☆ Performance measurements :-

- In this chapter we will look at how to measure the performance of model.

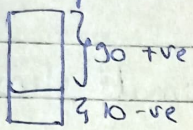
① **Accuracy :-** No. of correctly classified points / Total No. of points in Test set.

Accuracy lies b/w 0 to 1
bad ↳ good.

There are some problems with Accuracy

① Imbalanced data set :- Accuracy may be high for wrong prediction. (bad model)

Ex



And model predict all points as +ve :.
Accuracy = 90% even the model is not doing anything.

② Accuracy cannot use prob score

② **Confusion matrix :-**

Let's start with binary classification task : (0,1)

Predicted \ Actual	0	1
0	a	b
1	c	d

Dataset

x_1	y_1
x_2	y_2
x_3	y_3
⋮	⋮
x_n	y_n

actual class label

$\hat{y}_1$
$\hat{y}_2$
⋮
$\hat{y}_n$

predicted class

- Confusion matrix cannot process prob scores.

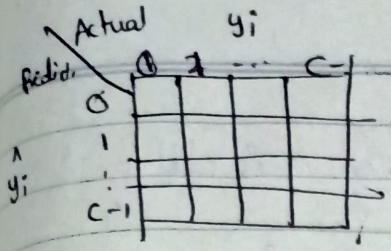
a = No. of points which are actually zero and predicted as zero.

b = No. of points which are actually 1 and predicted as zero

c = " " " " " " 0 " " " 1

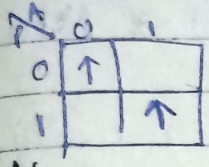
d = " " " " " " 1 " " " 1

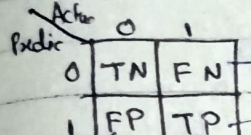
In a multi-class classification, we can draw a CxC matrix



multiclass

If a model is good. then diagonal value of confusion matrix should be high.



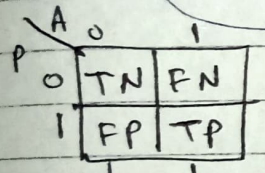


FN :- Predicted=Negativ → incorrect ∴ actual = 0

Predicte = 1 = Positiv → is it correct Yes T ∴ TP

TP
T → what is predicted label
actual are you correct

FP :- Predicted positive=1, incorrect predicts b/w actual = 0



$N + P = n$

True positive rate = $\frac{TP}{P}$

True negativ rate = $\frac{TN}{N}$

False positiv rate = $\frac{FP}{N}$

False negativ rate = $\frac{FN}{P}$

(11) · Precision, Recall & F1-score

Precision, Recall are very useful in information Retrivs.

Precision = $\frac{TP}{TP+FP}$ | of all the points the model predicted to be +ve what persentage of them are actually +ve.

Recall = True positive rate = $\frac{TP}{P}$

* Precision & Recall are only rating about +ve class.

we want Precision ↑, Recall ↑
can we combine the idea of precision & Recall.

$$\text{F1-score} = \left(2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}\right)$$

(iv) Reciever operatic characteristic Curve (ROC curve) & AUC (Area under the curve)

	y	$\hat{y}$ → predicted.
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

Let binary classifier not only give class label, it also gives a score. more score mean more chance of point to belong to that class.

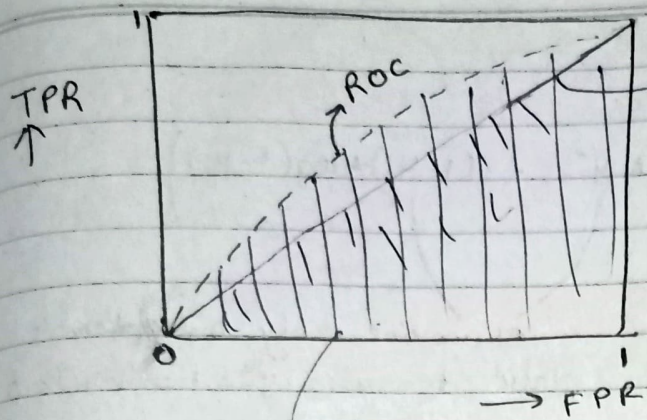
step 1: sort the data in decreasing order of $\hat{y}$.
step 2: Thresholding :- Initially the first value (max val) is threshold, T_1.

if $\hat{y} \geq T_1$
    label = 1
else
    label = 0

calculate the TPR, & FPR corresponding to the predicted class label

	y	$\hat{y}$	$\hat{y}, T=0.95$	$\hat{y}, T=0.92$	$\hat{y}, T_c=0.8$	...	$\hat{y}, T_c$ = least value.
x_1	1	0.95	1	1	1		
x_2	1	0.92	0	1	1		
x_3	0	0.80	0	0	1		
x_4	1	0.76	0	0	0		
x_5	1	0.71	0	0	0		
			↳ TPR_1, FPR_1	↳ TPR_2, FPR_2	↳ TPR_3, FPR_3		↳ TPR_N, FPR_N

plot these TPR_i, FPR_i.



→ AUC = Area under ROC curve
↳ 0 to 1
bad ↙ ↳ v.good

This is only useful to binary classification.
AUC have following property

(i) AUC can be impacted by imbalanced data : AUC have large value for even dumb model.

(ii) AUC didnot dependent on the $\hat{y}$ scores themselves. it is only dependent on ordering.

Ex:-

	y_1	$\hat{y}_1$	$\hat{y}_2$
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

↳ Orders of the value are same i.e. they are already in sorted order hence their ~~ROC~~ AUC will be similar.

(iii) AUC of a random model will be exactly 0.5.

(iv) if AUC < 0.5 just reverse the model outputs.

(v) Log-loss :-

It uses the probability scores.

	y		
x_1	1	0.9	→ P_1
x_2	1	0.6	→ P_2
x_3	0	0.1	→ P_3
x_4	0	0.4	→ P_4

Given a Test-set of n-points: log-loss $= -\frac{1}{n}\sum\left(\log(P_i)*y_i + (1-y_i)*\log(1-P_i)\right)$

Test-set of n-points:-

$$\text{log-loss} = -\frac{1}{n}\sum_{i=0}^{n}\left(\log(p_i) * y_i + (1-y_i) * \log(1-p_i)\right)$$

-ve of average

At one time only one of them is valid bcz when $y_i = 1 \Rightarrow (1-y_i) = 0$

$y_i = 0 \Rightarrow (1-y_i) = 1$

We want log-loss to as small as possible. it can lie b/w 0 to ∞.

We can easily extend log loss to multi class.

Let we have C class.

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

y_{ij} = 1 if x_{ij} b/w class j
= 0 otherwise

Only disadv of log loss is, its hard to interpret.

(41) R^2 or coefficient of determination

We have looked at various classification model measurement. Now look at regression performance measurement.

$$\text{error}, e_i = y_i - \hat{y}_i$$

$$\text{Total sum of square} = \sum_{i=1}^{n}(y_i - \bar{y})^2$$

$\bar{y}$ → average value of all y_i in data.

Simple mean model:- it for any query x_q it return the mean of $\bar{y}$.

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

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

y_i : actual, $\hat{y}_i$: predicted value

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

Case 1: $SS_{res} = 0$, { if there is no error } then $R^2 = 1$ (best value)

Case 2: $SS_{res} < SS_{total}$; $R^2 \equiv 0$ to 1

Case 3: $SS_{res} = S_{tot}$; $R^2 \equiv 0$ { It is a simple mean model }

Case 4: $SS_{res} > SS_{total}$; $R^2 < 0$ { model is worse than simple mean model }

to calculate R^2 we use SS_{res} which are mean hence R^2 is not very robust to outlier.

If we can image error as R.V we can calculate median of e_i's

$$MAD(e_i) = \text{media}\left(|e_i - \text{median}(e_i)|\right)$$

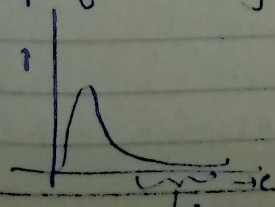
So, instead of using R^2 we can use Median of e_i or $MAD(e_i)$ as measure.

If median(e_i) is less model is good.

or $MAD(e_i)$ is less model is good.

→ Distribution of error:-

We can also use pdf, cdf of e_i's to calculate the performance of regression model.



{very few e_i are large hence model is good}