#Machine learning (IBM)

It is a branch of Artificeal Intelligence
Use of data and algorithms to imitate the way that human learn, gradually improving its accuracy

child $\xrightarrow[7\text{year}]{}$ young $\xrightarrow[20\text{ year}]{}$ Adult $\xrightarrow[30\text{year}]{}$ Old

model1 (Algo) $\longrightarrow m_2 \longrightarrow m_3 \longrightarrow m_4$ (data driven)

Q. ϕ input x (rules) define

```
if (x%2 == 0)
        #even
else
        #odd
```

	feature		target
floor	location	size	rent cost
7	Banani	1500	50,000
4	Tongi	2200	55,000
2	Banani	1200	30,000
5	Dhanmondi	1500	40,000
3	Uttara	1800	??

size ~~1500~~ ✗
location ✗
floor ✗
→ Not maintaining rules

When can not set the rules → we can use ML (Machine learning)

Types of Machine learning

- Supervised (label)
  - → Regression → continuous
  - → classification → discrete
    - binary and other classification
- Unsupervised (no label)
- semi-supervised (Unknown target) (target is known)
- Reinforcement learning

Example

Cat or Dog

Features	Target
image 1	cat
image 2	dog
image 3	cat
image 4	dog

im1	f_1	?
im2	f_2	?
im3	f_3	?
im4	f_4	?

label → labeled image

student exam pass/fail

study hour	decision	fail/pass
2h	fail	0
3h	fail	0
5h	pass	1
8h	pass	1

0/1 Classification

target

CGPA [1-4] → Continuous

3.14 3.42 2.92

2.03 1.23 4.00

Rent cost (0-30,000) any value

12k

13k

12k

target rent cost

possible

no idea

Continuous → Regression

discrete → classification

↳ cat or dog

Regression

Size	Rent cost
1200	25,000
1500	30,000
1800	35,000
2000	?? (40,000)

(0-10) (0-100) (500-700)

linear regression

$f(x) = 2x + 3$

x → independent (feature)

↓

dependent (target)

Rent cost = 20 × Size hypothesis

R = 20 S Randomly

prediction

$R = 20 \times 1200 = 24,000$

$R = 20 \times 1500 = 30,000$

$R = 20 \times 1800 = 36,000$

25,000	L = 1000 +	2000 / 3
30,000	L = 0 +	= 666.66
35,000	L = 1000	Loss

total Loss = 2000

$Loss = \sum |Actual - Prediction|$

Loss ↓ → Accuracy ↑

Loss = 2000 (3 sample)

We want to minimize loss

	sample	each	ΣL
m_1	N 3	1000	3000
m_2	N 1000	5	5000

↑ Dataset → Loss ↓

$loss = \frac{1}{N} \sum |Actual - Predict|$

$= \frac{3000}{3}$

$= \frac{5000}{5000}$

state of the art

2018 → Loss = 1000 ✗

2020 → Loss = 100 ✓

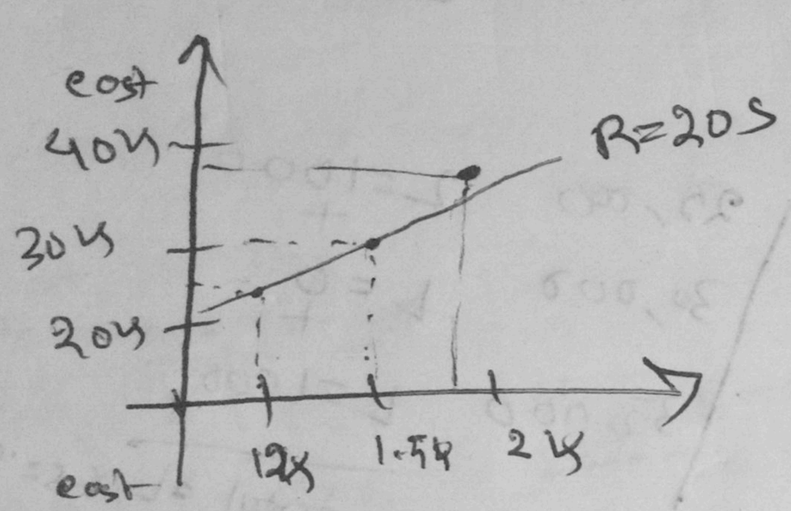
2023 → Loss = 10 ✓

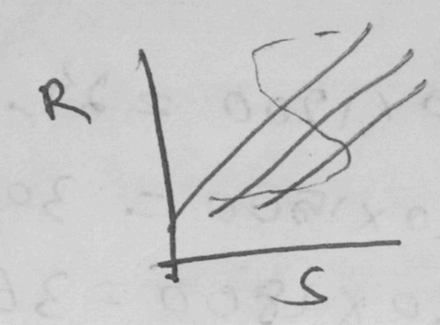
$R = 20S$

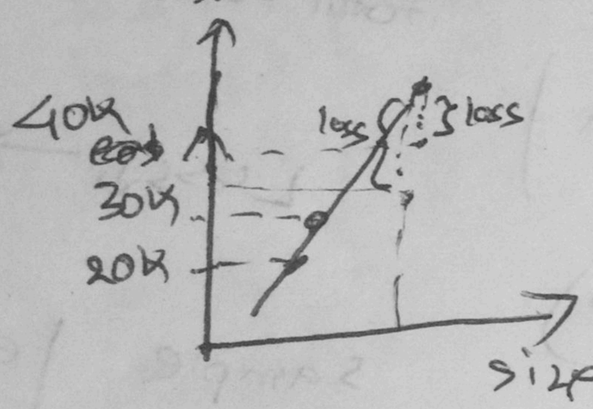
$y = 2(x) + 3$ which is linear → $y = mx + c$

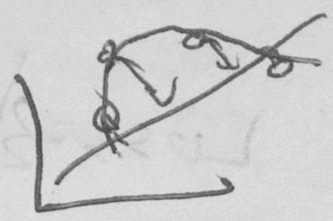
$y = 2x^2 + 3$

$y = x^{1/2}$









→ loss

fit (best)

$R = 20S$ Loss = 650 ✗ ↓ Reduce

min (Loss)

$R = 20S$

target ← $y = \omega x$ → feature

weight

gradient descent :

$$w_{new} = w_{old} - \alpha \frac{\partial}{\partial w}(Loss)$$

partial derivative

$y = 2x$

$y = x$

$y = \frac{1}{2}x$ ← (Best fits with line)

Size	y price	$\hat{y}$ prediction
1000	20000	21000
1200	25000	27000
1800	35000	31000

MAE (Mean absolute error)

$$MAE = \frac{1}{N}(y - \hat{y})$$

$$= \sum \frac{1}{N} |y - \hat{y}|$$

$$= \frac{1000 + 2000 + 4000}{3}$$

$$= \frac{7000}{3}$$

MSE (Mean ~~abso~~ square error)

$$MSE = \frac{1}{N}(y - \hat{y})^2$$

$$RMSE = \sqrt{\frac{1}{N}(y - \hat{y})^2}$$

$$= \sqrt{\frac{1000^2 + 2000^2 + 4000^2}{3}}$$

$$= \sqrt{\frac{21 \times 10^{\cdots}}{3}}$$

$$= - - - -$$

$$RMSE = \sqrt{MSE}$$

$$= \sqrt{BDT^2}$$

$$= BDT$$

(R²)

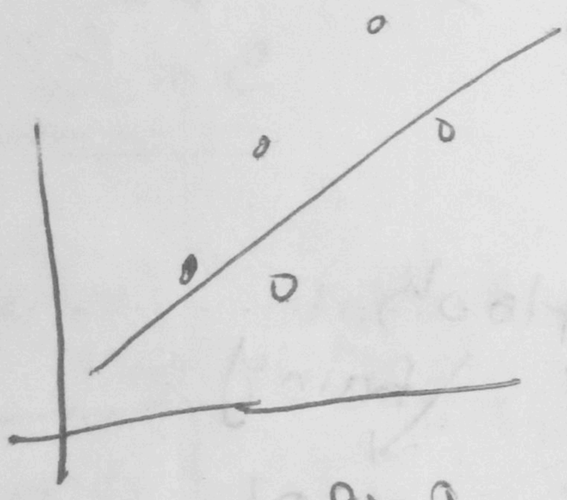


fig 1

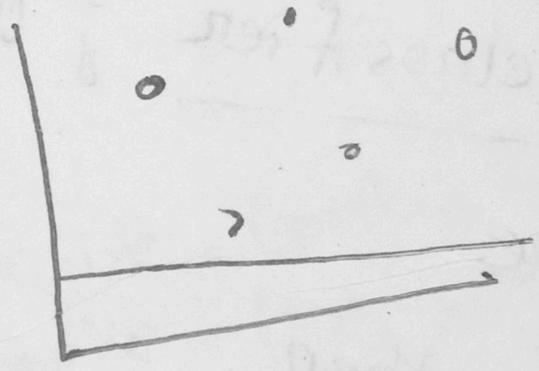


fig 2

$0 \leq R^2 \leq 1$

$R \Rightarrow 1$ → perfectly fitted

$R \Rightarrow 0$ → poorly fitted

$$R^2 = 1 - \frac{\sum (y - \hat{y})^2}{(y - \bar{y})^2}$$

$\bar{y}$ - Average

$\hat{y}$ - Predicted

y - Actual

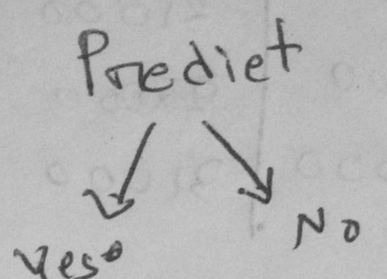
$R^2 = 0.8$

$R^2 = 0.2$

classification

Index	weather	outlook	Play 2D
1	hot	sunny	Y
2	cold	sunny	Y
3	cold	rainy	N
4	hot	sunny	Y
5	cold	sunny	Y

Predict



Outcome

Zero - R classifier

↳ count majority

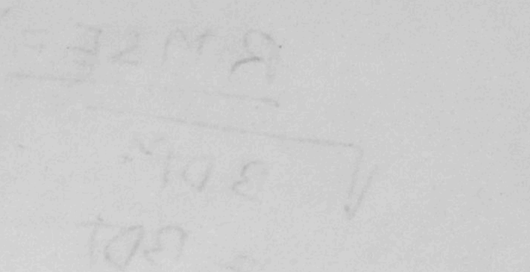
yes → 4

No → 1

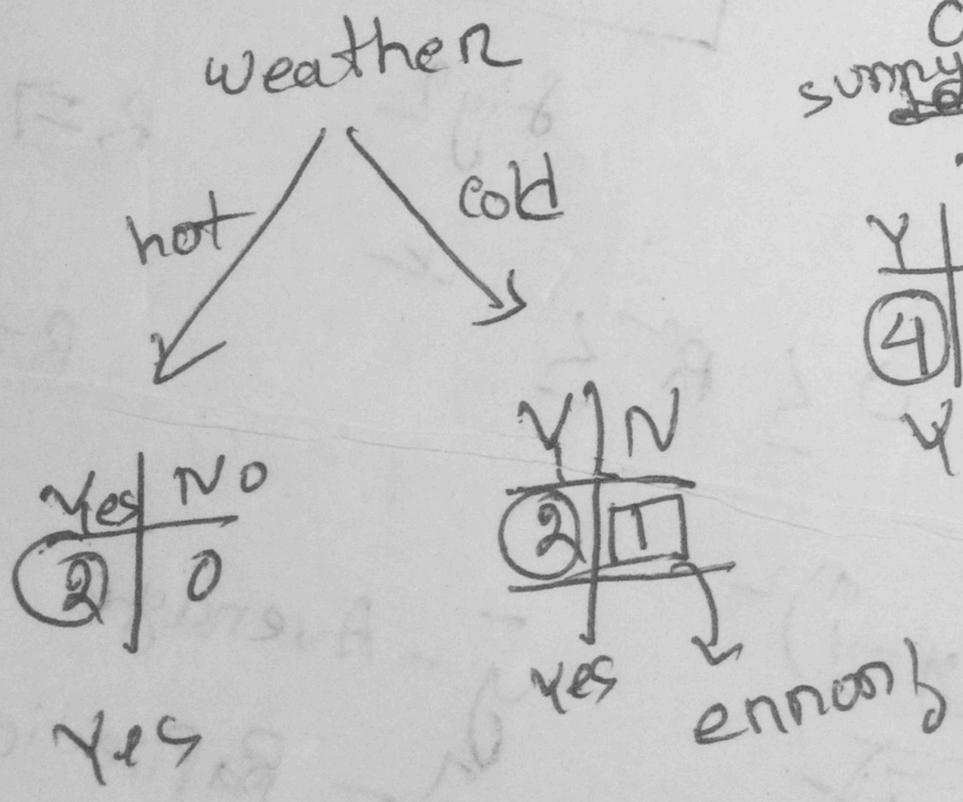
prediction All → Yes.

One_R classifier

Outlooks



Feature



If Y=N
Random choice

Evaluation metric

weather	outlook	Play
h	S	Y
h	S	Y
C	R	N
C	S	Y
h	S	Y

ZeroR

~~pe~~
prediction → "Yes"

$$\text{Accuracy} = \frac{1+1+0+1+1}{5}$$

$$= \frac{4}{5} = 0.8$$

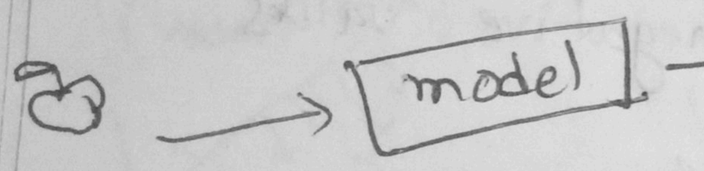
$$= 80\%$$

1000 apple

750 apple good } ZeroR

250 apple bad

$$\frac{250}{1000} = 25\%$$



factory

Confusion metric

positive - (good) P
negative - (bad) N

Actual value → / predicted value ↓	P (Yes) 00	N (No) 01
P	TP = 7	FP = 2
N	FN = 3	TN = 6
	10	11

Total = (7+2+3+6)
= 20

precision = $\frac{7}{7+2} = \frac{7}{11}$

(1) $\text{precision} = \frac{TP}{TP+FP}$

(2) $\text{recall} = \frac{TP}{TP+FN}$

FP↓ → ↑precision

FP↑ → ↓precision

FP=0 → ↑Precision (100%)

FN↓ → ↑recall

FN↑ → ↓recall

FN=0 → ↑ recall (100%)

precision → FP (medical test) cancer (Negative)
recall → FN

which metric should I use precision / recall?

Ans: Based on Target variable (P, N).

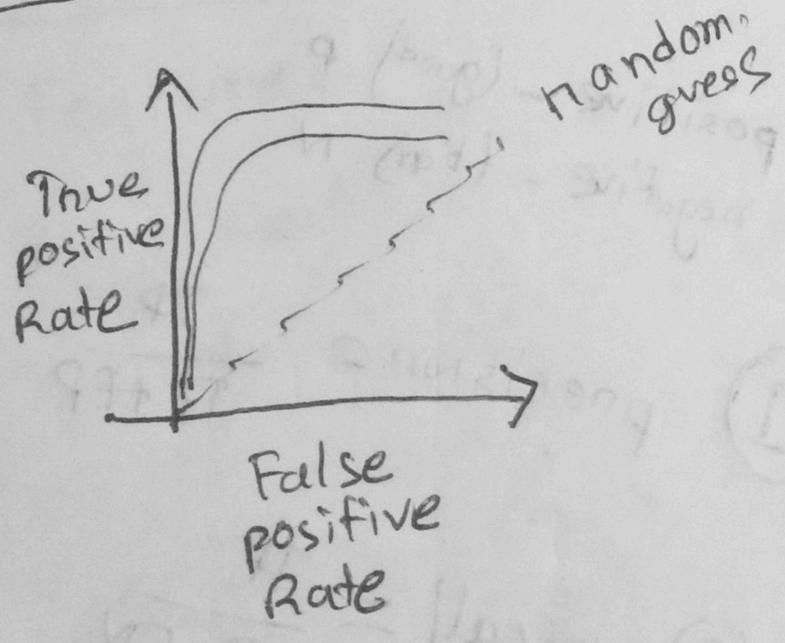
For FP on FN should be lessen than baseline value. For, example (spam email detection)

$$F1 \text{ score} = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}}$$

F1 measure → harmonic mean of precision and recall
↳ (close to smallest value)

↳ close to (mean) Neighten positive & non negative values
Mean

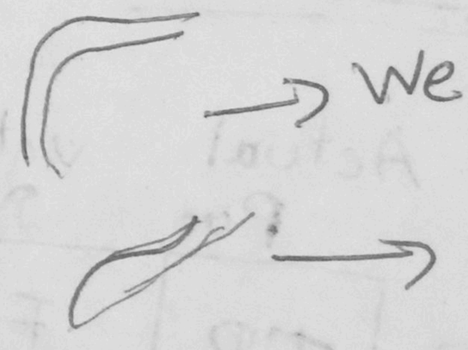
ROC → Receiver characteristics curve
ROC curve:



$TPR = \frac{TP}{TP+FN}$

$FPR = \frac{FP}{FP+TN}$

↳ FP / FP+TN



→ We want

→ Average

Under Random guess line, is considered bad values

AUC curve:

↳ Area under curve ⇒ must be high

⇒ must be low

* Divide into 3 parts

Train | Validation (can both)

60%, 80%, 70%

Test

10%, 20%, 30%

Train → Model → Train Accuracy
(Fitting)

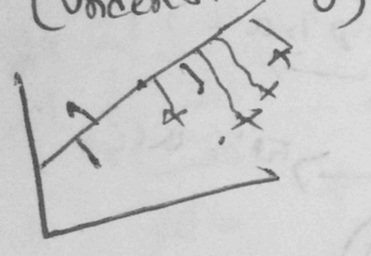
Val → Model → Validation Accuracy
(Validate)

Test → Model → Test Accuracy
Test

Test (must be unseen)

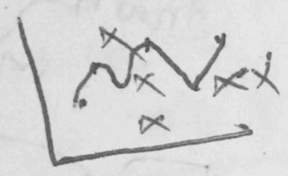
Overfitting and Underfitting

(Underfitting)



train → Low accuracy
test → low accuracy
(High bias, low variance)

(Overfitting)



train → High Accuracy
test → low Accuracy
(mixed bias, High variance)

Train → good Accuracy
Test → good Accuracy
Perfect fit
(low Bias and low variance)

Bias → Inaccuracy (Underfitting)
Variance → Inconsistant (Overfitting)

Bias ∝ $\frac{1}{\text{variance}}$ (Both cases)

Train acc →	20%	80%	30%	90%	85%	95%	72%
Test acc →	50%	70%	28%	82%	83%	perfect	perfect
	Overfit	good	underfit	Overfit	perfect		
				good ↙ ↘ overfit			

if test accuracy is greater than train accuracy,
then (different dataset, Wrong dataset split, ~~A~~ High complexity of model

How to tackle

Underfitting

(1) Training time ↑

(2) Increase the features ↑ (Feature Engineering)

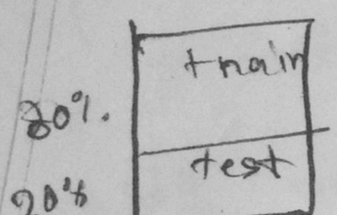
Overfitting

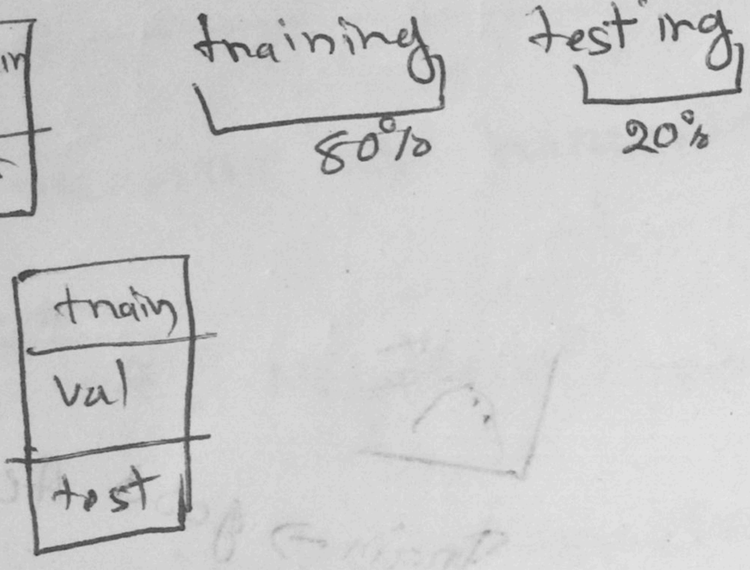
(1) Cross validation (Mandatory for cross validation)

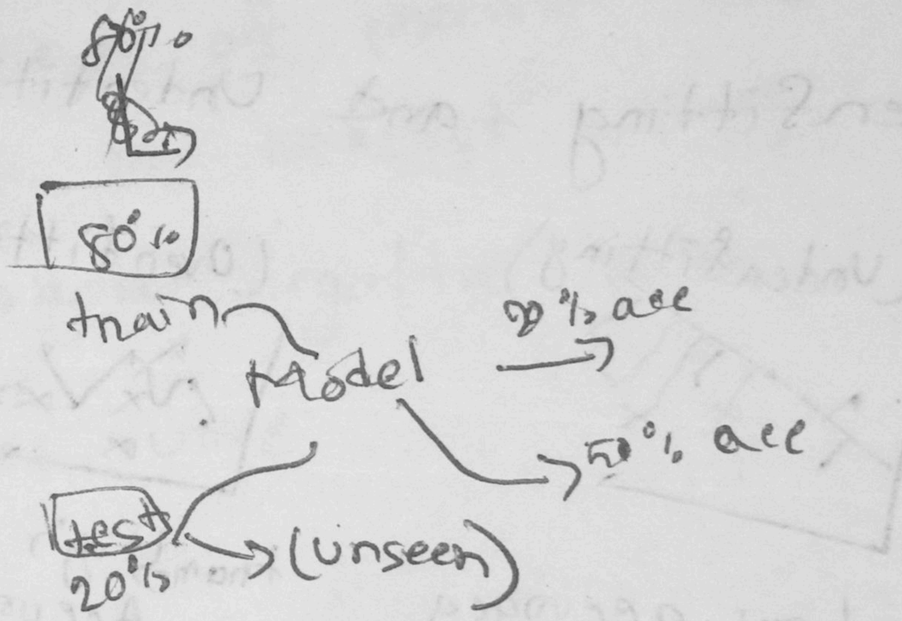
(2) Regularization

(3) Droput

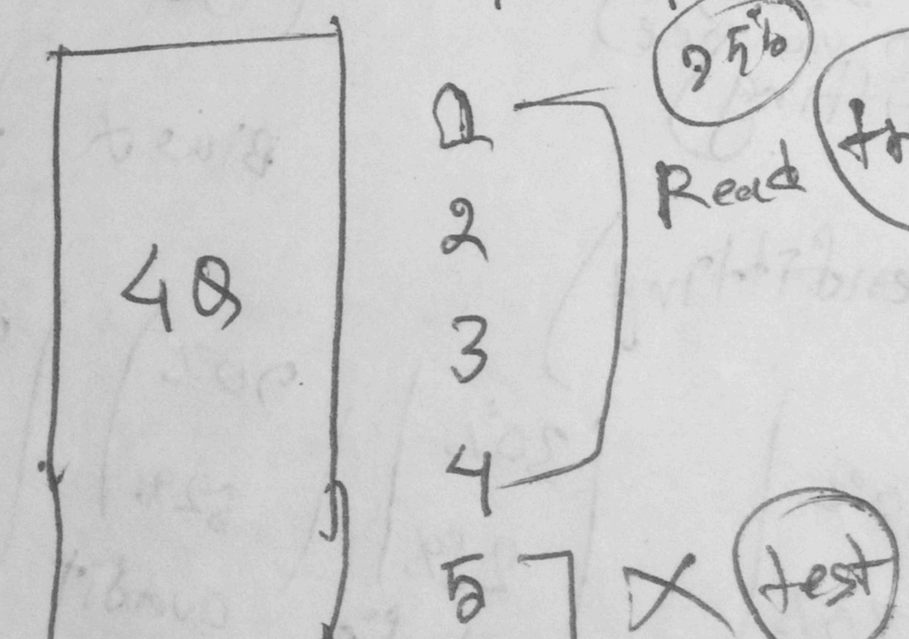
(Lasso, Ridge, elasticNet)

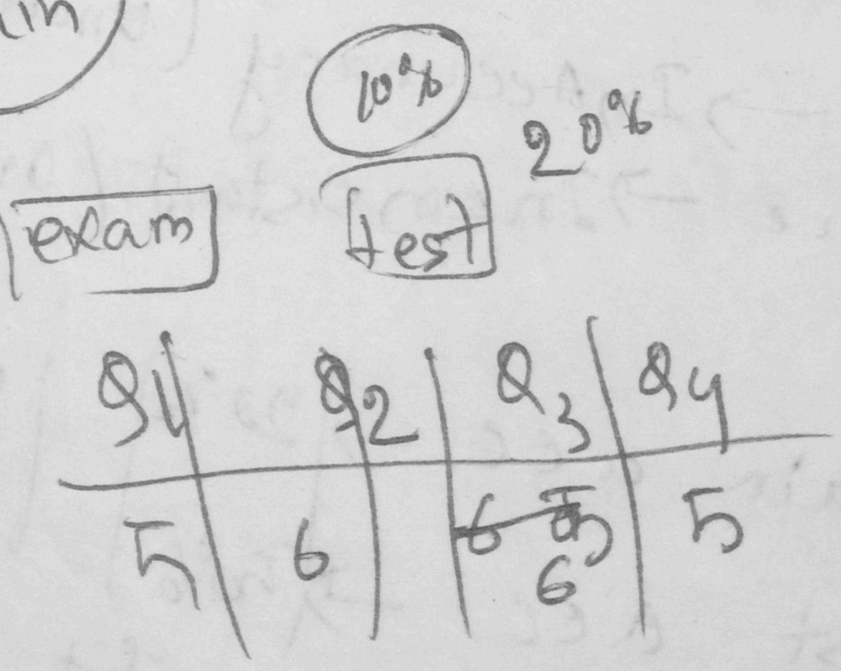






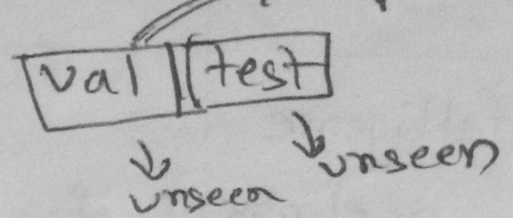
Math program --



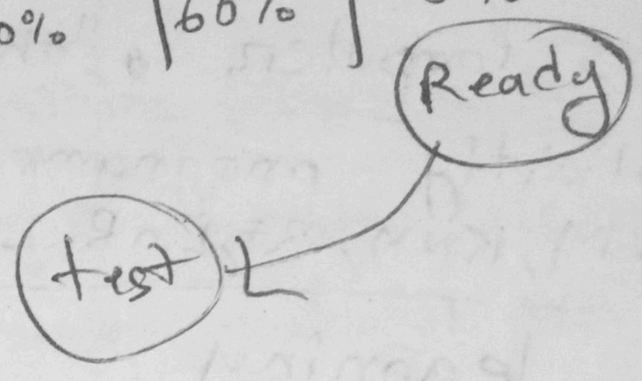


cross-validation (k-fold cross validation)

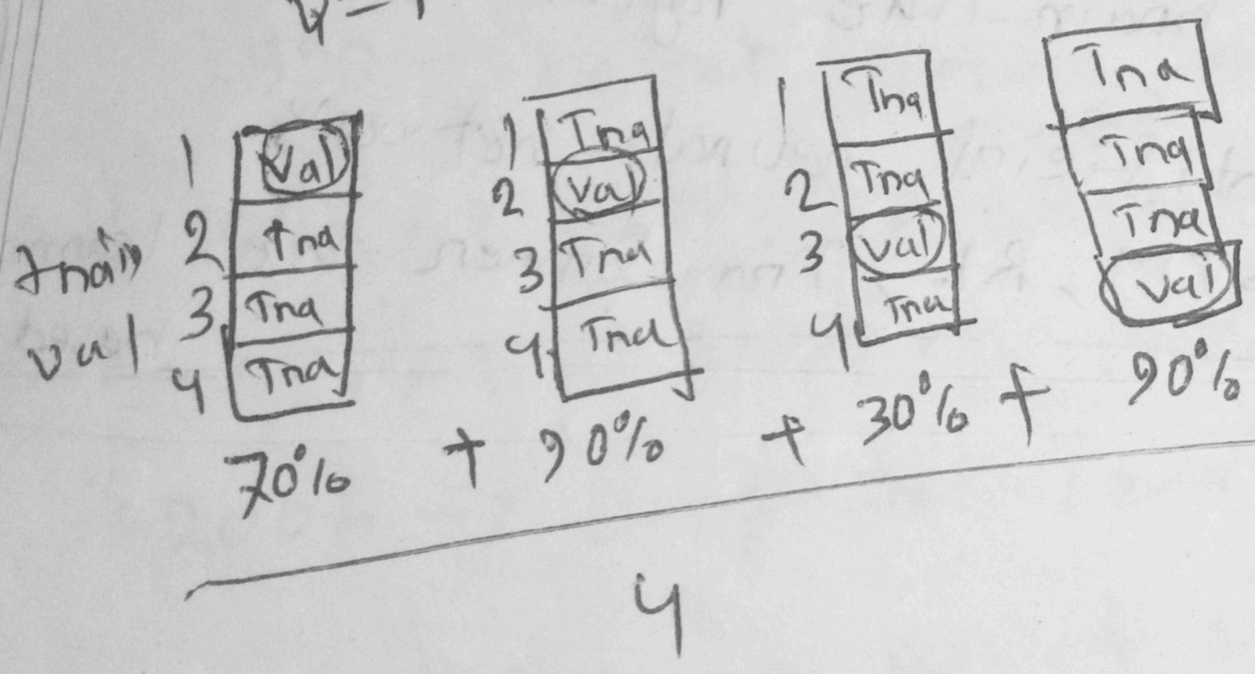
Isolated when training



train → reading → 70% | 90% | 90%
val → coaching → 50% | 60% | 80%



k=4



$$\frac{70\% + 90\% + 30\% + 90\%}{4} = 70\%\ (\text{Avg})$$