31/7/23

Machine Learning

A system is said to learn from experience E, with respect to a task T, measured by performance P, if performance P on T improves (as the experience E increases) with increase in Experience E.

Biasing: DC supply applied to transistor terminal, to work
DC + AC supply → Amplification.

- Voltage divider bias
- Fixed bias
- Self bias
- Potential divider bias

→ Transistor biasing methods

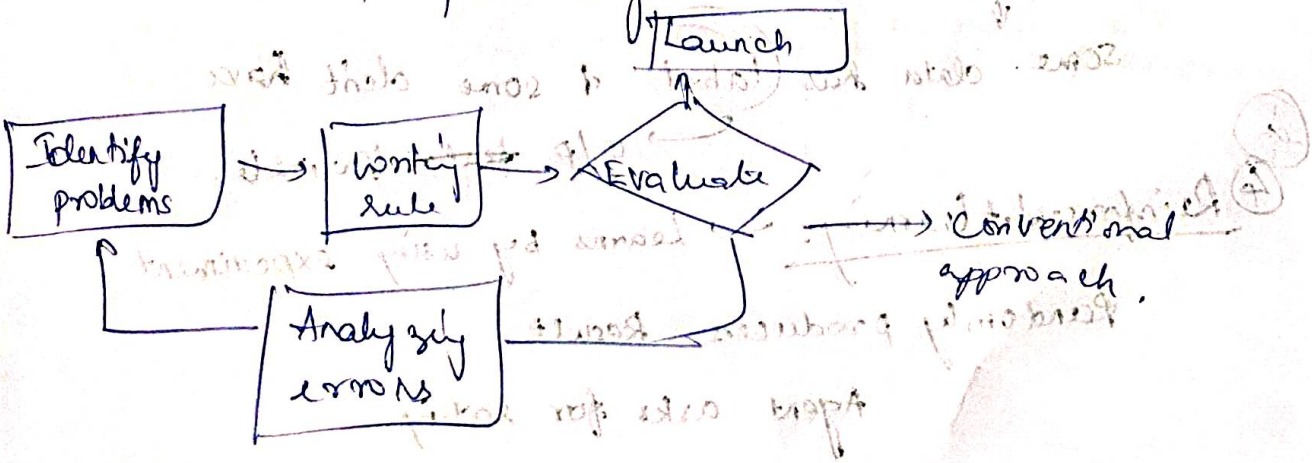
E↑ → P↑

Spam & Ham mail → Program to detect
↓
non spam.

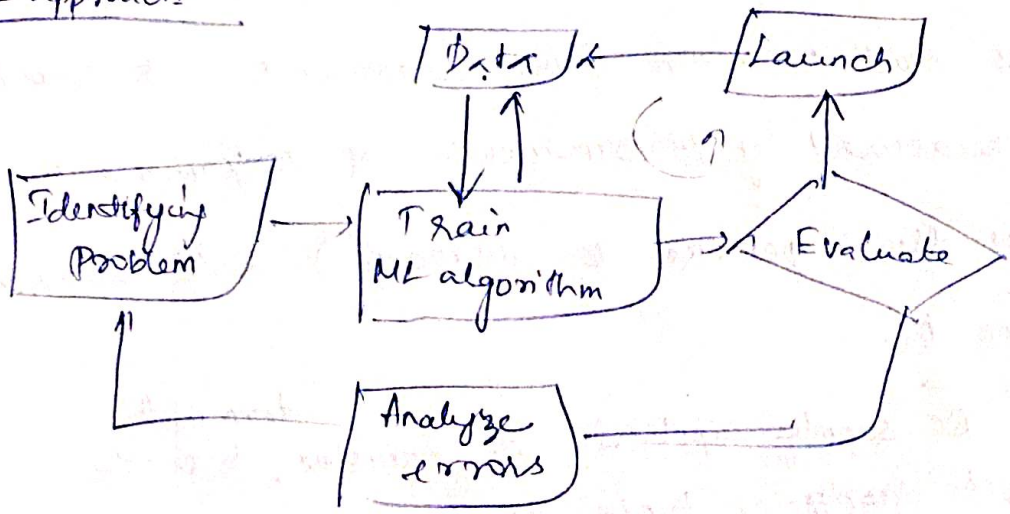
Conventional approach: check for keywords. if detected classify as spam.

if the msg keyword is changed next time, accuracy ↓.

ML approach: identifies frequent words & compares with next new spam mail. Generates keywords. Autoupdation of spam keyword database.



ML approach:



1/8/23

Classification is based on (Criteria)

1.) Data set o/p is present or not (whether teacher is present during the training or not)

Attributes, features → columns in data set.

i/p feature + o/p ("Teacher") → (1) Supervised learning

(best) (2) Unsupervised learning: only i/p variables (features) → creates related slots based on values

Row → record, Data point

without teacher, machine learns.

creates slots by classifying based on distance.

(3) Semi Supervised learning eg: google photos.

↓

some data has (label) & some don't have

↳ o/p ~~feature~~ variable.

(4) Reinforcement learning: → Learns by using experiment.

Randomly produces Result

Agent asks for rating

Based on rating, ~~gives~~ recommends some suggestions

eg: youtube video.

Types of ML:

* Based on Teacher
  - Supervised
  - Un supervised
  - Semisupervised
  - Reinforcement learning

* Based on learning on the fly:
  - Instance based learning
  - Model based learning

* Based on learning method:
  - Batch learning (offline)
  - Online / incremental learning.

Batch learning (offline):
once train the model then can't be changed
if any change, add records with the batch train again. Then put it after updating

Online / incremental learning.
As the data comes features are updated after learning.
data set is not required after this.
(Model : ~~log~~ general eqn.)

Instance Based:
Not learning entire data set.
Based on instance learn, by comparing with previous.

Model Based:
Watch entire data set.
↓
Generalize model.

(i/p) Housing area (sq. ft)	(o/p) Housing price (Lkhs)
150	300+75 = 375
500	1000+250 = 1250
750	1500+375 = 1875
200	400+100 = 500
550	1100+275 = 1375
1000	2000+500 = 2500
100	200+50 = 250
350	700+175 = 875
470	940+235 = 1175
1050	2100+525 = 2625

→ Scatter plot

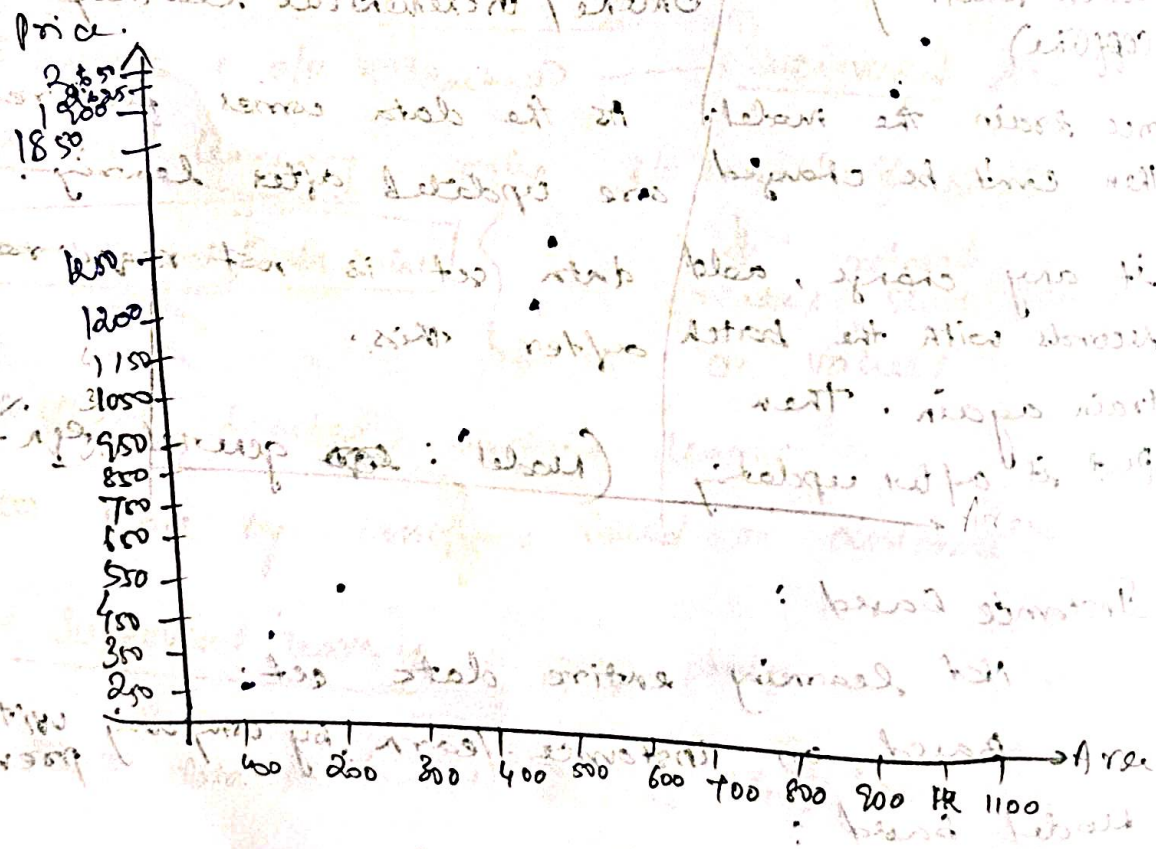
→ Fix eqn.

(Perfect fitting eqn → no error)

$y = 2x + \frac{k}{2}$

$y = 2x + k/2.$

Regression



12.25

29

A

18

25

31

69

17/8/25

Classification : either 0 or 1 based on i/p (only 4 categories)

o/p → discrete variable

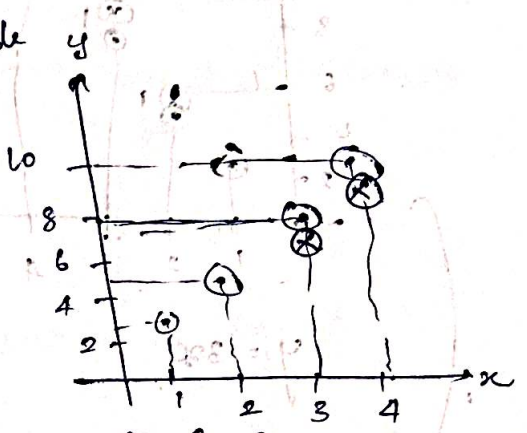
Regression : o/p value base on i/p.

→ continuous variable

Linear regression :

linear eqn. find best fitting curve

record no.	i/p variable x_1	o/p variable (y)
i=1	1	3
i=2	2	5
i=3	3	8
i=4	4	10



$MSE = \frac{0^2+0^2+1^2+1^2}{4} = \frac{2}{4} = 0.5$

linear eqn b/w y & x → $y = mx+c$

$y = \theta_1 x_1 + \theta_0$

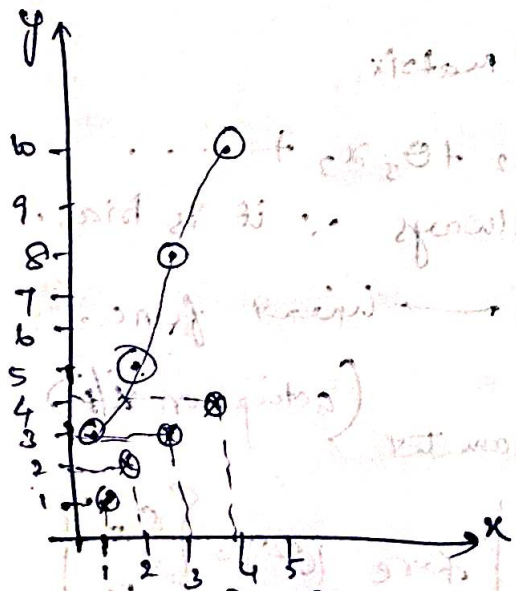
To make it vector

non linear → $x_0, x_1, x_2, x_3 \ldots$

$y = 2x + 1$

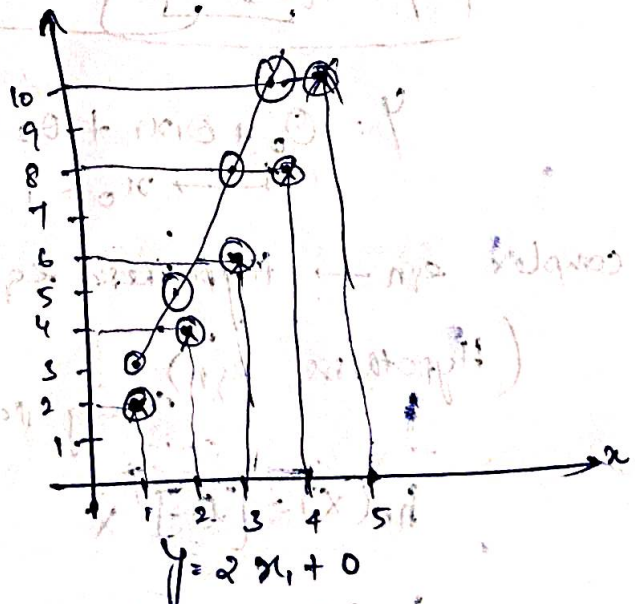
$y = \theta_0 + \theta_1 x_1$ → zero.

→ 1, 2, 3, ...



$y = 1(x_1) + 0$

$MSE = \frac{2^2+2^2+5^2+6^2}{4} = 12.25$



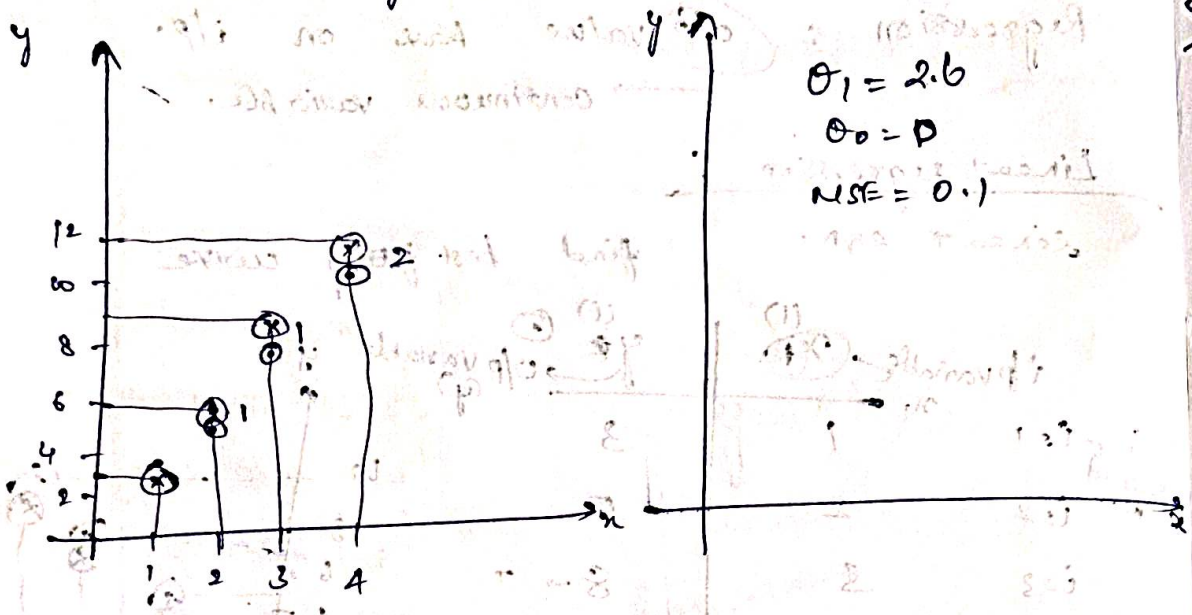
$y = 2x_1 + 0$

MSE =

⊙ → y_i from table (actuals)

⊗ → y_i derived from $y = \theta_0 + \theta_1 x_1$

θ_0, θ_1 → can be negative also.



$y = 3x$ $\quad \theta, \theta_1$ → Tuning parameter.

$y = \theta_0 + \theta_1 x_1$

→ y intercept → Bias in ML

$$MSE = \frac{0 + 1^2 + 1^2 + 2^2}{4}$$

$$= \frac{1+1+4}{4} = 6/4$$

MSE. = 1.5

$$\boxed{y = [\theta]^T x}$$ → matrix

$y = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3 + \ldots$

→ $x_0 = 1$ always ∵ it is bias.

complex eqn → Hypothesis eqn → linear func.

(Hypothesis eqn) (acting on x/p)

Tuning parameter

$h_\theta(x) = [\theta]^T X$

$h_\theta(x) = \theta_0 + \theta_1 x_1$

here $[\theta] = \begin{bmatrix} \theta_0 \\ \theta_1 \end{bmatrix}$

$[x] = \begin{bmatrix} 1 \\ x_1 \end{bmatrix}$ → x_0,

i) find [J] which will give min error:

ii) Graph b/w [θ_0, θ_1], error (↑ vertical axis)

Tuning param

→ ∞ combinations can be formed

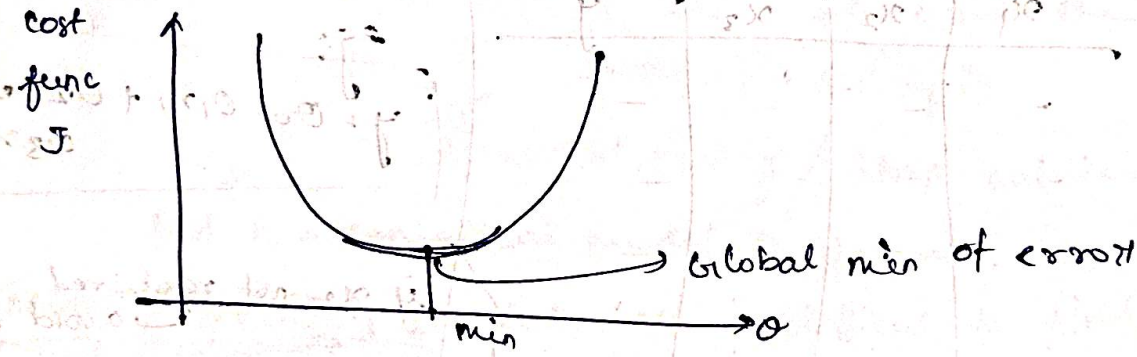
Gradient descend algorithm

Step size → θ_0, θ_1 combination ↓

error function / cost function → bowl like shape in 3d..

→ to minimize the error

for linear regression → cost function is bowl



cost func

$$J(\theta_0, \theta_1) = \frac{1}{M} \sum_{i=1}^{M} (\theta x_1^i - y^{(i)^2})$$

$$J_\theta = \frac{1}{M} \sum_{i=1}^{M} (h_\theta(x^i) - y^i)^2.$$

to find minima → differentiate the eqn.

Here wrt to θ_0, θ_1 (twice)

equate it to zero.

18/8/23

Linear Regression. → most simplest hypothesis

∴ linear line eqn

$y = mx + c$

$h_\theta(x) = \theta^T x$

hypothesis.

Cost func:

$$J_\theta = \frac{1}{m} \sum_{i=1}^{M} (h_\theta(x^i) - y^i)^2$$

where θ = Tuning params → $\begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \\ \vdots \\ \theta_n \end{bmatrix}$

x = features → $\begin{bmatrix} 1 \\ x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$

o/p → $y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix}$

Data set

	x_1	x_2	x_3	y^i
$i=1$	–	–	–	–
$i=2$	–	–	–	–
$i=3$	–	–	–	–
$i=M$	–	–	–	–

eg:

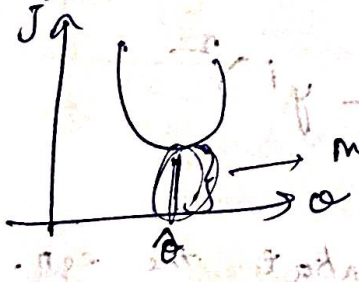
$y = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3$

if x_3 not required → avoid.

if x_1 – more impact more in eqn, x_1, x_1^2 etc.

disadvantage in NL.

for cost func:



step size is for tuning tuning parameters.

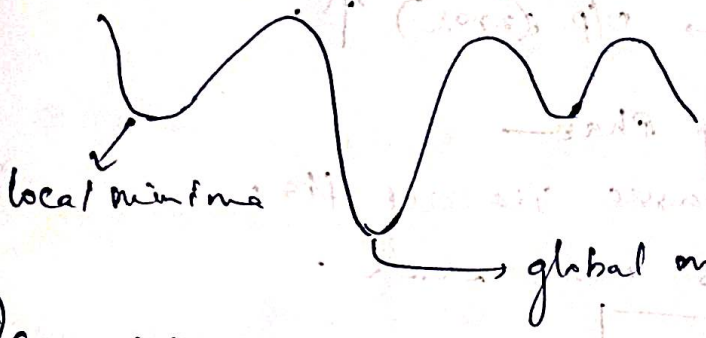
(θ_0, θ_1)

* Step size large: can't accurately detect.

} large step size.

* Small step size: Takes more iterations.

* Hence optimum step size is required for detecting global minima.



Constraints in ML:

Algorithm → model for data set : for ALL

Adv: when the model is fitted for data set

↳ give i/p → o/p is predicted using the model (line / eqn)

Mathematical mostly / Graphical model

Best mathematical model is found.

Missing values @ data set → difficult to find mathematical model

Quality: poor quality / accuracy.

Quantity → always should be huge

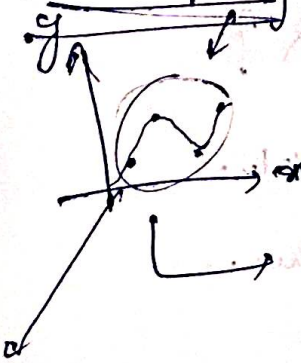
↳ less no. of data

more computational complexity for machine

(or)

algorithm should be as simple as possible so that machine avoids complexity.

over fitting — touches all points.



best fitting / good fitting.

(Non linear)

or

↳ Not good for predicting for new i/p.

∴ over fitting problem

18/8/22

Over fitting: creating a model → for the given data set the model exactly fits.

Disadv: i/p (new) → o/p (error) ↑.

↳ Testing phase

Under fitting: not passes via all pts.

Testing → throws more error.

$\theta = (X^T X)^{-1} X^T y$ → θ matrix for minimal error.

↓

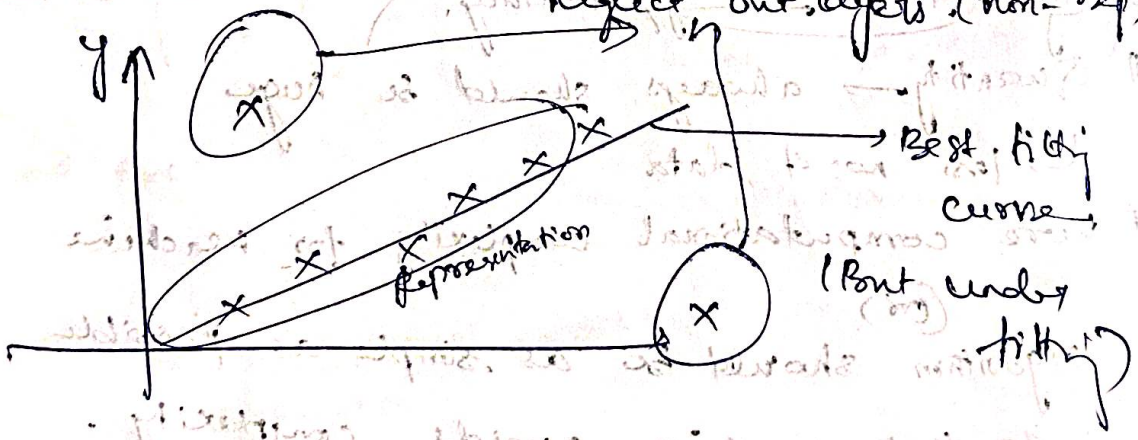
normal eqn.

Modeling: best fitting eqn / algorithm for relation b/w i/p & o/p.

Training: data set, error

Testing: data set, error

Non representative data: → focus on majority.



if we consider out layers → over fitting.

Q) what is training error & testing error?

NOISE → less accurate than correct value

(or) unwanted value.

21/8/23

No free lunch theory:

- if no insight to dataset
- no idea regarding algorithm accuracy.

θ → Tuning / Model param.

(7.) Bias (θ_0) — Error rate in training.

(8.) Variance → (Training - Testing) Error
↓ rate of change

error rate b/w Training & Testing error.

(9.) Challenge/constrain in ML: — Irrelevant feature.

Consider only normalization (Range) data.

eg:	1000 sq ft → 1	12 Lks → 12	
	2k sq ft → 2	14 lks → 14	

→ Identifying irrelevant feature is a big challenge
↳ creating complex Model
best is found in Testing phase.

(10.) Validate → small testing like (like Mock test)
↓ instead

Validation data set from Training dataset.
⇓ Test the model
find Error
↳ if more error → Tune model

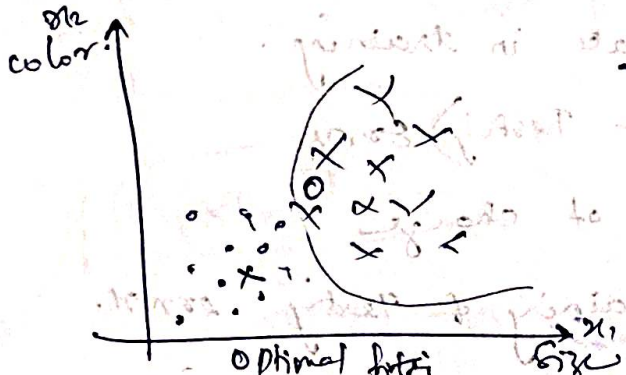
Cross validation → batches (K fold) of validation set
⇓
Test by batch by batch.
if any batch is v. less error
⇓
Go for testing

if validation is ok, then test

Training data set + validation data set.

→ Then go for final Testing phase

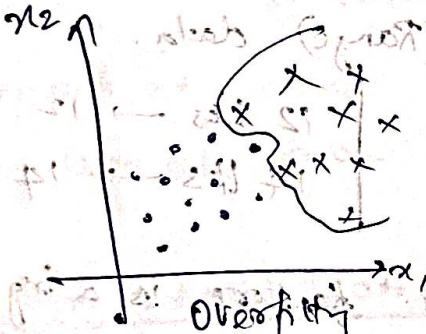
Regularisation: → All methods to avoid over fitting

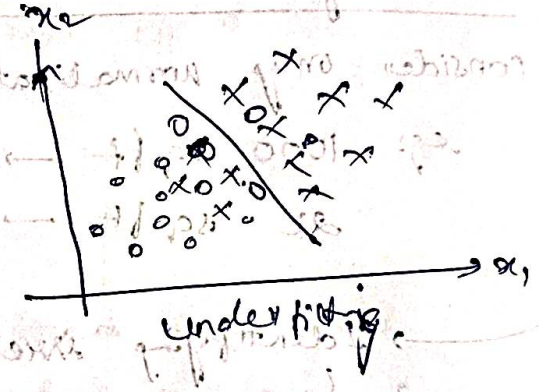


Size x_1	color x_2	Tumor (o/p)
—	—	T
—	—	NT
—	—	T
—	—	T

22/8/23

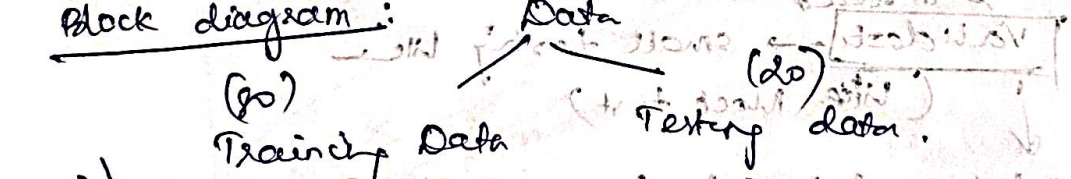
→ small error, ↑ accuracy

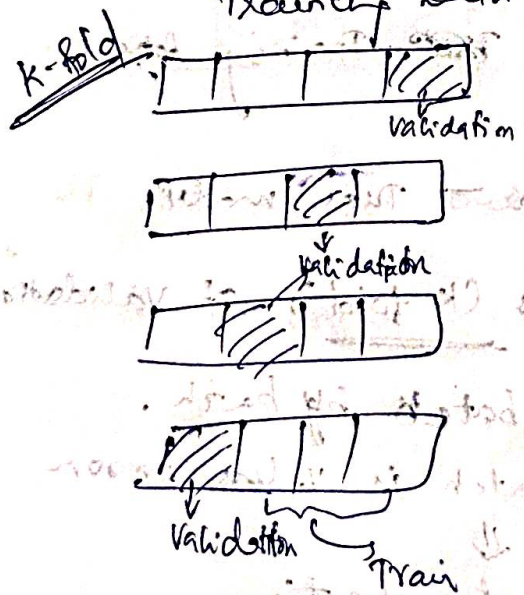




logistic eqn.

Block diagram:





Figures 2/Amk

more training data → desirable data — more accuracy.

Regression: predicting : o/p ⊙ continuous values

classification: o/p — discrete values.

Regularisation:

* ↓ model complexity.

* Model param (tuning param) For model ⚡ Hyper param → used for tuning for algorithm.

Linear regression

↳ other than tuning param, to have a model

other param ⟶ step size — Hyper param.

find o/p & i/p correlation — Heap / correlation map. (x_1, x_2, x_3)

to find relation b/w o/p & one of the i/p x_1, x_2, x_3.

feature selection: selecting x_1 & x_2 (features)

Feature extraction: New feature created from existing feature

sampling: while taking data consider generalise.

↓ not good or bad performing

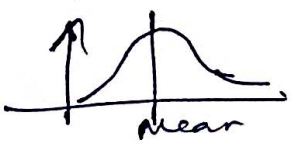
means model is good / bad.

↳ sampling noise (error) → sampling bias (model)

Generalised sample is good

mean, radian, skew, sckotosis.

Skewness: bell shaped curve tends towards one side due to sampling noise.



22/8/23

Unit-2

16m (X)

Steps (end to end) in ML

↳ & is a step

① Getting big pic of entire data & problem

① Kaggle, Keras → Dataset websites
② Aios
③ UCI repository
UCIrvine

Repository

② Visualization of data (Graph - plotting) → eg: scatter plot
↳ come to conclusion why there
↳ heat map - correlation b/w i/p & o/p.

convolution, correlation

↳ $x(k)\ h(n-k)$
i/p filter coeff

$x(k)\ h(n+k)$

$x_1(k)\ x_2(n-k)$

$$y(n) = \sum_{-\infty}^{\infty} x(k)\, h(n-k)$$

Convolution.

(or)

$$y(n) = \sum_{-\infty}^{\infty} x(k)\, x(n-k)$$

when 2 signals same ↔ high correlation.

correlation coefficient = 1.

↳ tends to 1

if 2 features same (highly correlated)
then 2 features not needed.

20/8/23.

Univariate : o/p – 1 variable

Multivariate o/p – many variables.

Instant / Model based – which is best suited?

Online / Offline → mostly

Sustaa plot - 3d.

individual feature has to be visualized.

Heat map. Histogram.

if 2 features are highly correlated

drop 1 feature

correlation coeff → tends to 1.

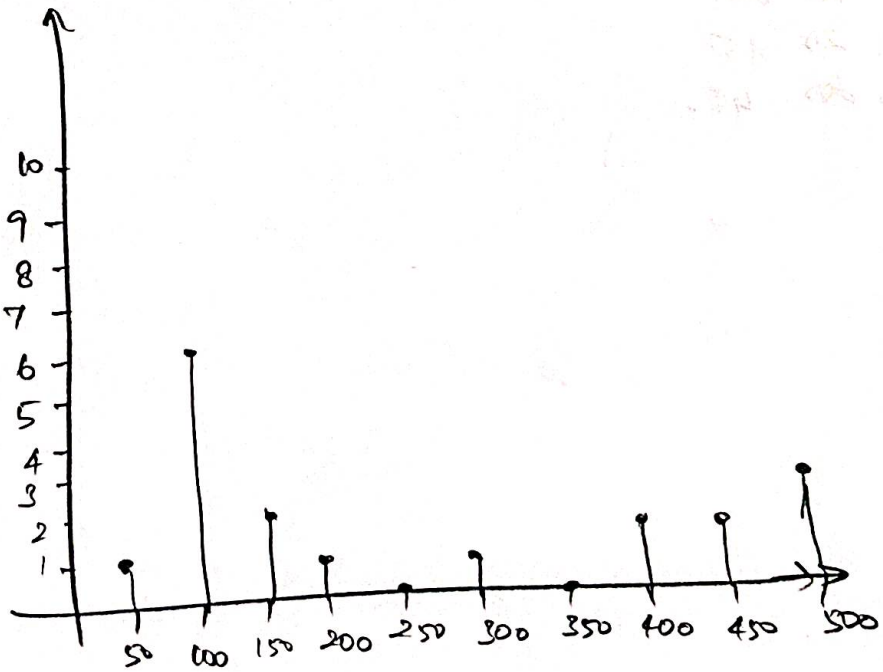
Histogram: x axis feature data

y axis frequency

for o/p (or) i/p -

x

1.) 300
2.) 100
3.) 150
4.) 100
5.) 150
6.) 200
7.) 50
8.) 150
9.) 400
10.) 450
11.) 300
12.) 100
13.) 100
14.) 450
15.) 250
16.) 100
17.) 500
18.) 500
19.) 100
20.) 400



Histogram

$y = 2x + (k) \rightarrow 5$.

	x	y
1.	5	15
2.	10	25
3.	5	15
4.	5	15
5.	5	15
6.	10	25
7.	10	25
8.	20	45
9.	20	45
10.	5	15
11.	5	15
12.	15	35
13.	5	15
14.	10	25
15.	5	15
16.	15	35
17.	20	45
18.	10	25
19.	20	45
20	20	45