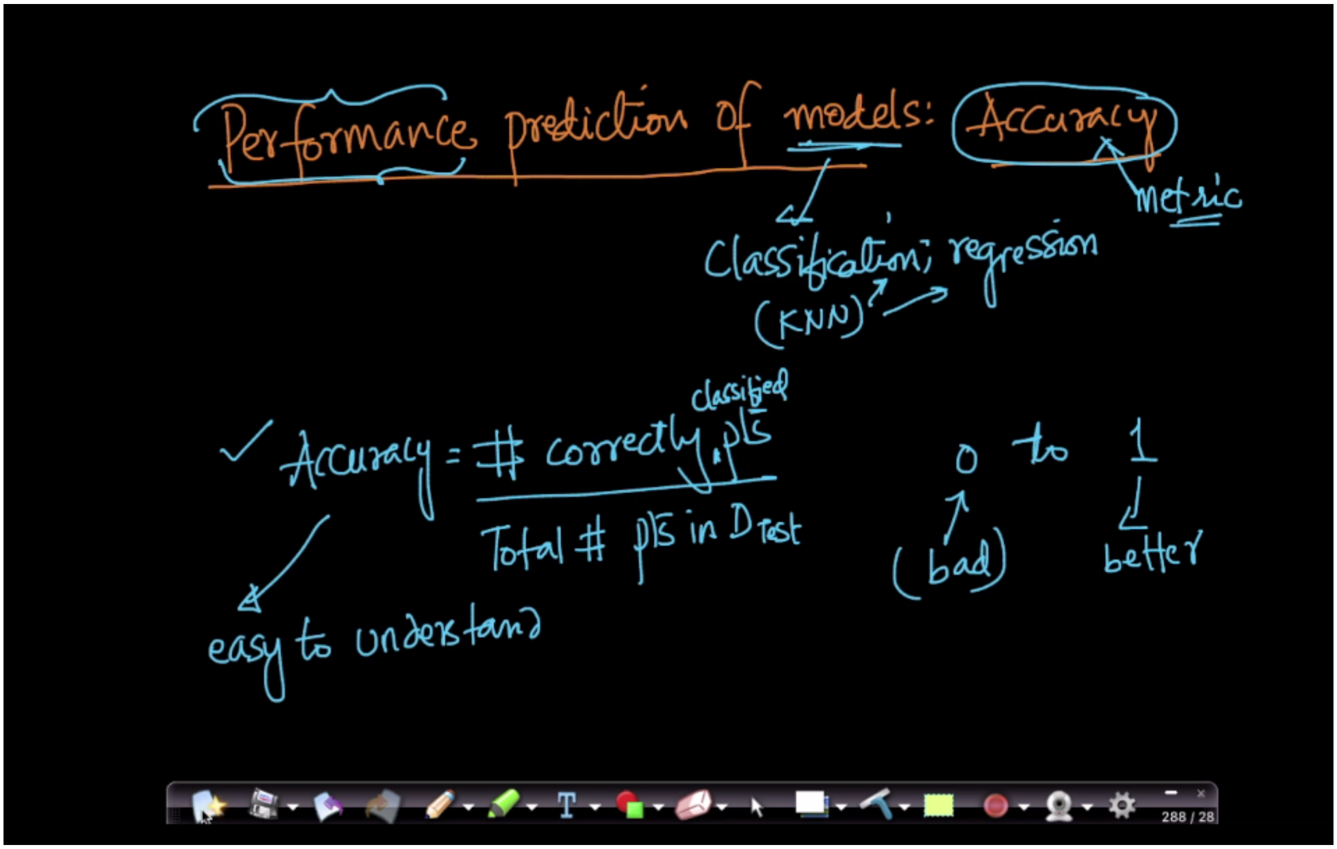
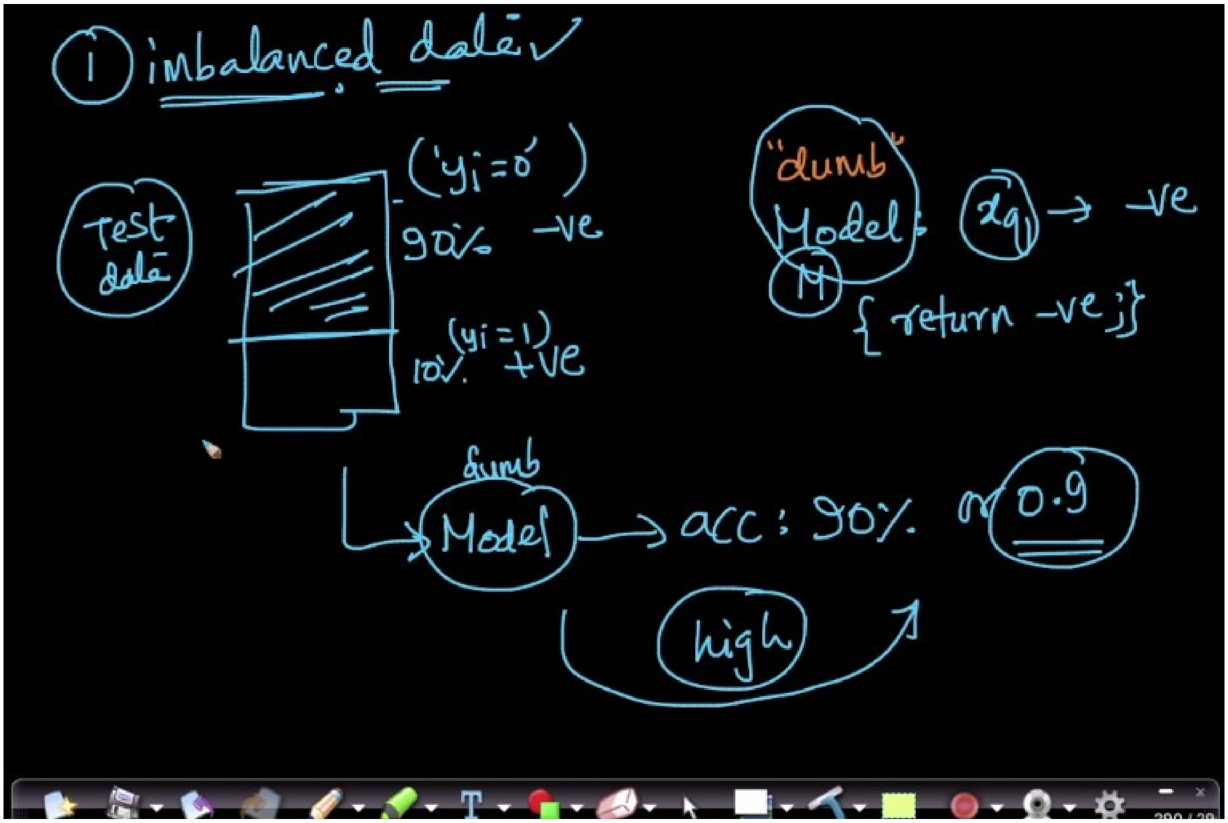
NOTES

Classification accuracy measures:
Performance measure of models: **Accuracy**
accuracy = Number of correctly classified points / Total number of points.

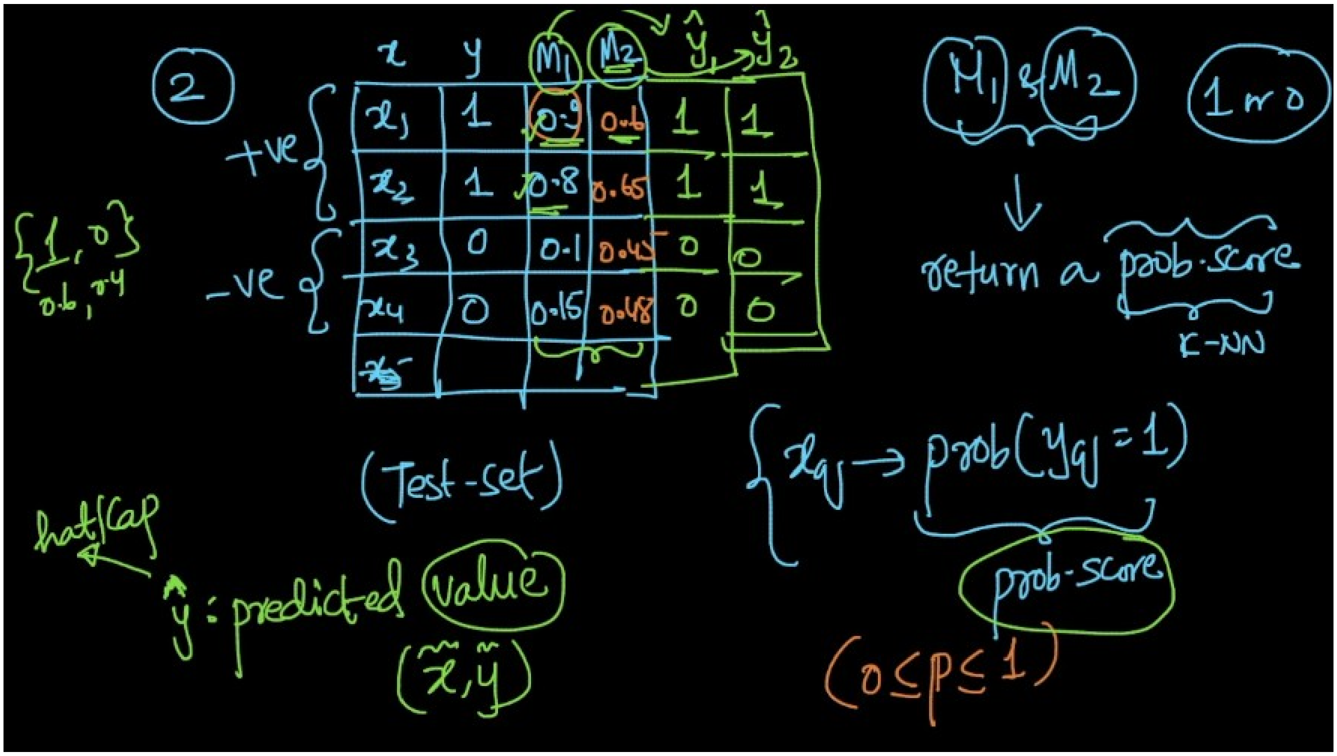

Performance prediction of models: Accuracy
metric
Classification; regression
(KNN)
✓ Accuracy = # correctly classified pts / Total # pts in D_{Test}
easy to understand
0 to 1
(bad)
better
288 / 28

Problems of Accuracy as the measure:

Accuracy is only done on the test data. Accuracy fails in case of imbalanced data.
In case of imbalanced data-set we should never use accuracy as the measure, because a DUMB model can give high accuracy.

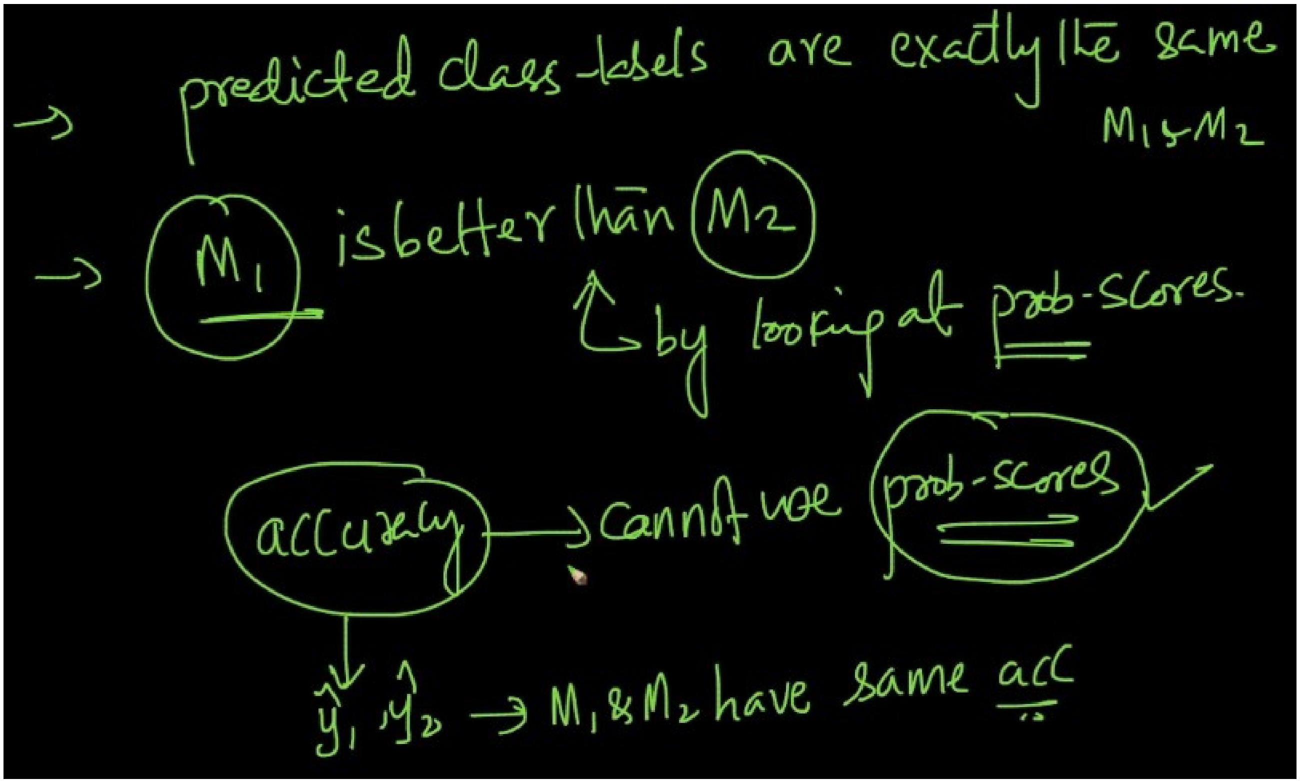


In case of the models output is a probability score, we assume a threshold to be and make the labels above that to be one class, below that is other class.(threshold ~ 0.5).



x	y	M_1	M_2	$\hat{y}_1$	$\hat{y}_2$
x_1	1	0.9	0.6	1	1
x_2	1	0.8	0.65	1	1
x_3	0	0.1	0.45	0	0
x_4	0	0.15	0.48	0	0
~~x_5~~					

The predicted class labels are exactly the same for the two models.(**m**1 and **m**2), though the models are different.



→ predicted class-labels are exactly the same M_1 & M_2

→ M_1 is better than M_2 by looking at prob-scores.

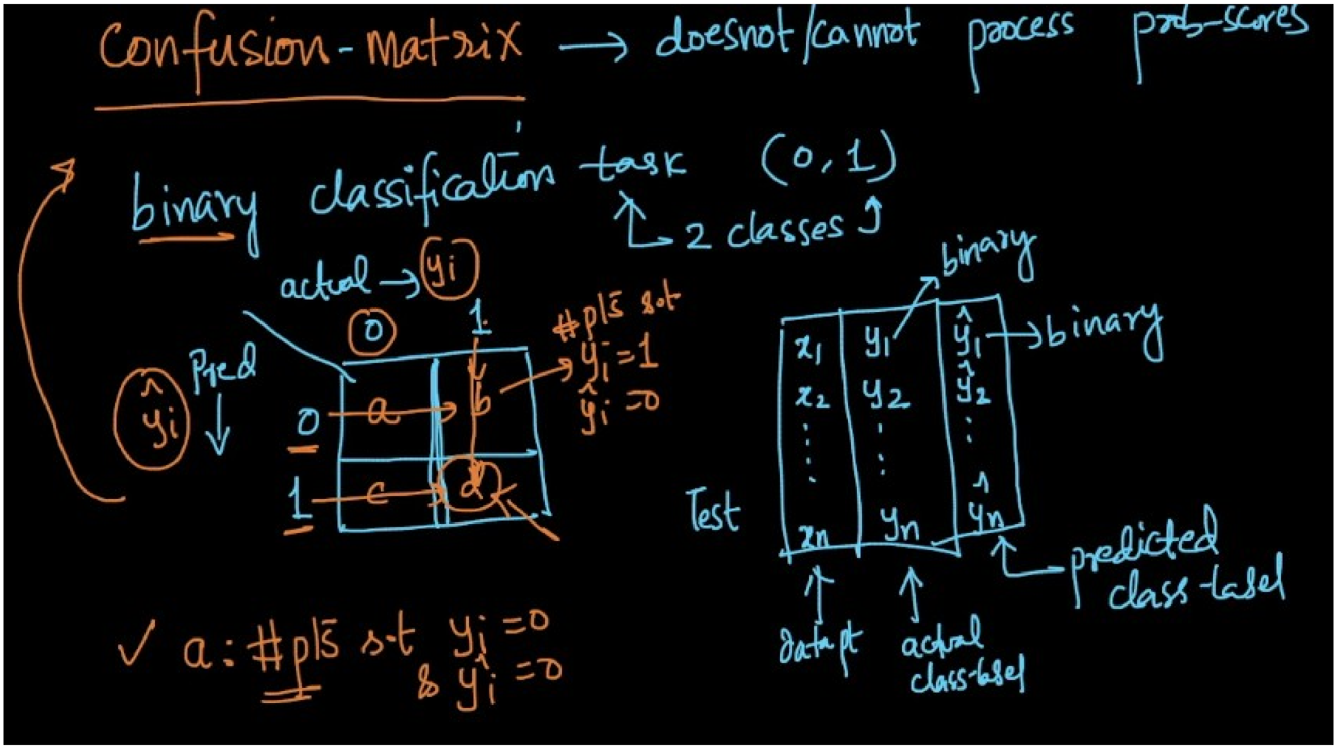
accuracy → cannot use prob-scores

$\hat{y}_1, \hat{y}_2$ → M_1 & M_2 have same acc

These are two major drawbacks for accuracy as the measure of the model.

Confusion – matrix:

It cannot take probability scores. They only take binary values.

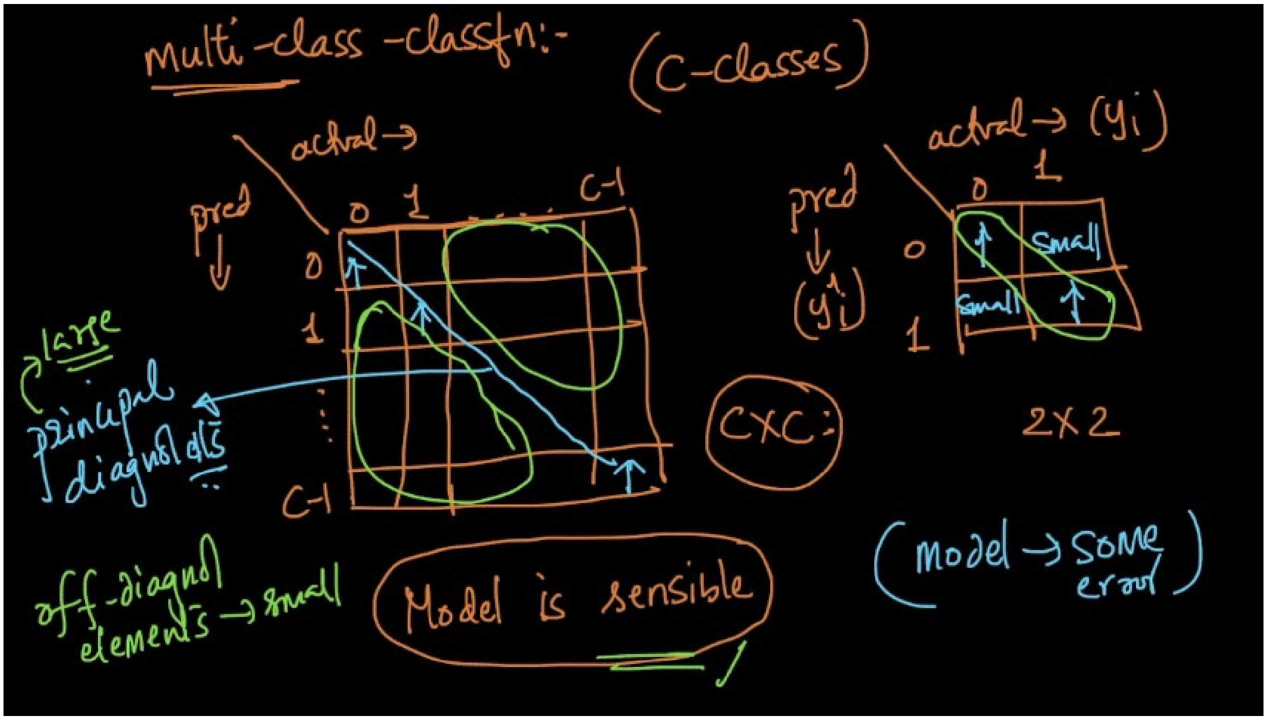


Confusion-Matrix → doesnot/cannot process prob-scores

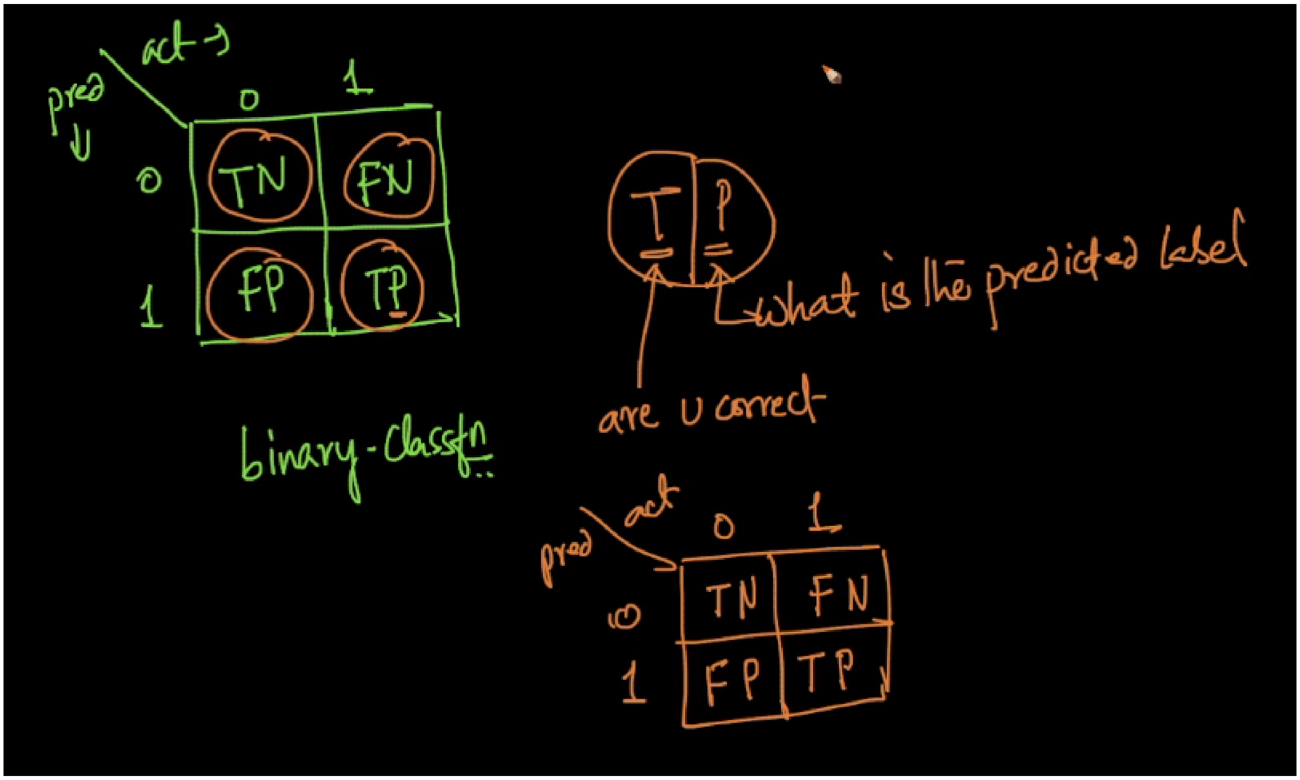
binary classification task (0, 1) ← 2 classes

✓ a: #pts s.t $y_i = 0$ & $\hat{y}_i = 0$

In case of multiple - class classification:
We can draw a matrix for the predicted and actual values.
If the model is sensible, the number of values along the principal diagonal must be more than the off diagonal elements.

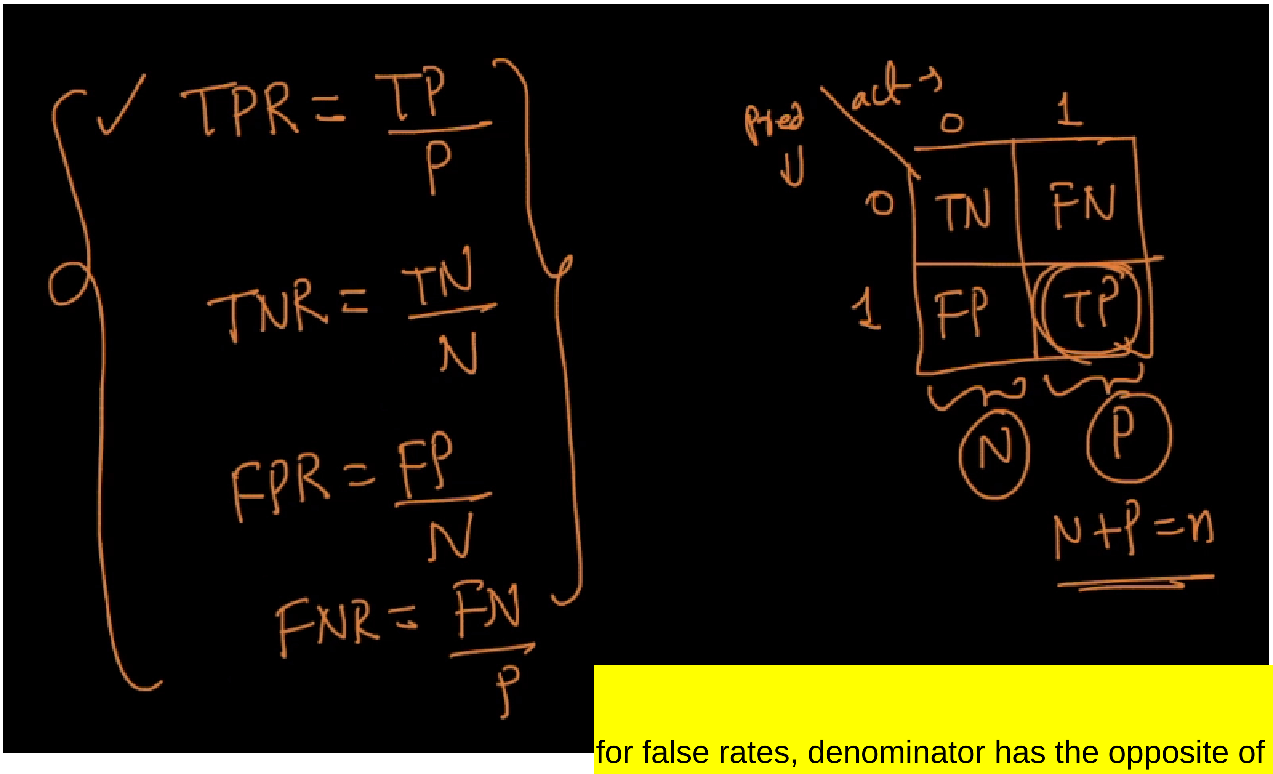


Important to remember the confusion matrix.



pred ↓ \ act →	0	1
0	TN	FN
1	FP	TP

Various confusion matrix measurements:

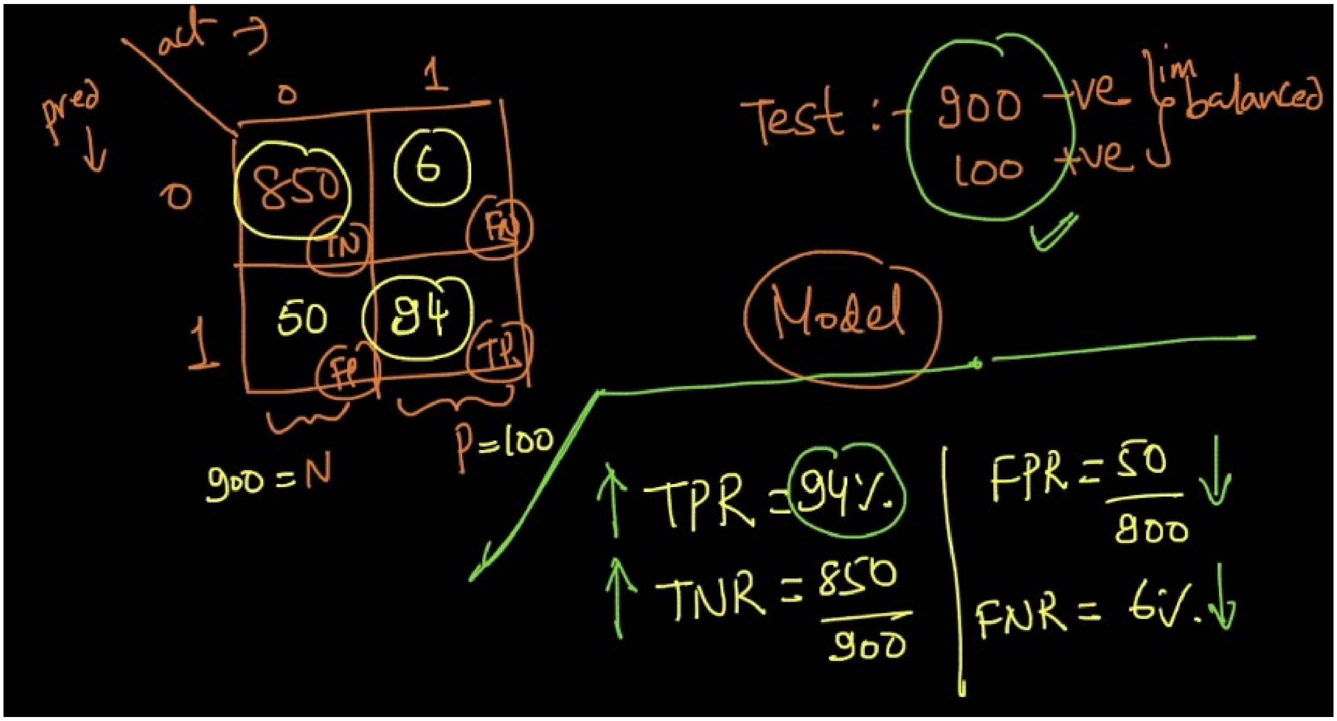


$$\left\{ \begin{aligned} \checkmark\ TPR &= \frac{TP}{P} \\ TNR &= \frac{TN}{N} \\ FPR &= \frac{FP}{N} \\ FNR &= \frac{FN}{P} \end{aligned} \right\}$$

$$\begin{array}{c|cc} & 0 & 1 \\ \hline 0 & TN & FN \\ 1 & FP & TP \\ & \underbrace{\quad}_{N} & \underbrace{\quad}_{P} \end{array}$$

$N + P = n$

for false rates, denominator has the opposite of



$$\begin{array}{c|cc} & 0 & 1 \\ \hline 0 & 850\ (TN) & 6\ (FN) \\ 1 & 50\ (FP) & 94\ (TP) \\ & \underbrace{\quad}_{900 = N} & \underbrace{\quad}_{P = 100} \end{array}$$

Test :- 900 -ve, 100 +ve } im balanced ✓

Model

$\uparrow TPR = 94\%$

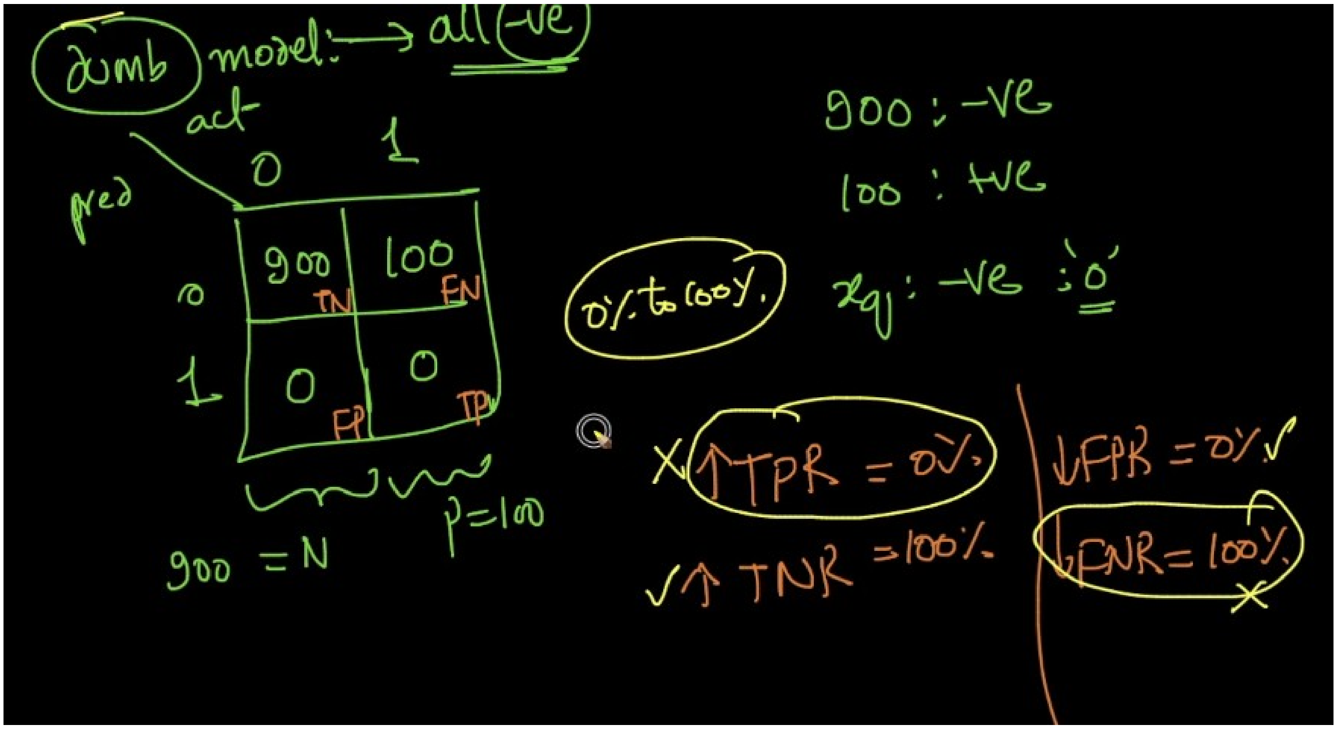
$\uparrow TNR = \frac{850}{900}$

$FPR = \frac{50}{800} \downarrow$

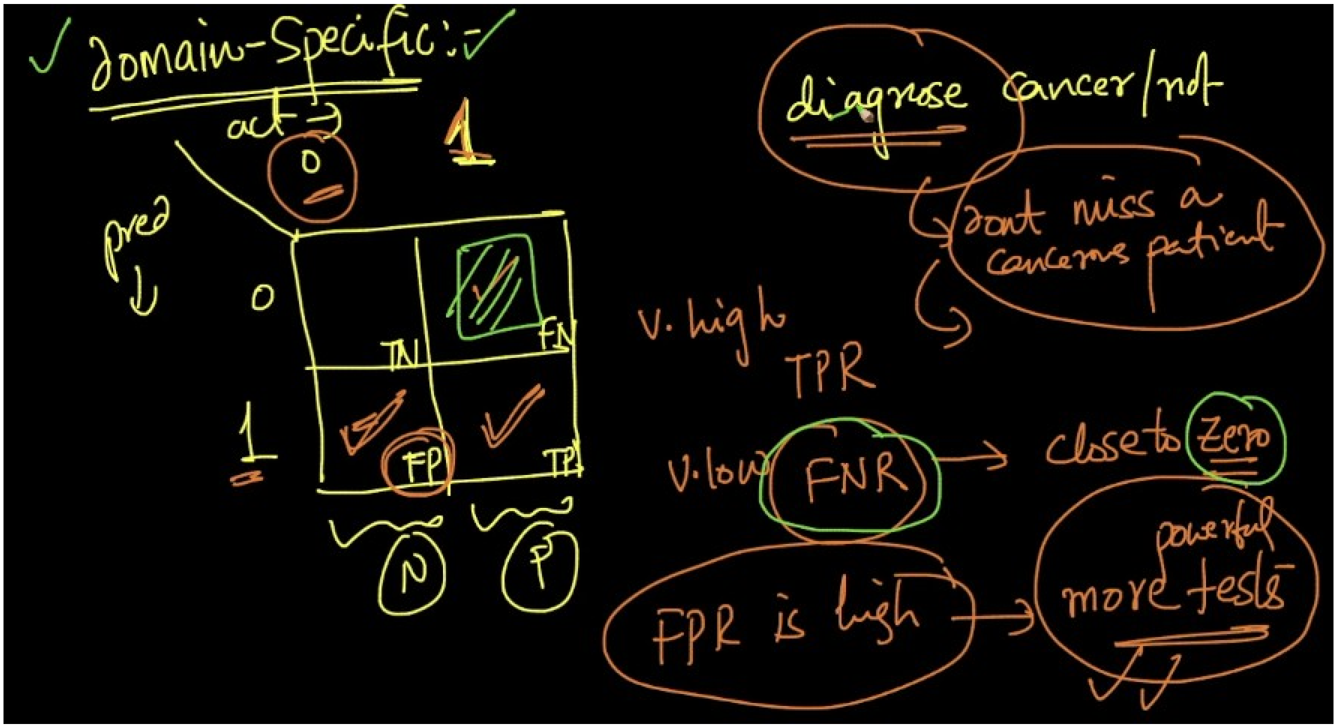
$FNR = 6\% \downarrow$

for a sensible model, true values need to be higher

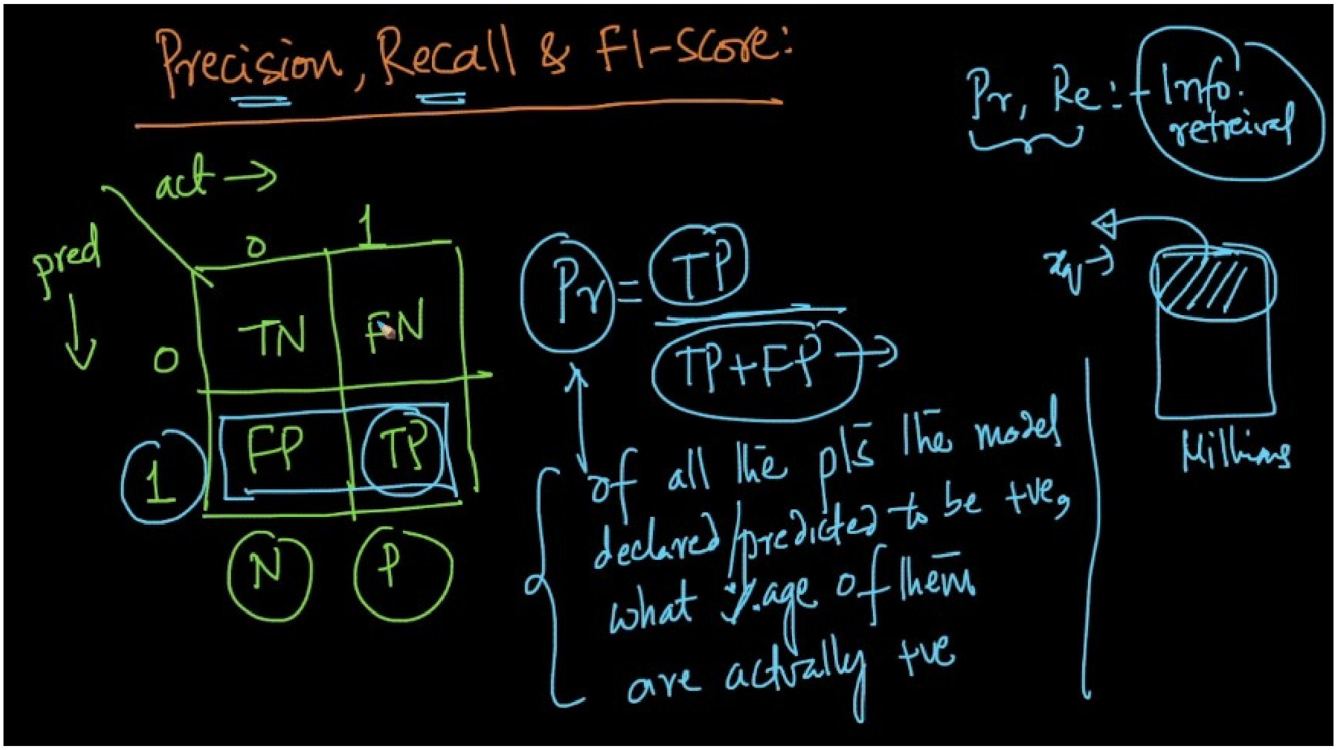
Cases of confusion matrix:



In case of imbalanced data-sets, confusion matrix helps in making good inference from the model. The importance of the various measures of the confusion matrix is more domain specific.


✓ domain-Specific: ✓
act
0
1
pred
0
1
TN
FN
FP
TP
N
P
diagnose cancer/not
dont miss a cancerous patient
v. high TPR
v.low FNR → close to Zero
FPR is high → powerful more tests

Precision, recall and F1 – score:



$$Pr = \frac{TP}{TP+FP}$$

of all the pts the model declared/predicted to be +ve, what %age of them are actually +ve

Precision: Precision only computes the positive class predicted rate.

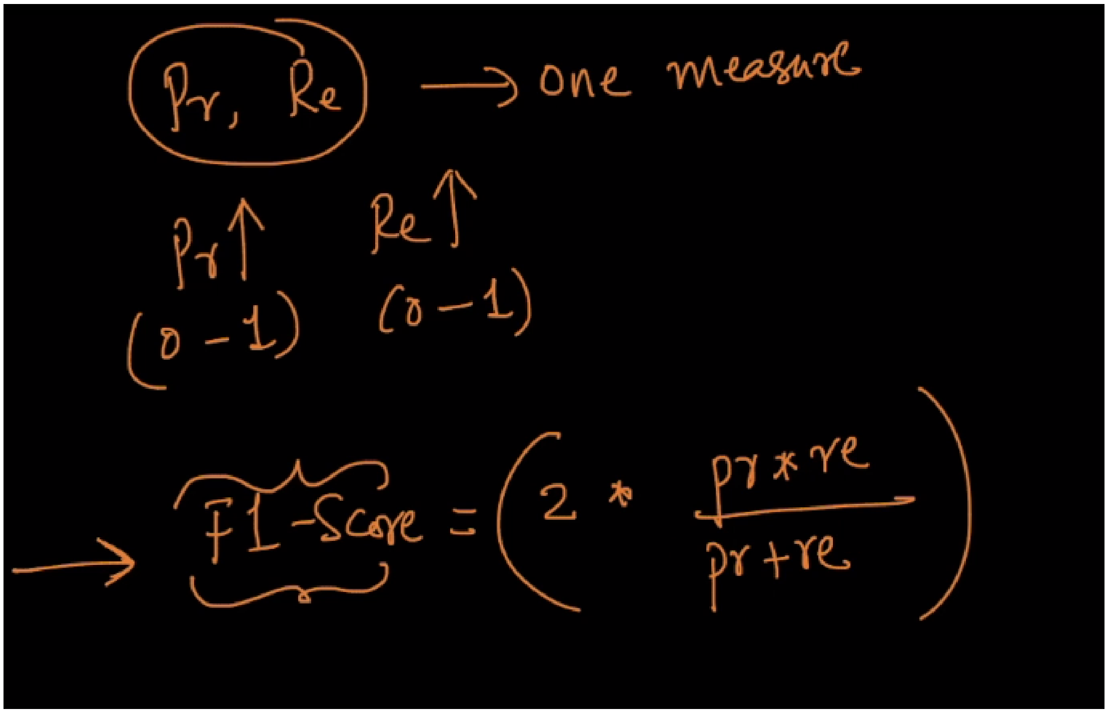
Recall: Recall only computes on actual positive class values rate.

recall = TPR = TF

These are only used, If we want to measure positive class performance.

F1- score:

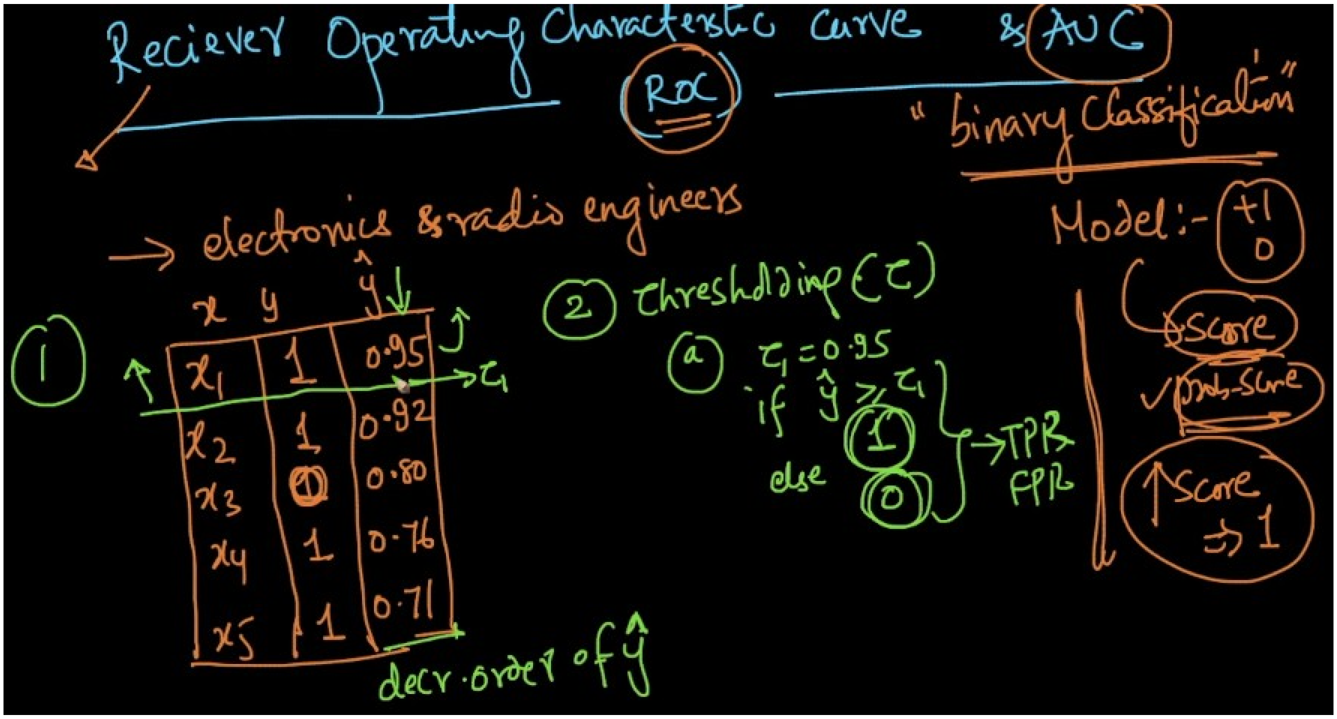
F1 – score is the harmonic mean of the precision and recall.



$$\rightarrow F1\text{-}Score = \left(2 * \frac{pr * re}{pr + re}\right)$$

Precision and recall are more interpret able, than f1 – score.

Receiver Operation characteristic curve and accuracy curve:
We use thresholds as the base in AUC and ROC curve. The values must be sorted in decreasing order of class scores.


Reciever Operating Characteristic Curve (ROC) & AUC

"binary Classification"

→ electronics & radio engineers

Model:- +1 / 0

score

✓ prob-score

↑score ⇒ 1

①

x	y	ŷ
x_1	1	0.95
x_2	1	0.92
x_3	0	0.80
x_4	1	0.76
x_5	1	0.71

decr. order of ŷ

② thresholding (τ)

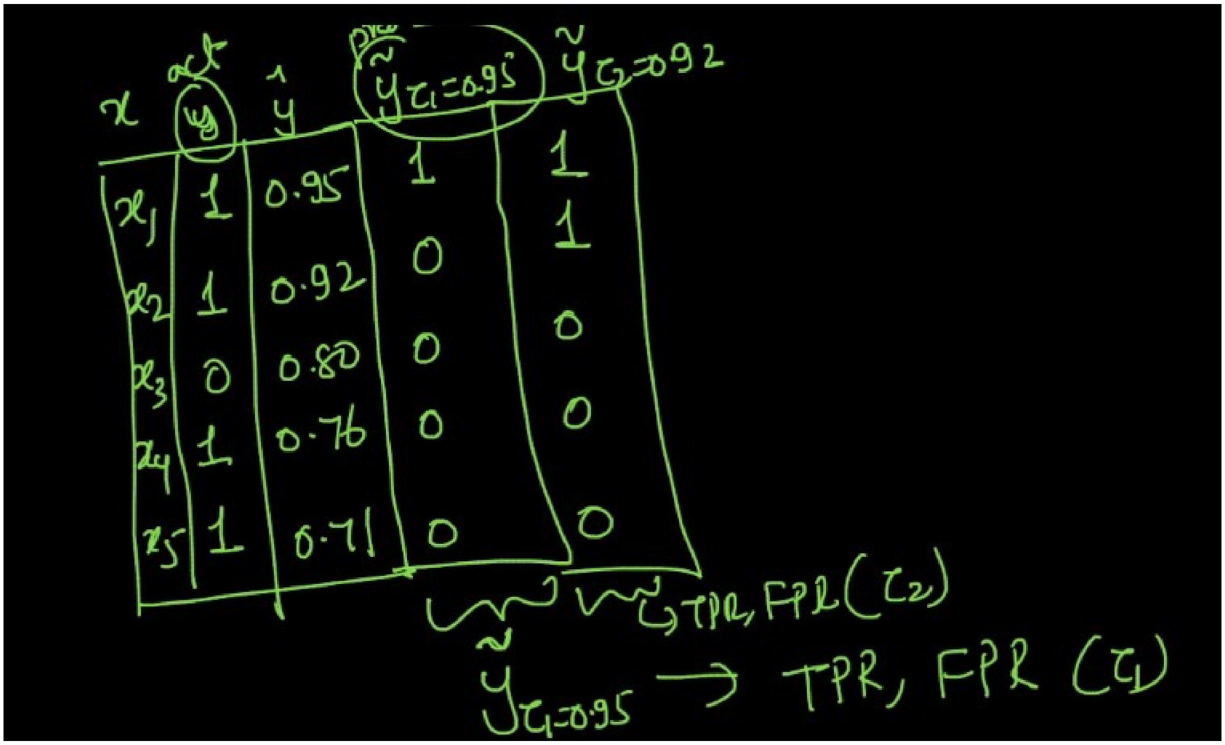
(a) $\tau_1 = 0.95$

if $\hat{y} \geq \tau_1$ → 1

else → 0

→ TPR, FPR

We can have ‘n’ thresholds for ‘n’ points in the data set.

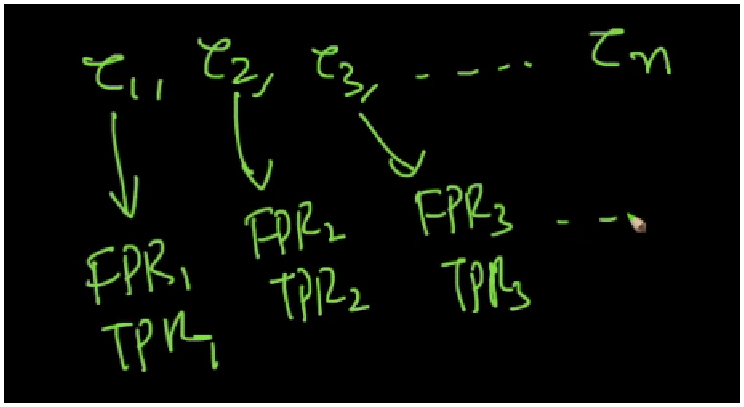

x	act y	ŷ	pred $\tilde{y}_{\tau_1=0.95}$	$\tilde{y}_{\tau_2=0.92}$
x_1	1	0.95	1	1
x_2	1	0.92	0	1
x_3	0	0.80	0	0
x_4	1	0.76	0	0
x_5	1	0.71	0	0

→ TPR, FPR (τ_2)

$\tilde{y}_{\tau_1=0.95}$ → TPR, FPR (τ_1)

we calculate TPR FPR for each threshold

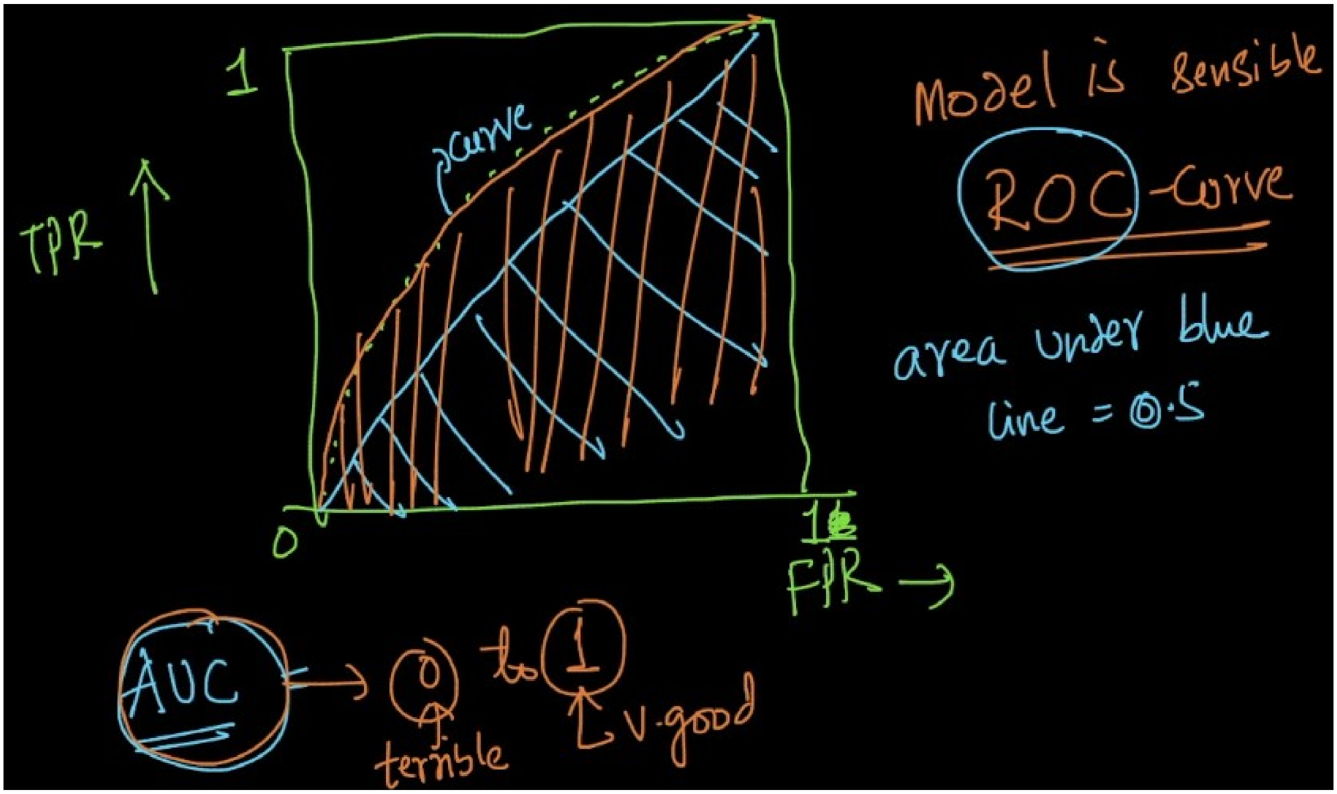
For each threshold we can compute the true positive and false positive rate.



Next, step is to draw the plot of TPR vs. FPR. This curve is called ROC curve.
The blue line breaks the total plot is broken into two halves.

AUC is the area of the curve under the ROC curve. ROC curve can only be used for the binary classification tasks.

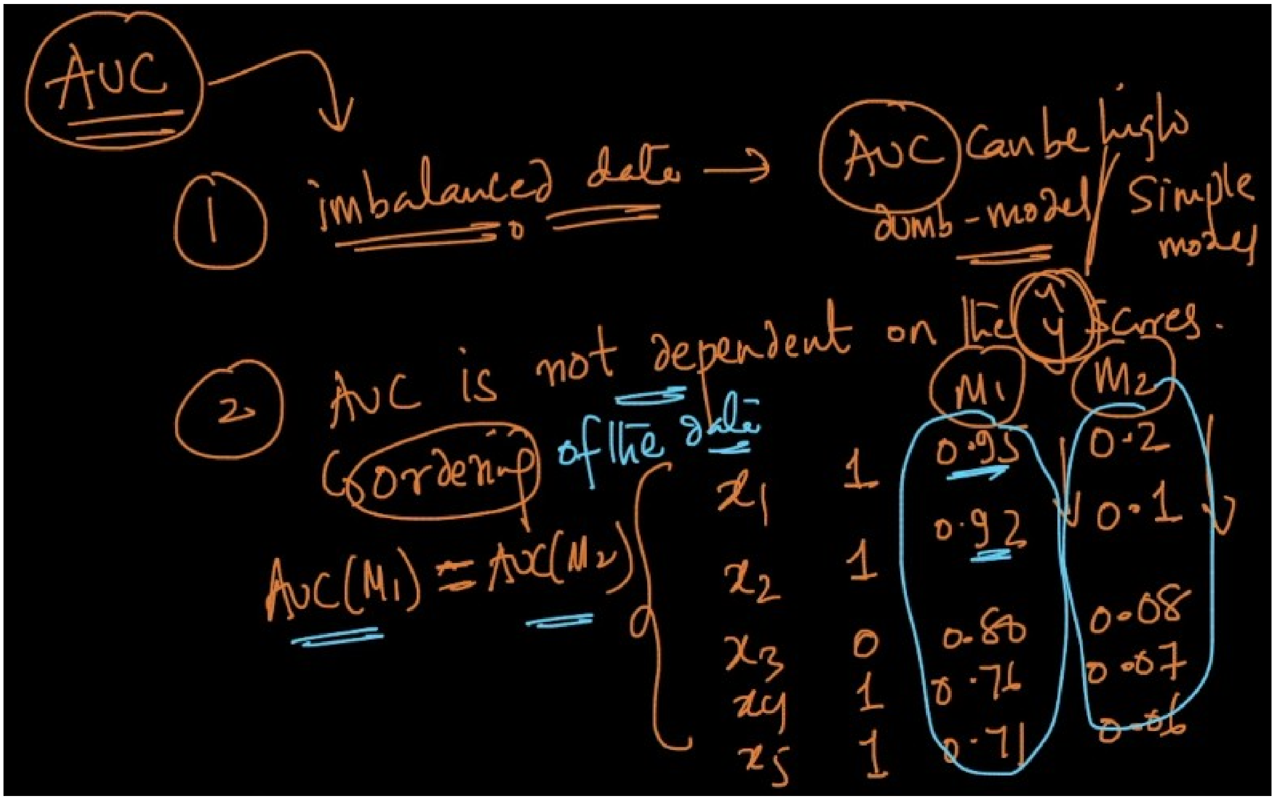
The values of AUC will lie between 0 and 1. The higher the better is the accuracy.



Properties of AUC:

1. In case of imbalanced data the AUC can he high.

2. AUC does not care of the actual value of the accuracy, It cares only the ordering of the class scores.



AUC

① imbalanced data → AUC can be high

dumb-model / Simple model

② AUC is not dependent on the $\hat{y}$ scores.

ordering of the data

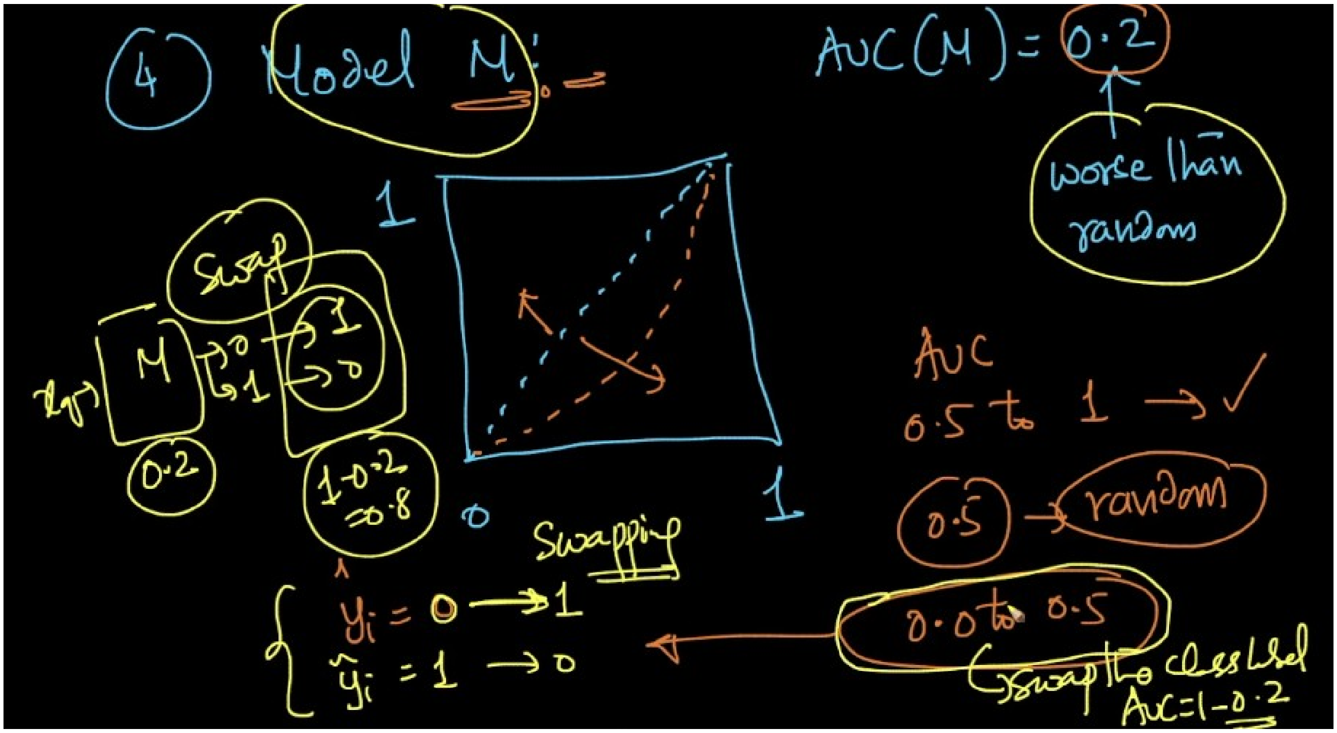
AUC(M_1) = AUC(M_2)

		M_1	M_2
x_1	1	0.95	0.2
x_2	1	0.92	0.1
x_3	0	0.80	0.08
x_4	1	0.76	0.07
x_5	1	0.71	0.06

AUC of several models can be the same.

If the model is random then the AUC will be same as the diagonal line, i.e, 0.5.

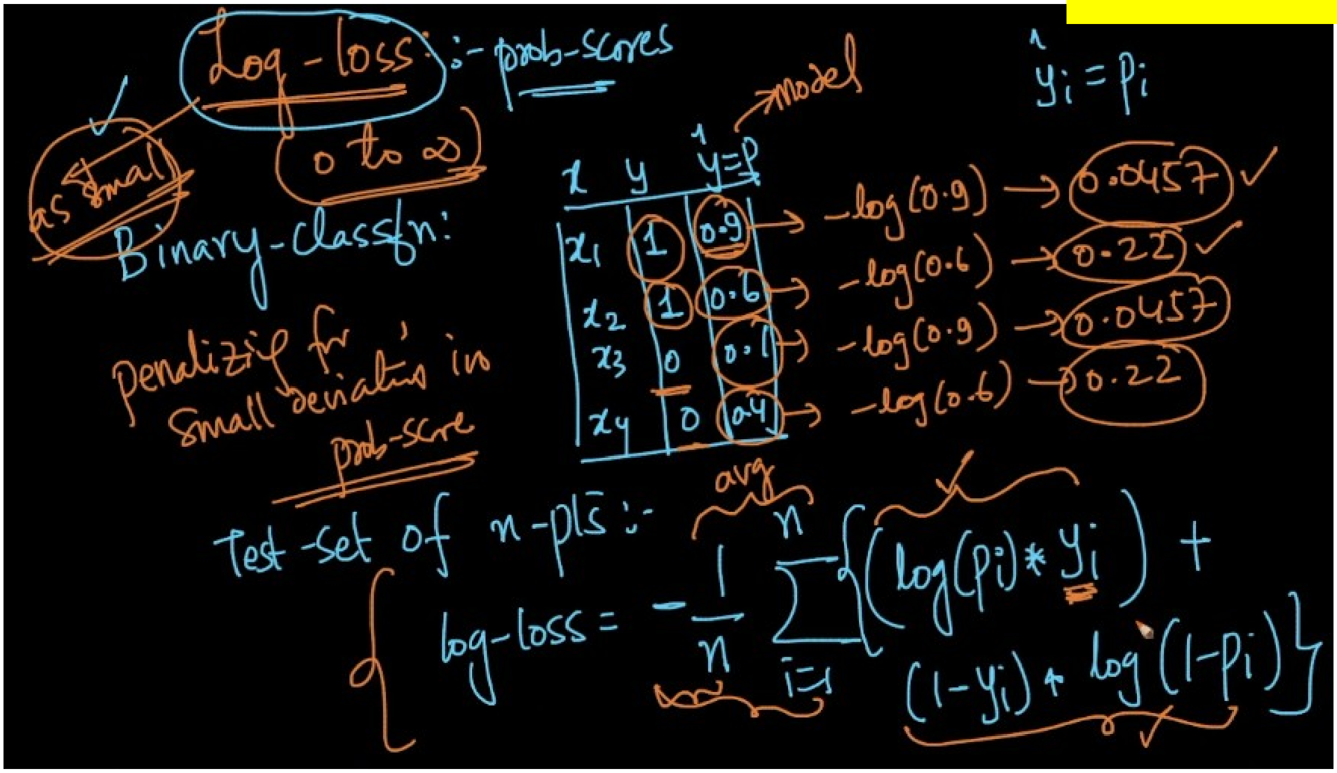
If the model gives the AUC value between 0 and 0.5 then we just swap the class values to get the good model.



Log – Loss: log loss uses probability scores

This model is penalizing small deviations in a probability score. We want the log – loss to be as small as possible. Here we use actual probability score unlike other interpretations.

predicted y values is called called

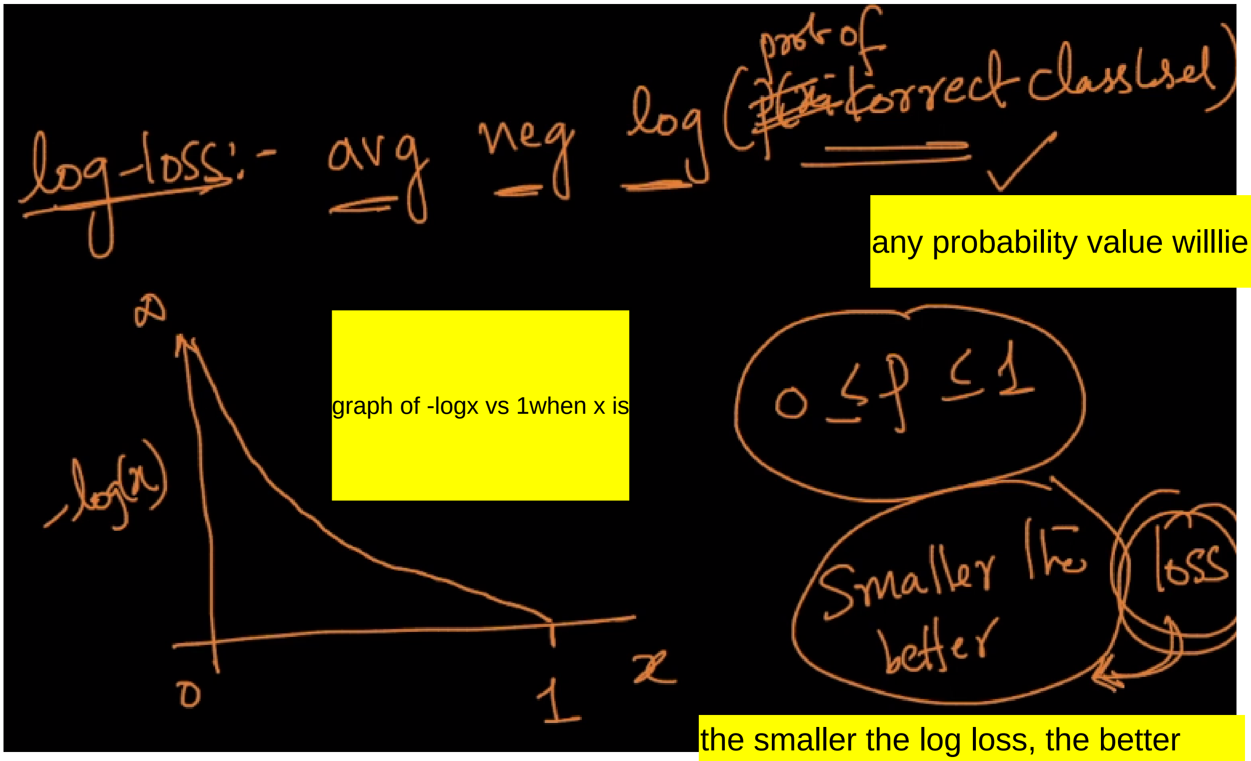


x	y	$\hat{y}=p$		
x_1	1	0.9	$-\log(0.9)$	0.0457 ✓
x_2	1	0.6	$-\log(0.6)$	0.22 ✓
x_3	0	0.1	$-\log(0.9)$	0.0457
x_4	0	0.4	$-\log(0.6)$	0.22

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

we want log loss to be as small as possible, log loss penalises for small deviatior when the class label is 0, the first term will be 0in the 2n

The smaller the better is the model.



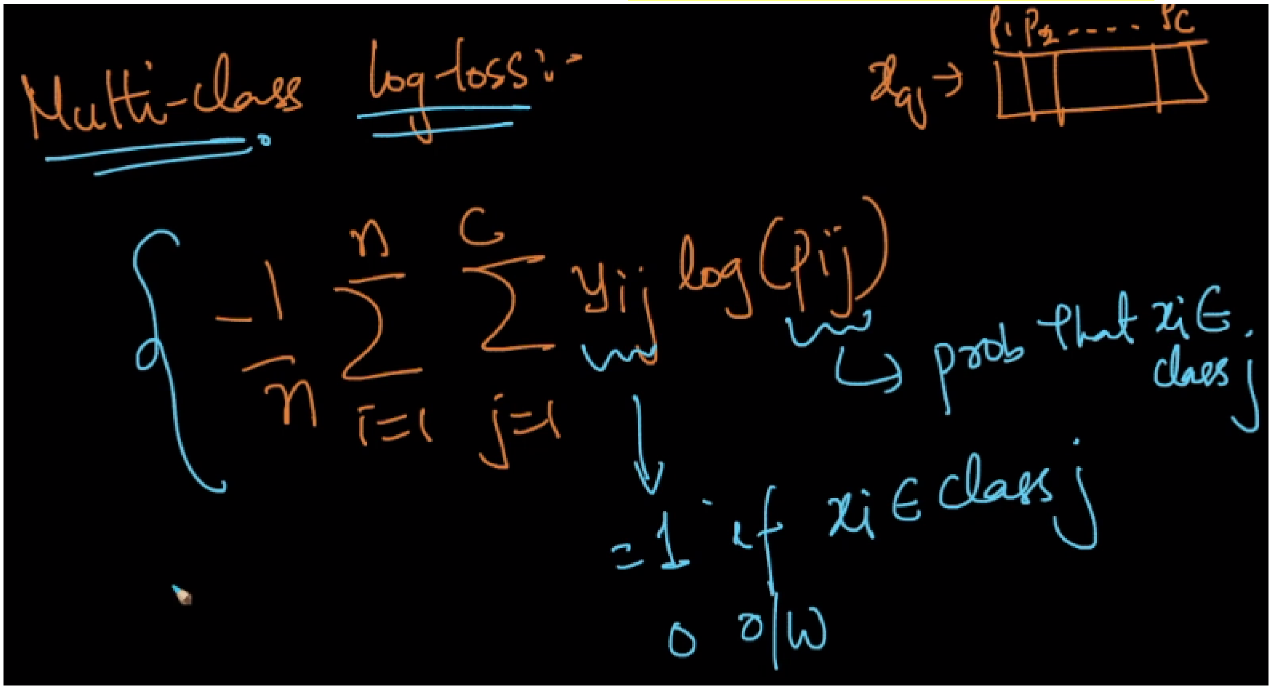
any probability value willlie b

graph of -logx vs 1when x is

the smaller the log loss, the better

Log loss function for multi – class classification:

if there are C classes, there will be C probab

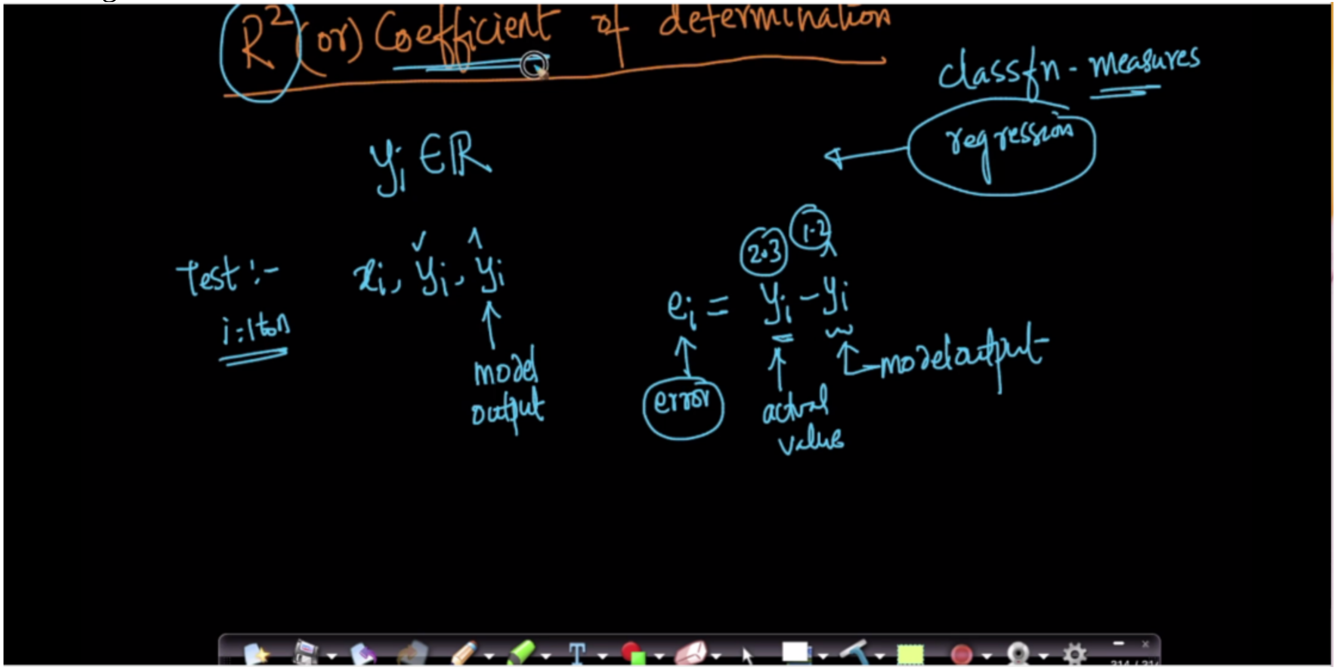

Multi-class log-loss:-
$x_q \rightarrow$ $p_1\, p_2 \ldots p_C$
$-\frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{C} y_{ij}\log(p_{ij})$
$\hookrightarrow$ prob that $x_i \in$ class j
$= 1$ if $x_i \in$ class j
0 o/w

The disadvantage of log – loss is we cannot interpret.

log loss can lie between 0 to infinityit's r

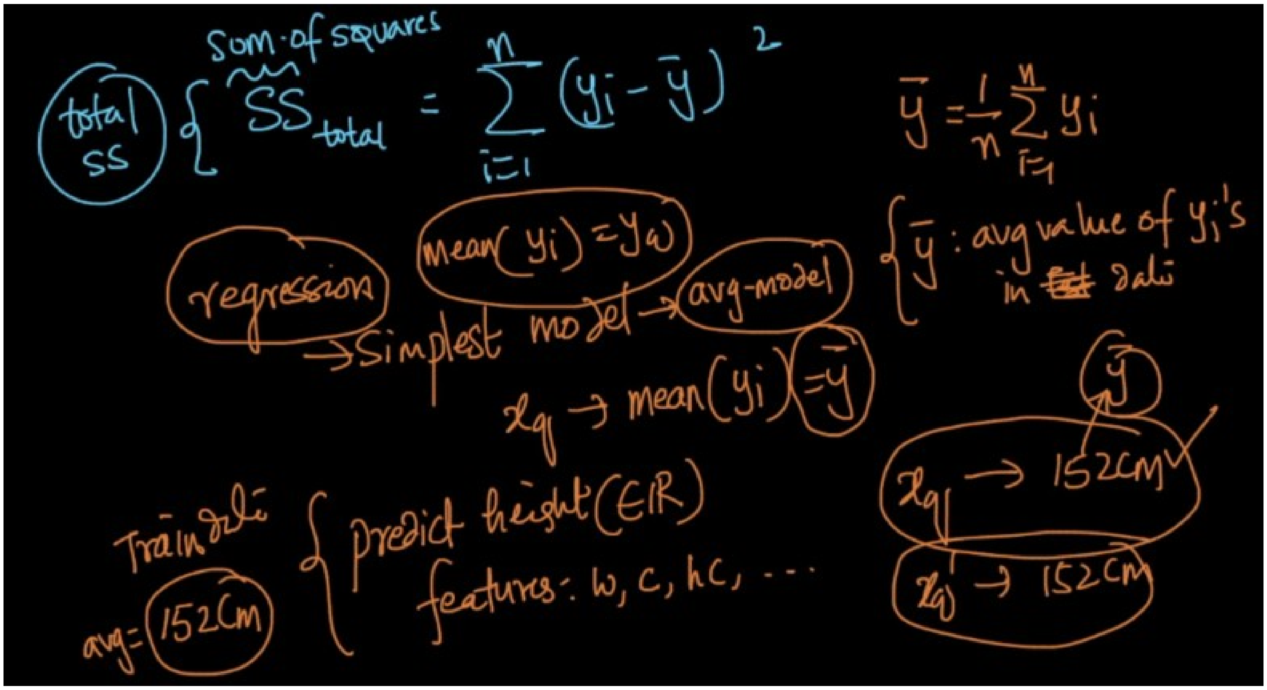
Log – loss is more useful for binary and as well as multi class classification tasks.

Accuracy measures for regression:
Defining the measures:

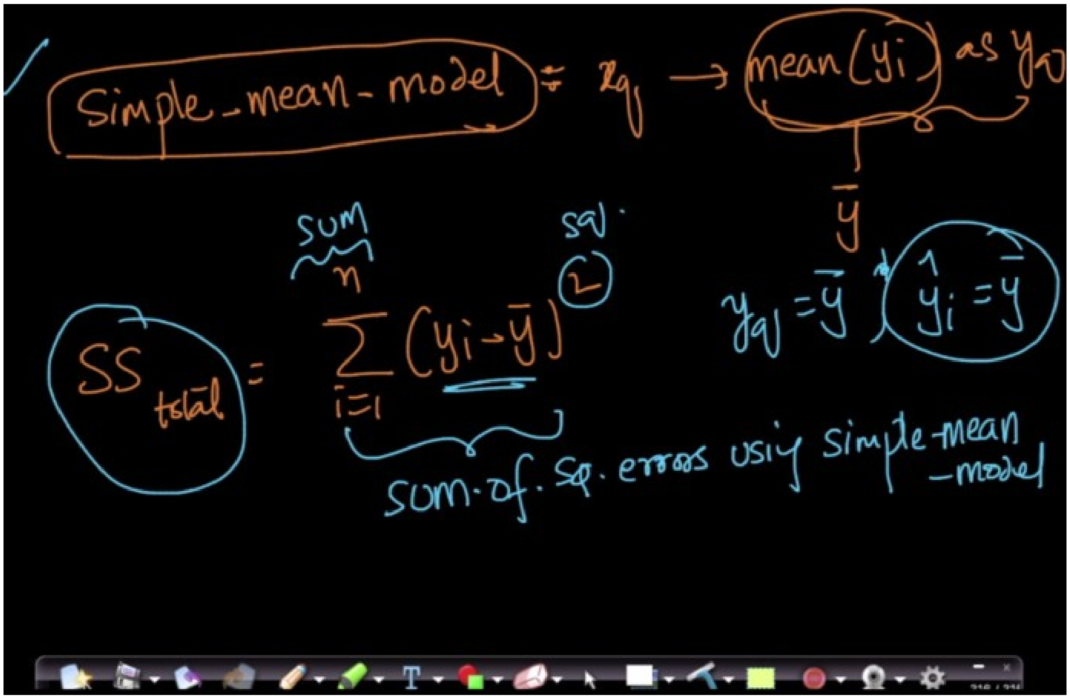


Steps in computing R^2:
Sum of squares:

mean is denoted by both yw and y


Sum of squares
total SS
$SS_{total} = \sum_{i=1}^{n} (y_i - \bar{y})^2$
$\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$
$\bar{y}$: avg value of y_i's in ~~[illegible]~~ data
regression
mean(y_i) = y_w
Simplest model → avg-model
x_q → mean(y_i) = $\bar{y}$
Train data
avg = 152 Cm
predict height ($\in \mathbb{R}$)
features: w, c, hc, ...
x_q → 152 CM
$\bar{y}$
x_q' → 152 Cm

The simplest model that can be constructed is reporting the mean of the data points for a query point xq. Herey(bar) is the average of all the data points in the data set.



Simple-mean-model : $x_q \to \text{mean}(y_i)$ as y_q

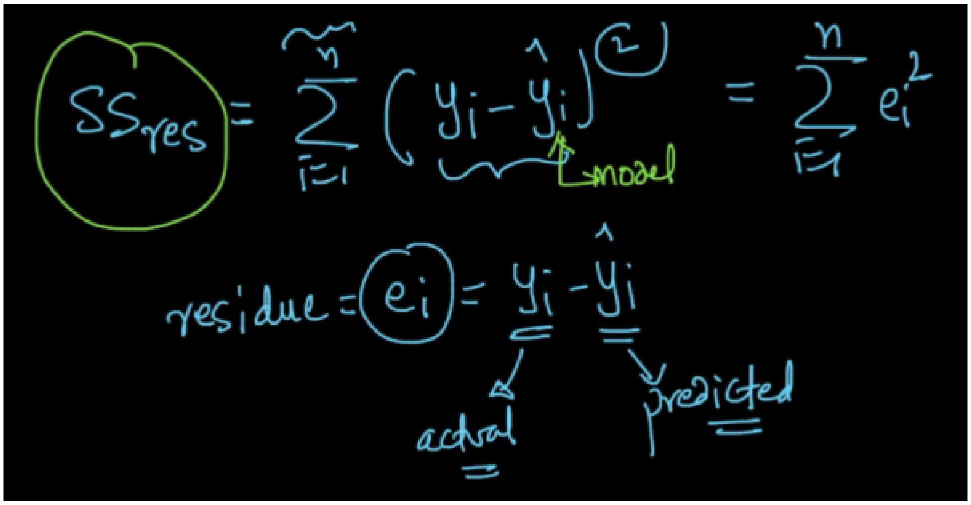
$\bar{y}$

$$SS_{total} = \underbrace{\overset{\text{sum}}{\sum_{i=1}^{n}} (y_i - \bar{y})^{\overset{\text{sq.}}{2}}}_{\text{sum. of. sq. errors usiy simple-mean-model}}$$

$y_q = \bar{y}$; $\hat{y}_i = \bar{y}$

Sum of squared of residuals:

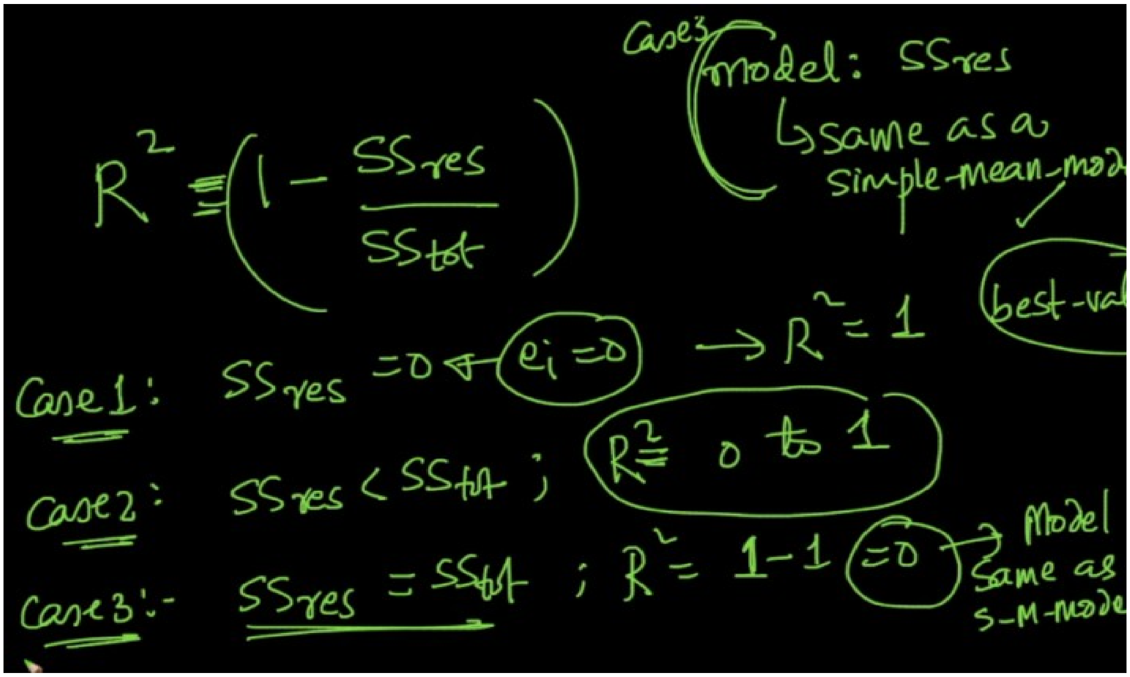
This is called sum of square errors for each point that is being predicted by the model.



$$SS_{res} = \sum_{i=1}^{n} (y_i - \underset{\text{model}}{\hat{y}_i})^2 = \sum_{i=1}^{n} e_i^2$$

$$\text{residue} = e_i = \underbrace{y_i}_{\text{actual}} - \underbrace{\hat{y}_i}_{\text{predicted}}$$

Defining R^2 term:



Case3 model: SS_{res}
↳ same as a simple-mean-mod ✓

$$R^2 = \left(1 - \frac{SS_{res}}{SS_{tot}}\right)$$

best-val

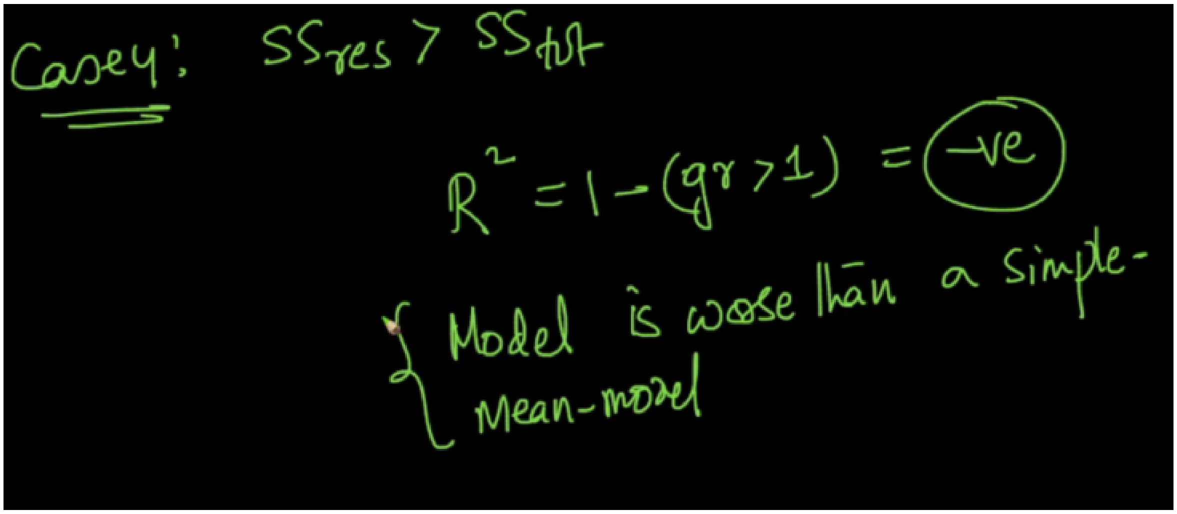
Case1: $SS_{res} = 0 \leftarrow \boxed{e_i = 0} \rightarrow R^2 = 1$

Case2: $SS_{res} < SS_{tot}$; $\boxed{R^2 = 0 \text{ to } 1}$

Case3:- $\underline{SS_{res} = SS_{tot}}$; $R^2 = 1-1 \boxed{=0} \rightarrow$ Model Same as S-M-mode

When the model residue and the total sum of squared is same, then the model is same as simple squared model.

Another case:

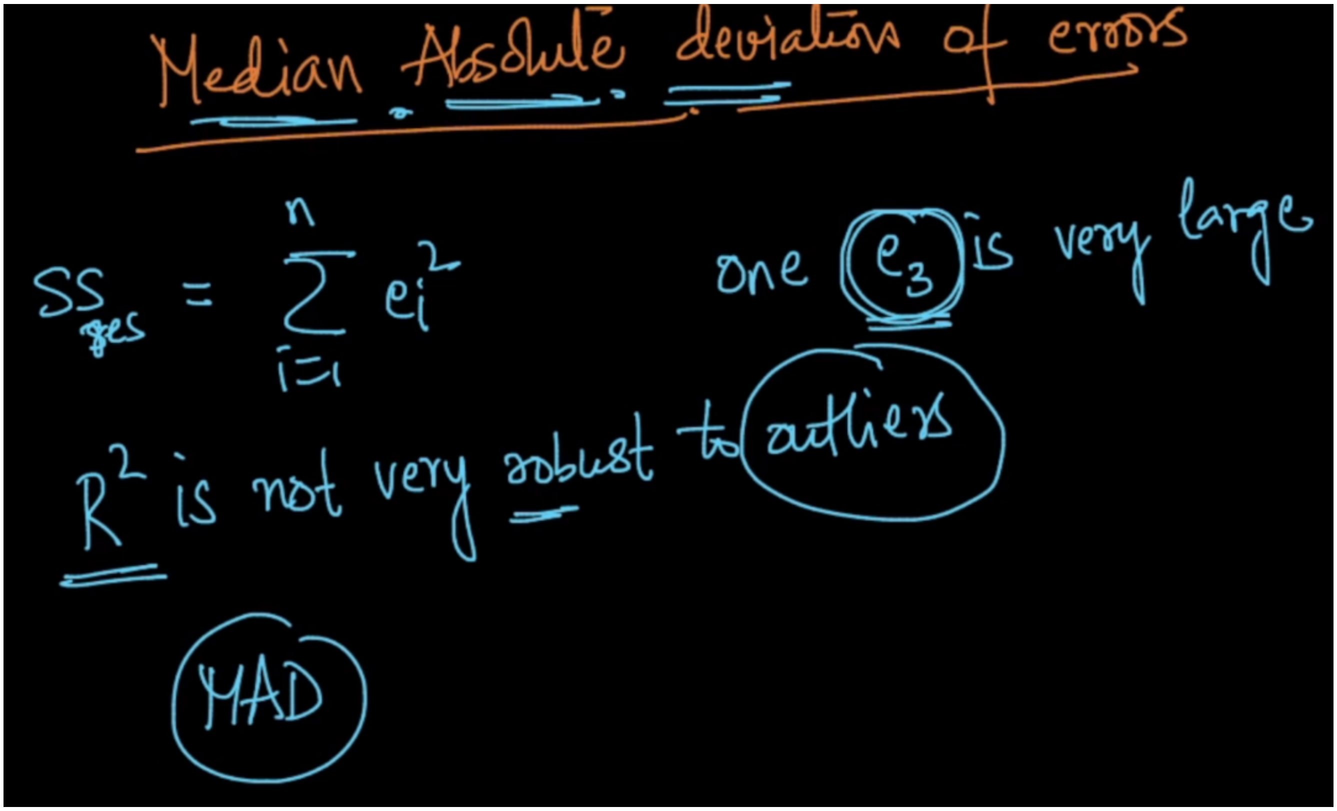


Case4: $SS_{res} > SS_{tot}$

$$R^2 = 1-(gr>1) = \boxed{-ve}$$

{ Model is wose than a simple-Mean-model

When one value is very large then R^2 can go wrong. The sum of residues can go for a toss in case of a large value as an outlier.



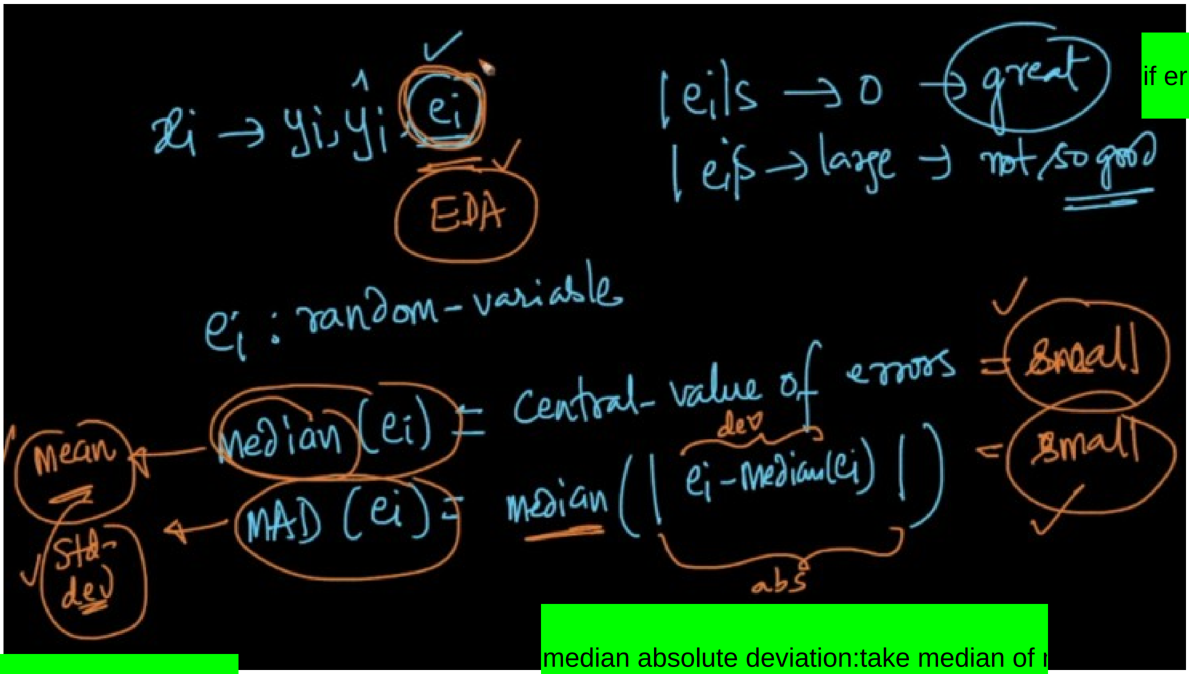
Median Absolute deviation of errors

$SS_{res} = \sum_{i=1}^{n} e_i^2$ one e_3 is very large

R^2 is not very robust to outliers

MAD

That is why we use median absolute deviation as the measure,because it is prone from outliers.



$x_i \to y_i, \hat{y}_i$ e_i ✓

EDA

$|e_i|s \to 0 \to$ great

$|e_i|s \to$ large $\to$ not so good

e_i : random-variable

$\text{Median}(e_i) = \text{central-value of errors} = \text{small}$ ✓

$\text{MAD}(e_i) = \text{median}\Big(\underbrace{\Big|\, \overset{\text{dev}}{e_i - \text{Median}(e_i)} \,\Big|}_{\text{abs}}\Big) = \text{small}$ ✓

Mean ← Median

Std-dev ← MAD

if errors are 0 or clo

median absolute deviation:take median of

median of errors is the central

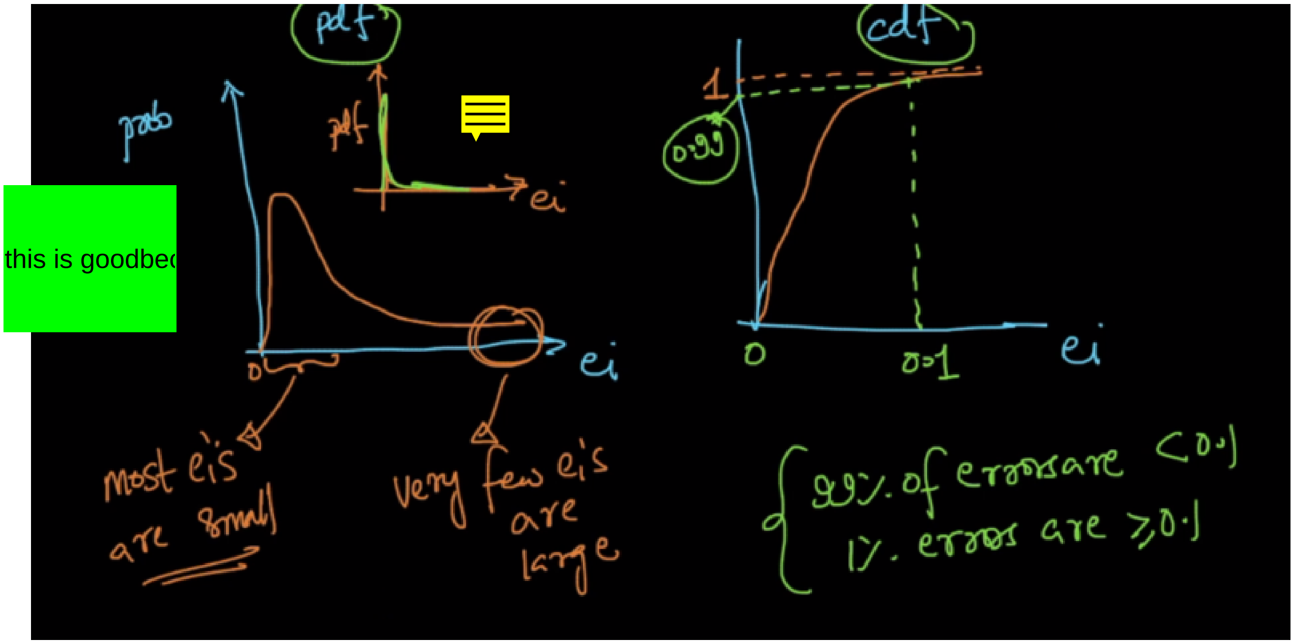
We could use the

Mean – standard deviation.
Median – median absolute deviation. (robust).

For measuring the errors, we can use above methods to infer the distribution of errors(residuals).

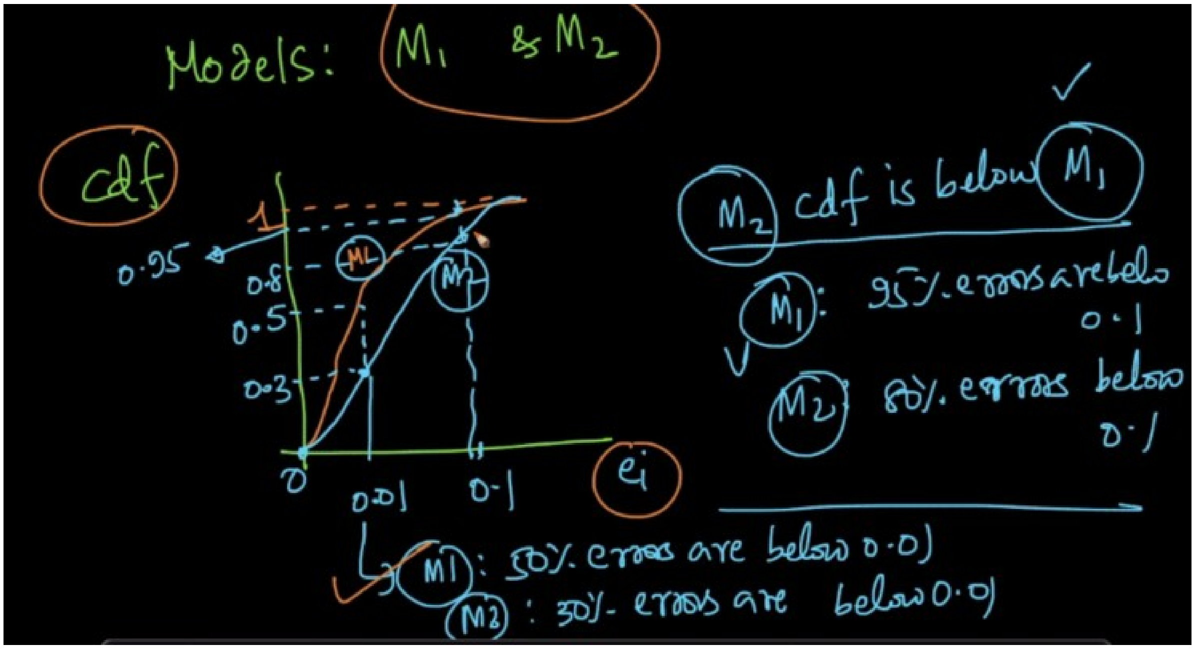
Distribution of errors:
We can use PDF and CDF for knowing the distributions of the errors.

this is goodbe



If all the ei’s are equal to zero then we performed best regressor.
Understanding the distribution of errors is important in case of regression.

Interpreting the errors of the two models:



Models: M1 & M2

M2 cdf is below M1

M1: 95% errors are below 0.1

M2: 80% errors below 0.1

M1: 50% errors are below 0.01

M2: 30% errors are below 0.01

By using CDF plots we can decide which model is better.