Performance measurement of models

☆ Accuracy ⇒ Measure of a metric how good our model is.

$$\text{Accuracy} = \frac{\#\text{ correctly classified points}}{\#\text{ total number of points}}$$

[0 to 1]

↓ Bad ↘ Best

100 pts → 60 +ve
40 -ve

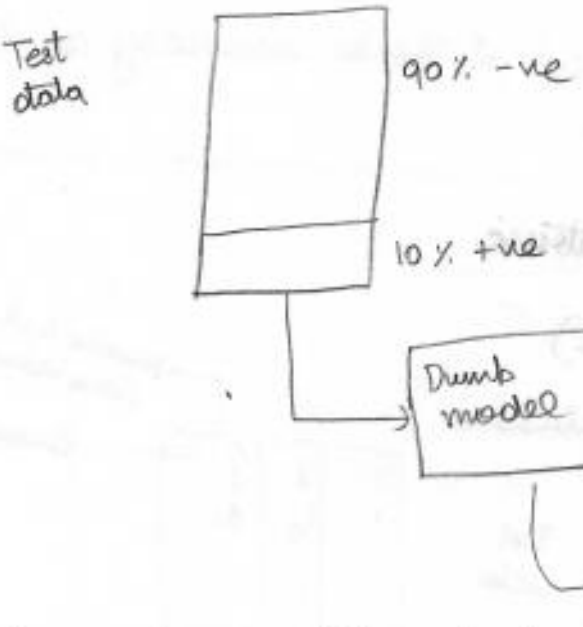
(M) → 53 +ve , 7 -ve
→ 35 -ve , 5 +ve

Correctly = 88 , Total errors = 12

$$\text{accuracy} = \frac{88}{100} = .88$$

⇒ Drawbacks of accuracy / Limitations of accuracy

Case 1 : Imbalanced data :



Model "dumb"

$x_q \rightarrow$ -ve

{ return -ve ; }

When you have imbalanced dataset never use accuracy.

Case 2:

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

Predicted value of model 1 and model 2

M_1 & M_2 return a prob. score

KNN

$x_5 \rightarrow$ prob $(\hat{y}_2 = 1)$ — probability score

(Test-set)

Probability values

Hat / Cap:

$\hat{y}$: predicted value

$(\bar{x}, \bar{y})$

predicted class label same for both models M_1 & M_2.

But looking at probability score M_1 is better than M_2

But accuracy cannot use probability score.

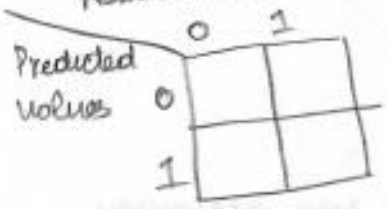
$\hat{y}_1, \hat{y}_2 \rightarrow M_1$ & M_2 have same accuracy.

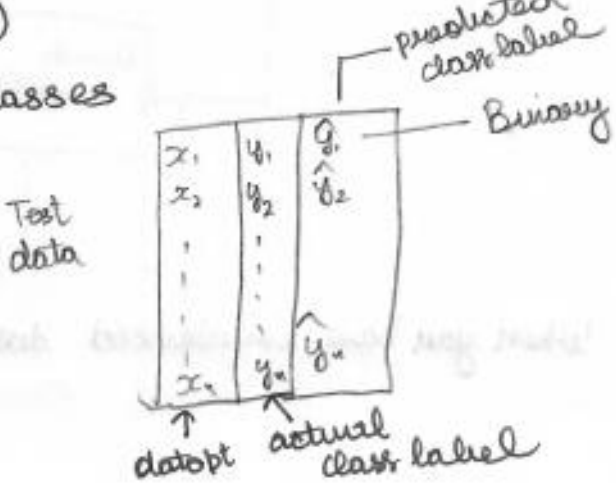
When given probability you should not choose accuracy as it gives both models above same importance

---

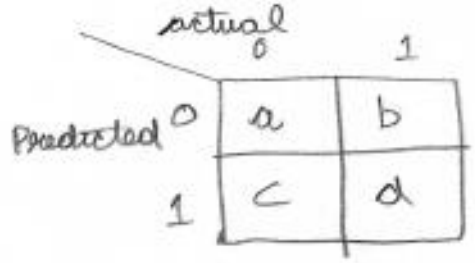
Confusion Matrix

☆ Binary classification task (0, 1) ← 2 classes





NOTE Confusion matrix cannot take probability scores.



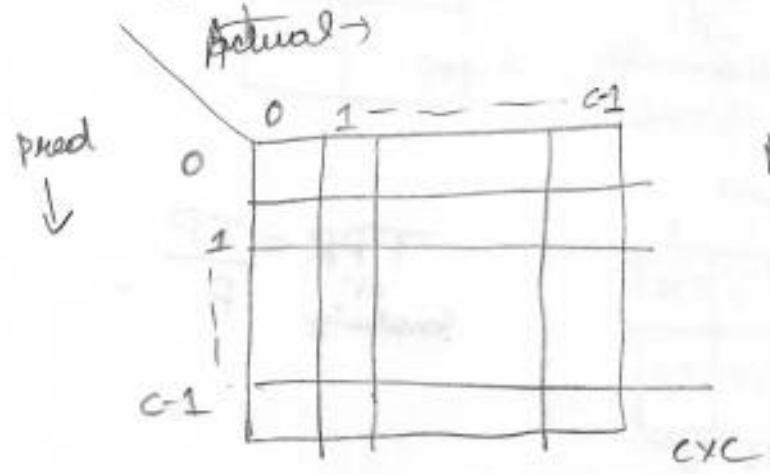
a: # pts s.t $y_i = 0$ (actual class label)
$\hat{y_i} = 0$ (predicted class label)

b: # pts s.t $\hat{y}_i = 0$ (predicted)
$y_i = 1$ (actual)

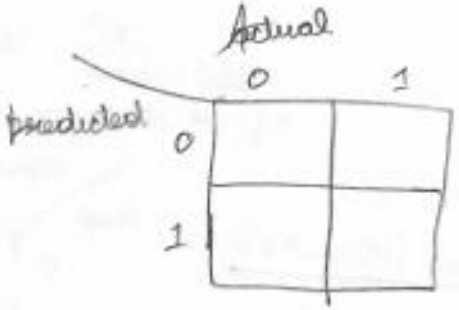
c: # pts s.t $y_i = 1$ (actual)
$\hat{y}_i = 1$ (predicted)

d: # pts s.t $y_i = 0$ (actual)
$\hat{y}_i = 1$ (predicted)

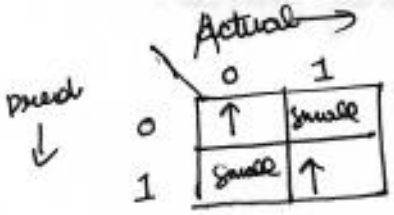
Multi Classification (C-Class)

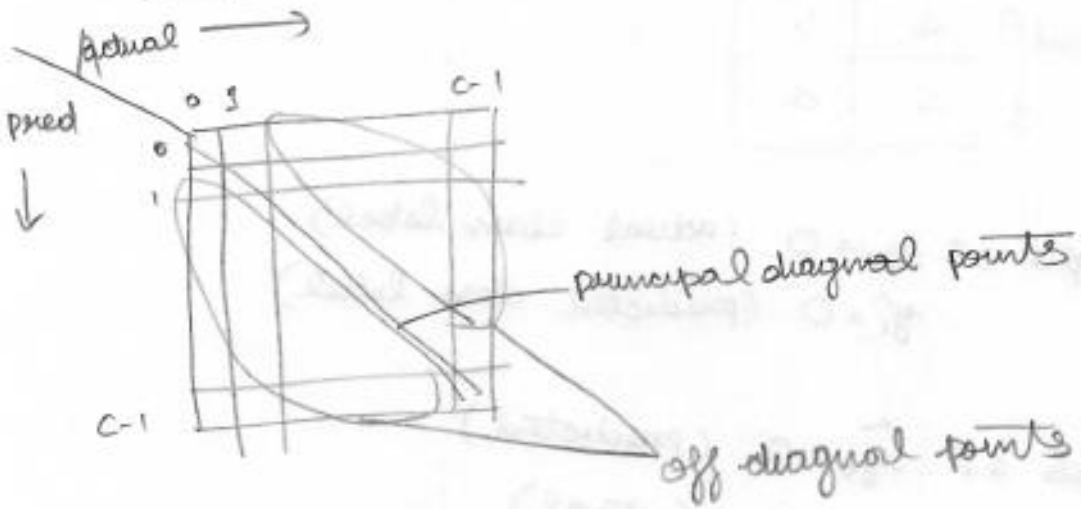


Binary setting

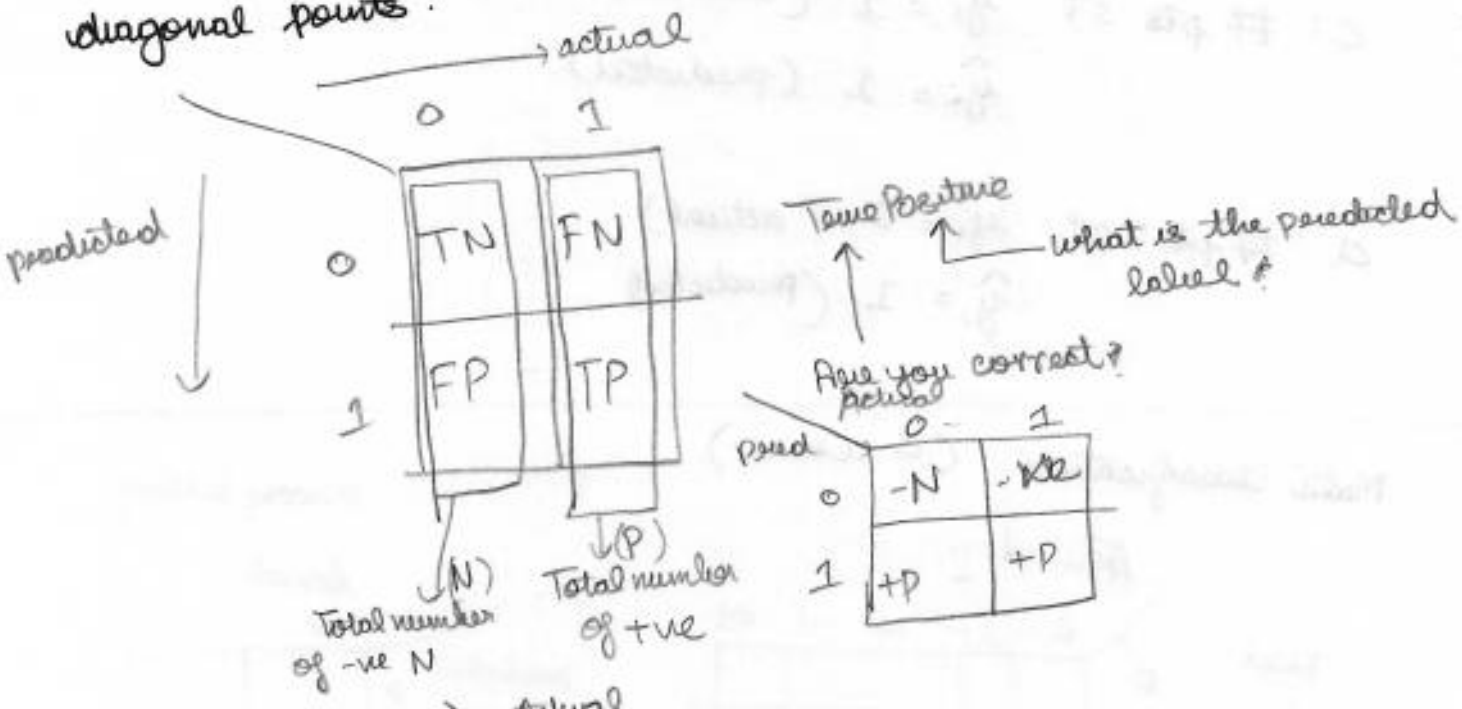


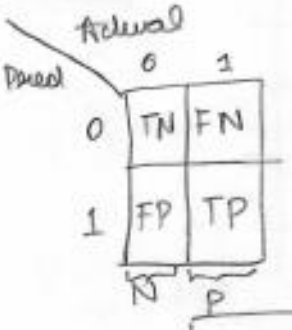
Model is sensible





Aim is that principal diagonal points must be larger than off diagonal points.





N + P = n

✧ True Positive Rate

$$TPR = \frac{TP}{P}$$

or Sensitivity

$$\text{True Negative Rate} = \frac{\text{True Negative}}{\text{Negative}}$$

or Specificity

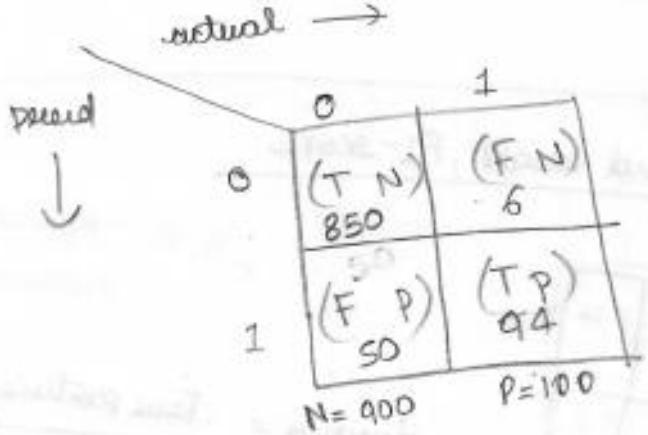
Predicted \ Actual	0	1
0	TN	FN
1	FP	TP

$$\text{False Positive Rate} = \frac{\text{False positive}}{\text{Negatives}}$$

$$\text{False negative Rate} = \frac{\text{False Negative}}{\text{positives}}$$

Example

Test :- 900 -ve, 100 +ve } Imbalance data



Model :

$TPR = 94\%$

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

$FPR = \frac{50}{900} =$

$FNR = \frac{6}{100} = 0.06 = 6\%$

Model said to be good if TPR and TNR are high.

Imagine Dumb model :- all -ve

pre \ act	0	1
0	900	100
1	0	0

N=900 P=100

900 -ve
100 +ve

↑TPR = 0 , ↑TNR = $\frac{900}{900}$ = 1

↓FPR = 0 , ↓FNR = $\frac{100}{100}$ = 1

Continued

↑ TPR = 0 %

↑ TNR = 100 %.

↓ FPR = 0 %

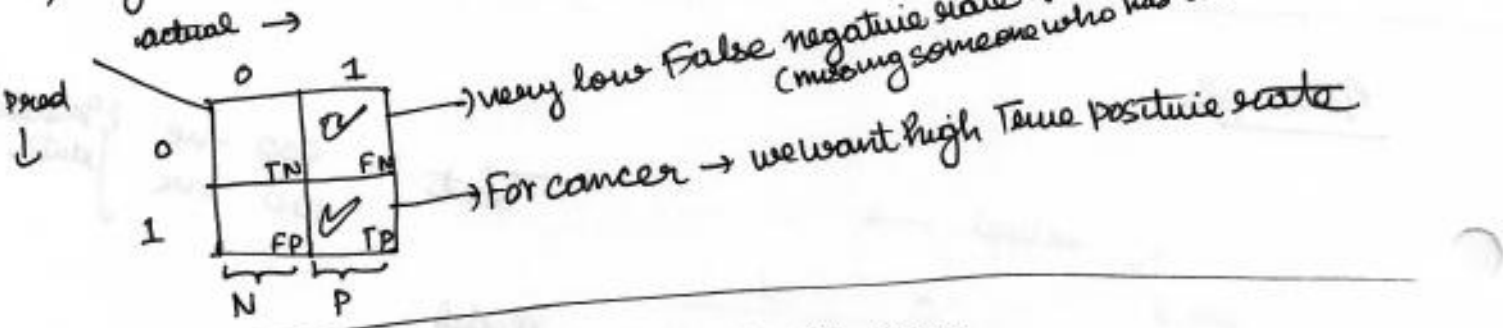
↓ FNR = 100 %

From this we can predict that the model is dumb.

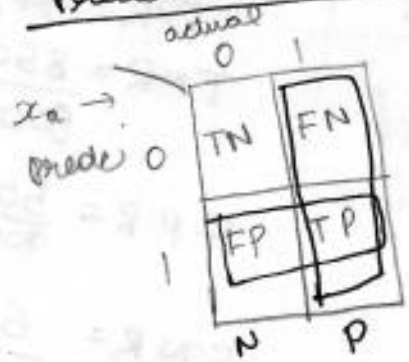
☆ Which number is more important among 4?
This is domain specific.

eg.

diagnose cancer / not



Precision and recall, F1-score



Pr, Re — Information retrieval

$$\text{Precision} = \frac{\text{True positive}}{\text{True positive + False positive}}$$

☆ Information retrieval → million of documents ~~use~~ objective to find some document which are relevant.

$$\text{Recall} = \frac{\text{True positive}}{\#\text{ positive}}$$

NOTE: Precision and Recall values when you care about the +ve class.

Precision and Recall into one measure

Precision ↑ (0-1) Recall ↑ (0-1)

$$\overbrace{\text{F1-Score}} = \left(2 \times \frac{\text{precision} \times \text{recall}}{\text{Precision} + \text{recall}}\right)$$

NOTE :→ Harmonic Mean →

$$F1 = \frac{2}{\left(\frac{1}{\text{recall}} + \frac{1}{\text{pr}}\right)} \left[\text{avg}\left[\text{inv-recall}, \text{inv-precision}\right]\right]^{-1}$$

NOTE :

Harmonic mean:

$$F1 = \frac{2}{\left[\frac{1}{\text{recall}} + \frac{1}{\text{precision}}\right]} = \left[\text{avg}\left(\text{inv-recall}, \text{inv-precision}\right)\right]$$

Used in kaggle competitions

Lies between 0 to 1

Medical domain, we want high recall. We don't want any patient who actually has a disease to be misdiagnosed and the model say that he/she doesn't have it.
But, we are allowed to have a low precision. Meaning, maybe, the patient actually doesn't have a disease but we are okay on saying that he/she has. That is not life threatening.

$$\text{Error rate} = \frac{FP+FN}{P+N}$$

☆ Why F-measures uses Harmonic mean?

It measures Test accuracy

Harmonic mean is applicable, if the numerators of the averaged values are the same. eg resistance in parallel

Arthematic mean is applicable, if the denominators of the averaged values are the same. eg resister in series

---

Harmonic mean ⇒?

Case I

Precession x recall = $0.1 \times 0.1 = 0.01$

f1score = $\frac{2}{\frac{1}{0.1} + \frac{9}{0.1}} = \frac{2}{20} = 0.1$ ↔ accurate value

---

Case II

$P = 0.8$

$r = 0.1$

$P \times r = 0.08$

f1score = $\frac{2}{\frac{1}{0.8} + \frac{1}{0.1}} = .177$

F1→Score gather information more accurately than PXr

Evaluating Classifier performance Receiver Operating Characteristics curve (ROC)

→ electronics & radio engineers discovered by

binary classification

Model → 1
0

↳ score
Prob score

x	y	$\hat{y}$ (probability score)
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

decreasing of $\hat{y}$

↑score ⇒1

ROC curve

Step-1 Sort in decreasing order of $\hat{y}$

Step-2 Thresholds (τ)

a.) $\tau_1 = 0.95$

if $\hat{y} \geq \tau_1$

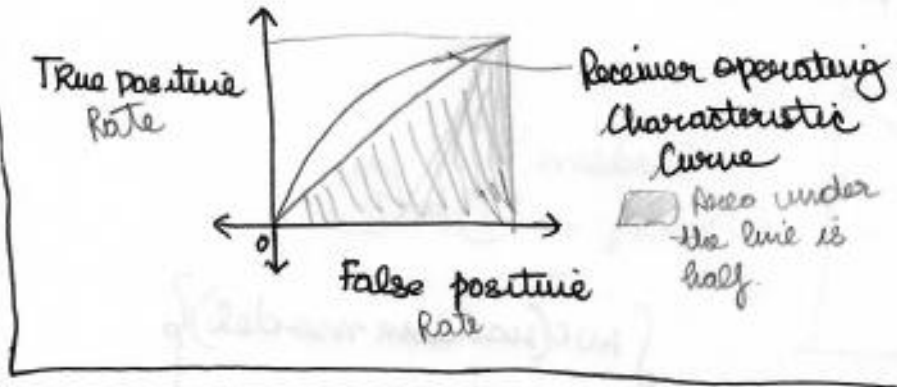
1

else

0

} TPR, FPR

ROC is ploted between



Thresholds are used for plotting different ROC curves.

ROC plots is plotted using different thresholds.

By setting different thresholds you get new True positive Rate and False positive Rate that is called ROC curve.

Area under the ROC curve can be 0 to 1

↑ Terrible ↑ V. good

☆ AUC brilliant properties

① Imbalanced data AUC can be high for dumb model

(2) AUC did not care about actual values. It depends on the ordering. (sorting)

		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

$AUC(M_1) = AUC(M_2)$

↓

As sorted value is same. both models are equivalent

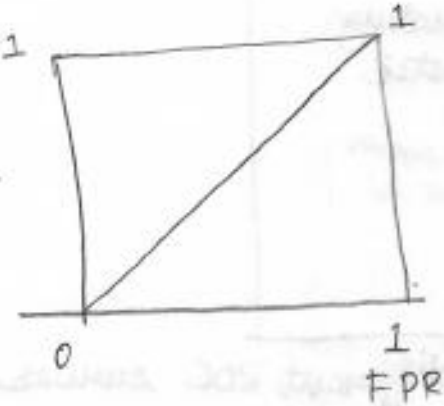
5.) Classifier A is better than Classifier B if A's AUC is larger than B's AUC

6.) A perfect classifier has AUC of 1

7.) A classifier making random guesses have AUC of 0.5

(3) TPR ↑

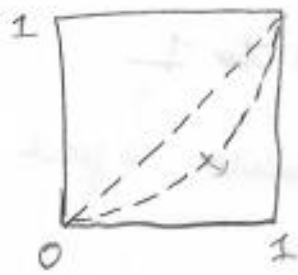
ROC



Random → x_q → (1) or (0)

{AUC(random model) = 0.5}

(4) Model M:



AUC (M) : = 0.2 ↑ worse than random

AUC value 0.5 to 1 → normal

0.5 → random

0.0 to 0.5 → swap class labels

Change the class label to opposite to it: $\hat{y}_i = 0 \rightarrow 1$, $\hat{y}_i = 1 \rightarrow 0$

Log Loss : probability Scores

Binary classification :

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

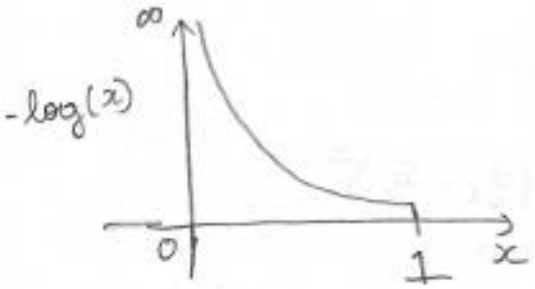
Test - set of n points :

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

If model is good P_i values should be equal to 1 that means model is good.

log loss should be as small as possible.

log loss : avg. neg log (probability of correct class label)



$0 \le p \le 1$

smaller the better
all loss function aim/objective is to be small

Multi Class 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 o/w.

P_{ij} → prob that $x_i \in$ class j

best case = 0

log loss (M) = 1.0

R^2 Squared / Coefficient of determination

$y_i \in \mathbb{R}$

Test :- $x_i, y_i, \hat{y}_i$ ← model prediction / output.

$i \to 1$ to n.

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

↑ error for pt i

✧ Sum of squares (SS)

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

$\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$ { (mean / average values of y_i's in test data)

{ example predict height ($\in \mathbb{R}$)

- features - w, c, hc, ...

mean can be average value . that can be used to build simplest model.

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

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

|| squared errors using simple mean model.

Residue = $e_i = y_i - \hat{y}_i$

(e_i: error, y_i: actual, $\hat{y}_i$: predicted)

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

case 1: $SS_{res} = 0 \leftarrow e_i = 0 \rightarrow R^2 = 1$ (best - value)

case 2: $SS_{res} < SS_{total}$; $R^2 = 0$ to 1

case 3: $SS_{res} = SS_{total}$; $R^2 = 1 - 1 = 0 \rightarrow$ Model is same as simple mean model.

case 4: $SS_{res} > SS_{total}$

$R^2 = 1 - (\text{ugt} > 1) = -ve$

{ Model is worse than a simple mean model

Median absolute deviation (MAD)

$SS_{Residue} = \sum_{i=1}^{n} e_i^2$ one e_3 is very large whole R^2 will be effected.

R^2 is not very robust to outliers.

MAD → (Median absolute deviation)

e_i : random variable.

median (e_i) = central - value of errors. = small

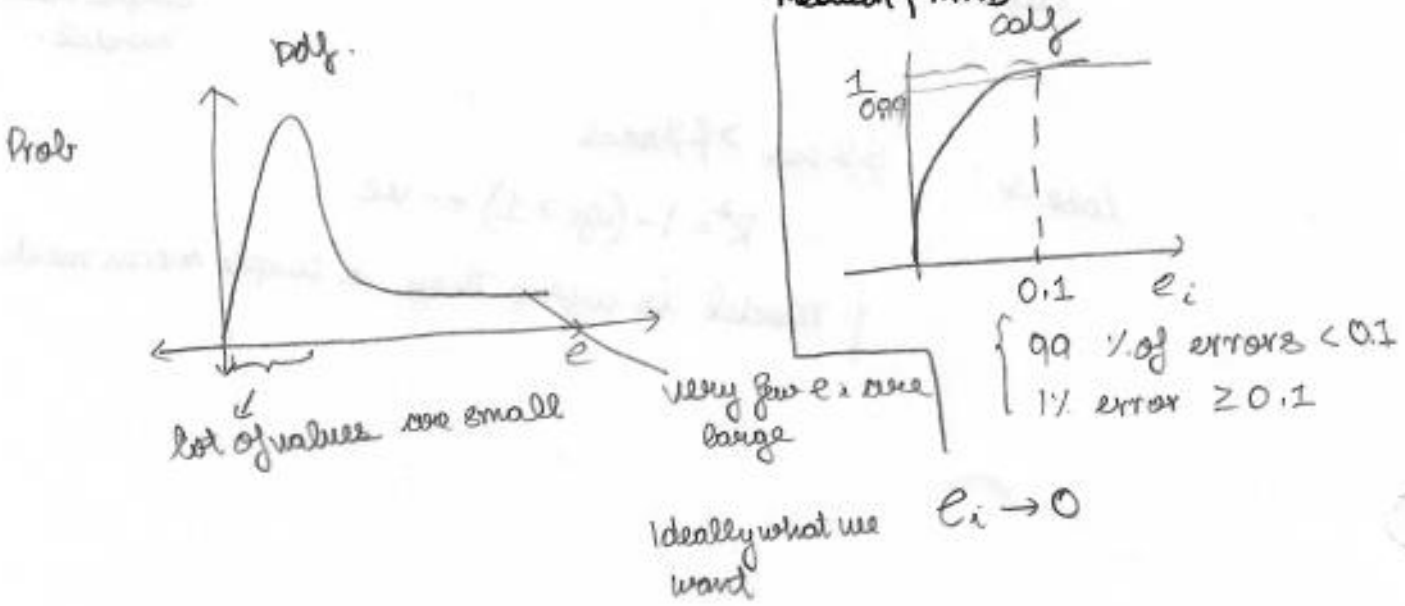
$MAD(e_i) = \text{median}\left(|e_i - \text{median}(e_i)|\right)$ = small → no outliers

Central tendency mean or median of e_i s

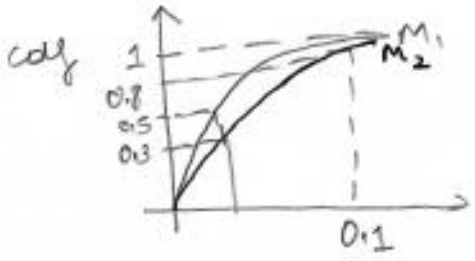
std dev ↓ MAD ↓ robust to outliers

Distribution of errors :

$x_i \rightarrow y_i, \hat{y}_i$, $e_i \rightarrow$ mean, std - dev median, MAD



Model M_1 & M_2



M_2: cdf is below model1

M_1: # errors 95% errors are below 0.1

M_2: 90% errors below 0.1

M1: 50% errors are below 0.01

M2: 30% errors are below 0.01

M1 is better than M2.