Ch5: Performance Measurement of Models

done understood

Classification Metrics

VI. Accuracy

Some classification Metrics:

Accuracy is performed on Test.

$$\text{Accuracy} = \frac{\text{Total \# of pts correctly classified}}{\text{Total \# of pts. in } D_{test}}$$

correct Predictions / No. of D pts. Test

Sensitivity. $\frac{pos}{pos+neg}$

Specificity. $\frac{neg}{pos+neg}$

(Check soln @ end)

$\frac{TP+TN}{D_{test}}$

Accuracy $\in [0, 1]$

worst — best

$\frac{TP}{(TP+FP+TN+FN)}$

Accuracy is easy to understand measure

It is measured on test set usually.

Let test pts = 100

		correct	errors
60	+ve	53 +ve	7 -ve
40	-ve	35 -ve	5 +ve

error = 12

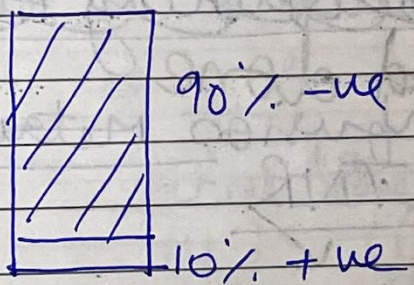
correctly classified = 88

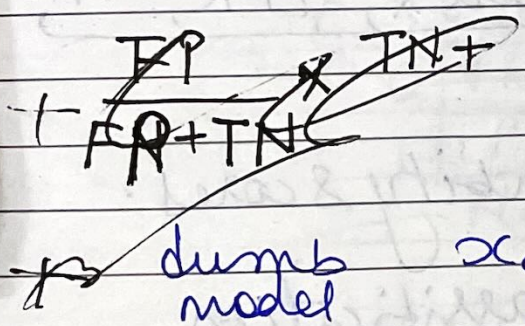
$$\text{Accuracy} = \frac{88}{100} = 0.88$$

Problems with accuracy :-

1) Doesn't work well with imbalanced data.

Suppose,

Test data has 



Accuracy = 0.9.

So, never use accuracy as a performance measure for imbalanced data.

2) Assume M1 & M2 return probability score (Refer KNN to see how we calculate Prob. score)

Test set has :-

x	y	Model 1	Model 2	$\hat{y}_{M_1}$	$\hat{y}_{M_2}$
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	0.16	0.48	0	0

① We can clearly see that M_1 is better than M_2 by looking at the prob. score.

② But since predicted class labels are same, we get same accuracy.

~~So, we can use prob. score in calculating accuracy.~~

understood done ✓

V2) Confusion Matrix, TPR, FPR, FNR, TNR ✓

It doesn't process probability scores.

Consider a binary classification task. 2 classes, either 0 or 1.

Predicted $\hat{y}_i$ \ Actual y_i	0	1
0	a	b
1	c	d

Dtest ↓

x_i	y_i	$\hat{y}_i$
x_1	y_1	$\hat{y}_1$
x_2	y_2	$\hat{y}_2$
⋮	⋮	⋮
x_n	y_n	$\hat{y}_n$
↓ data point	↓ actual class label	↓ predicted class label

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

b: # of pts. s.t. $y_i = 1$ & $\hat{y}_i = 0$

c: # of pts. s.t. $y_i = 0$ & $\hat{y}_i = 1$

d: # of pts. s.t. $y_i = 1$ & $\hat{y}_i = 1$

Actual on top
Predicted on left

This can be extended to multiclass classification.

✓ For a sensible model, principal diagonal elements are higher and off diagonal elements are small.

$\hat{y}_i$ \ y_i	0	1
0	TN	FN
1	FP	TP

eg T P
Actual (y_i) Predicted ($\hat{y}_i$)

N ⇒ P ⇒ (column total)

Negative then Positive row wise
True on diagonal
False on off diagonal.

N = Total Negatives
P = " Positives
n = N + P

✓ TPR = True Positive Rate (Sensitivity) $= \frac{TP}{FN+TP}$ ⇐ $\frac{TP}{P}$ — column total

✓ TNR = True Negative Rate (Specificity) $= \frac{TN}{N}$

✓ FPR = False Positive Rate $= \frac{FP}{N}$

✓ FNR = False Negative Rate $= \frac{FN}{P}$

eg-

$\hat{y}_i$ \ y_i	0	1
0	850	6
1	50	94
	900 (N)	100 (P)

$TPR = 94\% = \frac{94}{100}$ $\quad FPR = \frac{50}{900}$

$TNR = \frac{850}{900}$ $\quad FNR = 6\% = \frac{6}{100}$

So we have high TPR & TNR, which is desired.

Test set has 900 -ve, 100 +ve } imbalanced.

Ideally, TPR ↑↑ FPR ↓
TNR ↑ FNR ↓

Let us now assume, a dumb model which gives o/p label always -ve. Then

$\hat{y}_i$ \ y_i	0	1
0	900 $_{TN}$	100 $_{FN}$
1	0 $_{FP}$	0 $_{TP}$

$TPR = 0\% = \frac{0}{100}$

$FPR = 0\% = \frac{0}{900}$

$TNR = 100\% = \frac{900}{900}$

$FNR = 100\% = \frac{100}{100}$

So, we can defect dumb models by using these 4 metrics together.

Now, which of the 4 metrics is more important?

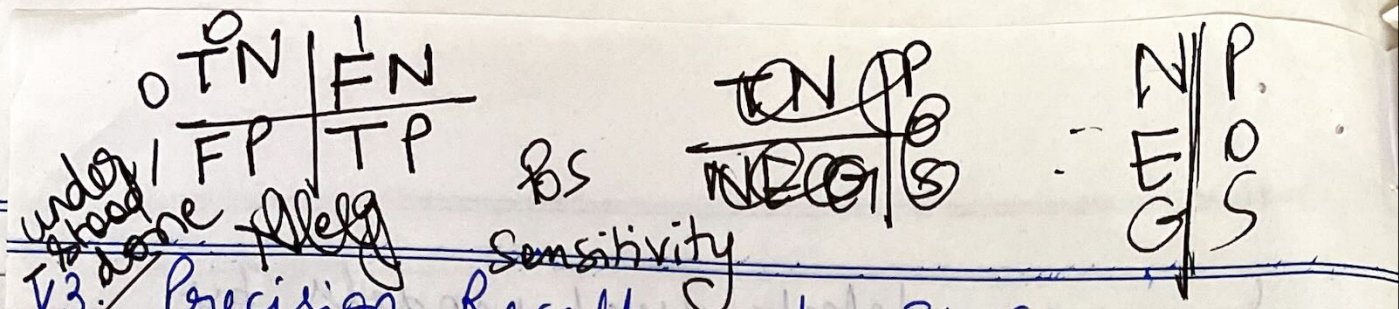
It is domain specific.

Eg- Diagonosing Cancer.

We must have very near or zero FNR as we can't afford to say an achual patient that he doesn't have cancer if he has.

Why, Little high FPR is acceptable as even if we say a non-cancer patient that he has cancer, later through powerful tests that can be classified.

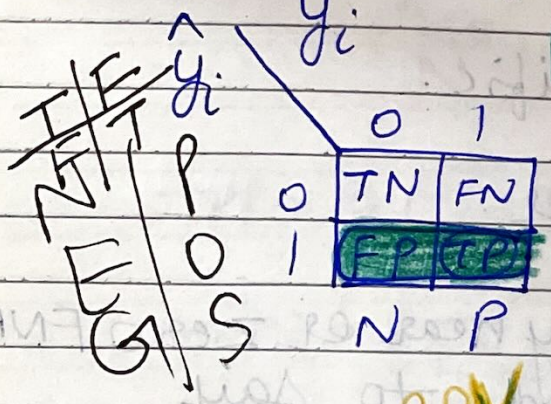
Don't miss a cancer patient!



V3. Precision, Recall and F1-Score

This is based on Information Retrieval.

eg- collecting only 10% out of Millions of datapoints



PGH **Precision** = $\frac{TP}{TP+FP}$ (only in context of positive class)

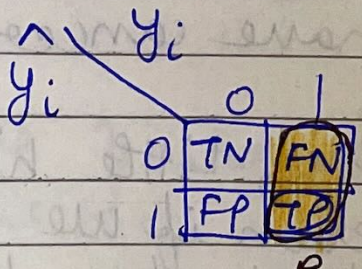
of all the points the model declared to be +ve, what %age of them are actually +ve.

BRV

Recall = **TPR** = $\frac{TP}{P}$ (only in context of positive class)

(Sensitivity)

of all the points which actually belong to class +ve, how many are predicted to be +ve.



We want both Pr & Re to be high.

[0,1] , [0,1]

F1-Score combines them both :-

$$\text{F1-Score} = 2 * \left[\frac{Pre * Rec}{Pre + Rec}\right]$$

* Num
+ Deno

Formula is derived from P.T.O

Specificity = $\frac{\text{True Negative}}{\text{True Neg + False Positive}}$
(Negative)

Harmonic mean of Pr & Re ⇒ F1-score lies b/w 0 & 1

$$\frac{1}{F1} = \frac{1}{2}\left(\frac{1}{Pr} + \frac{1}{Re}\right)$$

V4: Receiver Operating charecterestic Curve ROC & AUC (Area Under this Curve)

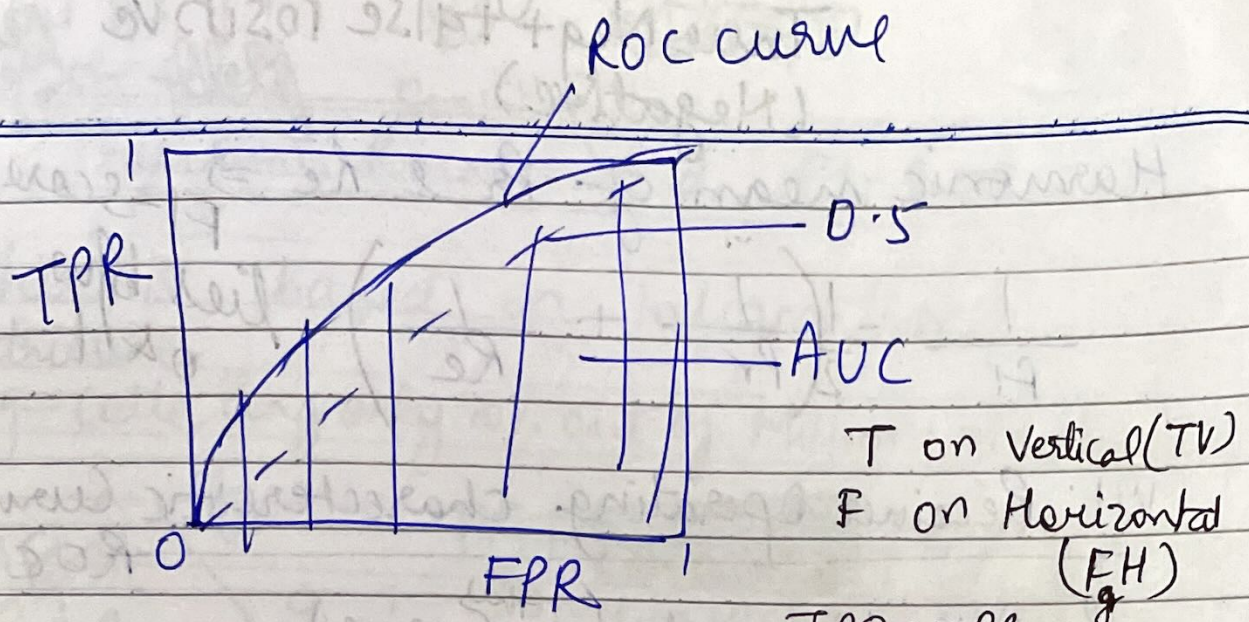
Used for binary classification only. Assume model outputs a score which can be interpreted similar to probability of class '1' score.

x_i	y_i	$\hat{y}_i$
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

Steps for ROC :-
1) Sort entries in decreasing order of $\hat{y}_i$
2) Thresholding (τ)
3) Find TPR & FPR for each τ

So, $\tau_1 = 0.95$ if $\hat{y}_i \geq \tau$ label = 1
else label = 0.

x_i	y_i	$\hat{y}_i$	$\tilde{y}_{\tau_1=0.95}$	$\tilde{y}_{\tau_2=0.85}$
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



T on Vertical (TV)
F on Horizontal (FH)

TPR ; FPR
PR in both.

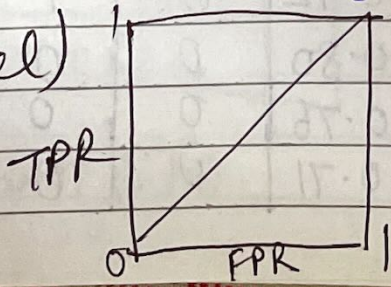
AUC lies b/w [0, 1]
worse
best

for a sensible model this will have ~~have~~ the curve like above.

AUC properties:-

1) For imbalanced data, AUC can be high, ~~and~~ also for dumb/simple model also.
2) AUC is not dependent on the $\hat{y}$ score. It ~~only~~ only depends on ordering of data
3) AUC(random model) = 0.5 which might be 0.2
4) If AUC(M) $\in [0, 0.5]$, then, simply compliment the o/p of the model. (swapping) / change the class label or reverse the class o/p. Simple swapping

AUC(Random Model)



vs: Log loss (uses probability score)

On Test set of n- pts :-

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

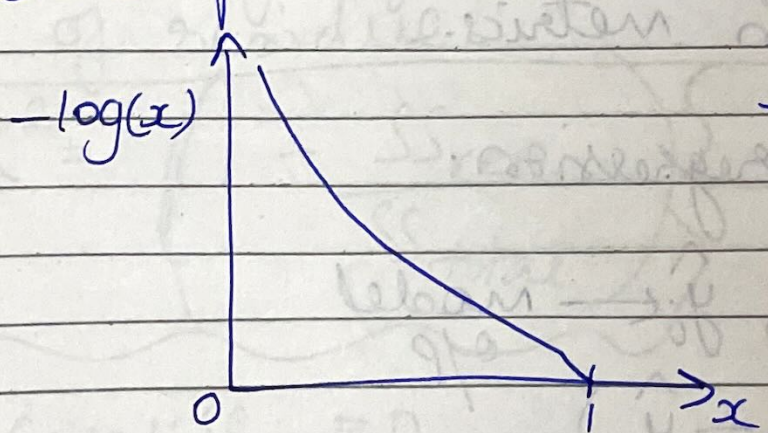
either of the term is valid, the other becomes zero

negative of average of log loss

Average of negative of log of probabilities of correct class label.

All log loss, smaller the better.
" prob., higher " "

Log loss is very powerful because unlike all previous metrics, it makes use of actual class probabilities.



So, more closer the correct class probability as to 1, lesser the loss.

For multiclass classification,

total pts. — n c — classes

$$\text{log loss} = \frac{-1}{n} \sum_{i=1}^{n} \sum_{j=1}^{c} y_{ij} * \log(p_{ij})$$

y_{ij}: 1 if $x_i \in$ class j, 0 Otherwise

p_{ij}: Prob. of $x_i \in$ class j

1) One key disadvantage of log loss is its hard to interpret, as it can go as high as ∞.
2) No upperbound, can't determine, how bad.

Regression Metrics

V6: R^2 or Coefficient of determination

Some Regression metrics.

$y_i \in R$ is regression.

Test set i=1 to n

$x_i \rightarrow y_i \rightarrow \hat{y}_i \leftarrow$ model o/p

$e_i = (y_i - \hat{y}_i)$

e_i / error for pt i

Just like in classification, a simple model would be one that returns the label of the majority class for all inputs.

Similarly, for regression, a simple model would be one that returns the mean of y_i's.

Sum of squared errors for the simple mean model = $\sum_{i=1}^{n} (y_i - \bar{y})^2$

(Sum, error, squares)

Also called SS_{Total}

Note: $\bar{y} = \frac{1}{n} \sum_{i=1}^{n} y_i$

Residue = $\hat{e}_i = (y_i - \hat{y}_i)$

Sum of squares of residue = $SS_{res.} = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$

$$R^2 = \left(1 - \frac{SS_{res.}}{SS_{total}}\right)$$ — Actual metric

SS_{res} → closer to 1 better it is

Case 1: $SS_{res} = 0$

$\Rightarrow e_i = 0$

$\Rightarrow R^2 = 1$ (Best value)

Case 2: $SS_{res.} < SS_{total}$

$\Rightarrow R^2 \in [0, 1]$

Case 3: $SS_{res.} = SS_{total}$

$\Rightarrow R^2 = 0$

$\Rightarrow$ Same as simple mean model.

Case 4: $SS_{res.} > SS_{total}$

$\Rightarrow R^2 = 1 - (\text{some value greater than } 1)$

$\Rightarrow$ Negative.

$\Rightarrow$ Means it is worst than a simple mean model.

Read Wikipedia → Coefficient of determination

V7: Median Absolute Deviation of Errors [MAD]

$R^2 \Rightarrow$ Relies on mean, hence not very robust to outliers (if one e is very large it be an outlier)

For each $x_i \xrightarrow{\text{we have}} y_i, \hat{y}_i, e_i$

If e_i = random variable.

median(e_i) = Central value of errors

(It is robust measure of mean)

MAD(e_i) = Median of $(|e_i - \text{median}(e_i)|)$

(It is a robust measure of Std dev.)

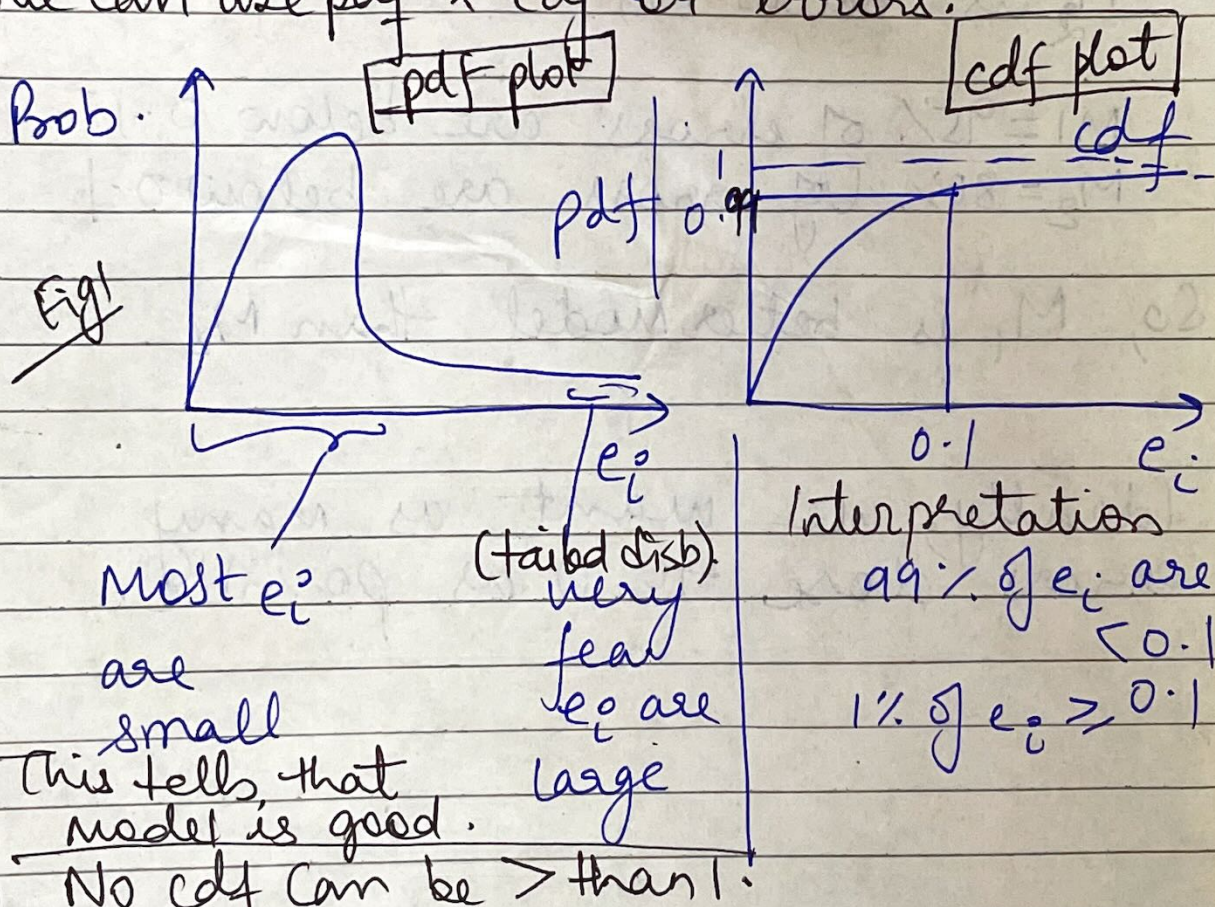
Defined for a set.

Use median & MAD if outliers are present.

Use mean & Std. dev. if no outliers

V8: Distribution of Errors

We can use pdf & cdf for errors.



MOST e_i are small. This tells that model is good.

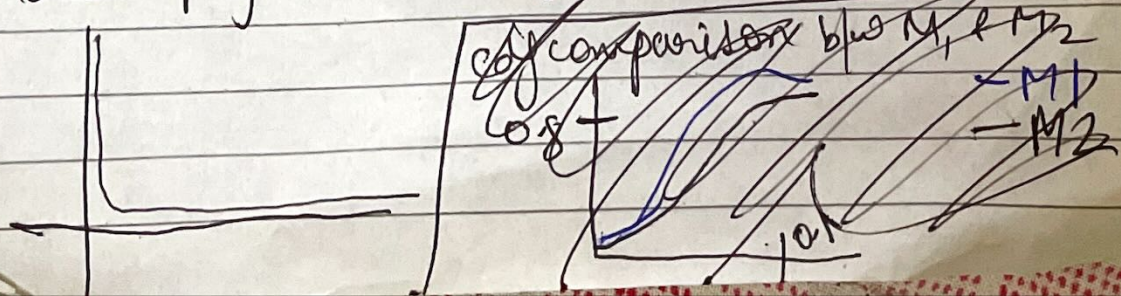
very few e_i are large

Interpretation
99% of e_i are < 0.1
1% of $e_i \geq 0.1$

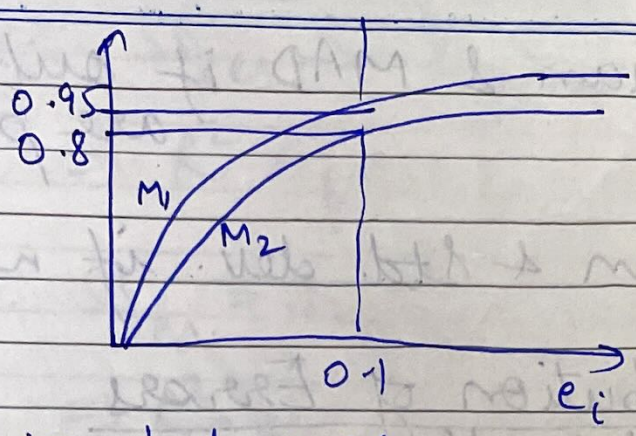
No cdf can be > than 1.

So, we can use these standard statistical measures to get an idea of distribution of errors.

Ideal pdf in Disb of errors should be :-



eg -



M_2 is below M_1,

$M1 = 95\%$ of errors are below 0.1
$M_2 = 80\%$ of errors are below 0.1

So, M_1 is better model than M_2.

Ideally, we want as many errors close to 0 as possible.