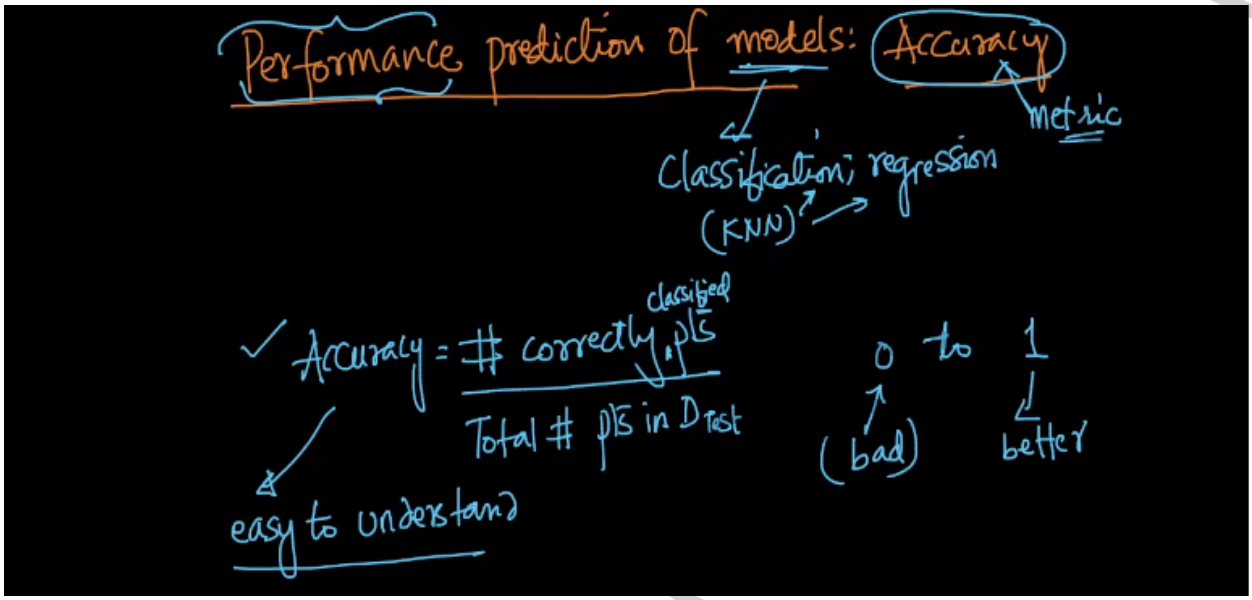
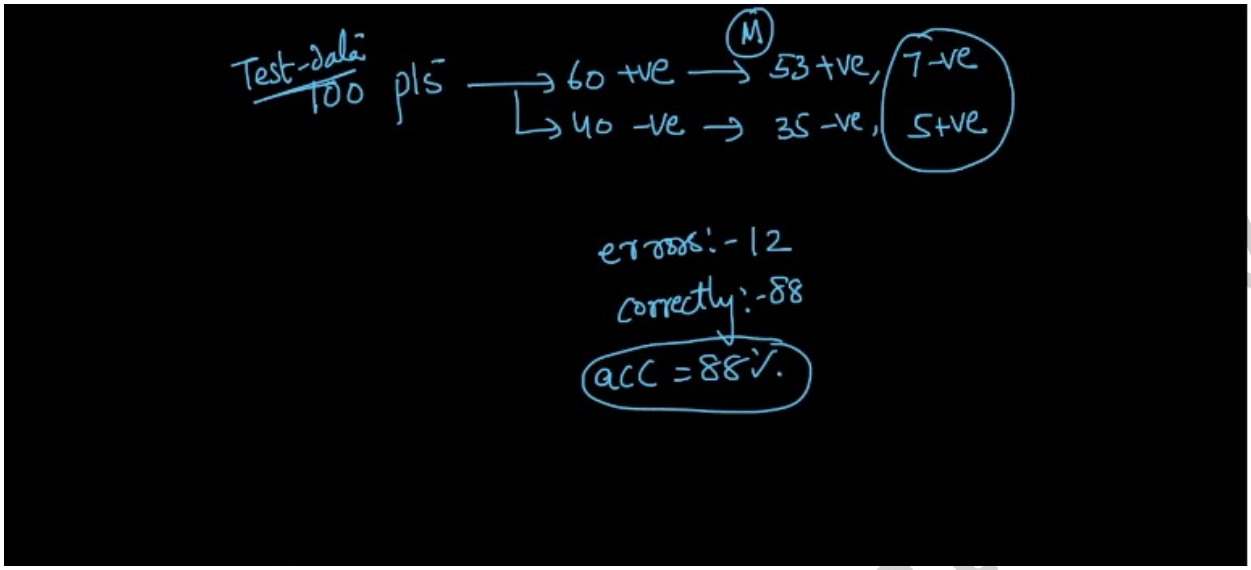
In this chapter we will look at various metrics by which we can assess the performance of our models. We will also look at the advantages and disadvantages of them. We will mainly look at metrics for classification and regression models.

30.1 Accuracy



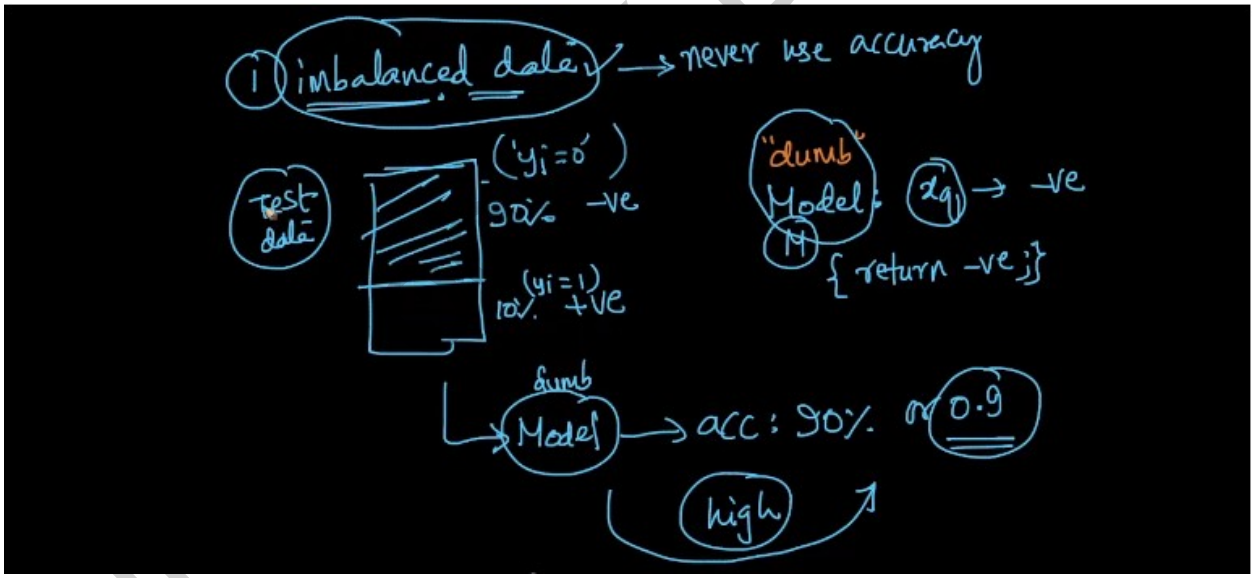
Timestamp 2:07

Accuracy can be defined as the ratio of the correctly classified and the total number of points, as shown in the image. It is generally used for classification. It ranges from 0 to 1, where 0 means that the model hasn't predicted any of the data points correctly and 1 means all the data points were classified correctly. It is very easily interpreted.



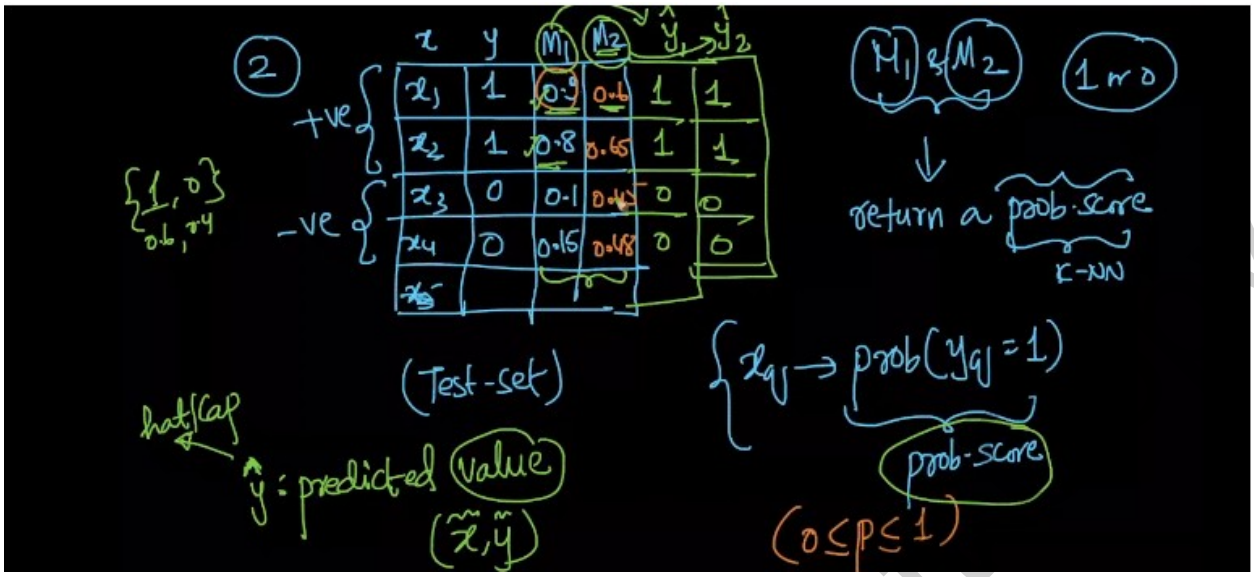
Timestamp 3:07

In the above image we can see how accuracy is calculated. For our final prediction we mainly compute our metrics on test data.



Timestamp 6:00

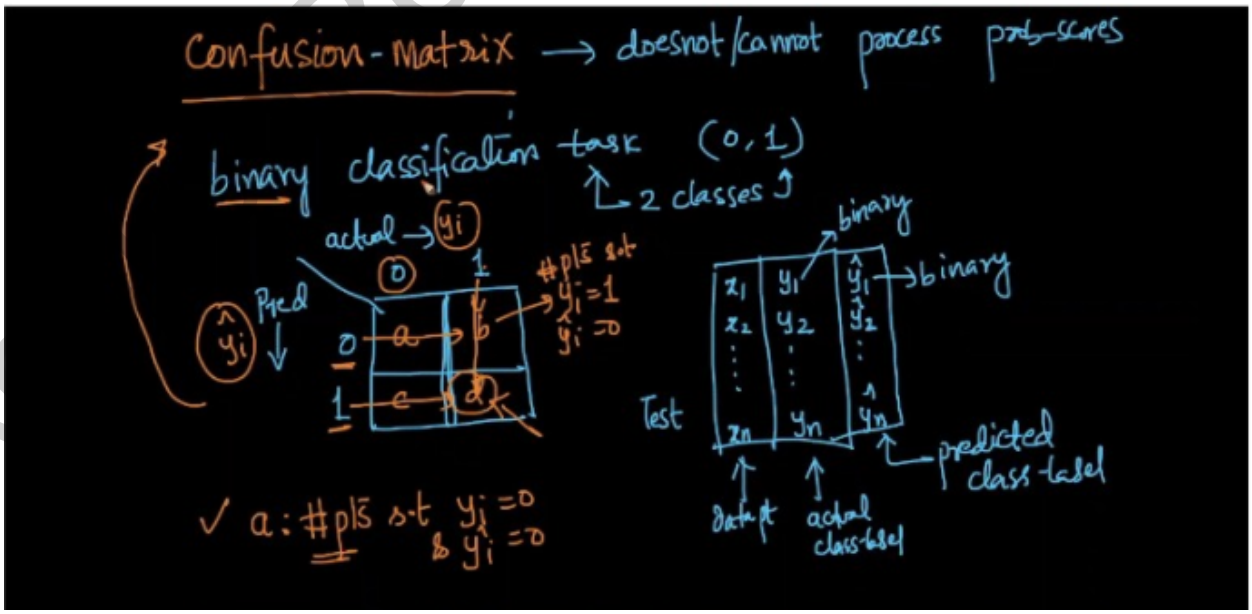
One of the disadvantages of using accuracy is that it performs very poorly for an imbalanced dataset. As you can see from the image, if my dataset has one majority class, say negative(say 90%) and we make a model such that we predict the majority class(negative) for every data point. This model will result in having a very high accuracy giving us a wrong conclusion that my model has performed very well. Even though my model hasn't learnt anything.



Timestamp 12:08

Another disadvantage of using accuracy is that it doesn't deal with probability scores directly. Consider KNN which can output probability scores. Let's say we have two models M1, M2. We have trained these models on x's and our true classes are y's. As shown in the image. Now we can see the probabilities predicted by these models. Let's say these values represent the probability that the predictions are 1. So according to that we have filled the predictions($\widehat{y}$) for both the models. Now we can calculate the accuracy for both these models using our $\widehat{y}$'s. We will see that both have equal accuracy even though M1 gave much better predictions.

30.2 Confusion Matrix, TPR, FPR, FNR, TNR

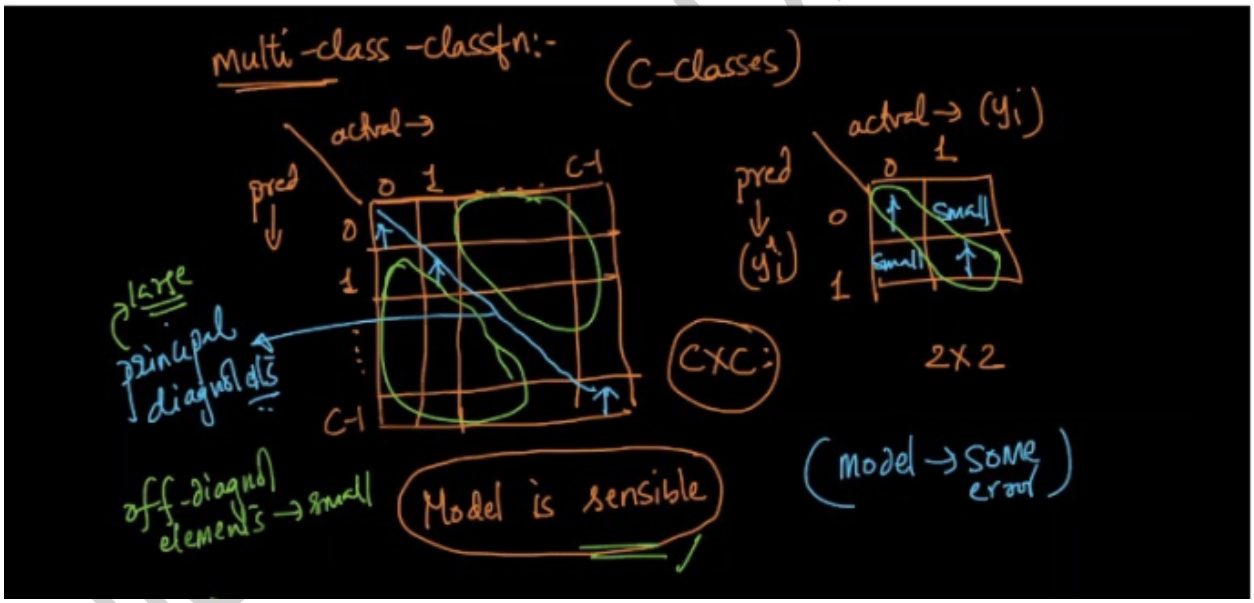


Timestamp 4:29

Here we will look at the confusion matrix, which is a very useful way of summarizing the predictions of the model. It works on actual class predictions and not probabilities. Consider a binary classification problem where y_i's(ground truth) $\in$ {0, 1}. Now for a binary classification problem, the combinations can have 4 possibilities. These 4 possibilities are represented using the cells of the 2-D matrix as shown in the image.

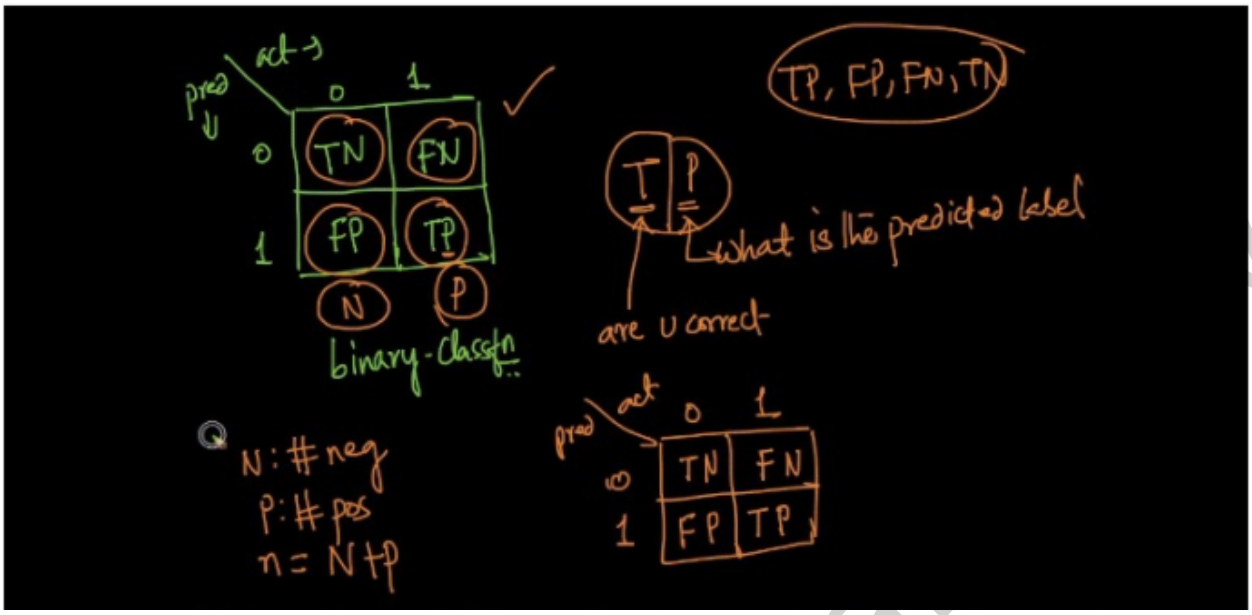
1. a: #pts such that y(actual) = 0 and $\widehat{y}$(predicted) = 0.
2. b: #pts such that y(actual)= 1 and $\widehat{y}$(predicted) = 0.
3. c: #pts such that y(actual)= 0 and $\widehat{y}$(predicted) = 1.
4. d: #pts such that y(actual)= 1 and $\widehat{y}$(predicted) = 1.

Also keep in mind that the headers of the cell are important like in which direction you have actual values and in which direction you have predicted values. In some textbooks these can change.



Timestamp 7:41

Similarly the confusion matrix can be extended for c classes. These can be represented using a square 2-D matrix of c dimensions. Note in the matrix the diagonal elements represent the correct predictions whereas off diagonal elements represent the wrong predictions. A good or sensible model will have large diagonal elements and lower off diagonal elements.



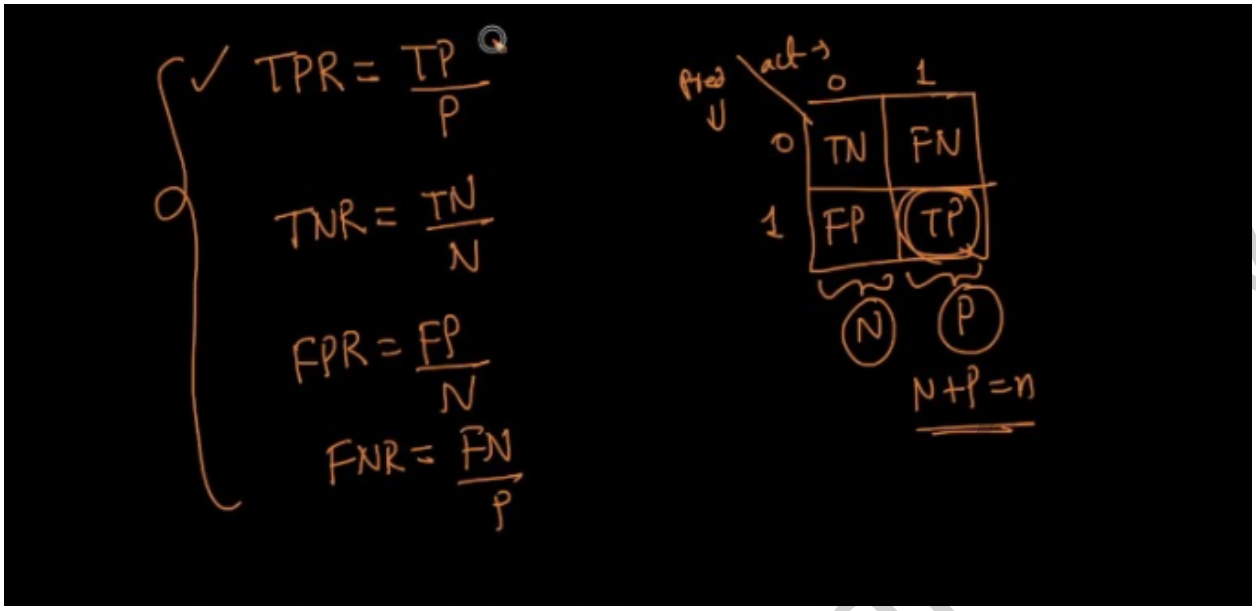
Timestamp 11:05

We can name our cells as shown below:
True Negative(TN) - Ground truth: 0 and Predicted: 0
False Negative(FN - Ground truth: 1 and Predicted: 0
False Positive(FP) - Ground truth: 0 and Predicted: 1
True Positive(TP) - Ground truth: 1 and Predicted: 1

One easy to trick to remember this is to break it into two parts, where the first part says whether we are correct and the second part says our prediction
For e.g. - In TP, T - represents whether our prediction was correct, in this case it was correct i.e. True and P represents what is our prediction, in this case positive.

If we add FN + TP , we get our total positive points, similarly TN + FP gives us the total negative points that were present in the dataset.

If we add both positive and negative data points we get our total no of data points in the dataset.



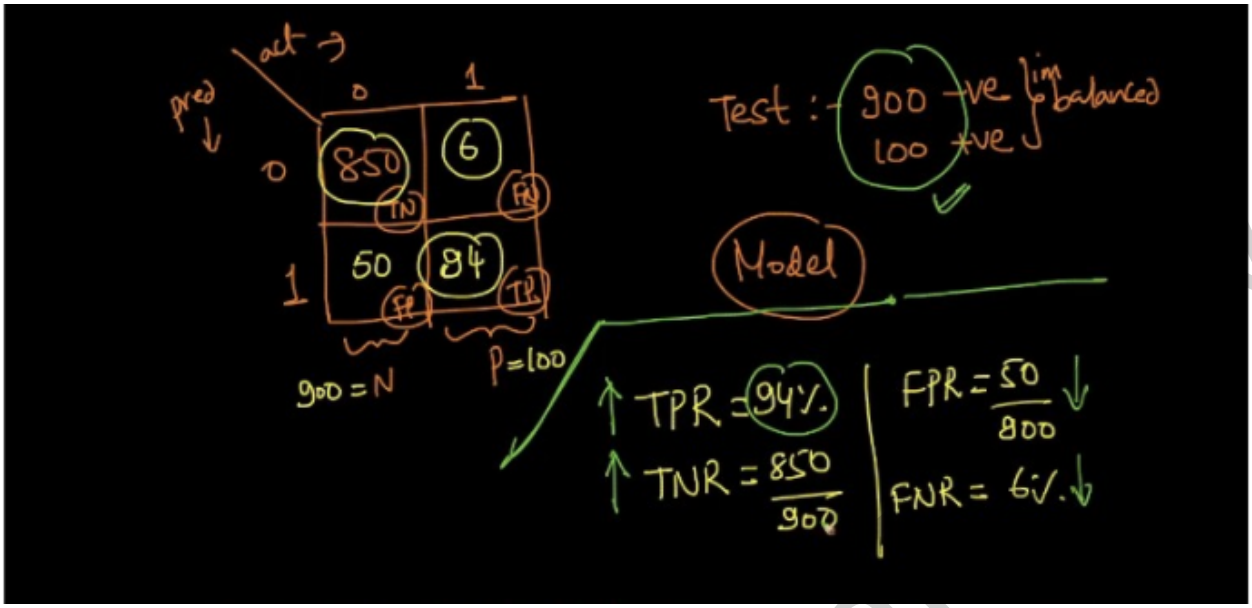
Timestamp 13:03

Here we will define a few more terms.

True Positive Rate (TPR) = TP/P
True Negative Rate(TNR) = TN/N
False Positive Rate(FPR) = FP/N
False Negative Rate(FNR) = FN/P

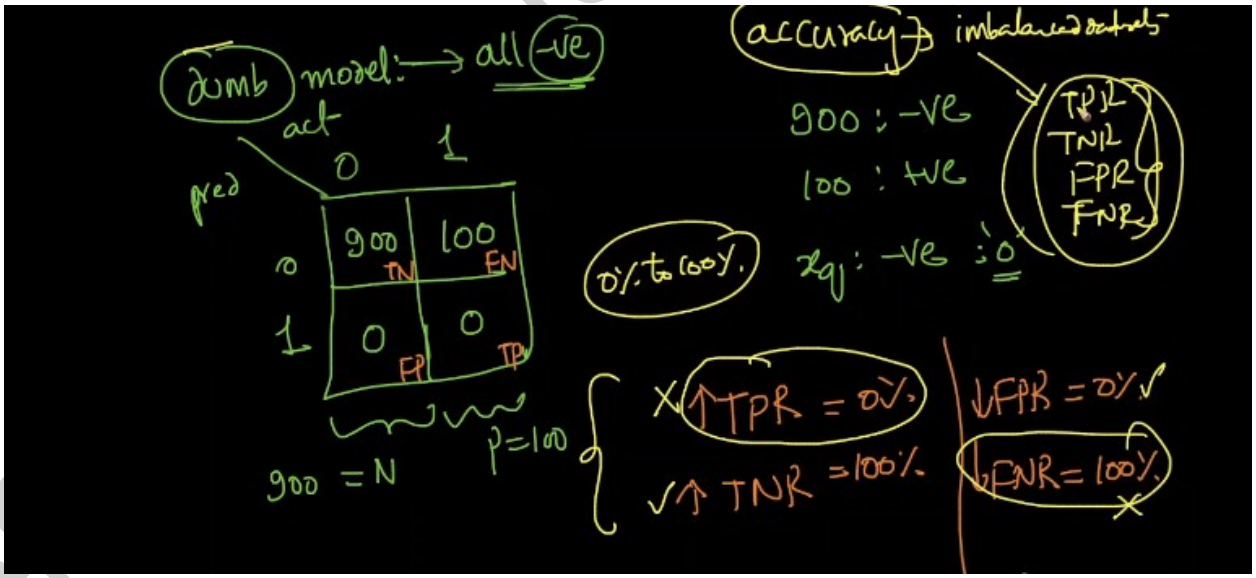
Here P = #positive points, N = #negative points.

We can relate it to the diagram in the image above to help remember these formulas better.



Timestamp 16:19

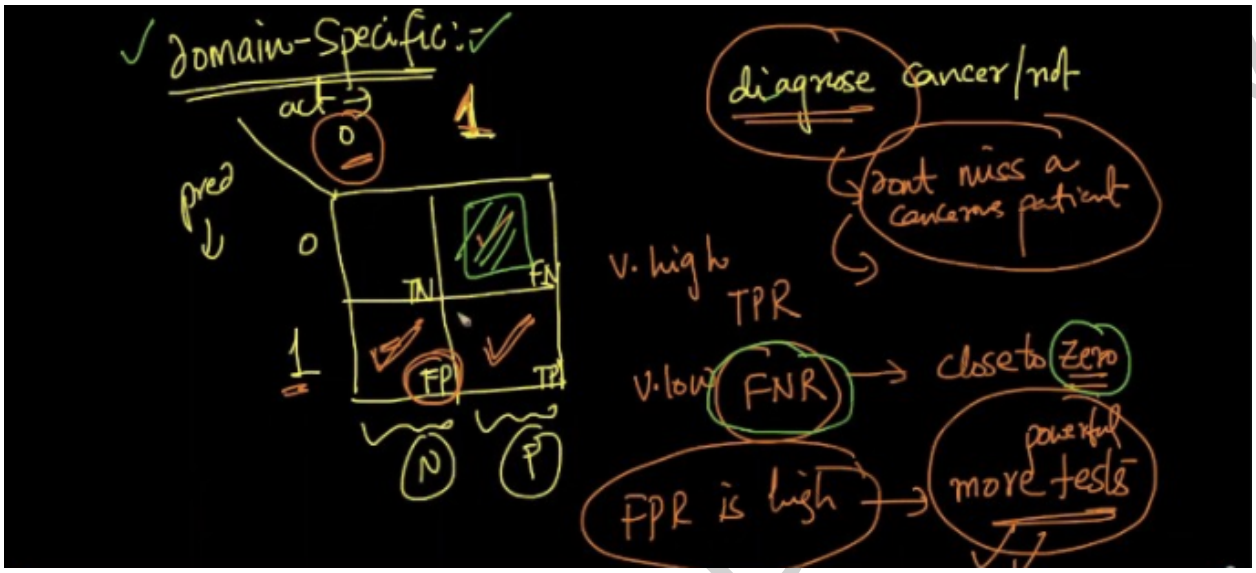
In the above image it can be seen that we have taken an imbalanced dataset. Suppose our model has made predictions for all the data and based on that we have filled the cells of confusion matrix and calculated various metrics. It can be clearly observed that the diagonal values are high whereas off diagonal values are low. This means that our model is good. Good models have high TPR,TNR whereas FPR, FNR should be low.



Timestamp 20:02

Here we have again taken an imbalanced dataset and used dumb model. Dumb model predicts the majority class without learning anything. Now, we have filled the confusion matrix according

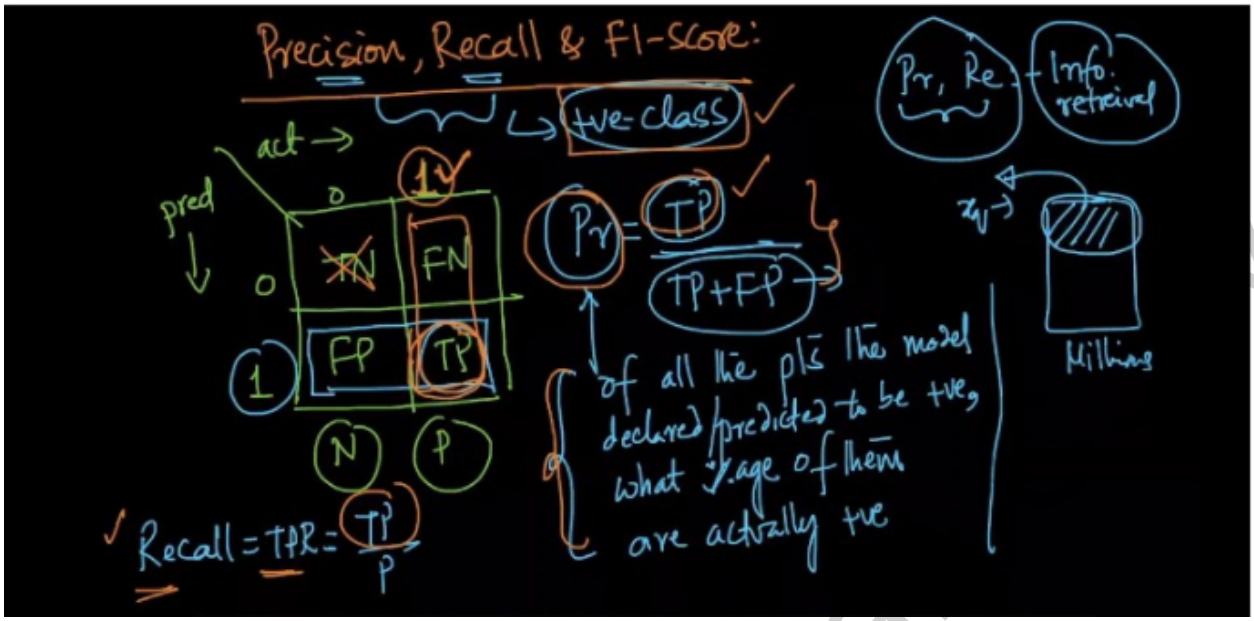
to this. We can now observe that TNR is high, but TPR is low. Also FNR is high, indicating that my model is not good and we need to improve it. This type of observation was not possible in case of accuracy. But on the contrary now we have to deal with 4 numbers instead of one like in accuracy.



Timestamp 24:50

Now the question comes whether we should use a confusion matrix and deal with 4 numbers or use just accuracy and deal with just one number. The answer is it is domain specific. It depends on the problem at hand. Consider a case of cancer diagnosis, where we have built a model that predicts whether a person has cancer. Now in this case we want to make sure that if a patient has cancer, our model detects it i.e. high TPR and also low FNR so that we don't miss a person that has cancer. It is very important to have low FNR. This wouldn't have been possible if we would have used accuracy.

30.3 Precision, Recall & F1-Score

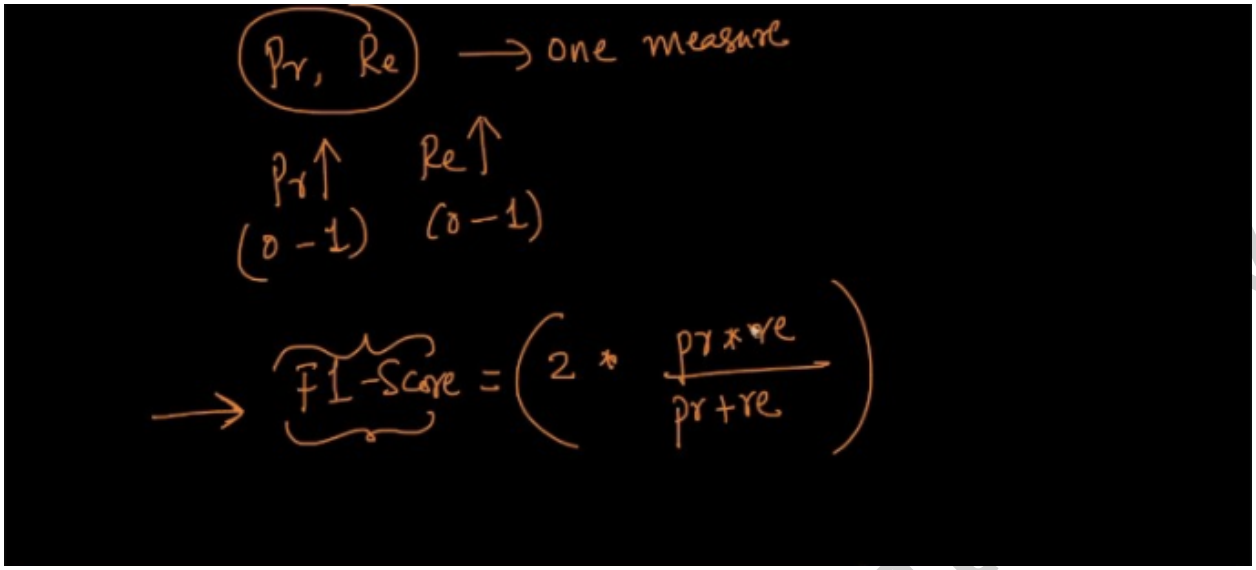


Timestamp 5:17

Here we will talk about precision and recall. Both these metrics have come from information retrieval, which focuses on retrieving certain documents from millions of documents based on certain conditions. Both these metrics focus on positive class. So for any problem we can use these metrics by changing our requirement to positive class.

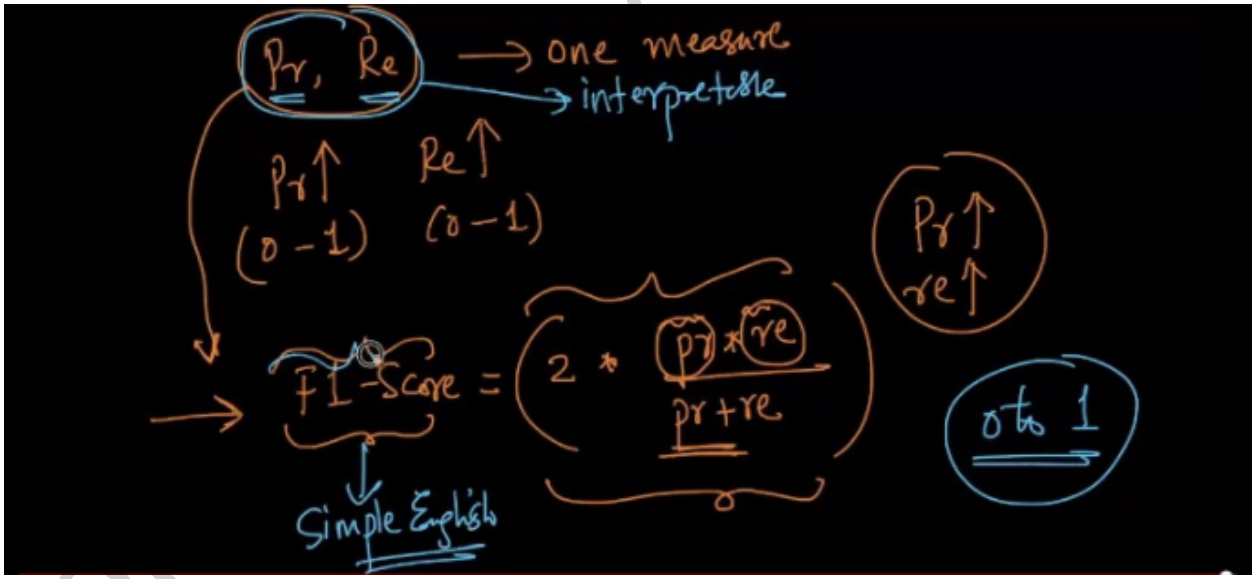
Precision = TP/(FP + TP). Intuitively it means that out of all the points our model predicted to be positive, what percentage of them are actually positive. It's a number between [0,1], with large values meaning high precision.

Recall = TPR = TP/(FN + TP). Intuitively it means that out of all positive points that were present in the dataset, how many of them were predicted successfully. It's a number between [0,1], with large values meaning high recall.



Timestamp 6:23

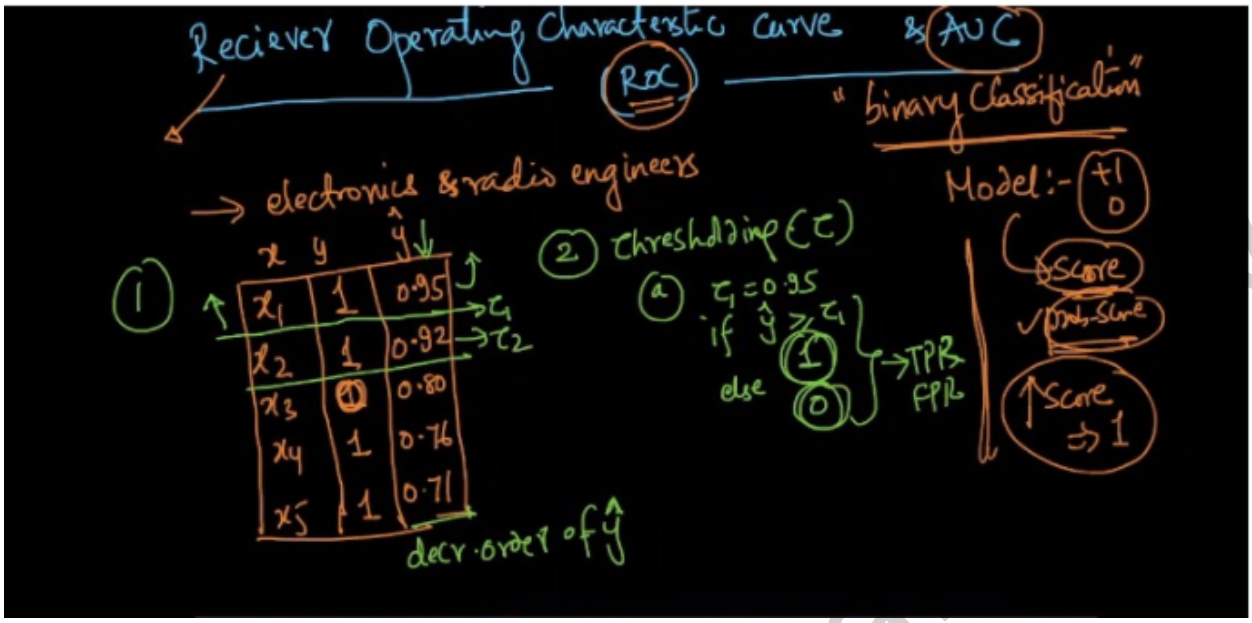
Both precision and recall can be combined to give us one score. Generally we want both precision and recall to be high. We combine them using something called F1-Score which is the harmonic mean of precision and recall.



Timestamp 9:07

F1-Score's are less interpretable than precision and accuracy. It is widely used in kaggle competitions. It ranges between 0 & 1, where a higher F1 score means a better model.

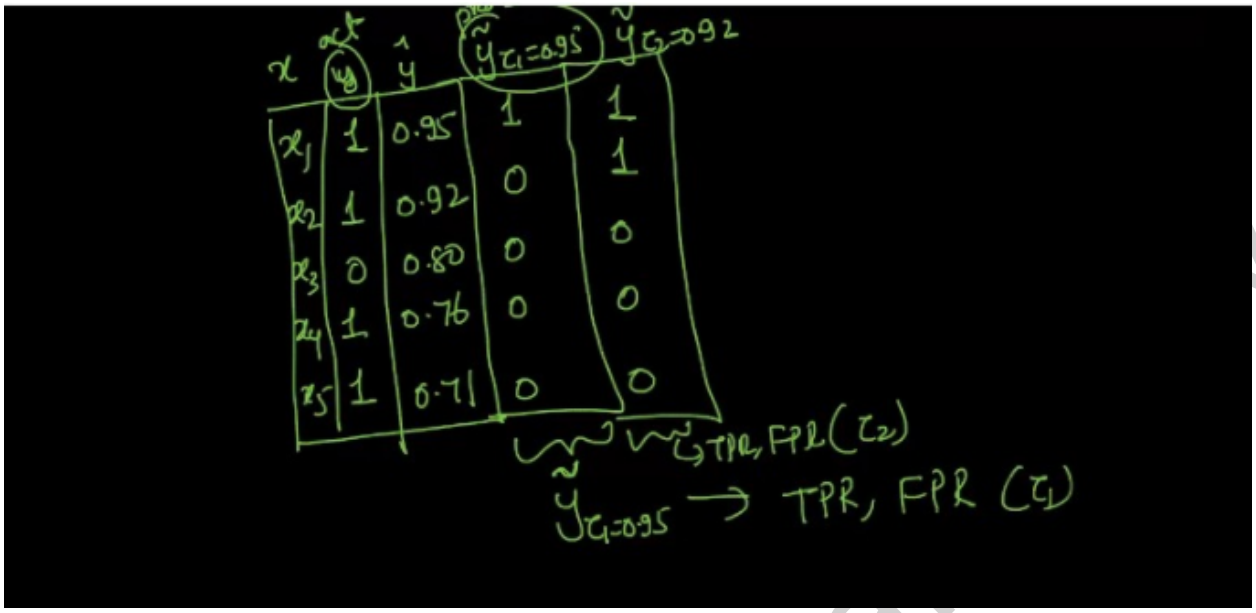
30.4 Receiver Operating Characteristic Curve (ROC) curve and AUC



Timestamp 6:15

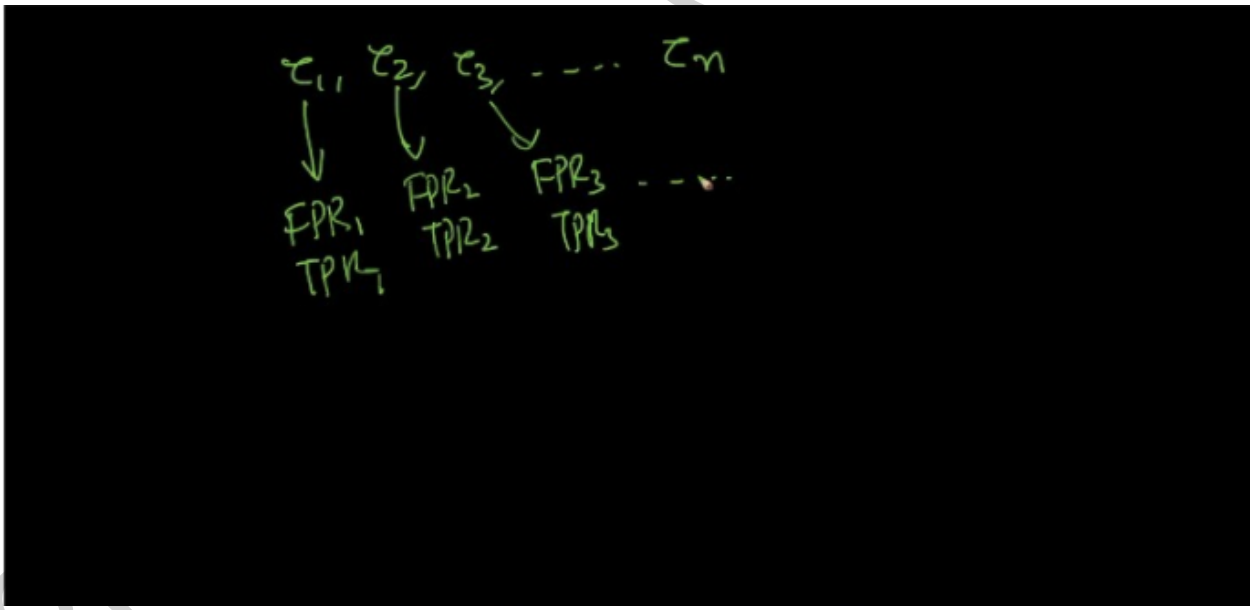
Receiver Operating Characteristic Curve or ROC is a metric which is mainly used for binary classification problems. It was developed during the world war mainly by electronic and radio engineers.
Suppose we have a model that outputs probability scores as output, remember ROC can work on any scores. Now in order to calculate the ROC we need to follow the below steps.

1. In this step we sort the data table based on the decreasing value of predictions($\hat{y}$). As it can be seen in the above image marked by 1.
2. Thresholding - In this step we first select the topmost value say ς_1, in our example we have ς_1 = 0.95, and use this as the threshold i.e. any value greater than or equal to this value will be assigned a class label of 1 and the other ones will labeled as 0.



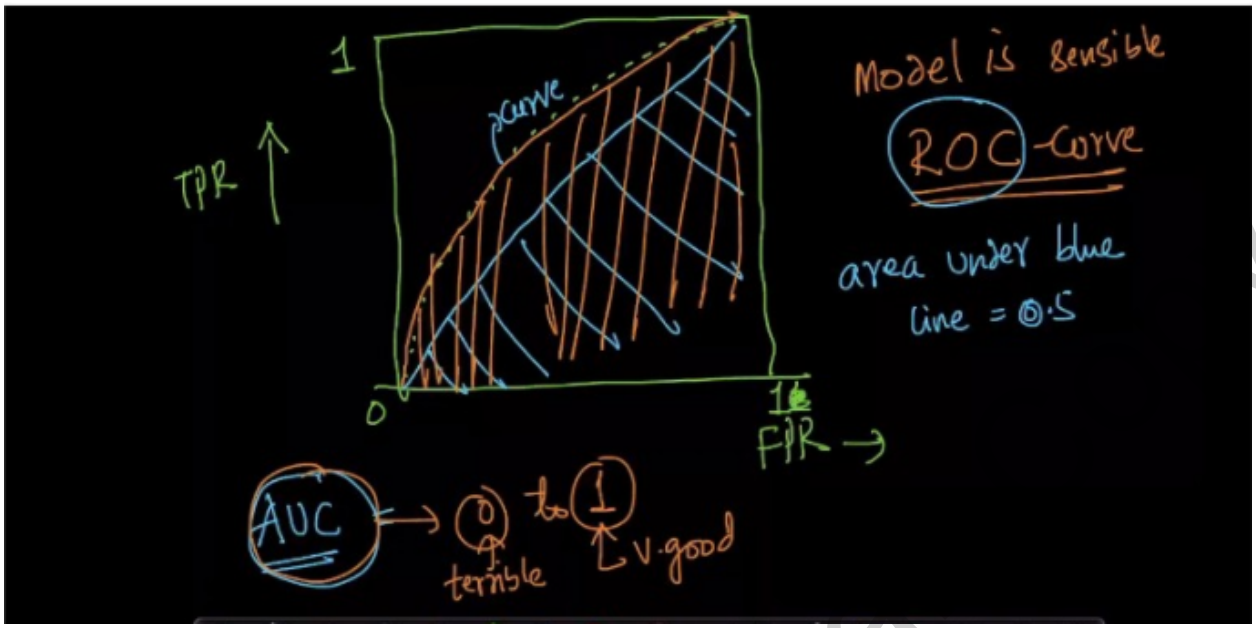
Timestamp 6:40

Similarly these thresholding process should be followed for other $\hat{y}'s$ based on their ordering. For each threshold we will get a different list of predictions as shown in the above image. Now, for each of these predictions w.r.t different thresholds, we calculate their TPR and FPR.



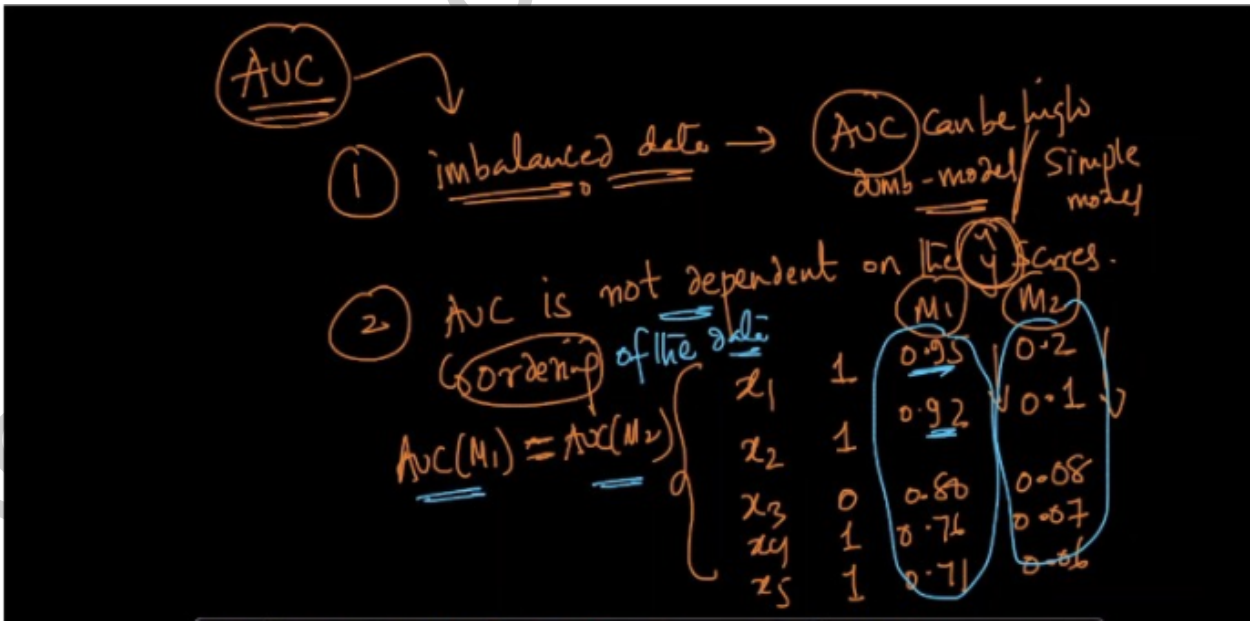
Timestamp 7:27

Now, if our dataset had n rows, then we will have n different thresholds, and for each of these thresholds we will have different TPR and FPR.



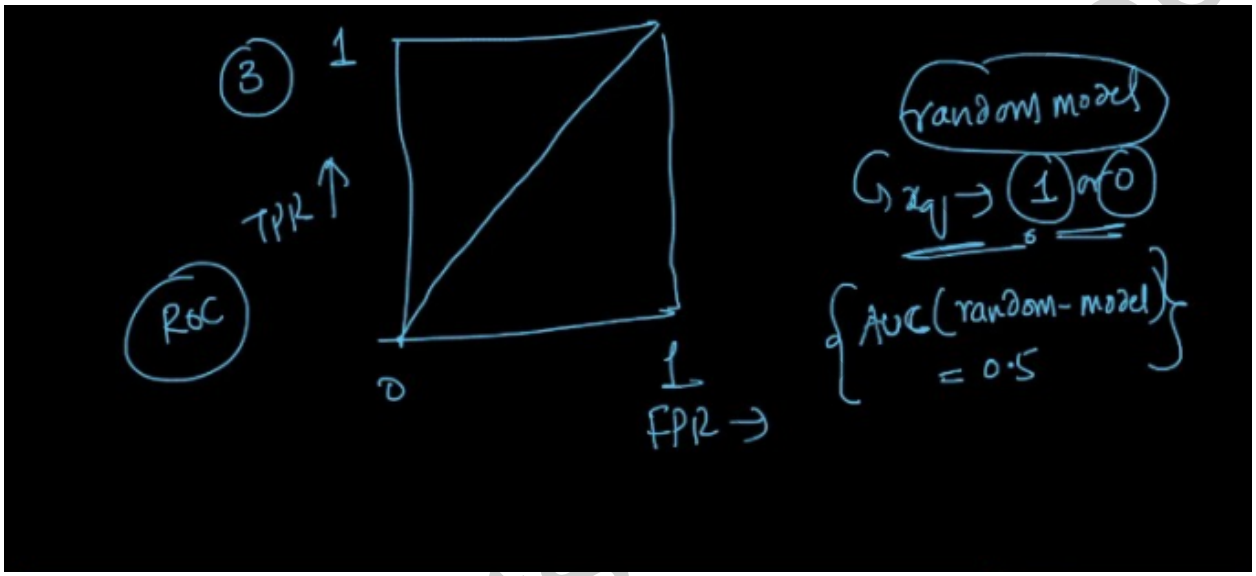
Timestamp 10:15

Now we have a list of FPR's and their corresponding TPR's. Also keep in mind that their values lie between 0 &1. We can now plot these values in a graph with FPR on x-axis and TPR on y-axis. After plotting the points we will get a curve shown in red line, it is called the ROC Curve. The total area under the curve can be 1, as the maximum value is 1 on both axes. The blue line represents the line under which half of the area lies. Now, the area under the ROC Curve is called AUC(Area under the curve). It's value can range from 0 to 1, with higher value meaning better model.



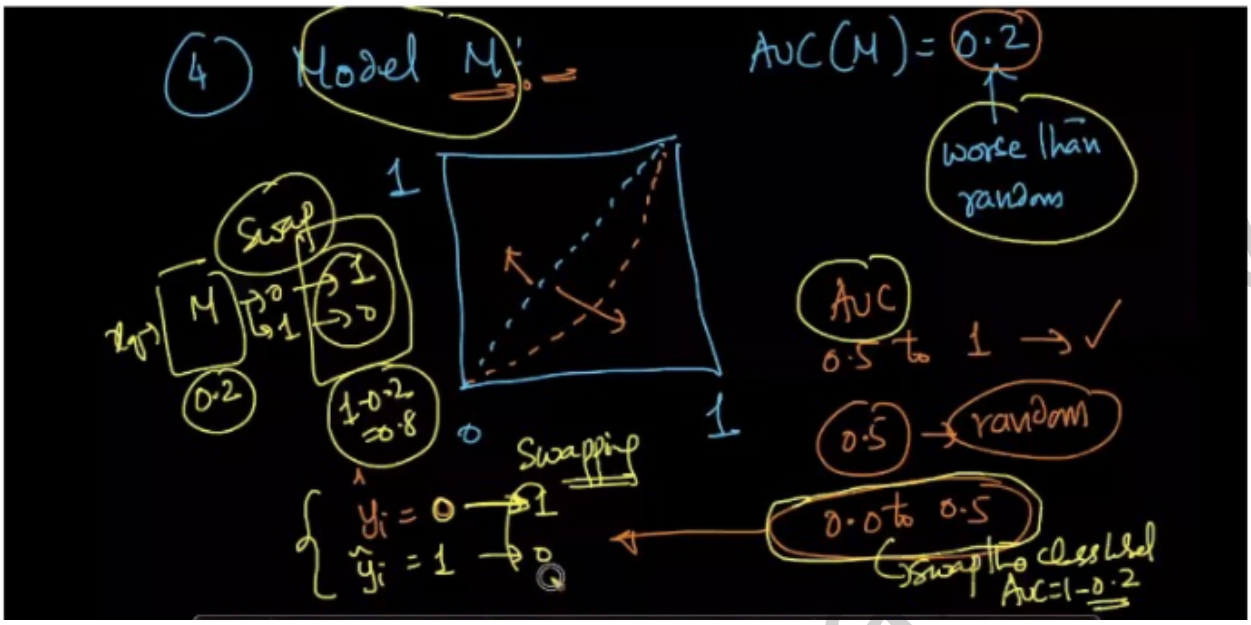
Timestamp 13:30

Now let's look at several properties of AUC.
1. AUC is impacted by imbalanced dataset i.e. even a dumb or simple model can have high AUC.
2. AUC is not dependent on the actual predictions but rather on the ordering of the predictions. Consider two models M1 and M2 making different predictions as shown in the image above. If we calculate the AUC for both these models, it will be the same. But it can be clearly observed that M1 did much better at predicting than M2.



Timestamp 14:53

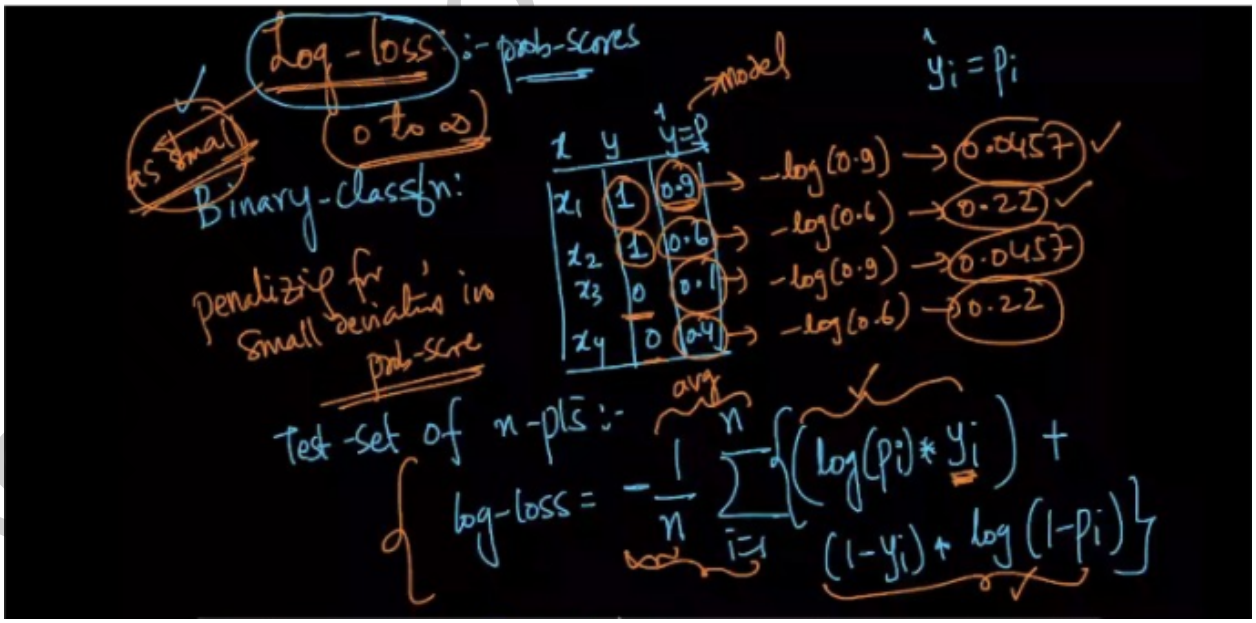
3. AUC for a random model will be 0.5. A random model is such that for each prediction it will select class 0 or class 1 with equal probability, same as tossing a coin.



Timestamp 18:18

4. Suppose we have our model M, that has an AUC of 0.2 as shown in the image by the red line, which is worse than a random model. In that case we can deploy a trick. Suppose my model predicts class 0, then we switch our predictions and say that the prediction is 1 and vice-versa. We can just switch our predictions. Doing this will change our AUC score from 0.2 to 1 - 0.2 = 0.8, which is a pretty good auc score.

30.5 Log Loss



Timestamp 6:15

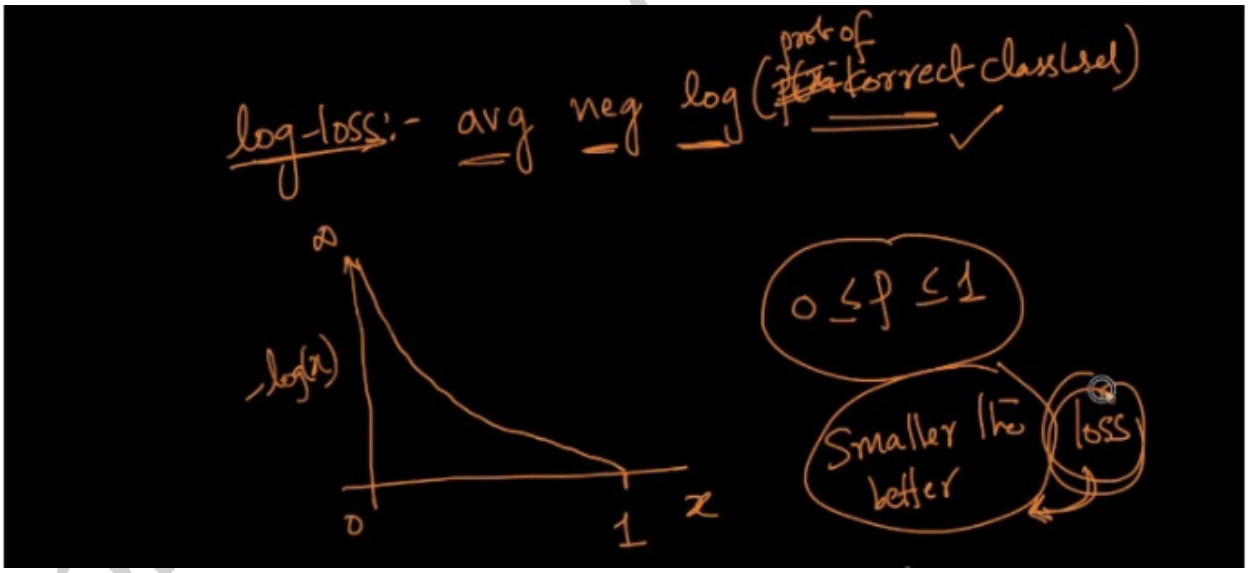
Log loss is loss metric which lies between [0, ∞). The lower the loss value the better the model. Log loss deals with direct probability scores unlike other metrics that were defined till now. Consider a set of n points, log loss for these data set can be defined as:-

$$\text{Log-loss} = -\frac{1}{n}\sum_{i=1}^{n}\{(\log(p_i)*y_i) + (\log(1 - p_i)*(1 - y_i))\}$$

Here p_i =Probability that the class belongs to 1,

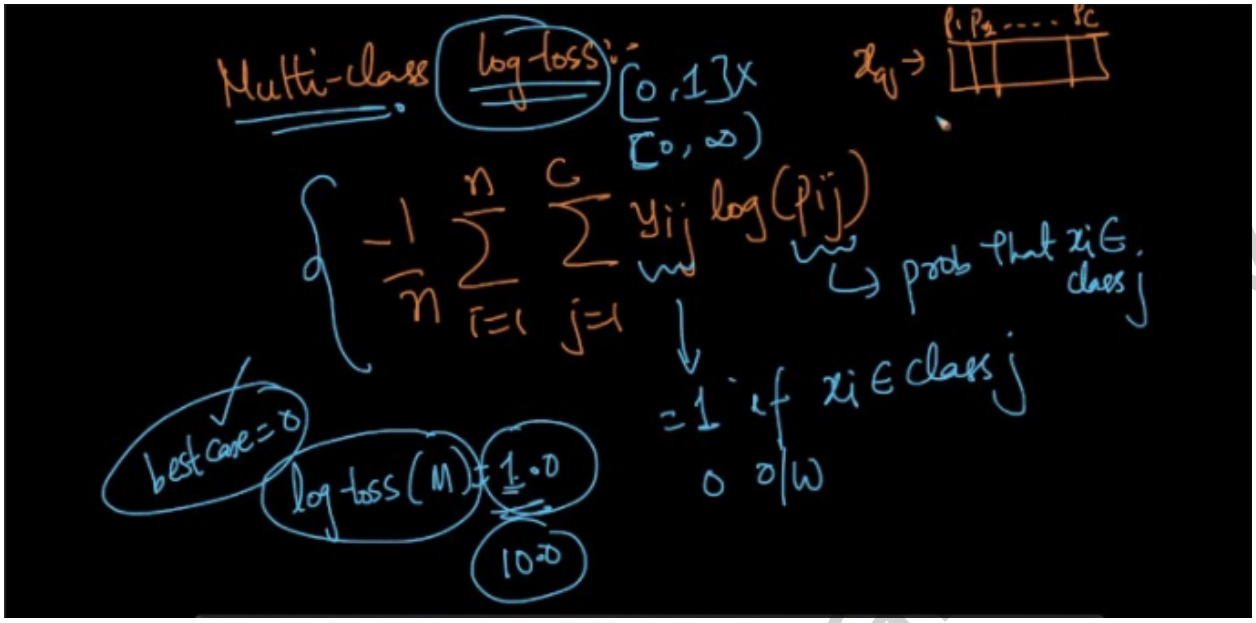
y_i = Actual class label, n = total no of datapoints

As it can be seen from the above formula it is a negative average of all points. This formula can be thought of in two parts.
Suppose our class label y_i= 1, then the second part vanishes as 1 - y_i =0, so the formula that remains is -log(p_i)* y_i. Now here since y_i =1, we have -log(p_i), which is the probability that y_i belongs to class 1. Now greater the value of p_i smaller will be the negative log value, hence smaller the loss. This can be verified for the class 0 case also.
In the above image we have calculated various log values and shown how it varies w.r.t change in probability values for different classes.



Timestamp 8:16

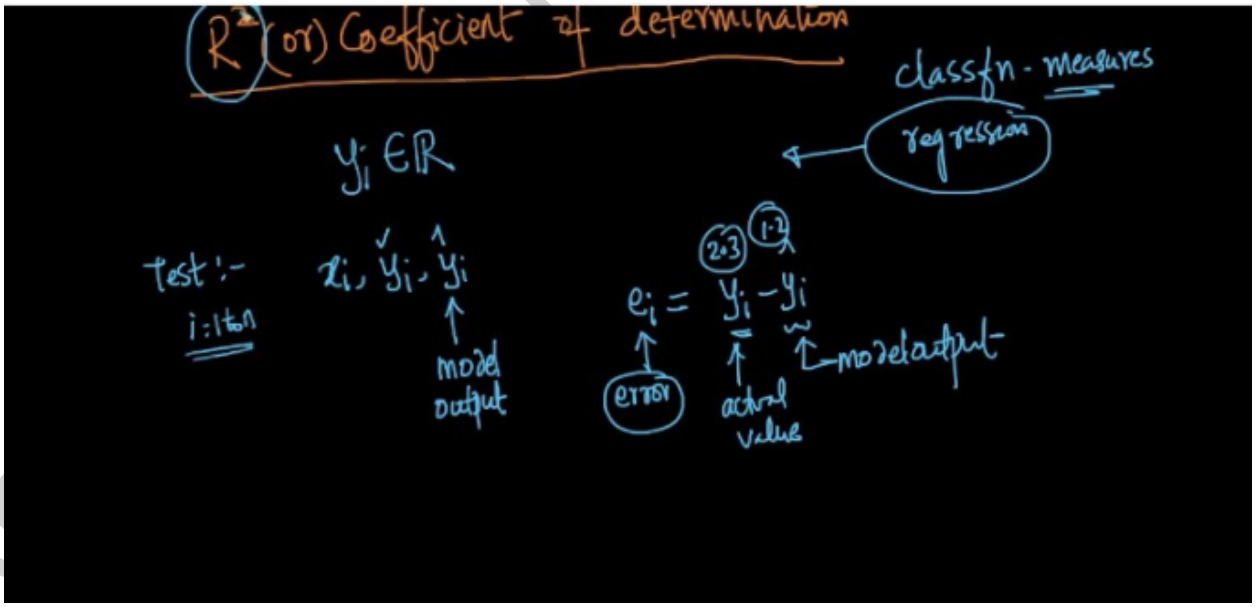
In the above image we can see a graph which shows the relationship between the probability scores and the negative log of those scores. As the scores rise the negative log value decreases. We always strive for smaller loss values.



Timestamp 11:02

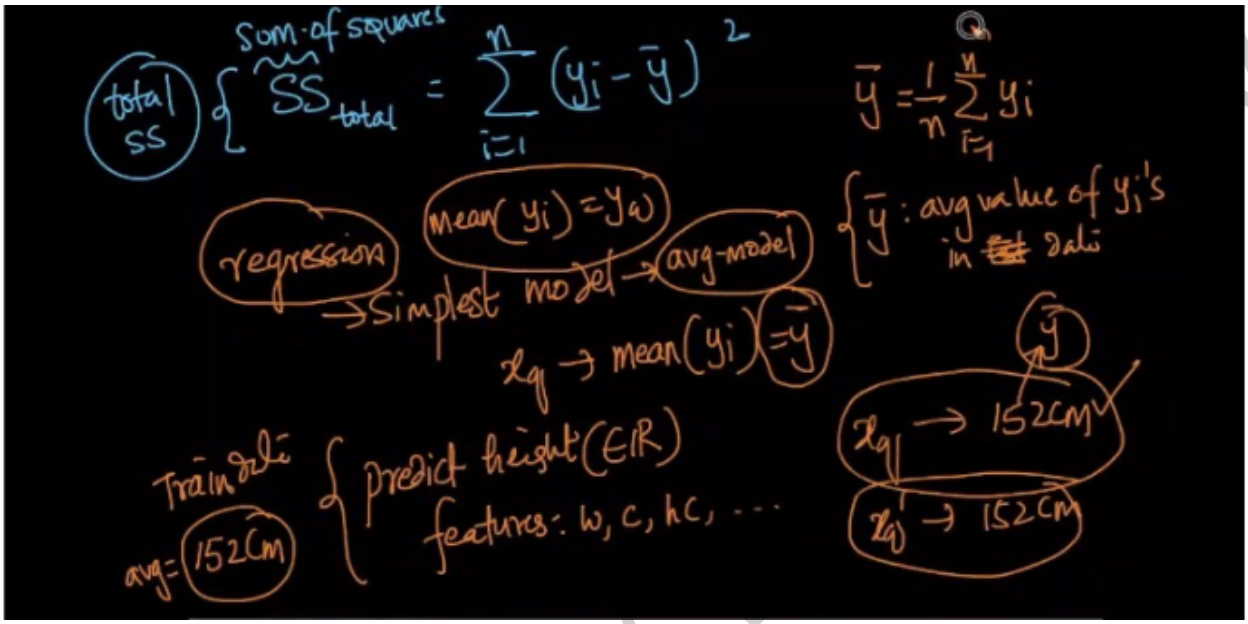
The formula for log loss can be easily extended to a multiclass setting as shown in the above image. Here c represents the number of classes that we have. We can verify this formula by setting c = 2, and observing that it reduces to binary log loss.
One of the disadvantages of log loss is that it is very difficult to interpret.

30.6 R-Squared/Coefficient of determination



Timestamp 2:02

Here we will talk about the metrics for regression problems. Our y_i's ∈R i.e. set of real numbers unlike classification tasks. Now, we can define error as the difference between the ground truth or actual value and predicted value as shown in the above image. We represent it as e_i.



Timestamp 6:32

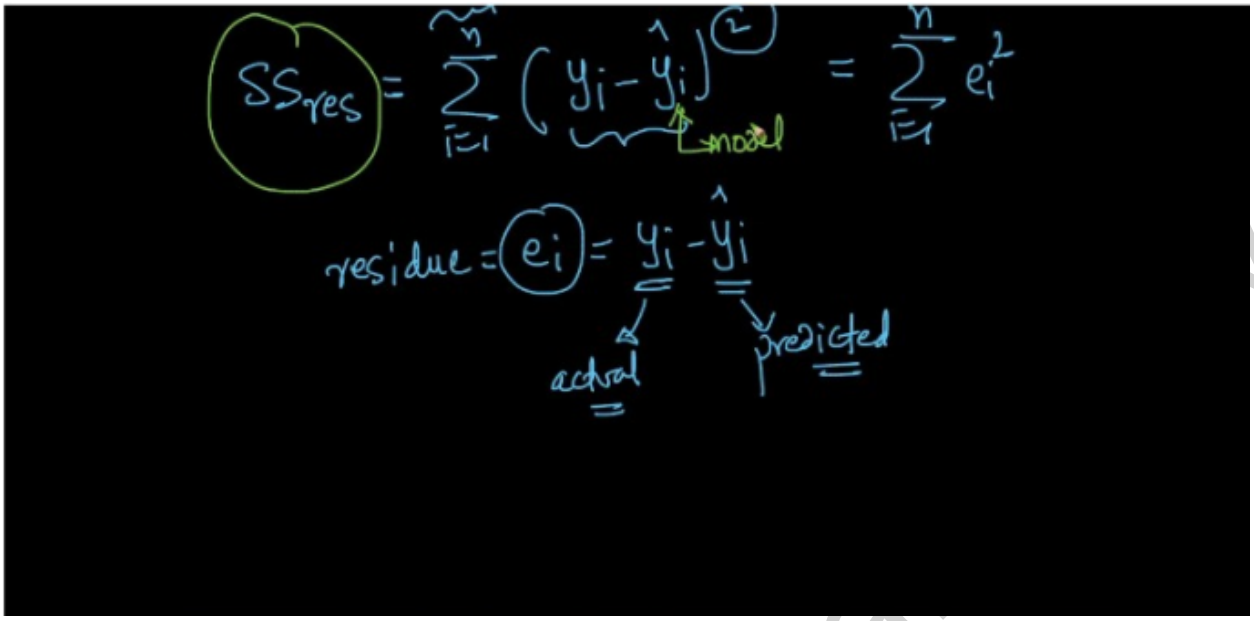
Here we will define a term SS_{total} ,total sum of squares,

$SS_{total} = \sum_{i=1}^{n} (y_i - \bar{y})^2$

y_i = ground truth

$\bar{y}$ = average value of all the y_i's in the train dataset.

This term represents the sum of the square of the errors between y_i and $\bar{y}$. Here for prediction we have predicted $\bar{y}$ for every datapoint in train data, which is the mean of y_i's. This is a very basic model where we haven't learnt anything useful. This is also called the average-model or simple-mean-model.



Timestamp 8:53

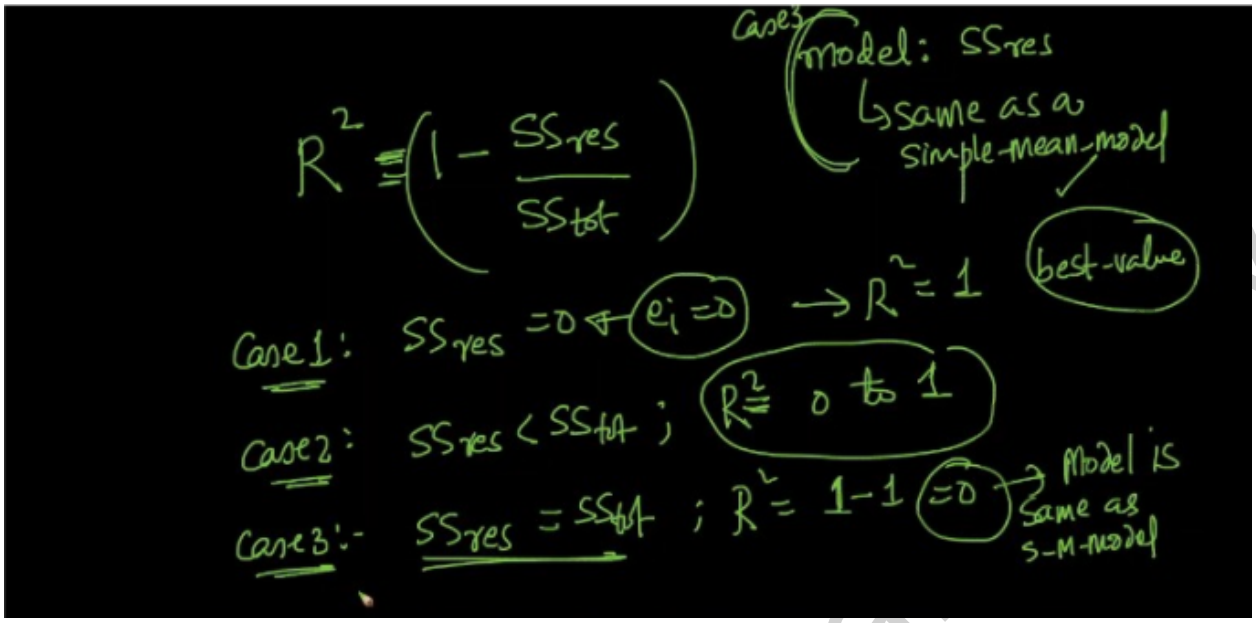
Here we will define a term SS_{res} ,residual sum of squares,

$SS_{res} = \sum_{i=1}^{n} (y_i - \hat{y})^2$

y_i = ground truth

$\hat{y}$ = predicted value based on a trained model.

This term represents the residual sum of the square of the errors between y_i and $\hat{y}$. Here we have trained a model on the dataset and based on that we are giving predictions.



Timestamp 11:30

Now we can define our coefficient of determination(R^2),

= (1 - SS_{res}/SS_{total})

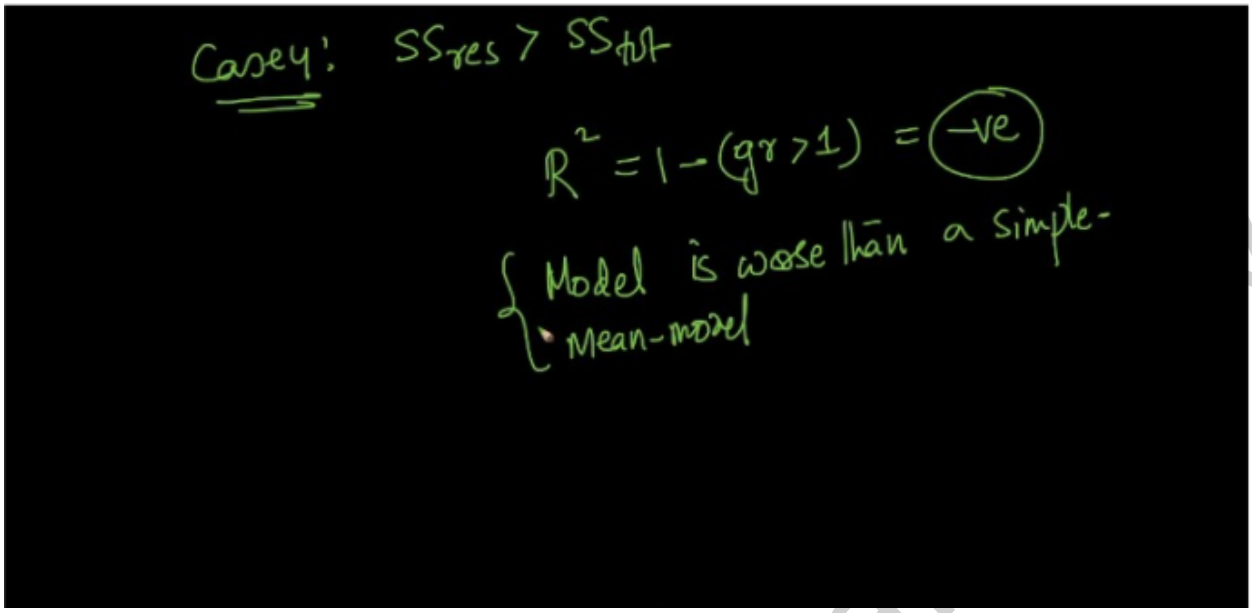
Let's understand this based on cases:

Case 1: SS_{res} = 0, If SS_{res} is 0, this means that our predicted model has made no errors and predicted everything correctly.

In that case R^2= (1 - 0/SS_{total})= 1, this is the best value i.e. R^2= 1

Case 2: SS_{res} < SS_{total}; R^2 lies between 0 and 1.

Case 3: SS_{res} = SS_{total}; This means that our model is the same as the simple mean model. Here R^2= 0.



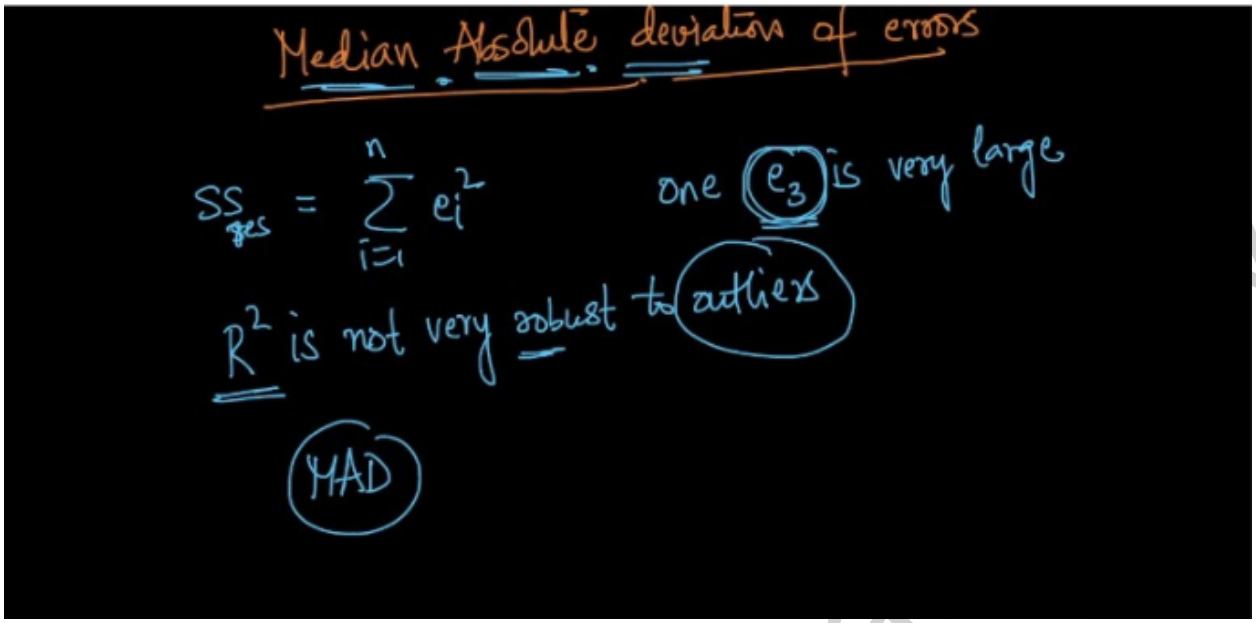
Timestamp 12:07

Case 4: $SS_{res} < SS_{total}$; R^2 = 1 - (something which is greater than 1) = -ve.

So, if R^2 is negative then our model is worse than a simple mean model.

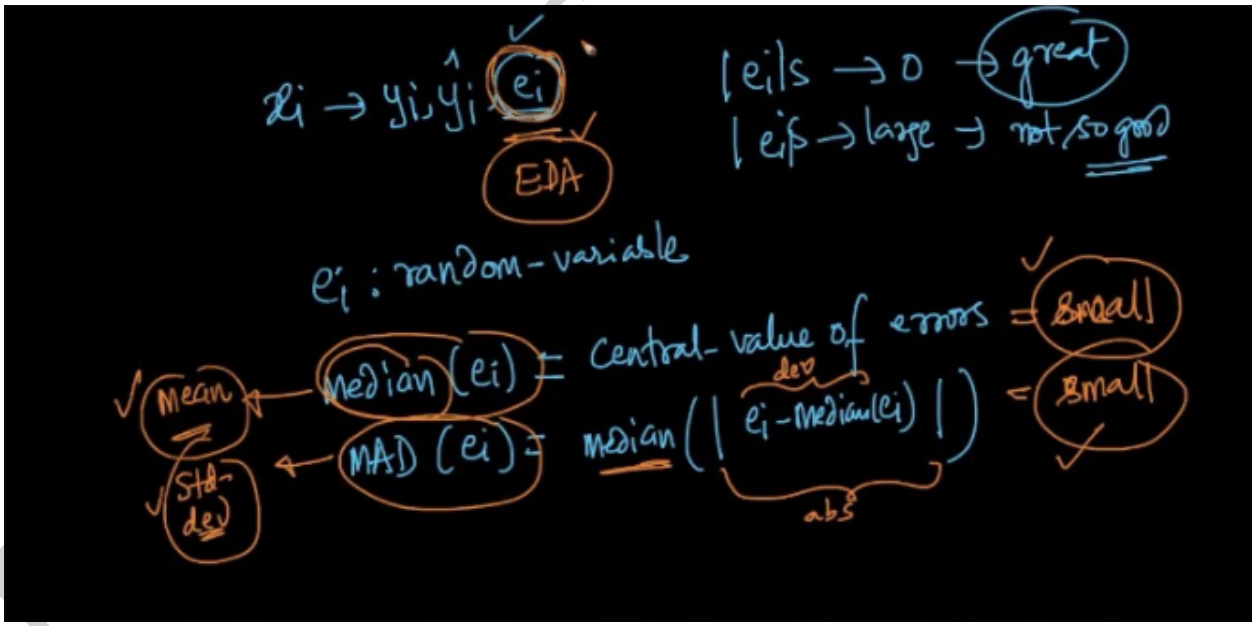
So, based on the values of R^2, we can determine how my model is performing with reference to a simple mean model.
Also there are a lot of different interpretations of coefficients of determination, please read the wikipedia article to know more about it.
https://en.wikipedia.org/wiki/Coefficient_of_determination

30.7 Median absolute deviation (MAD)



Timestamp 1:21

There is one issue with R^2, i.e. it is not very robust to outliers. Suppose there are outliers in my dataset and due to which one my error terms gets very large, this will impact our R^2, giving us a wrong picture.

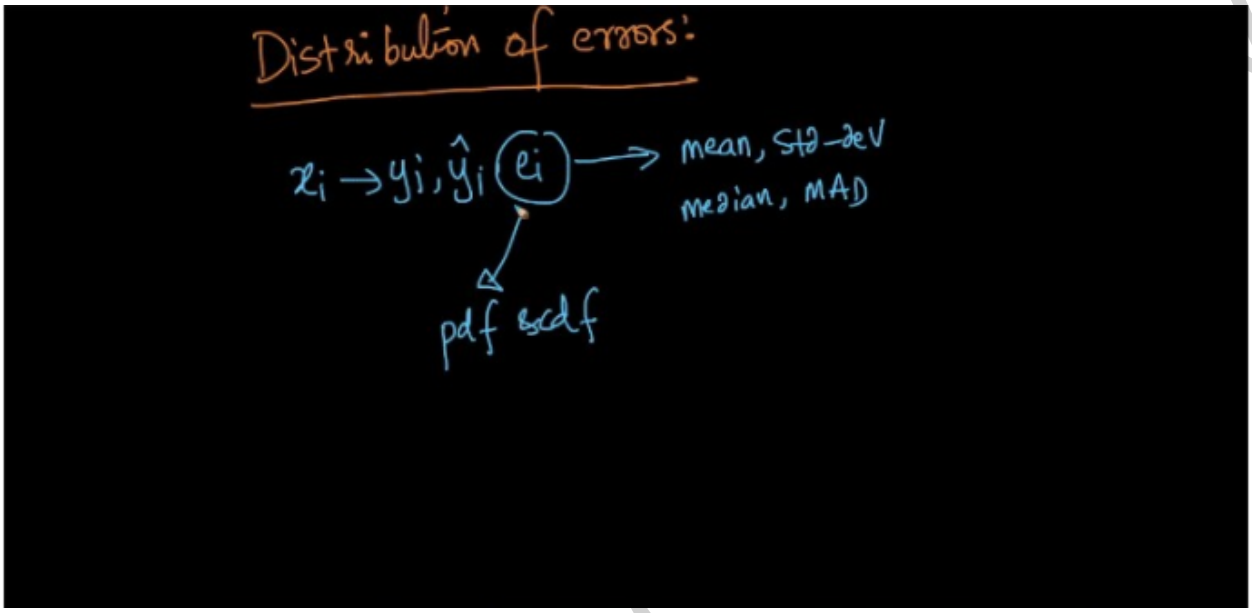


TImestamp 4:15

Now, we saw in earlier chapters that we calculated the mean of the residual errors to get the total error. Mean's are severely affected by outliers. So instead of using means for the central tendency we can use medians. So for total error we calculate the median of total errors. Also if

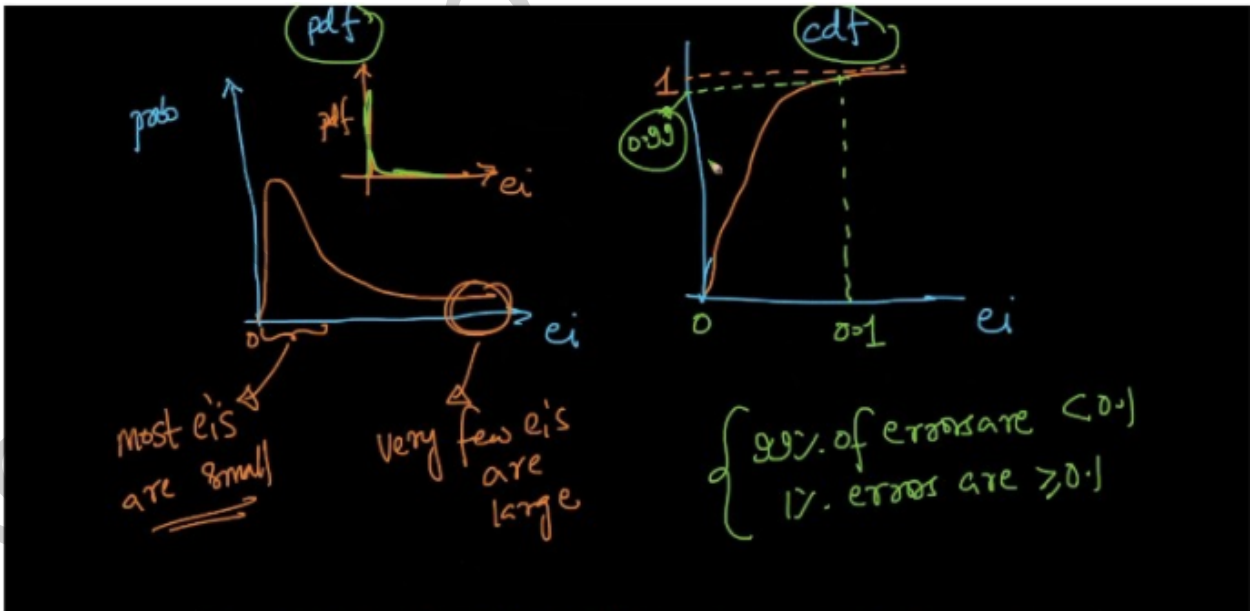
you want to calculate the standard deviation we can calculate the Mean Absolute Deviation, which is defined as the absolute difference between the error terms and their median. Please refer to the image above. If these values are small we can conclude that our model is good.

30.8 Distribution of errors



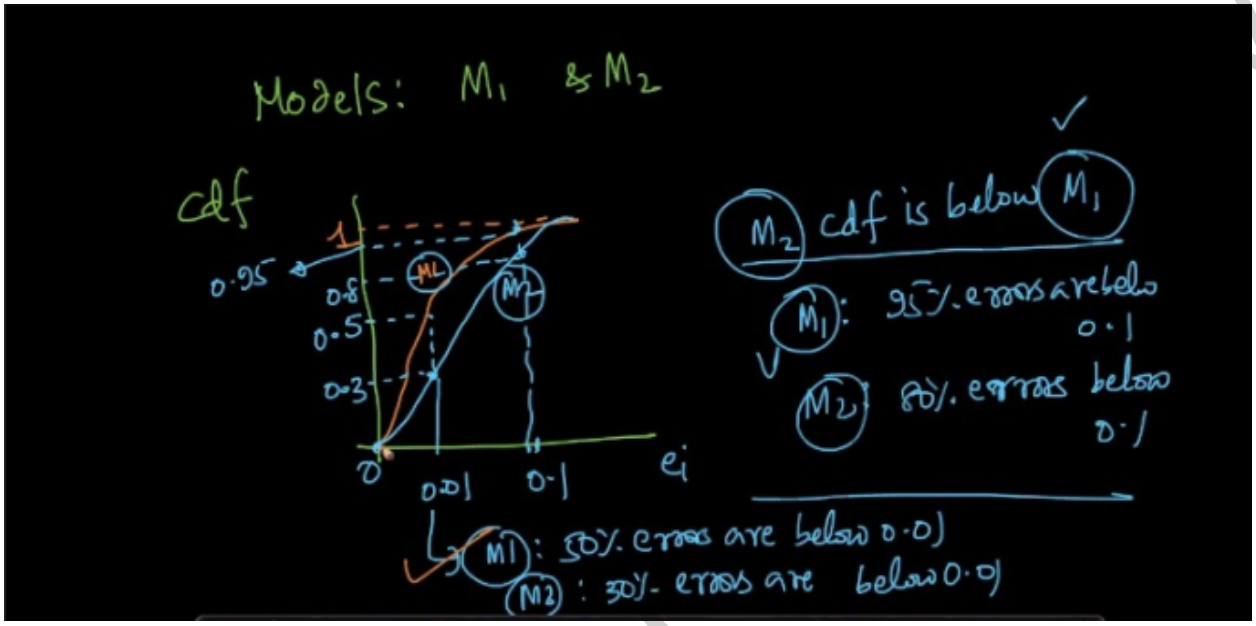
Timestamp 0:33

In addition to using central tendencies to analyze our errors. We can also use PDF's & CDF's.



Timestamp 3:31

Suppose we did a regression and we got our $e_i's$. In the above image we have drawn the distribution of $e_i's$. Now if we look at the graph of PDF, we can conclude that most of $e_i's$ are small and very few of them large. Ideally we want each of them to be as close to 0 as possible. The same can be concluded from the graph of CDF's also.



Timestamp 6:30

We can compare two models using the distribution of their errors. Consider a case where we trained two models M1 & M2. Let's say the distribution of CDF observed by model M1 is shown with the red curve whereas for model M2 with the blue curve, as shown in the above image. Now we can take any value of e_i and observe the CDF's for these curves. Higher value of CDF will mean that more points have error below this point, meaning that particular model is better. We can see other examples from the image.