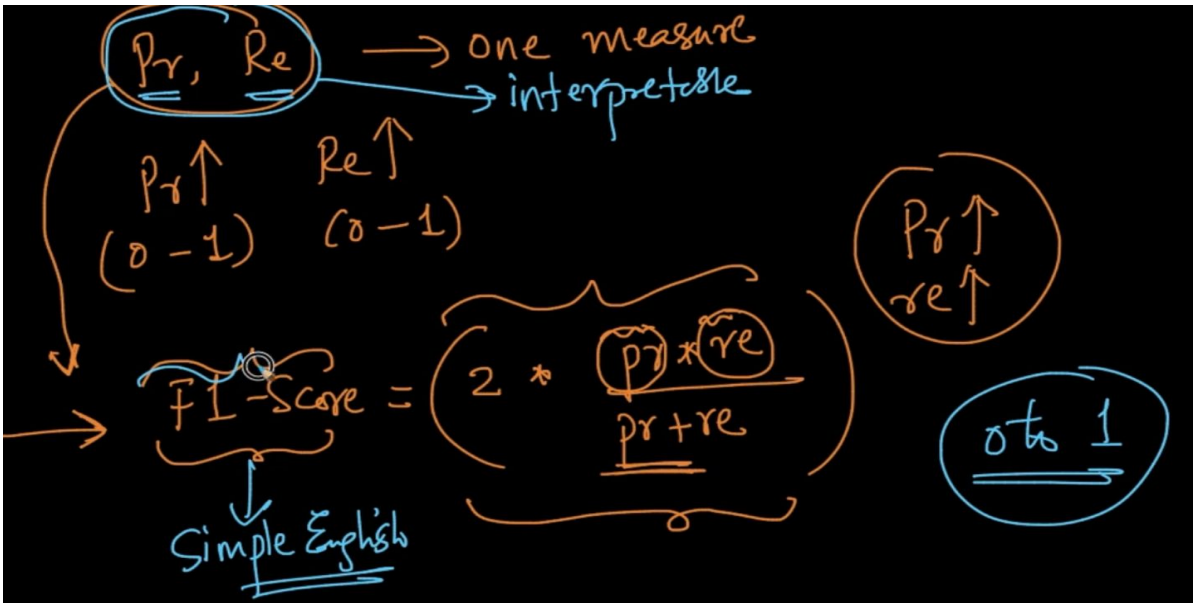
PRECISION, RECALL & F1-SCORE



Precision and Recall are used for positive class.

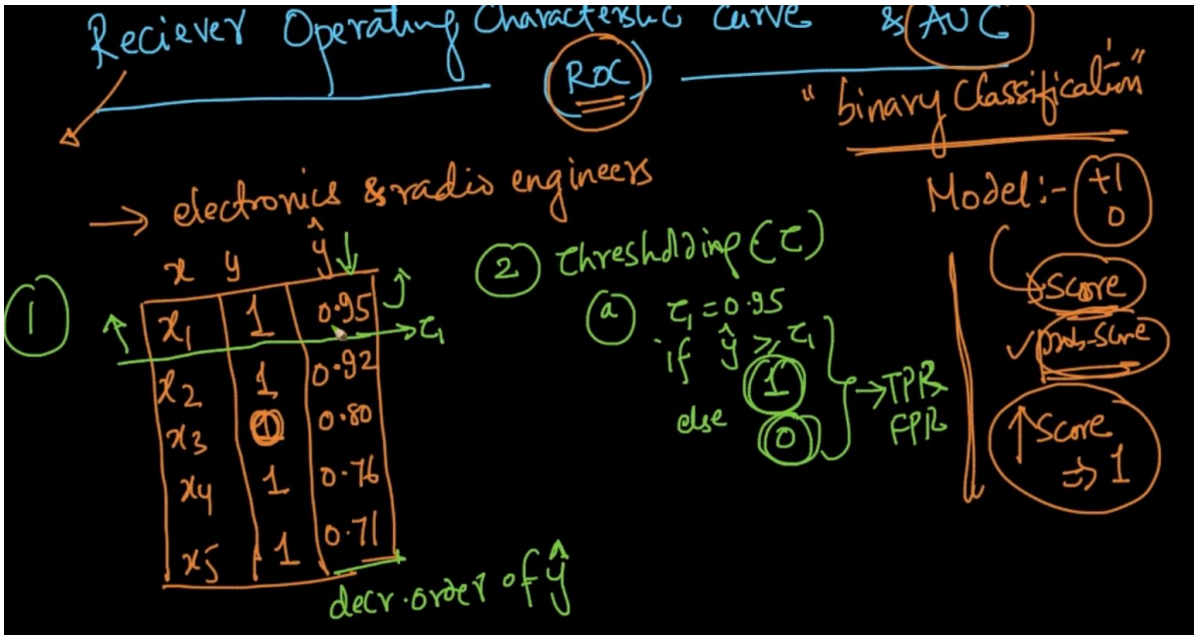
Precision: From all the predicted positive points, how many percentages are actually positive

Recall: From all the positive points in dataset what percentage are actually predicted to be positive



F1-Score = $avg[(inv-precision),\ (inv-recall)]^{-1}$ .It is harmonic mean of Pr AND Re

RECEIVER OPERATING CHARACTERISTIC (ROC) CURVE and AREA UNDER CURVE

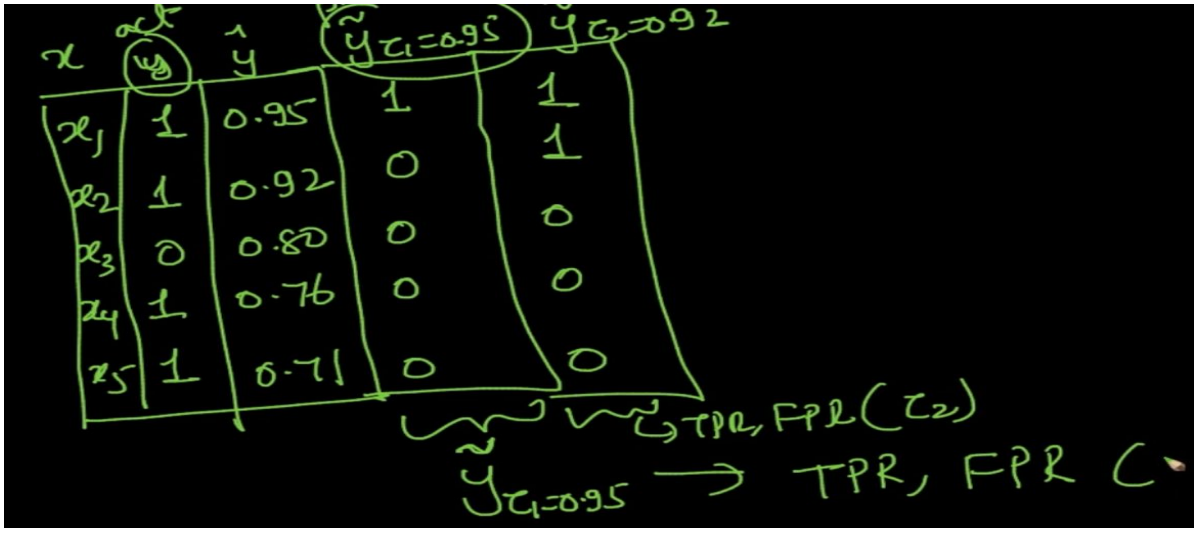


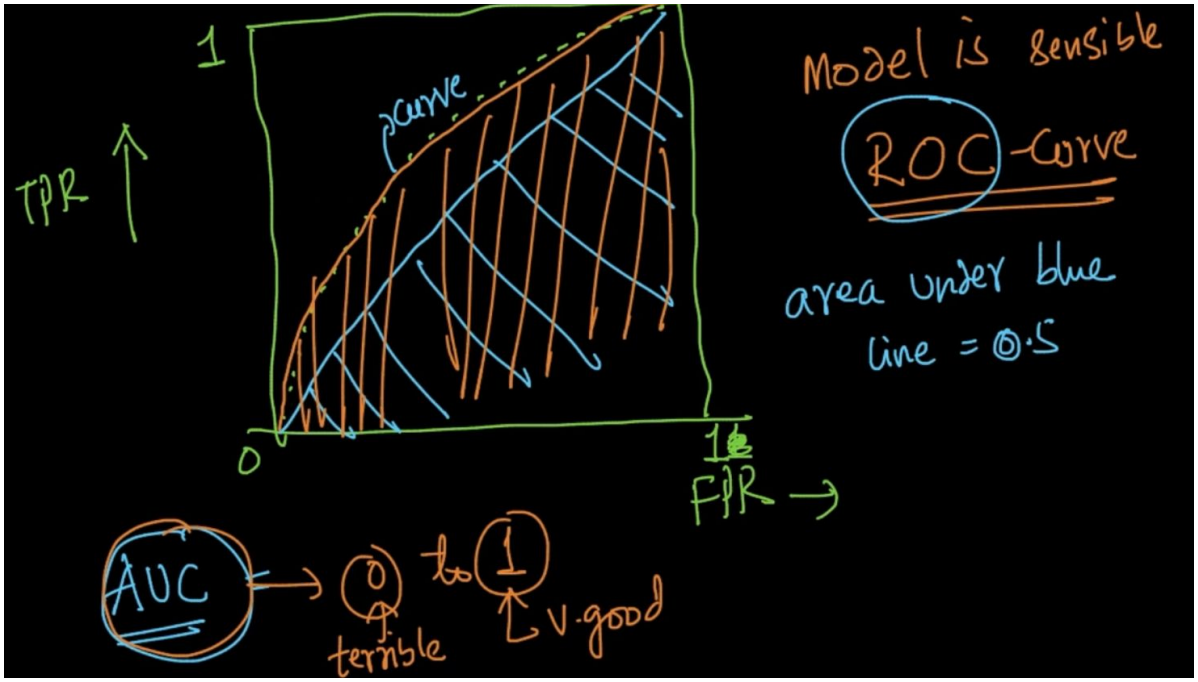
We are doing binary classification, the Model apart from values 0 or 1 is also giving us scores like prob-score and higher the score the more it can be 1

1) So our model gives predicted values $\hat{y}$ and we sort them on decreasing order

2) Thresholding (τ): We put a threshold value and if our predicted $\hat{y}$ is $\geq \tau$ then 1 else 0.

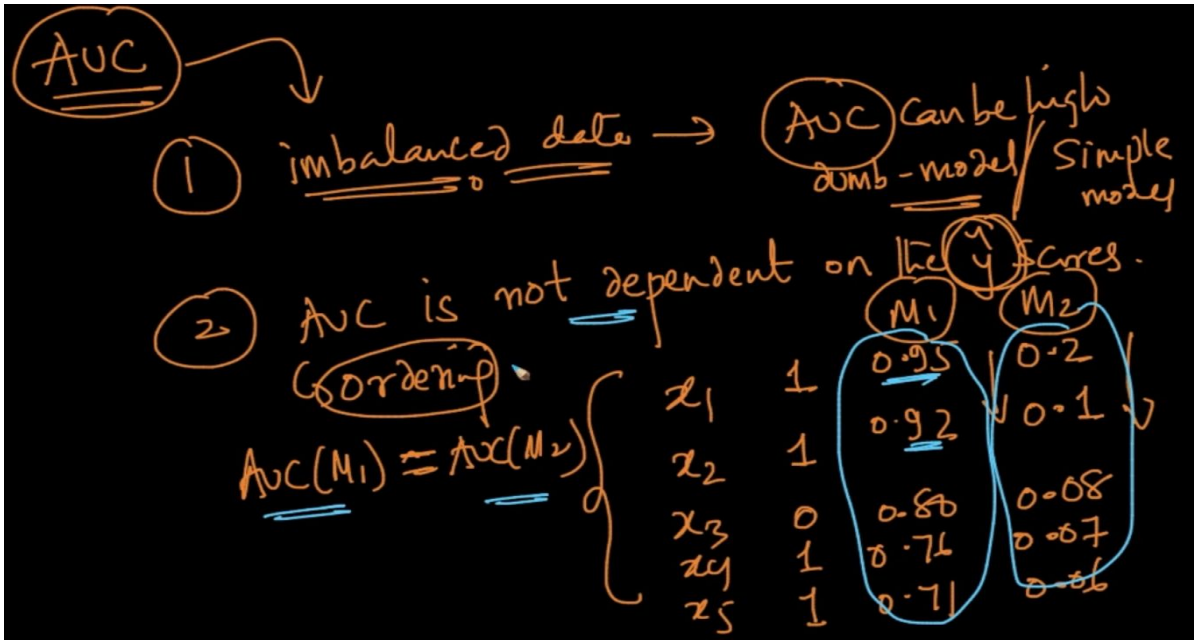
We put each $\hat{y}$ as threshold and get the predicted values as seenfor each threshold and we'll get the corresponding FPR and TPR





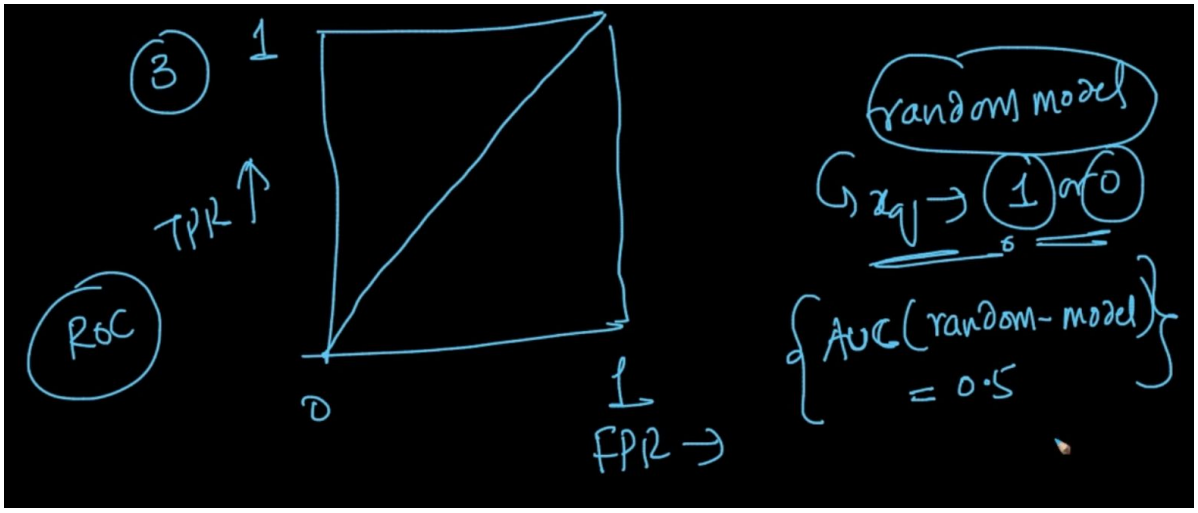
We get the TPR and FPR of all the threshold predicted values and then we plot them.

If our model is sensible then we'll get a curve like above which is also called ROC. In AUC, 0 is terrible and 1 is Excellent.

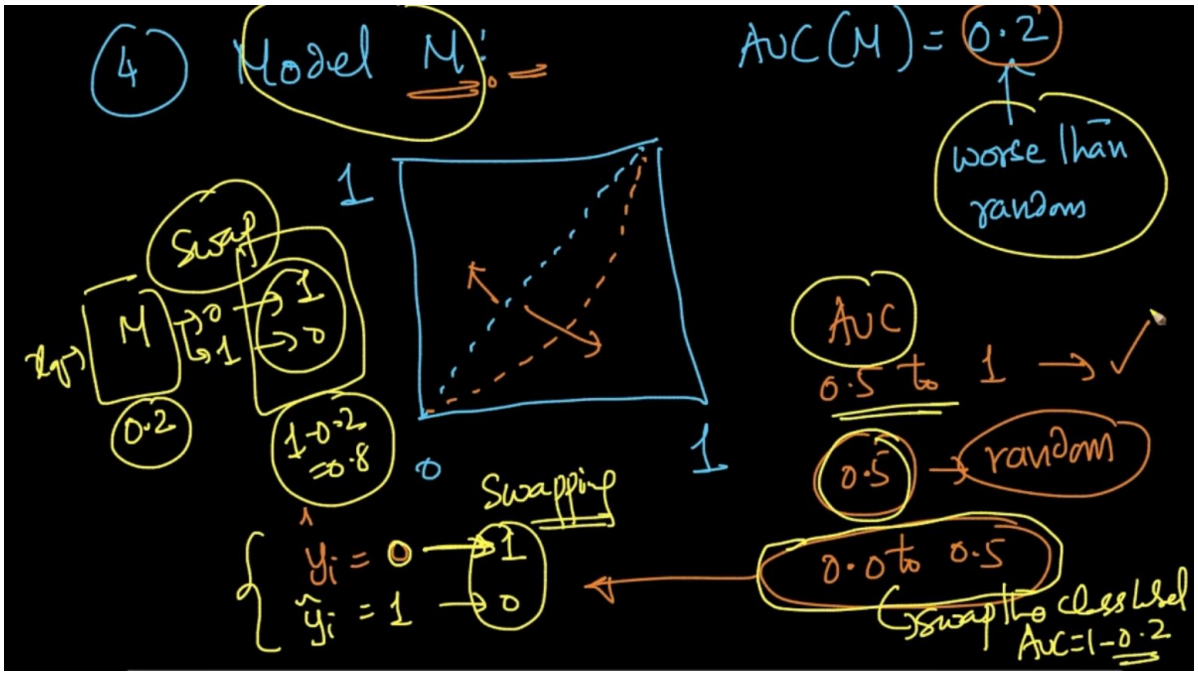


1) In imbalanced data -> AUC can be high with dumb model/simple model

2) AUC is not dependent on $\widehat{y}$ scores it is based on ordering as seen above

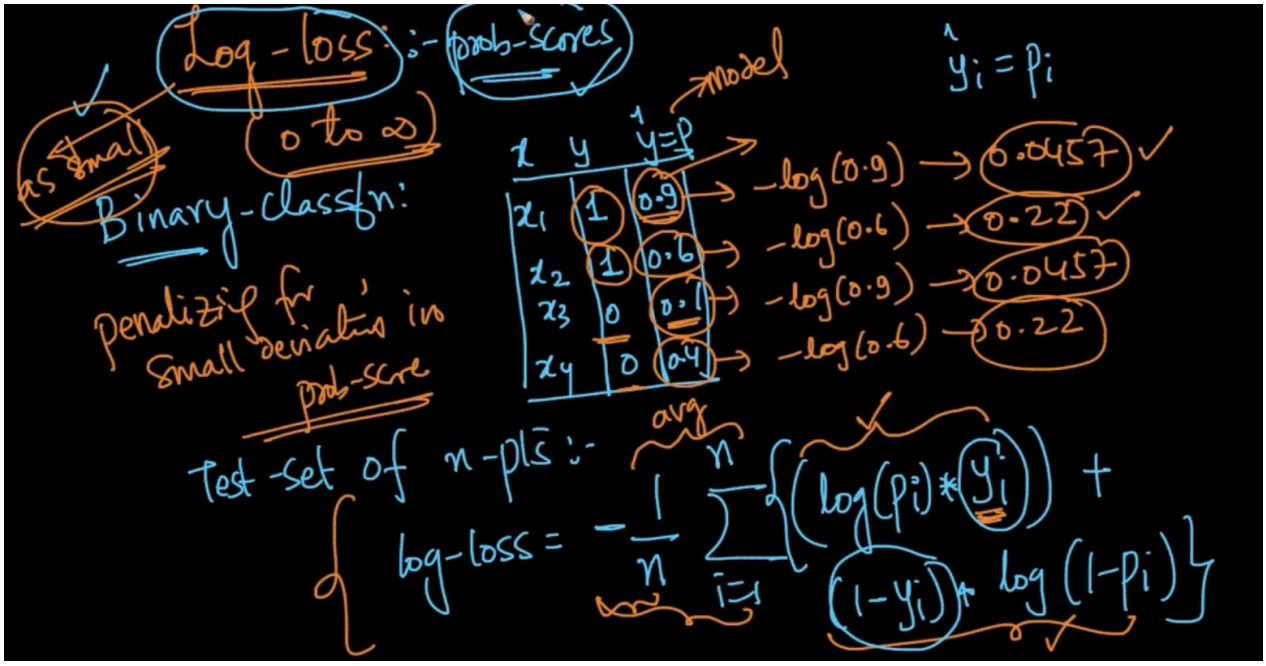


If a model is random i.e for a given x_q it'll give 0 or 1 . The AUC = 0.5



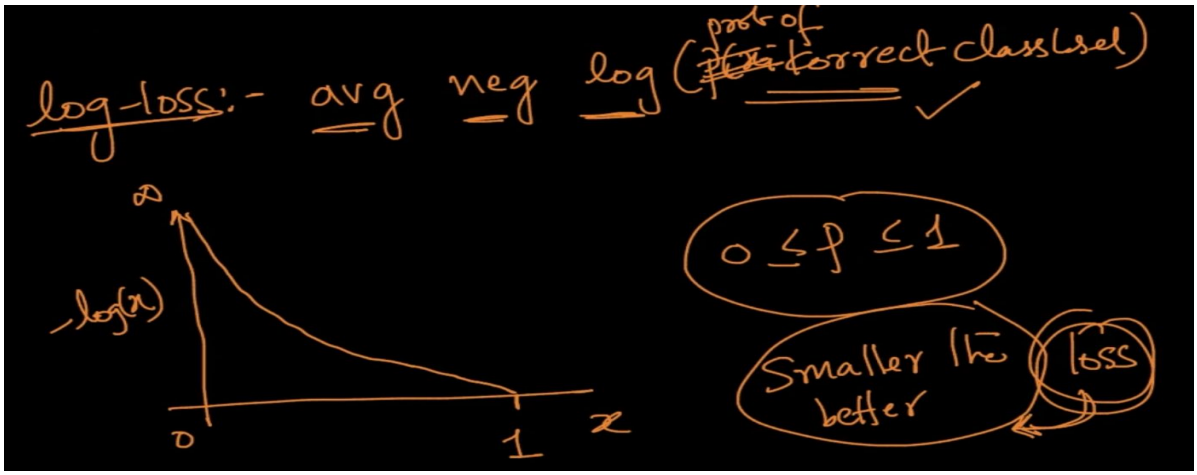
If our AUC is between 0.0 to 0.5 .The curve will like above which is not good as AUC should be high or between 0.5 - 1.0 . So, what we do is we swap the predicted class label values i.e 0 -> 1 and 1 -> 0

LOG - LOSS

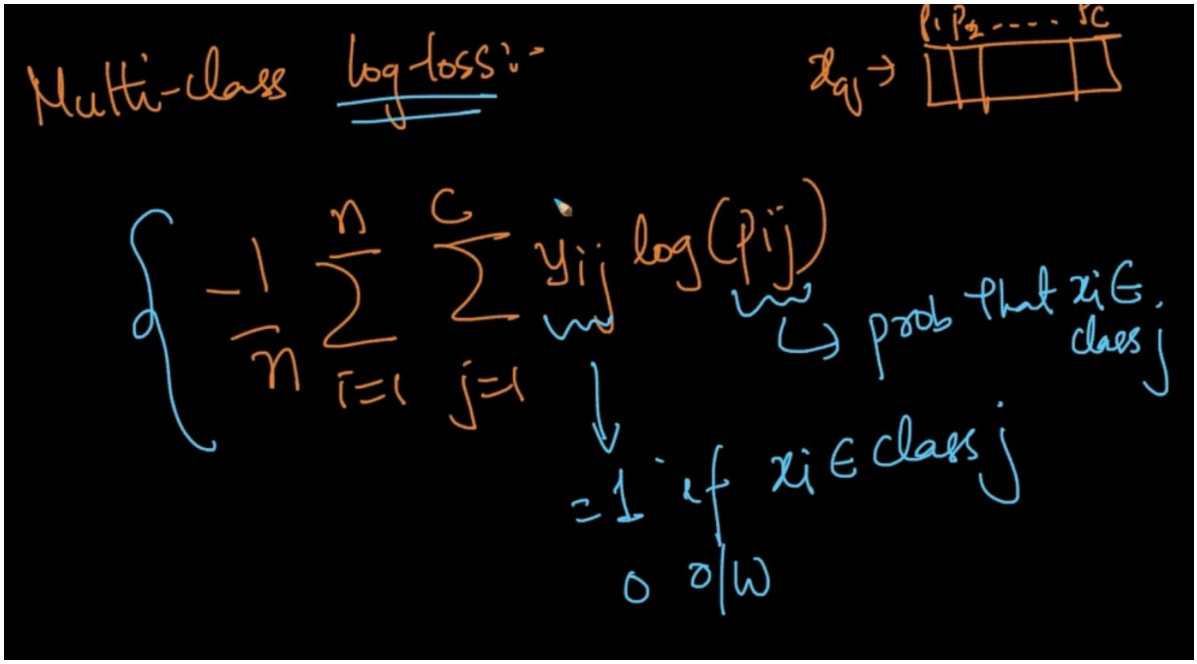


It works with probability scores $\hat{y}$. For x_1 log loss is low because prob-score is high and it's y = 1 as well but with x_2 it is higher as it's y = 1 but prob-score is low i.e 0.6 so it's penalizing it . Same analogy for x_3 but here y = 0 and if our prob-score is low then log loss will also be low

The formula is formulated such that it it'll try to give the correct log-loss for every class label as it's seen for x_1 and x_3

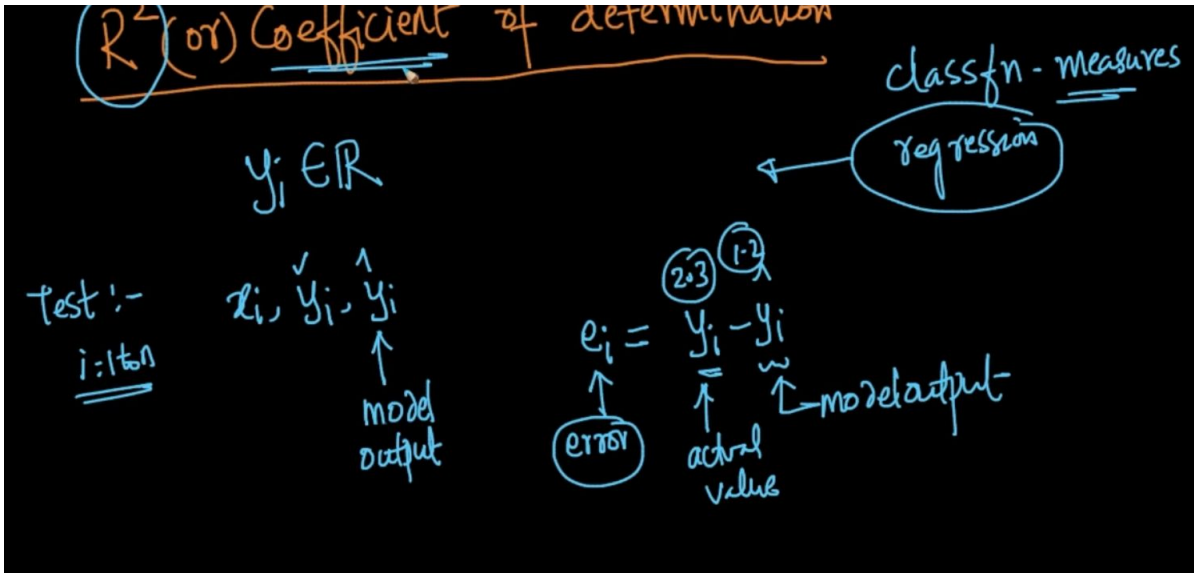


Log-loss -> Average negative log(probability of correct class label)

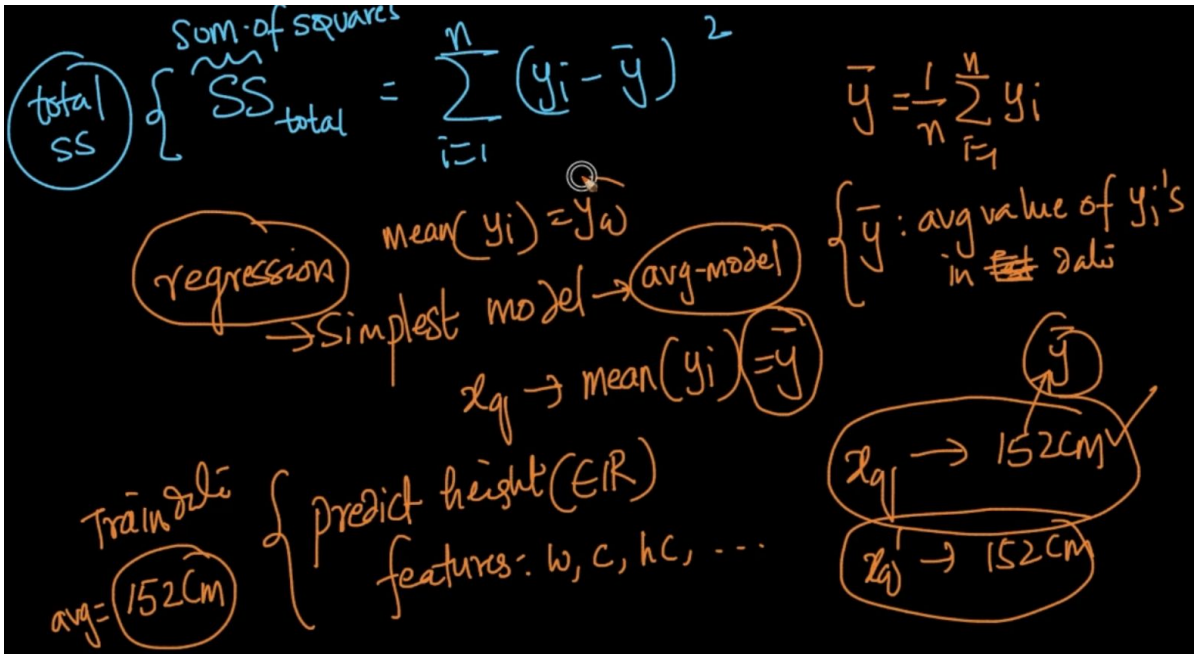


For multiclass , For a new query x_q model will give probability scores for all classes

R- SQUARED OR COEFFICIENT OF DETERMINATION

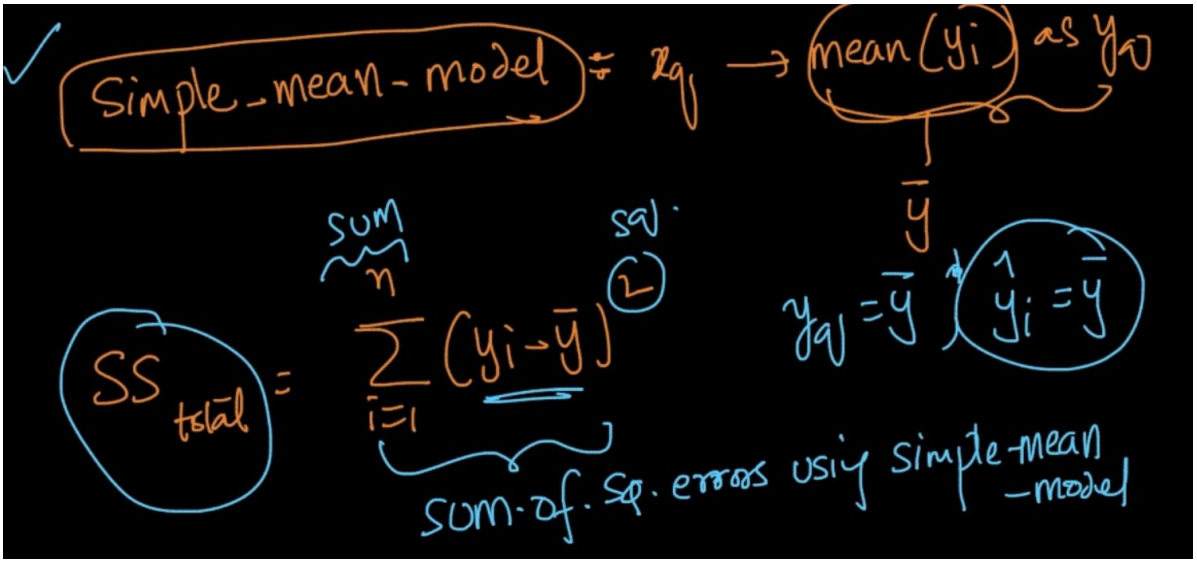


Error = Actual value (y_i) - Predicted Value($\hat{y}_i$)

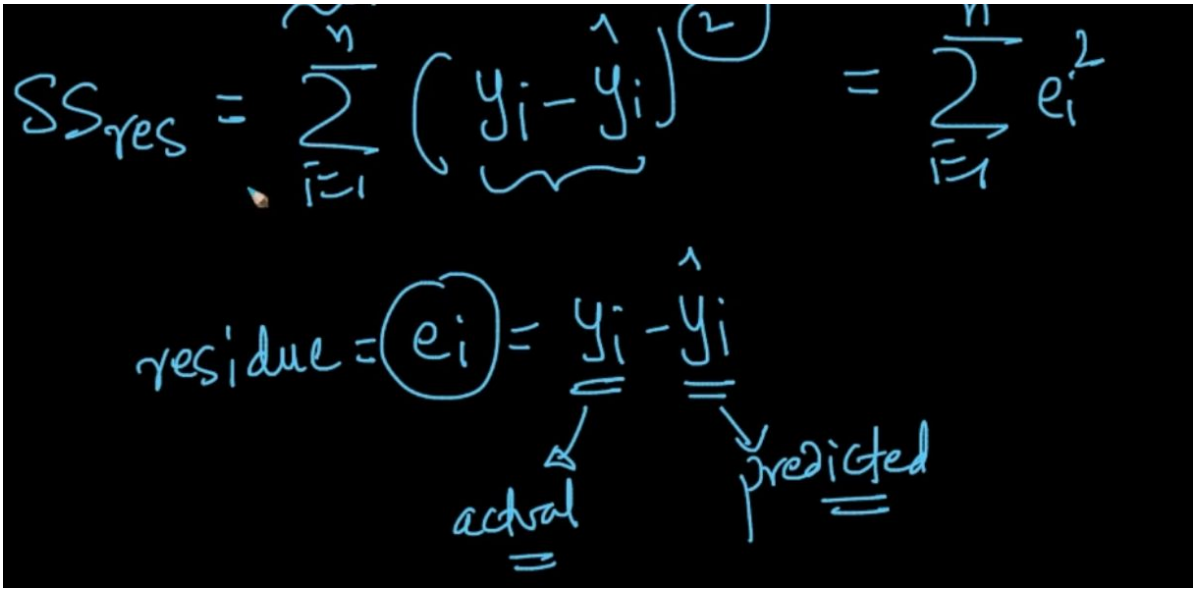


$\bar{y}$ is average of all y in data and Sum of square = $\sum_{i=1}^{n} (y_i - \bar{y})^2$ so we are replacing the predicted $\hat{y}$ with simple average $\bar{y}$. This is the simplest Regression Model

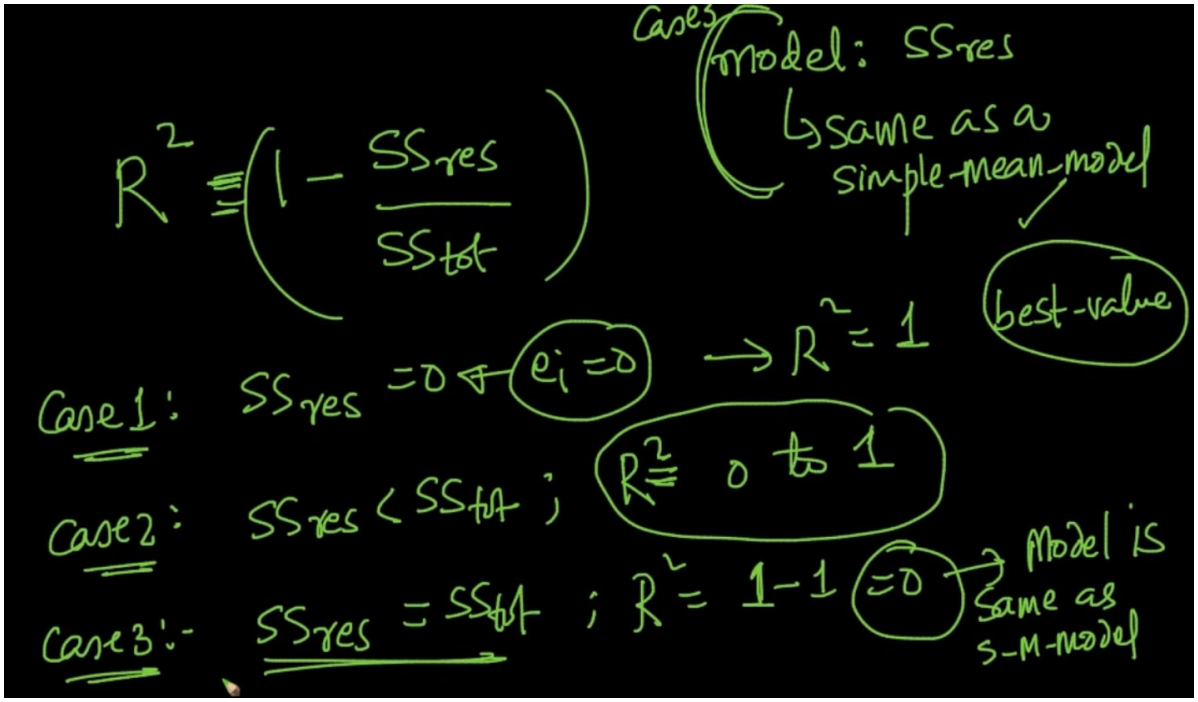
The explanation of above is given with the below image



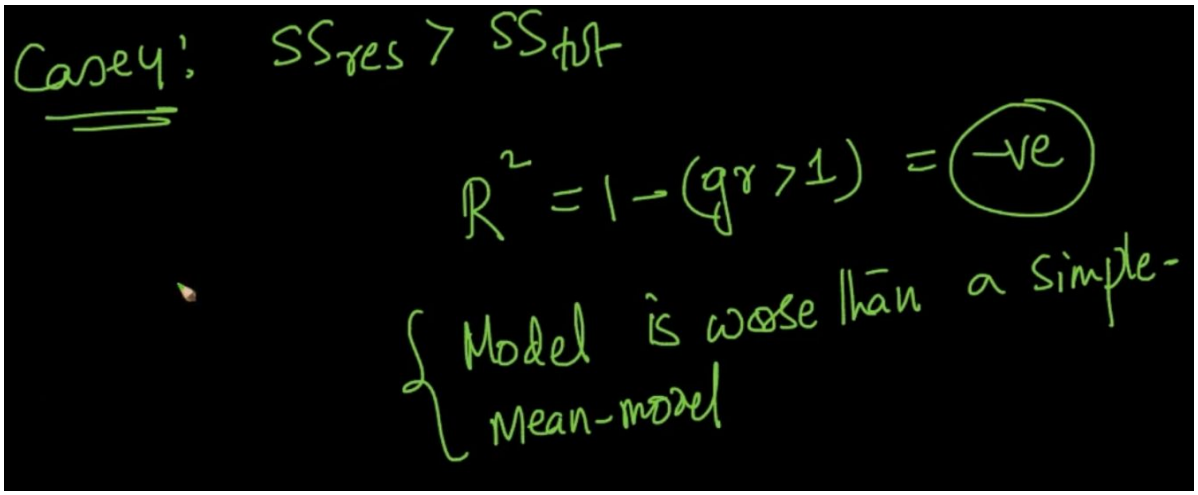
$SS_{residue}$



Here, we are using the actual predicted class labels $\hat{y}$ instead of $\bar{y}$

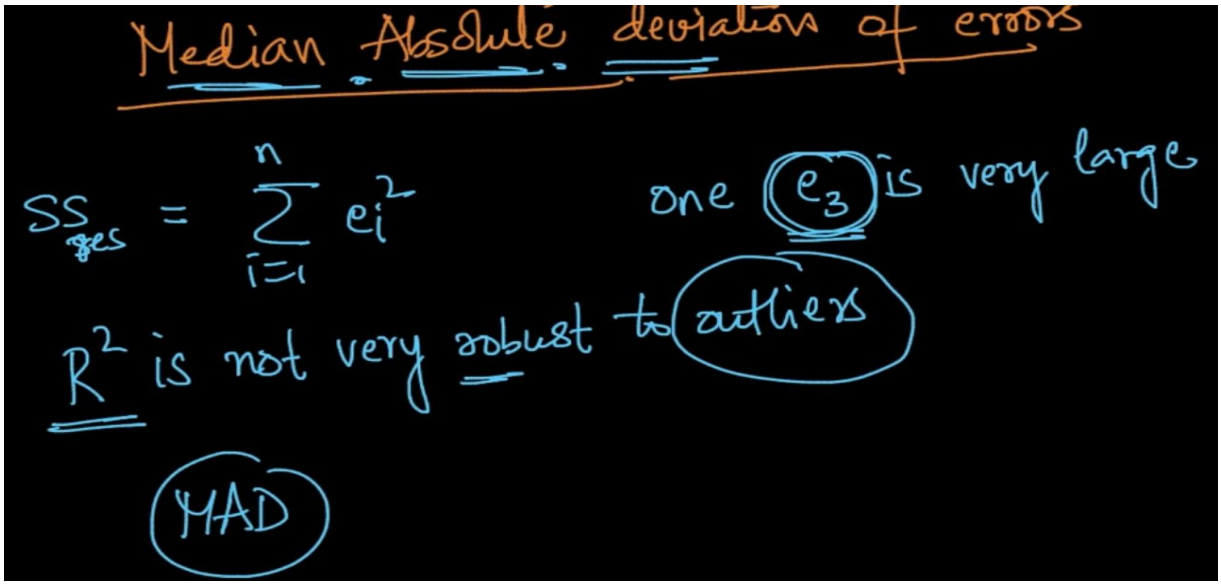


Our R^2 should be closer to 1 . The closer the better



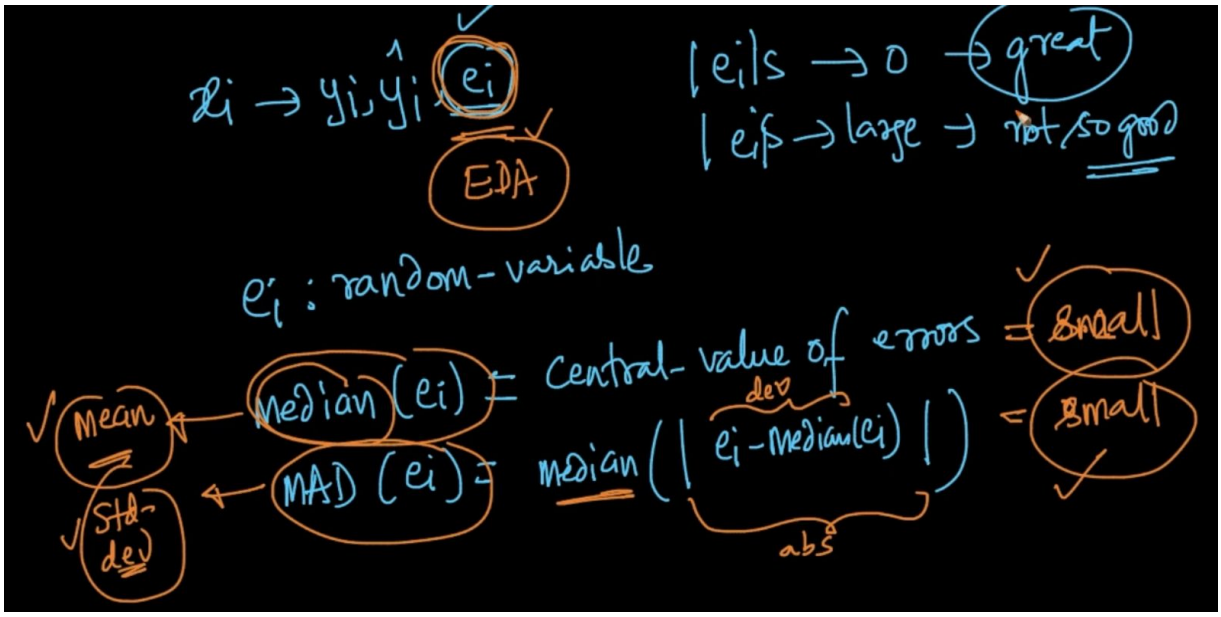
If $R^2 < 0$ it is worse than a simple mean model

MEAN ABSOLUTE DEVIATION OF ERRORS



In our Sum of Squares calculation (SS) if one error is very large then what should be done

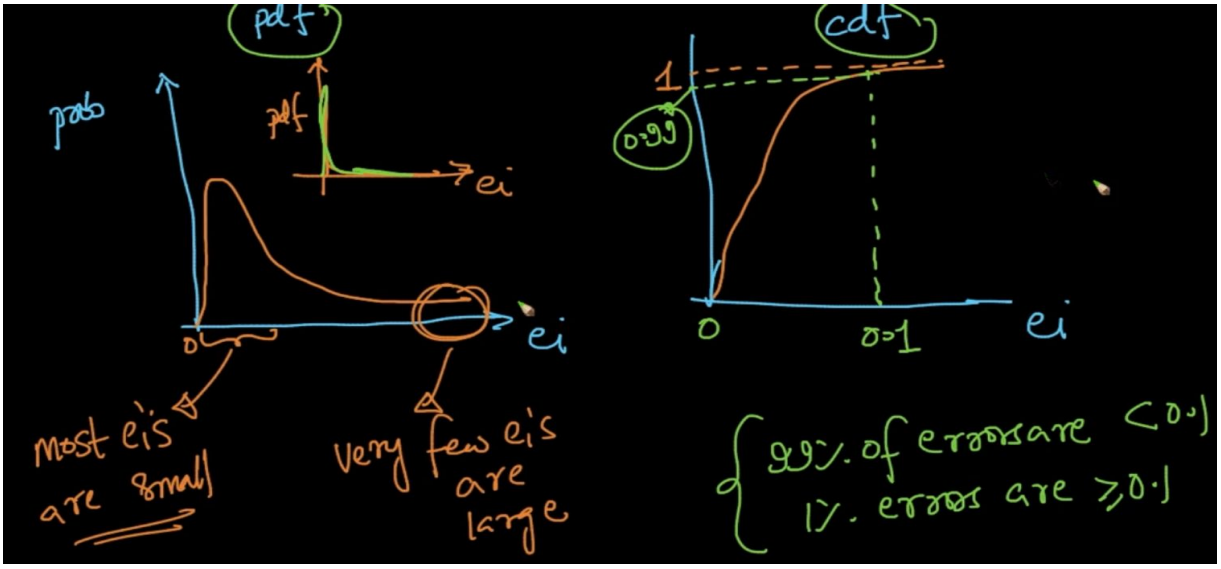
We use a technique called MAD here



MAD is mentioned above. If MAD is small then we can say that we are not prone to outliers and our model is working great

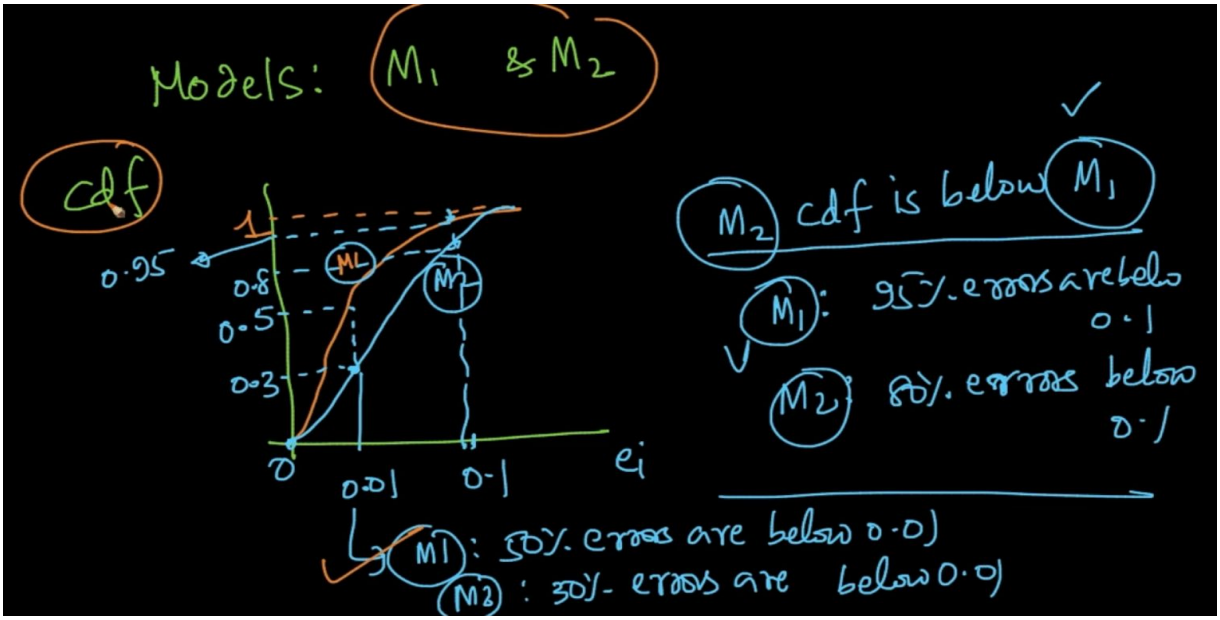
DISTRIBUTION OF ERRORS

We can use Probability Density Function (PDF) and Cumulative Distribution Function (CDF) For getting the distribution of errors



From PDF, we can infer that only few errors are large and most are small

From CDF, If error is 0.1 & we are getting it at 0.99 (y-axis) we can infer that 99% errors < 0.1



If we are comparing 2 models M1 and M2 with CDF for errors we can infer that M1 is better than M2 because of 95% errors < 0.1 in M1 but in M2 only 80% errors < 0.1