CHP 7

- Bias
- variance
- underfitting
- overfitting
- Bias-variance tradeoff.

last year question paper

30 question

5 years → memorize

Exam → ≃100%

P_1	Q_1	≃	Ans 1
P_2	Q_1	≃	Ans 2
P_3	Q_1	≃	Ans 3

Vary

Overfitting :

model is not learn the reason behind the O/P for the I/P data point

learn

Raw data → useful information + noise

- Model is not generalized.
- training set accuracy :- ≃100%
identification → testset accuracy << training accuracy

Data → Train, testing

variance : for an unknown I/P ~ Ans. = Error → Variance

reason : - less number of sample data

model → Training data → over fit

Predict data set (training data set → high

But unknown data set → accuracy low

underfitting : not learn too much.

- training set accuracy is less
- test set accuracy is also less.
- Error introduced because of underfit is called bias.

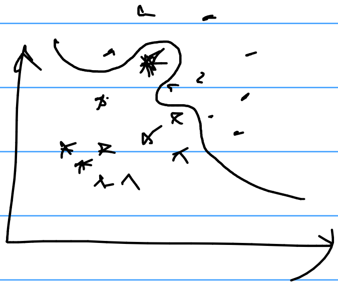
- reason :
  - few number of parameters.
  - less training itteration

underfit ↑ overfit ↗ Bias-variance trade off

H/W : Q_1 write 4 diff of bias vs variance

Q_2 write 4 diff of Underfit and overfitting of a model.

Q_3 : what do you mean by bias-variance trade off



Ch10 KNN algorithm

↓

K - nearest neighbour algorithm

↓

?

KNN algorithm ← supervised classification algorithm

- Basic KNN
  - Discreate o/p variable
  - Real/Continuous o/p value variable
- weighted KNN
  - Discreate o/p variable
  - Real/continuous o/p value variable

Basic KNN (Discreate)

eg:

Soln

	Height	weight	target	distance	nearest neighbour
K=3 given	150	50	M	8·06	
	155	55	M	2.24	1
	160	60	L	6·71	3
	161	59	L	6·40	2
	158	65	L	11·05	
unknown →	157	54	?		

among K=3 neighest neighbour
M → 1
L = 2

So
157 54 L

note: Distance : Euclidean distance

X	Y
x_1	y_1
x_2	y_2
x_3	⋮
⋮	
x_n	y_n

$$E(x,y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$$

Basic KNN (Continuous/Real data)

Euclidean distance → Solution Part

Qs

K = 3

Height	Weight	Target	Distance	nearest-neighbour
150	50	1.5	8.06	
155	55	1.2	2.24	1
160	60	1.8	6.71	3
161	59	2.1	6.40	2
158	65	1.7	11.05	
157	54	?		

So, o/p for $(157, 54) = \frac{1.2 + 1.8 + 2.1}{3}$

$= 1.7$

- How to decide value of K

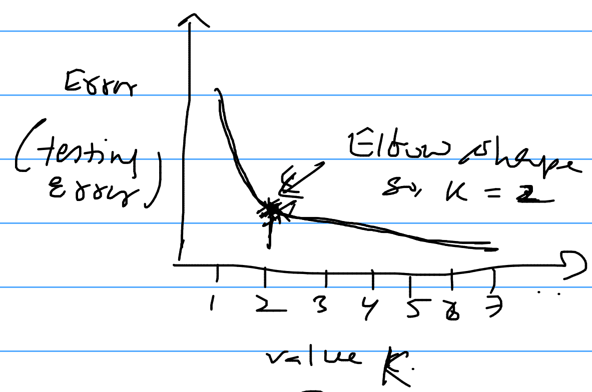
Ans: Approach 1 $K = \sqrt{n}$ ← Sample size

Approach 2 K ← odd number (to avoid any tie)

Approach 3 : Elbow method.

- Consider odd K near to Elbow shape

	X	Y	Target
Training data	x_1	y_1	M
	x_2	y_2	L
	x_3	⋮	M
	⋮		⋮
	x_{50}	y_{50}	M
Testing data	x_{51}	y_{51}	L
	x_{52}	y_{52}	L
	⋮		M
	x_{60}	y_{60}	L



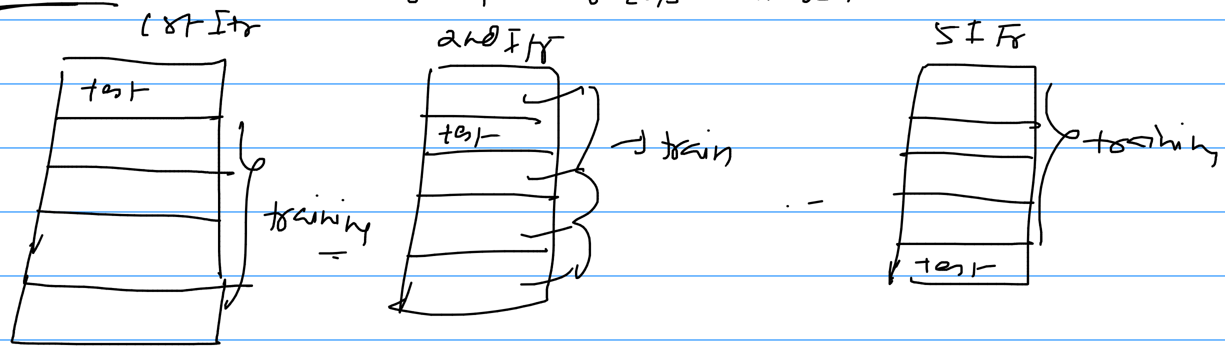
K = 2

① - find o/p label for all testing data points

② - find test accuracy. (By comparing given o/p level for test data and model predicted o/p for test data

$$\text{Error} = (100 - \text{Accuracy})\ \%$$

Approach: use 5-fold cross validation

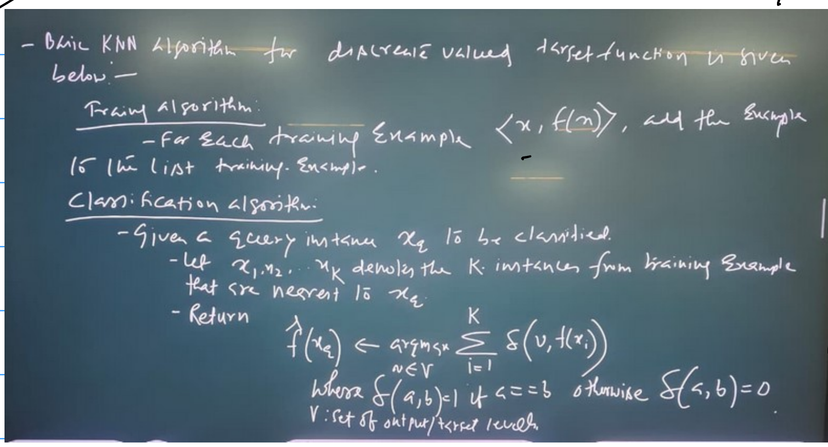


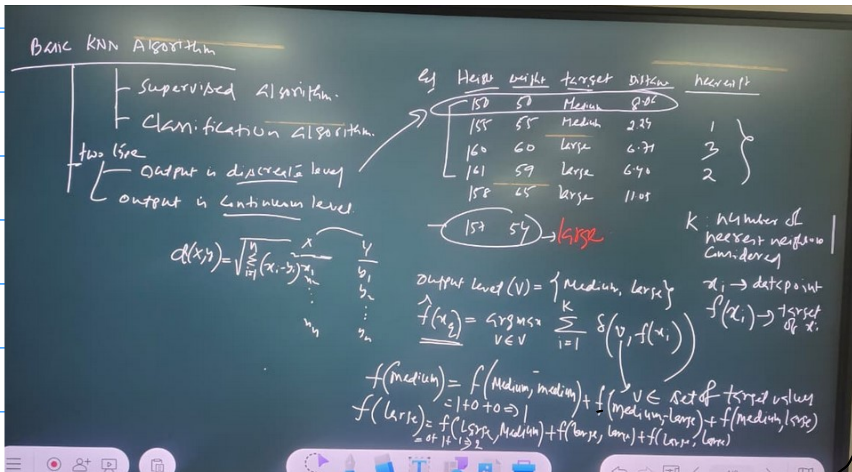
$k = 2$

$$\text{test accuracy} = \frac{\sum_{i=1}^{n} (\text{test accuracy Itr}_i)}{5}$$

$$\text{Error} = (100 - \text{test accuracy})\ \%$$

- plot the elbow curve





Dataset

Soln

Height	weight	target	d's	neighbour
150	50	M	8.06	
155	55	M	2.24	1
160	60	L	6.71	3
161	59	L	6.40	2
158	65	L	11.05	
157	54	? L		

i/p → o/p

$\langle x, f(x) \rangle \leftarrow$ training sample

$x : (150, 50)$

$f(x) = M$

$x_q : \langle 157, 54 \rangle$

for $k = 3$

$x_1 : (155, 55)$

$x_2 : (161, 59)$

$x_3 : (160, 60)$

Predicted $\hat{f}(x_q)$: o/p of x_q

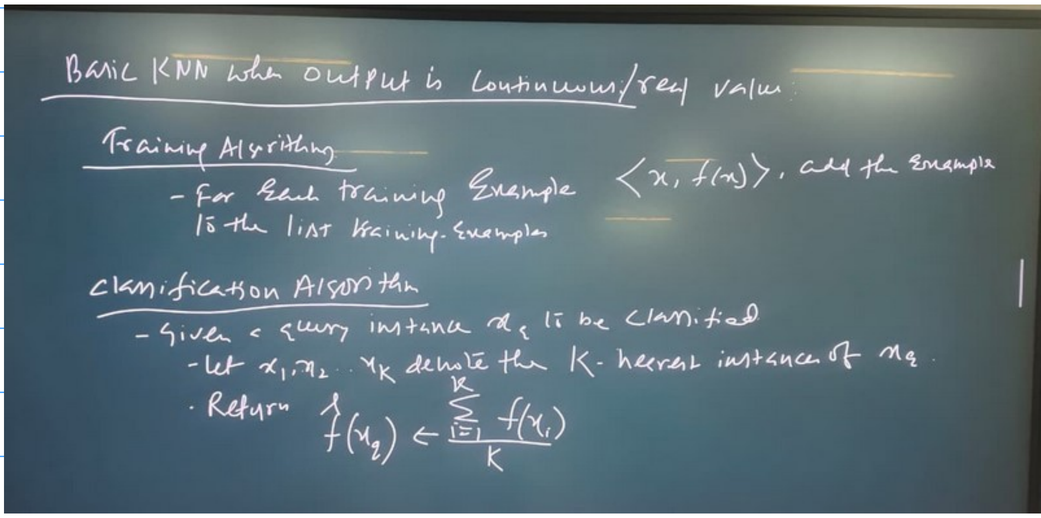
V : set of target o/p

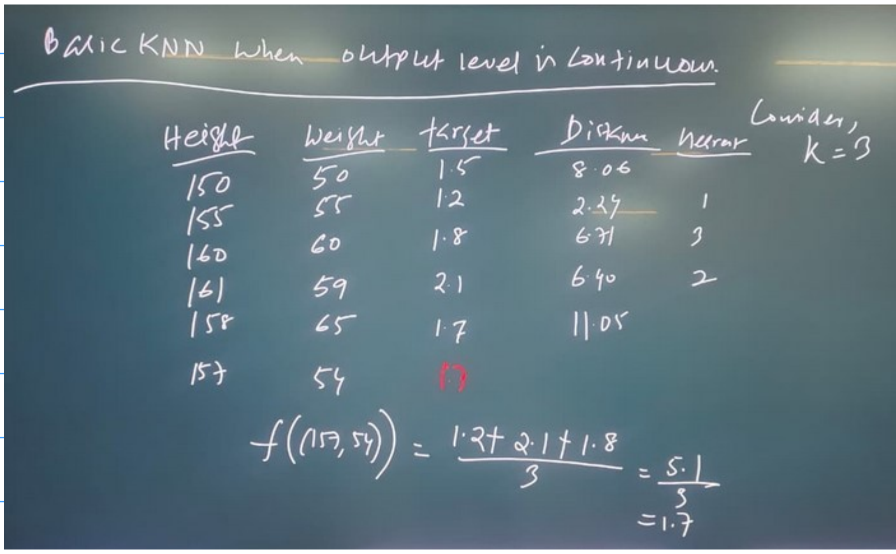
so, $V \Rightarrow \{M, L\}$

$$\text{argmax} \begin{cases} \delta(M,M) + \delta(M,L) + \delta(M,L) = 1 \\ \delta(L,M) + \delta(L,L) + \delta(L,L) \end{cases}$$

= 2

= L





$f(\langle 153, 54 \rangle) =$

Discreate o/p

Distance weighted KNN Algorithm

- Supervised algorithm
- Classification algorithm
- output types
  - Discreate
  - real valued

for discreate output value

$$\hat{f}(x_q) \leftarrow \underset{v \in V}{\arg\max} \sum_{i=1}^{K} w_i \, \delta(v, f(x_i))$$

where, $w_i = \frac{1}{d(x_q, x_i)^2}$

Height	Weight	target	distance	nearest pt	$w_i = \frac{1}{(\text{distance})^2}$
150	50	M	8.06		
155	55	M	2.24	1	$1/(2.24)^2 \approx 0.45$
160	60	L	6.71	3	0.15
161	59	L	6.40	2	0.16
158	65	L	11.05		
157	54	M			

$$f((157,54)M) = 0.45 \times f(M,M) + 0.15 \times f(M,L) + 0.16 \times f(M,L)$$
$$= 0.45 \times 1$$
$$= .0.45$$

$$f((157,54)L) = 0.45 \times f(L,M) + 0.15 \times f(L,L) + 0.16 \times f(L,L)$$
$$= 0.15 \times 1 + 0.16 \times 1$$
$$= 0.31$$

$\therefore 0.45 > 0.31$

Distance Weighted KNN for real valued output data

→ continuous

$$\hat{f}(x_q) \leftarrow \frac{\sum_{i=1}^{K} w_i f(x_i)}{\sum_{i=1}^{K} w_i}$$

where, $w_i = \frac{1}{d(x_q, x_i)^2}$

Basic KNN when output level is continuous

Consider, $k = 3$

Height	Weight	target	Distance	nearest
150	50	1.5	8.06	
155	55	1.2	2.24	1
160	60	1.8	6.71	3
161	59	2.1	6.40	2
158	65	1.7	11.05	
157	54	1.7		

$$f((157,54)) = \cancel{\frac{1.2 + 2.1 + 1.8}{3}} = \cancel{\frac{5.1}{3}} = 1.7$$

$w_i = \frac{1}{(\text{distance})^2}$

$\frac{1}{(2.24)^2} = 0.45$

$1/(6.71)^2 = 0.15$

$1/(6.40)^2 = 0.16$

$f(157, 54)$

$\Downarrow$

$$\frac{0.45 \times 1.2 + 0.15 \times 1.8 + 0.16 \times 2.1}{0.45 + 0.15 + 0.16}$$

$= 1.507$ ✓

$$\sqrt{(157-150)^2 + (54-50)^2}$$

= ✓

Note :

(x)
↑
- KNN algorithm is based on distance. So, if the i/p are in different scale then it might produce incorrect result.

⇓ Soln is make all inputs are in same scale

- z - normalization ✓
or
- min - max standardization ✓ (in Exam for standardization you can prefer either of this two method).

⇓ then apply KNN algorithm.