$\lambda \uparrow \Rightarrow \beta \downarrow$
bcoz entire term should be ↓ so β↓.

Lasso Regression:

$$\text{Loss function} = \sum_{i=1}^{N} (y_i - \beta_0 - \sum_{j=1}^{p} \beta_j * x_{ij})^2 + \lambda \sum_{j=1}^{p} |\beta_j| .$$

λ is tuning parameter - decided the amt of regularization needed.

Logestic Regression

- Mainly used for classification, but formula is derived from regression.
- Points on line indicates value of y. → Regression.

$p(x) = \beta_0 + x \cdot \beta_1$ (wrong) — We get any value for y bcoz method should return probability. (0 to 1)

$P(Y=1|x) \Rightarrow P(x)$
$P(Y=0|x) = 1 - P(Y=1|x)$

The transformed func is called as logestic (logit) func.

$$\log\left[\frac{p(x)}{1-p(x)}\right]$$

$p(x) = P(y=1/x)$

Value in the range of 0 to 1.

$$\log\left[\frac{p(x)}{1-p(x)}\right] = \beta_0 + x \cdot \beta_1$$

$$\frac{p(x)}{1-p(x)} = e^{\beta_0 + x\beta_1}$$

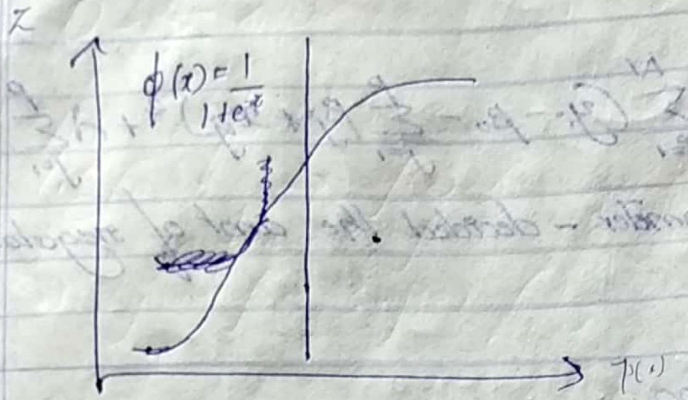
$$p(x) = e^{\beta_0 + x\beta_1} - p(x) \cdot e^{\beta_0 + x\beta_1}$$

$$p(x)(1 + e^{\beta_0 + x\beta_1}) = e^{\beta_0 + x\beta_1}$$

$$p(x) = \frac{e^{\beta_0 + x\beta_1}}{1 + e^{\beta_0 + x\beta_1}}$$

$$p(x) = \frac{1}{1 + e^{-(\beta_0 + x\beta_1)}}$$ ⇒ logistic func / sigmoid func

$z = \beta_0 + x\beta_1$



$p(x) = \max \Rightarrow$ class label = 1
$= \min \Rightarrow$ class label = 0

Derivative of sigmoid func.

$$g'(z) = \frac{d}{dz}\frac{1}{1+e^{-z}}$$

$$= \frac{1}{(1+e^{-z})^2}(e^{-z})$$

$$= \left(\frac{1}{1+e^{-z}}\right)\cdot\left(1-\frac{1}{1+e^{-z}}\right)$$

$$= g(z)\,(1-g(z))$$

Multiple Logestic Regressions

$$\log\left[\frac{p(x)}{1-p(x)}\right] = \beta_0 + x_{11}\beta_1 + x_{12}\beta_2 \ldots$$

Max likelihood estimation.

Decision boundary is, (linear classifier)

$\beta_0 + x\beta_1 = 0.$

. Cost func

$$l(\beta_0,\beta_1) = \prod_{i:y_i=1} p(x_i) \prod_{i':y_{i'}=0} 1-p(x_{i'})$$

$y = 1$ (dog) $= 1$ $\qquad$ $y = 0$ (cat) 0.

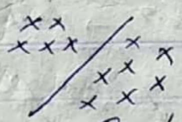
Classification Algo:

- Artificial Neural N/w
- Support Vector Machine
- Decision Trees
- Linear Discriminant Analyses.
-

ANN:

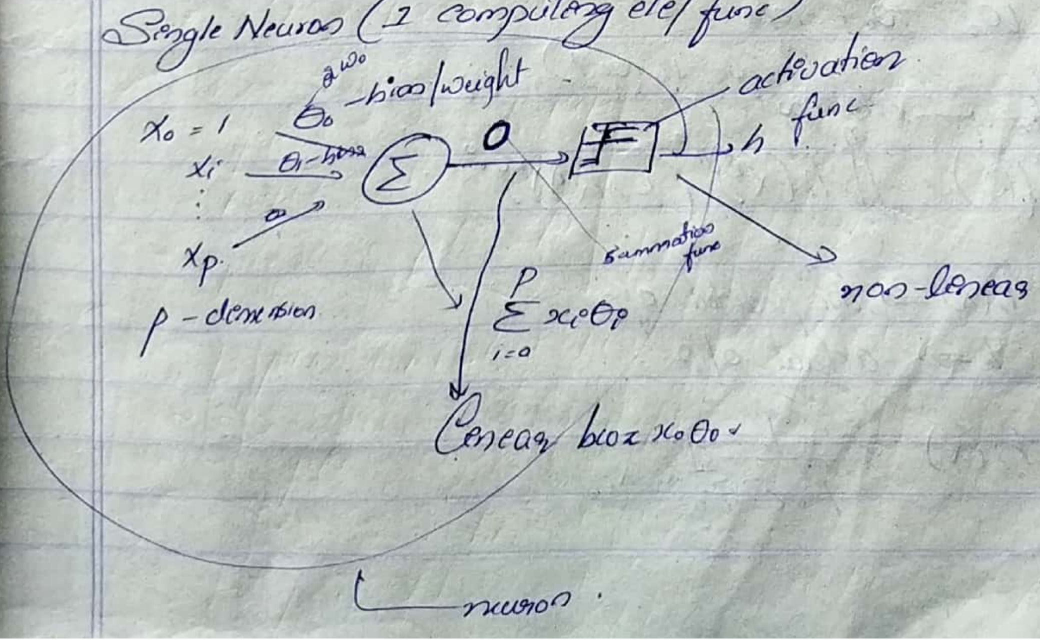
- Derived from human brain architecture.
- Neurons — main structural component of brain.
- Particular neuron will respond not all depending on i/p. Chemical component gets updated with i/p.
- ~~Neuro~~ Human brain is composed of 100 billion nerve cells called neurons.
- They are connected to other 1000 cells by Axons.
- A/u human: Multitasking / ‖ task at a time

Linear classification: Can draw line that separate 2 classes.



Non-linear classification: Can't draw line that separate 2 classes

Single Neuron (1 computing ele/func)



LSE → to know β_0, β_1 → Linear reg.

not on the

$$O = \sum_{i=0}^{d} x_i w_i$$

Activation function.

1) Linear ⇒ $f(x) = x$.

2) Sign func ⇒ $f(x) = \begin{cases} 1 & \text{if } x \geq 0 \\ 0 & \text{otherwise} \end{cases}$ (non-linear)

3) Sigmoid func ⇒ $f(x) = \frac{1}{1+e^{-z}}$ (0 to 1) $z = \Sigma w x$

calculate error for each example

Stochastic Online Gradient Descent.

— for getting weights.

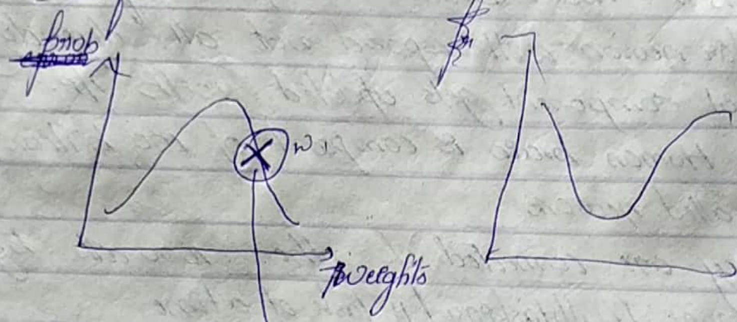
— learning weights

— Cost func / error func.

or → min

prob → max.

Cost func.



Gradient Descent.

1) Initialize weights with a Random values.

(move to dir → peak)

2) Compute the slope of the point.

3) Update w.

if sloped, max ⇒ then w should be ↓

$w_0 = w_0 + \text{slope}$

↳ (-ve, ↓, +ve ↑)

~~Cost func~~ — Likelihood.

$$L(w) = \prod_{i=1}^{m} h(x^i)^{y^i} (1-h(x^i))^{1-y^i}$$

$h(x)$ ⇒ the predicted o/p.

y ⇒ actual o/p.

$$h(x) = \frac{1}{1+e^{-\sum_{i=0}^{d} w_i x_i}}$$

if $y=1 \Rightarrow h(x^i) = 1$ $\quad L(w)=1$

$y=0$ $\quad L(w) = 1$

$\log(ab) = \log a + \log b$

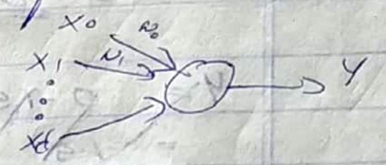
$l(w) = \log(L(w))$

$\Rightarrow \sum_{i=1}^{m} y_i \log(h(x^i)) + (1-y^i)\log(1-h(x^i))$

$\frac{\partial}{\partial w_j} L(w) = (y - h(x^i))\, x_j$ → slope

$w_j = w_j + (y - h(x^i))\, x_j$ → jth feature

0 ← actual o/p, predicted o/p



$(w_0 = w_0 + (y - h(x))\, x_0)$

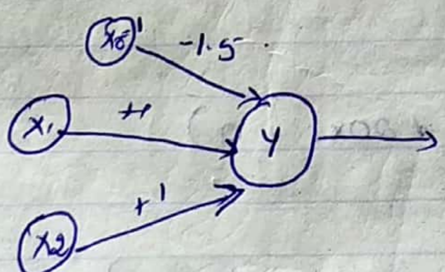
$w_j = w_j + \gamma(y - h(x))\, x_j$

γ → learning rate

↳ you are updating variable with small value

1) $Y = S(x_1 + x_2 - 1.5)$

↓ sign $\quad$ ↳ Bias



$W = [+1 \ +1 \ -1.5]$

$y = \text{sign}(0 \times 1 + 0 \times 1 - 1.5)$

$= \text{sign}(-1.5) < 0$

$= 0$

x_1	x_2	y
0	0	0
0	1	0
1	0	0
1	1	1

found

implements functionality of ~~add~~ and →

if weight changed no and.

2) $w = [-30 \quad +20 \quad +20]$

$$\text{Sigmoid} = \frac{1}{1+e^{-z}}$$

x_1	x_2
0	0
0	1
1	0
1	1

$$\Sigma wx = (-30)\times 0 + 20\times 0$$
$$= \underline{0}$$

$$\cancel{y = 0 - \cdot 5 = -1.5}$$

$$y_1 = \text{Sigmoid}\left(\overset{x_0 \text{ (bias)}}{-30\times 1} + \overset{x_1}{20\times 0} + \overset{x_2}{20\times 0}\right)$$
$$= \text{sigmoid}(-30)$$
$$= \frac{1}{1+e^{+30}}$$
$$= 9.35\times 10^{-14}$$
$$\cong \underline{0}$$

$$y_2 = \text{Sigmoid}(-30\times 1 + 0\times 20 + 20\times 1)$$
$$= \text{Sigmoid}(-10)$$
$$\cong \underline{0}$$

$$y_3 = \text{sigmoid}(-30\times 1 + 20\times 1 + 0)$$
$$\cong 0$$

$$y_4 = \text{sigmoid}(-30\times 1 + 20\times 1 + 20\times 1) = 0.99$$
$$\cong \underline{1}$$

∴ AND Gate

for 2 class label, Sigmoid func : $\frac{1}{1+e^{-z}}$

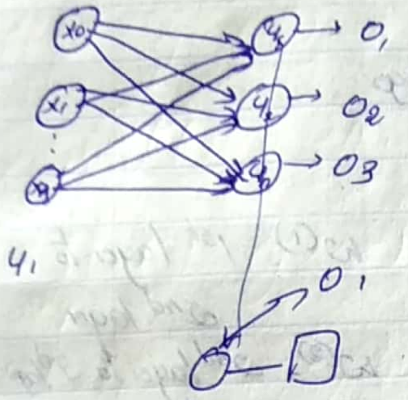
> 2 — sigmoid func — softmax func

Multiclass ⇒ No of class > 2

i.e Binary ⇒ 0, 1

Multiclass ⇒ 0, 1, 2, 3, . .

3 class label 3 neuron



By find prob we assign class.

$O_1 = \Sigma \omega x$

Softmax func, $y_1 = \frac{e^{O_1}}{e^{O_1}+e^{O_2}+e^{O_3}}$ $n = 3$

Total weight

$y_2 = \frac{e^{O_2}}{e^{O_1}+e^{O_2}+e^{O_3}}$

wx for first neuron.

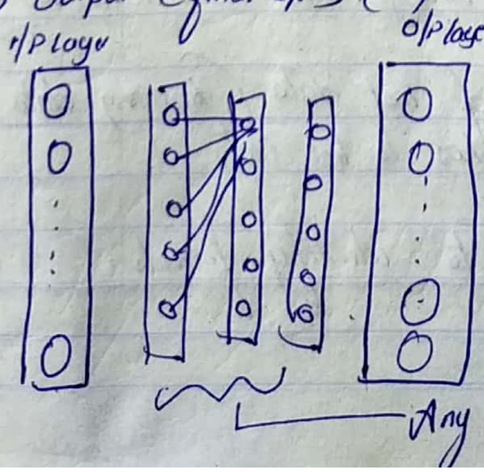
ANN:

- Non-linear task → many neurons must be arranged.

(1) neuron $\xrightarrow{\text{weighted sum}}$ $\boxed{A \cdot F}$

Layers : ← depending on i/p feature.

1) Input (dummy neuron (doesn't do any computation, just forward to next layer))
2) Hidden
3) Output (final o/p) (depending on class, 2 class – 1 neuron
> 2 class – No of class = no of neuron)



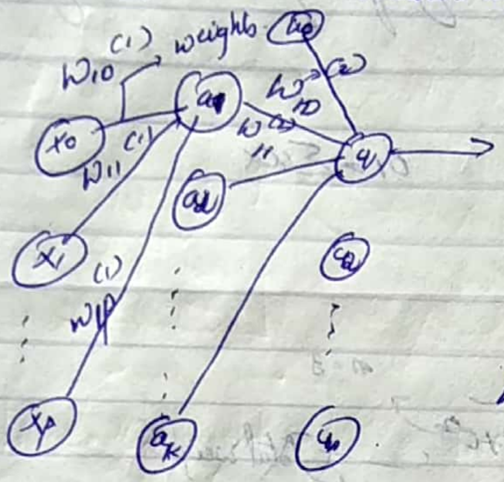
Complex prob solved ⇒ ↑ layer & neuron.

Any no. of hidden layer.

D/u of ANN
— lot of computational power

Shallow n/w
— 1 Hidden Layer.

Deep neural n/w.
— 2, .. 100 .. Hidden Layers.



$w^{(1)}$ = 1st layer to 2nd layer.
$w^{(2)}$ = 2nd layer to 3rd layer.

$W_{ij}^{(1)}$ = ith neuron in 2 layer connected with jth neuron in 1 layer.

Backpropagation algo.
— > 1 layer > 1 neuron
— for training
— same as Gradient descendent.
— But weight Tool is diff.

Training of single neuron — Gradient descent.

K - nearest neighbour Algorithm (KNN) / Lazy Learner
— No parameters for model. (no training) learns only when test data comes.
— Divide training & testing data.

(1) Compute neighbours of test data (training data)
(2) Apply label of neighbor to test data

Euclidean distance to identify friends.

$(x, y) \Rightarrow \begin{matrix} x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \end{matrix}$

$d = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + (x_3 - y_3)^2}$

Diff b/w $x^{(i)}$ & $x^{(j)}$ where X is 'p' dim.

$d_{ij} = \sqrt{\sum_{k=1}^{p} (x_{ik} - x_{jk})^2}$

X_1	X_2	Class
7	7	Bad
7	4	Bad
3	4	Good
1	4	Good

Test data (3, 7)

Distance:

$d1 = \sqrt{(7-3)^2 + (7-7)^2} = \sqrt{16} = \underline{4}$

$d2 = \sqrt{(7-3)^2 + (4-7)^2} = \sqrt{16+9} = \underline{5}$

$d3 = \sqrt{(3-3)^2 + (4-7)^2} = \sqrt{9} = \underline{3}$

$d4 = \sqrt{(1-3)^2 + (4-7)^2} = \sqrt{4+9} = \underline{\sqrt{13}} = 3.4$

$k = 3 \Rightarrow$ 3 ~~neigh~~ nearest neighbor

3 4 1

↓ ↓ ↓

Good Good Bad

Good

B — LR
weights — ANN } model
tree — DT

DT

- internal node → test condn
- leaf node → class label

Person ⇒ fit
unfit

Age
Eating
Exercise

$\boxed{Age < 30}$

40

$\boxed{\text{Eats lot of pizza}}$

Process of building tree = tree induction

Tree induction:
- looking for attribute which to select at each level

Entropy:
To measure purity of attribute

$$Gain(D,A) = Entropy(D) - \sum_{i=1}^{v} \frac{|D_i|}{|D|} Entropy(D_i)$$

for entire database

count (no. of examples)

Gini Index
$G =$ —

1) $k=2$

$C_1 = 9$

$C_2 = 5$

$$E(D) = -\sum_{i=1}^{2} P_i \log_2 P_i$$

$$= -\frac{9}{14} \log_2 \frac{9}{14} - \frac{5}{14} \log_2 \frac{5}{14}$$

$$= \cancel{0.883}\ 0.940$$

$$IG(D, Wind) = Entropy(D) - \left[\frac{6}{14} Entropy(Wind = Strong) + \frac{8}{14} Entropy(Wind = Weak)\right]$$

only rows with wind strong consider not all

$$Entropy(Wind = = strong)$$

$$= -\left(\frac{3}{6} \log_2 \frac{3}{6} + \frac{3}{6} \log_2 \frac{3}{6}\right)$$

$$= 1$$

$$Entropy(Wind = weak)$$

$$= \left(\frac{6}{8} \log_2 \frac{6}{8} + \frac{2}{8} \log_2 \frac{2}{8}\right)$$

$$= 0.811$$

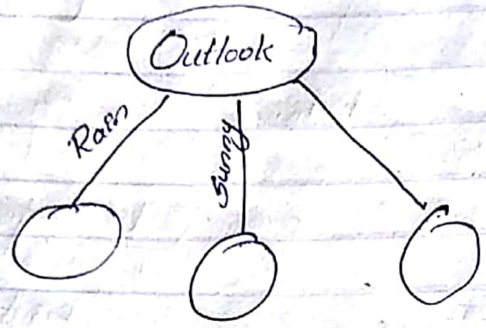
IG (D, Outlook) = 0.246

IG (D, Temp) = 0.029

IG (D, Humidity) = 0.15

IG (D, Wind) = 0.048

IG of Outlook is more. Therefore, Outlook will be tested first.



Hidden Markov Model (HMM)

— Sequential / Order is imp.

— to identify sequence

HOW

$H \rightarrow O \rightarrow W$ identification order

Markov Process / Chain

— Set of seq rep set of states

What is the probability that t is after h or o is after h?

t, o, h $\rightarrow$ each separate state

1st order Markov

$$P(q_{t+1} = S_j \mid q_t = S_i, q_{t-1} = S_k, \ldots) = P(q_{t+1} = S_j \mid q_t = S_j)$$

t $\rightarrow$ tmrw depends time. today yesterday

If I know todays weather, I can predict tmrw.

Transition probabilities

$$a_{ij} = P(q_{t+1} = S_j \mid q_t = S_i)$$

tmrw. todays

H. $\leftarrow$ P(O) not b4 H

Initial probabilities

The Initial prob.

$$\pi_i = P(q_1 = S_i)$$

I need to know current state to predict next state, not all prev state — Markov process.

$$\pi_1 \rightarrow (1) \xrightarrow{a_{12}} (2)$$

— HMM

— probability for combinatorial prob & follows Markov process

$$\Pi = [0.5,\ 0.2,\ 0.3]^T \qquad A = \begin{array}{c|ccc} & R & B & G \\ \hline R & 0.4 & 0.3 & 0.3 \\ B & 0.2 & 0.6 & 0.2 \\ G & 0.1 & 0.1 & 0.8 \end{array}$$

(R, B, G under the entries of Π)

Initial — Transition prob

$O = \{S_1, S_1, S_3, S_3\}$
(R, R, B, B)

red to red — Given red to green

$$P(O \mid A, \Pi) = P(S_1) \cdot P(S_1 \mid S_1) \cdot P(S_3 \mid S_1) \cdot P(S_3 \mid S_3)$$
$$= \pi_1 \cdot a_{11} \cdot a_{13} \cdot a_{33}$$
$$= 0.5 \cdot 0.4 \cdot 0.3 \cdot 0.8 = \underline{0.048}$$

Hidden Markov Model

- States (class labels) are hidden — prob of carrying umbrella in raining season / sunny / cloudy.
- We know observation (features)

$$b_j(m) \equiv P(O_t = V_m \mid q_t = S_j)$$

umbrella present / not

Deep Learning:

Feature Extraction — noise & all removed — ML
So in DL, it may learn noise (no feature extraction).
So we need to give large info.

— CNN
- many layers (100..)
- last 3 layers similar to ANN.
- working of 1st layer: filtering, compression, convolutional
  - 1 - layer — edge detection
  - 2 layer — bg detected

1 pt → classifier → Knn

Eg: Google: large set of data ⟶ large set of computation power

Unit-4

Unsupervised Learning

Clustering

Grouping data into diff clusters based on similarities

Eg:

— Euclidean Distance

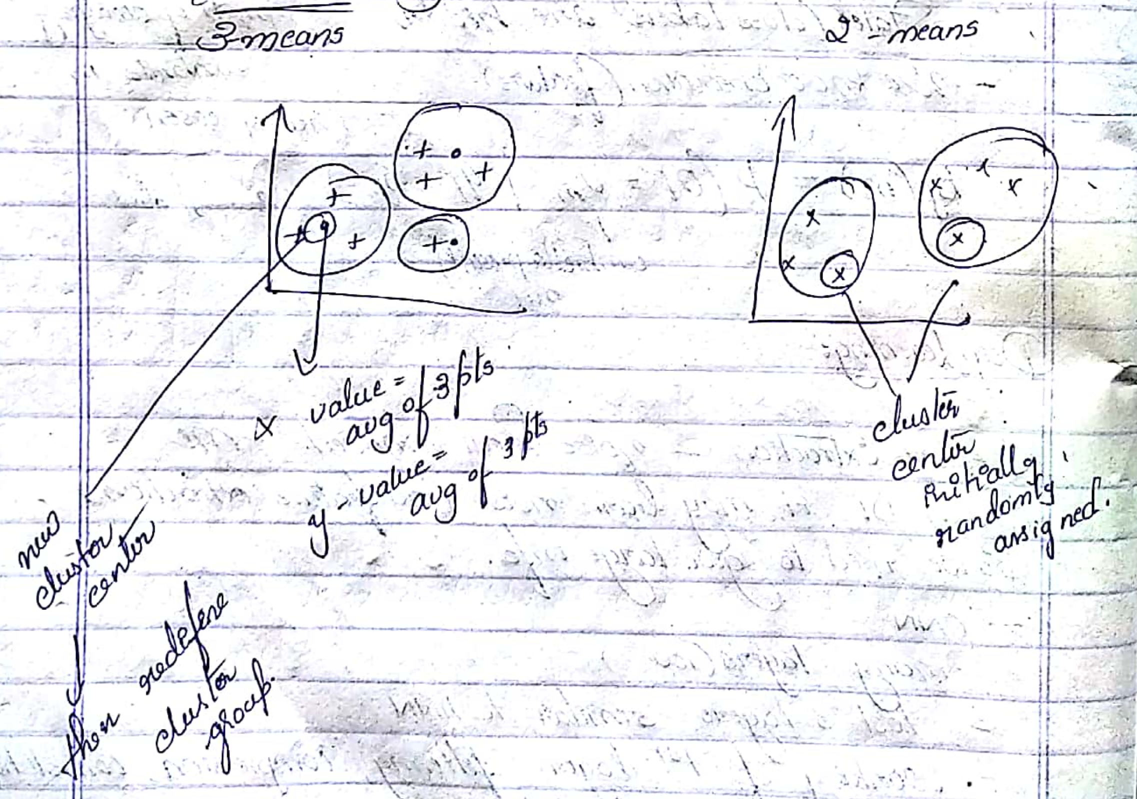
2 clustering techniques:

1) K-means clustering
2) Hierarchical clustering

user defined parameter

avg

K-means clustering

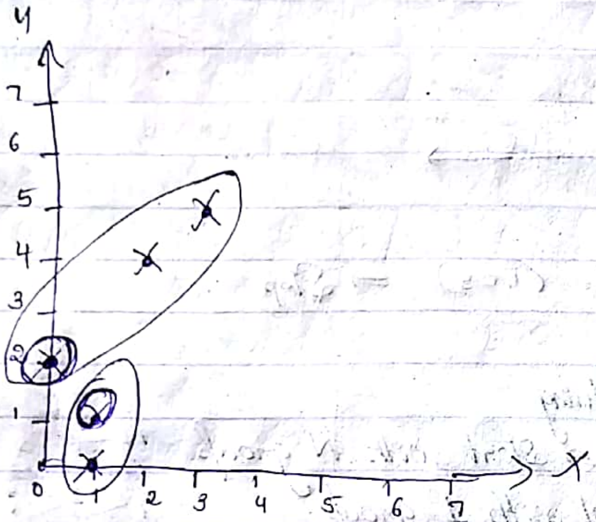


Initial cluster center → Group → Compute cluster center → Group ... until cluster center is const.

1)

Points	X_1	X_2
A	1	1
B	1	0
C	0	2
D	2	4
E	3	5

Initial cluster center A & C



nearest to A

Data	Distance C1 (1,1)	C2 (0,2)	C1	C2
A	$\sqrt{0} = 0$	$\sqrt{(1-0)^2+(1-2)^2} = \sqrt{2}$	0	1.4
B	$\sqrt{0+1} = 1$	$\sqrt{1+2^2} = \sqrt{5}$	1	2.2
C	$\sqrt{1^2+1^2} = \sqrt{2}$	$\sqrt{0}$	1.4	0
D	$\sqrt{1+3^2} = \sqrt{10}$	$\sqrt{2^2+2^2} = \sqrt{8}$	3.2	2.8
E	$\sqrt{2^2+4^2} = \sqrt{20}$	$\sqrt{3^2+9} = \sqrt{18}$	4.5	4.2

$C1 = \{A, B\}$. $C2 = \{C, D, E\}$

Calculate new center, recompute cluster

$C1 = \{A \; B\}$ — (1,1) (1,0) $\qquad C2 = \{C, D, E\}$ — 0,2 2,4 3,5

$= \frac{1+1}{2}, \frac{1+0}{2} \qquad = \frac{0+2+3}{3}, \frac{2+4+5}{3}$ $\quad 11/3$

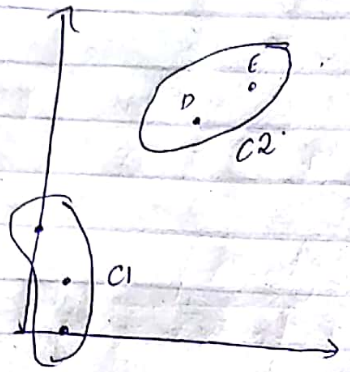
$= (1, 0.5) \qquad = (1.70, 3.7)$

1, 0.5   1.7, 3.7

Data	C1	C2
A	0.5	2.7
B	0.5	3.7
C	1.8	2.4
D	3.6	0.5
E	4.9	1.9

C1 = {A, B, C}

C2 = {D, E}



Next time of C1 = (1, 0.5) ⇒ Stop

Hierarchical Clustering

1) Agglomerative ⇒ Start with N group   N → 1

2) Divisive ⇒ Start with 1 group.   1 → N

	A	B	C	D	E
A	0	1	1.41	3.16	4.47
B	1	0	2.23	4.12	5.38
C	1.41	2.25	0	2.82	4.24
D	3.16	4.12	2.82	0	1.41
E	4.47	5.38	4.24	1.41	0

C1 (A, B), C2 (C), C3 (D), C4 (E)

Given

Points	X_1	X_2
A	1	1
B	1	0
C	0	2
D	2	1
E	3	5

$AB = \sqrt{1-1+1} = \underline{1}$ $AC = \sqrt{(1-0)^2 + (1-2)^2} = \sqrt{2}$.

$AD = \sqrt{(1-2)^2 + 0} = \underline{1}$

A & B — Bcoz diff b/w A & B 1 (Smallest of all (whole table)) ignore 0

min(1.41, 2.23)
AC BC

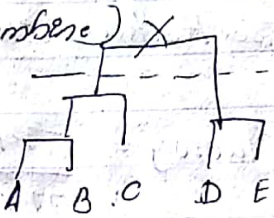
1) Single link clustering → min
2) Complete k → max

	A&B	C	D	E
A&B	0	1.41	3.16	4.47
C	1.41	0	2.82	4.24
D	3.16	2.82	0	1.41
E	4.47	4.24	1.41	0

Smallest → 1.41

ABC DE (At a time 2 combine)

	A&B&C	D	E
A&B&C	0	2.82	4.24
D	2.82	0	1.41
E	4.24	1.41	0

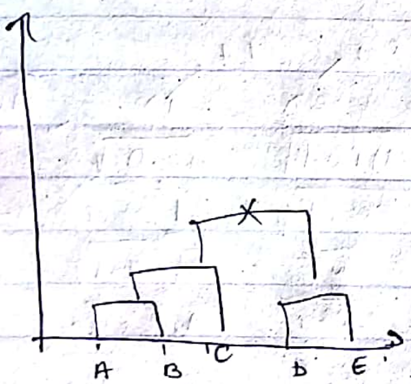


2 cluster

finally: 1 cluster

	A&B&C	D&E
A&B&C	0	2.82
D&E	2.82	0

	A&B&C&D&E
A&B&C&D	0



Evaluation measures & combining learners

- while practically implementing ML model ⇒ evaluation req.
- To check accuracy.

Diff evaluation techniques:

- Confusion Matrix

Dataset → Training — Model
Dataset → Testing — Evaluate

Justified as don — Statistics created

~~Prediction~~ Prediction

actual / +ve

	Yes	No
Yes	TP	FN
No	FP	TN

classified as what

correct/wrong

True Positive : Its +ve class & classified as +ve.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

$$= \frac{\text{No. of correctly classified}}{\text{Total no. of examples}}$$

Date / /20

Resampling & K-fold cross validation

dataset divided into k different set.
- One of the set used as testing set/validation.

1st itr $V_1 = X_1 \quad T_1 = X_2 \cup X_3 \cup \ldots \cup X_k$

2nd itr $V_2 = X_2 \quad T_2 = X_1 \cup X_3 \cup \ldots \cup X_k$

⋮

k itr $V_k = X_k \quad T_k = X_1 \cup X_2 \cup \ldots \cup X_{k-1}$

Avg of all accuracy = Accuracy of dataset.

5X2 Cross Validation

Dataset → Training (X_1), Testing (X_2)

1 itr: $T_1 = X_1^{(1)} \quad V_1 = X_2^{(2)}$ randomnly select x_1 & x_2

swap $T_2 = X_1^{(2)} \quad V_2 = X_1^{(1)}$

dataset

2 itr $T_3 = X_2^{(1)} \quad V_3 = X_2^{(2)}$ randomly select x_1 & x_2.

$T_4 = X_2^{(2)} \quad V_4 = X_2^{(1)}$

5 itr 10 accuracy each iteration 2

Change in seed value, diff random no.

Bootstrapping

- When you have small dataset

1) Random Sampling with replacement — allowed again
2) Random Sampling w/o replacement
 ↳ keep aside.

$$\left(1 - \frac{1}{N}\right)^N \approx e^{-1} = 0.368$$

K-nearest Neighbour Algorithm:

- lazy learner
- no parameters, no training
- 1st compute neighbour using training data
- neighbour calculated using Euclidean distance

~~(x, y)~~ (x_1, y_1)

(x, y) $\{x_1 \ x_2 \ x_3$

$y_1 \ y_2 \ y_3$

$x_1 - y_1 \quad x_2 - y_2 \quad x_3 - y_3$.

$$d = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + (x_3 - y_3)^2}$$

d_{ij} $x_i^{(i)}$ & $x_j^{(j)}$ where x is p dimension.

$$d_{ij} = \sqrt{\sum_{k=1}^{p} (x_{ik} - x_{jk})^2}$$

x_1	x_2	class
7	7	Bad
7	4	Bad
3	4	Good
1	4	Good

Test data $(3, 7)$ — $y_1 \ y_2$

$K = 3$

$2 - 0$

$\sqrt{(\text{testdata} - x_1)^2 + (x_2 - \text{testdata})^2}$

$$d_1 = \sqrt{(7-3)^2 + (7-7)^2}$$
$$= \sqrt{4^2 + 0^2} = 4$$

$$d_2 = \sqrt{(7-3)^2 + (4-7)^2} = \sqrt{16+9} = 5$$

$$d_3 = \sqrt{(3-3)^2 + (4-7)^2} = 3$$

$$d_4 = \sqrt{(1-3)^2 + (4-7)^2} = \sqrt{4+9}$$
$$= \sqrt{13} = 3.4$$

$K = 3$ d_3, d_4, d_1 $\therefore$ good

good   good   bad

Bayesian classifier:

Bayes Theorm:

$C_1, C_2 - - C_k$ - class

$P(C_1|x)$

$P(C_2|x)$

⋮

$P(C_k|x)$

assign the test data x to class label C_j which has max value for $P(C_j|x)$

$P(C_i|x)$ = ? computed using Baye's theorm

A & B events not independent

$P(A \cap B) = P(A)\,P(B|A)$ or $P(B)\,P(A|B)$

Independent $P(A \cap B) = P(A)\,P(B)$

A = { got value 4 } , B = { even no }

$P(A) = \frac{1}{6}$ $P(A|B) = \frac{1}{3}$

$$P(B|A) = \frac{P(A \cap B)}{P(A)}$$

$$P(B|A) = \frac{P(B)\,P(A|B)}{P(A)}$$ ⇒ Baye's theorm

Feature (x) class label (C_i)

$$P(c_i|x) = \frac{P(c_i)\,P(x|c_i)}{P(x)}$$

multidimensional:

$P(x|c_i) = P(x_1|c_i) \cdot P(x_2|c_i) \ldots P(x_d|c_i)$

C_1 = Yes $\quad C_2$ = No $\quad$ 4 attributes

x' = (Outlook = Sunny, Temp = Cool, Humidity = high, Wind = strong)

$P(C_1|x') = ?$
$P(C_2|x') = ?$ } consider whichever is high

14 - Examples
9 - Yes
5 - No

$P(C_1) = 9/14$

$P(C_2) = 5/14$

Compare:

$$\frac{P(C_1)\,P(x'|C_1)}{\cancel{P(x')}} \quad \& \quad \frac{P(C_2)\,P(x'|C_2)}{\cancel{P(x')}}$$

$$P(x'|C_1) = P(x_1'|C_1)\,P(x_2'|C_1)\,P(x_3'|C_1)\,P(x_4'|C_1)$$

Outlook	Yes	No
Sunny	check given table 2/9	3/5
Overcast	4/9	0/5
Rain	3/9	2/5

out of 9 yes, 2 are sunny yes

P(Outlook = Sunny | Yes) = 2

P(Outlook = Sunny | No) = 3. out of 5 no, 3 are for sunny

P(Outlook = Rain | Yes)

P(" " | No)

P(" = Overcast | Yes)

P(Outlook = Overcast | No)

Similarily for Temp, humidity, wind

Temp: Cool, mild, hot
humidity: high, normal
wind: Strong, weak

$P(C_1|x') = P(C_1) \cdot P(x'|C_1)$

$P(Yes|x') = P(x'|Yes)\,P(Yes)$

$= P(Outlook = Sunny|Yes)\,P(T = Cool|Yes)\,P(Hum = high|Yes)\,P(Wind = Strong|Yes)\,P(Yes)$

27
14
108
27
378

$= 2/9 \times 3/9 \times 3/9 \times 3/9 = \frac{2}{9 \times 27} \times \frac{9}{14} = \frac{2}{378}$

$= 0.0053$

Temp	Yes	No
Hot	2/9	2/5
mild	4/9	2/5
Cool.	3/9	1/5.

~~Humidity~~ Wind	Yes	No
Weak	6/9	2/5
Strong	3/9	3/5

Humidity	Yes	No
High	3/9	4/5
Normal	6/9	1/5

Naive Bayes Classifier — Assume that attributes are independent of each other

$P(X)$ = Prior probability
$P(c|x)$ = Posterior Probability
$P(x|c)$ = class conditional probability

P(outlook=sunny|no) P(temp=cool|no) —

$P(No|x') = (3/5)(1/5)(4/5)(3/5) \times 5/14$

$\frac{3}{5} \times \frac{1}{5} \times \frac{4}{5} \times \frac{3}{5} \times \frac{5}{14}$

$= \frac{18}{875} = 0.0205$

$P(no|x') > P(Yes|x')$
= Hence x' belongs to C_2 (class No)

$P(x|c) = P(x_1|c)\, P(x_2|c) \ldots P(x_p|c)$
$= P(x_1 x_2 x_3 \ldots x_p | c)$

If X(input) is continous-valued i/p attribute:

$$P(x_j | c = c_i) = \frac{1}{\sqrt{2\pi}\,\sigma_{ji}} \exp\left(-\frac{(x_j - \mu_{ji})^2}{2\sigma_{ji}^2}\right)$$

(imp)

μ_{ji} = mean of attribute values x_j of egs for which $c = c_i$
σ_{ji} = std deviation " " "

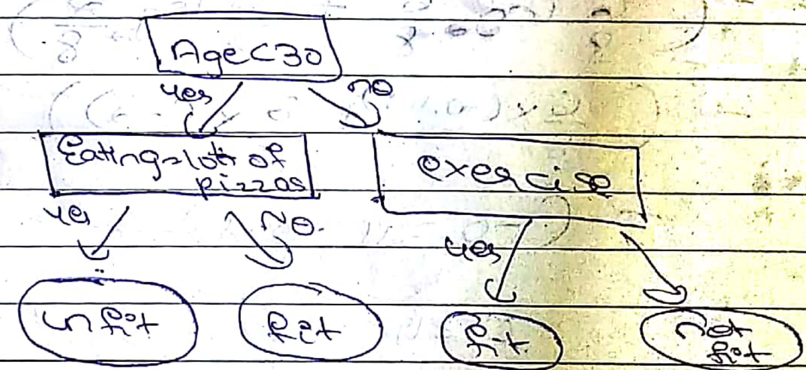
$z = \frac{x - \mu}{\sigma}$ Guassion distribution ↓ avg value will have more no of ppl

P(class=no|X)

(Income, Class=no)
Semple mean - 110 sample variance = 2975

$\sigma^2 = \frac{\Sigma(x - \mu)^2}{n - 1}$

Decision Trees:
· internal nodes - test condition
leaf " - class labels
Heirarchical model for supervised learning
Person - fit or unfit.
3 attributes Age, Eating habits, Exercise



model - tree

$Entropy(D) = \sum_{i=1}^{K} -p_m^i \log_2 p_m^i$ where $0\log 0 = 1$

$Gain(D, A) = Entropy(D) - \sum_{i=1}^{v} \frac{|D_i|}{|D|} Entropy(D_i)$
D ↓ Dataset, A ↓ Attribute

attribute which has highest gain is selected as attribute for testing.

$C_1 = \text{Yes}$ $\quad k = 2$

$C_2 = \text{No}$

Entropy (D)

$$= -\overset{\text{for yes}}{9/14 \log_2 9/14} - \overset{\text{for no}}{5/14 \log_2 5/14}$$

$$= \underline{0.939}$$

Information Gain (D, Wind)

$$= \text{Entropy}(D) - \left[\frac{6}{14}\,\text{Entropy}(\text{Wind} = \text{Strong})\right.$$

(2)

$$\left. + \frac{8}{14}\,\text{Entropy}(\text{Wind} = \text{weak})\right]$$

(6 - strong, 8 - weak)

Entropy (Wind = Strong) $\quad$ 3 - Yes

$$= -\left(\frac{3}{6}\log_2\frac{3}{6} + \frac{3}{6}\log_2\frac{3}{6}\right)$$

$$- \left(-3/6 - 3/6\right)$$

$$= \underline{1}$$

Entropy (wind = weak)

$$= -\left(\frac{6}{8}\log_2\frac{6}{8} + \frac{2}{8}\log_2\frac{2}{8}\right)$$

$$- \left(\frac{6}{8}\times(-0.415) + \frac{2}{8}\times(-2)\right)$$

$$- \left(-0.311 - 0.5\right)$$

$$\underline{0.811}$$

(1)

$$= 0.94 - \left[\frac{8}{14}\times 0.811 + \frac{6}{14}\times 1\right]$$

$$= 0.94 - [0.4634 + 0.428]$$

$$= \underline{0.048}$$

Scanned by CamScanner

Hidden Markov Models:

$IG(D, \text{Outlook}) = 0.246$
$IG(D, \text{Temp}) = 0.029$
$IG(D, \text{Humidity}) = 0.151$
$IG(D, \text{Wind}) = 0.048$

Hidden markov models :- for sequential type of data
move from 1 state to another.
N states: $S_1\ S_2 \ldots S_N$
state at time t, $q_t = S_i$

1st order markov:
$P(q_{t+1} = S_j \mid q_t = S_i, q_{t-1} = S_k \ldots)$
$= P(q_{t+1} = S_j \mid q_t = S_i)$

Transition probabilities:

today's weather

$a_{ij} = P(q_{t+1} = S_j \mid q_t = S_i)$ $\quad a_{ij} \geq 0$

tomo weather depends on

$\sum_{j=1}^{N} a_{ij} = 1$

Initial probabilities:

$\pi_i = P(q_1 = S_i)$
$\sum_{i=1}^{N} \pi_i = 1$

You should only know current state to predict next state. Previous states are not needed.

Eg: Balls & Urns
3 urns each full of balls of 1 color
S_1: red $\quad S_2$: blue $\quad S_3$ = green
$\pi = [\underset{S_1}{0.5}, \underset{S_2}{0.2}, \underset{S_3}{0.3}]^T$

Transition probability

$$A = \begin{bmatrix} 0.4 & 0.3 & 0.3 \\ 0.2 & 0.6 & 0.2 \\ 0.1 & 0.1 & 0.8 \end{bmatrix}$$

red red green green

$O = \{S_1, S_1, S_3, S_3\}$ 1st ball red & 2nd ball red

$P(O|A,\pi) = P(S_1)\,P(S_1|S_1)\,P(S_3|S_1)$

$= \pi_1 \cdot a_{11} \cdot a_{13} \cdot a_{33}$ $P(S_3|S_3)$

$= 0.5 \times 0.4 \times 0.3 \times 0.8$

$= 0.048$

Emission probabilities:

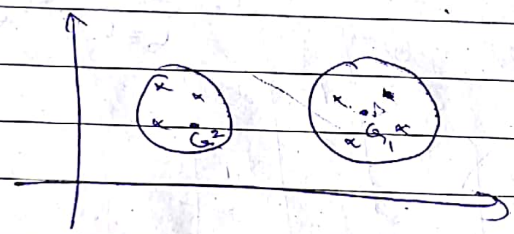
$b_i(m) = P(O_t = v_m \mid q_t = S_i)$

time

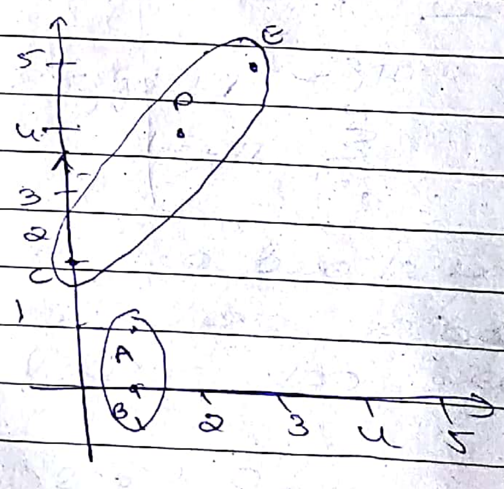
Unit - IV

1) K-means clustering
2) Hierarchical clustering

K-means clustering



Points	X_1	X_2
A	1	1
B	1	0
C	0	2
D	2	4
E	3	5



Initial cluster centers are A & C

Data	C_1		C_2
A	0	<	$\sqrt{2}$
B	1	<	2.2
C	1.4	>	0
D	3.2	>	2.8
E	4.5	>	4.2

$C_1 = A \quad x_i - y_i$

$C_2 = C$

$A(1,1) \quad C(0,2)$

A & C_2

$(1,1) \quad (0,2)$

$= \sqrt{(1-0)^2 + (1-2)^2}$

$= \sqrt{1+1}$

$= \sqrt{2}$

$B(1,0)$ & $C_1 (1,1)$

$= \sqrt{(1-1)^2 + (1-0)^2}$

$= \sqrt{1} = 1$

$B(1,0)$ & $C_2(0,2)$

$= \sqrt{1^2 + 2^2} = \sqrt{5}$

$C_1 = \{A, B\}$

$C_2 = \{C, D, E\}$

Recompute cluster mean:

$C1 = \{A, B\} = \frac{1+1}{2} = (1, 0.5)$
$(1,1) \; (1,0)$

$C2 = \{C, D, E\}$

$(0,2) \; (2,4) \; (3,5)$

$\frac{0+2+3}{3} \qquad \frac{2+4+5}{3}$

$= \frac{5}{3} = 1.6 \qquad \frac{11}{3} = 3.36$

$C_2 = (1.6, 3.6)$

$C_1 = (1, 0.5) \qquad C_2 = (1.6, 3.6)$

$A(1,1) \quad C_1 = (1, 0.5)$

$= \sqrt{(1-1)^2 + (1-0.5)^2}$

$= \sqrt{(0.5)^2} = 0.5$

A & C_2

$\sqrt{(1.6-1)^2 + (3.6-1)^2}$

$= \sqrt{(0.6)^2 + (2.6)^2}$

$= 2.7$

Data	C_1	C_2
A	0.5	2.7
B	0.5	3.7
C	1.8	2.4
D	3.6	0.5
E	4.9	1.9

$C_1 = \{A, B, C\}$

$C_2 = \{D, E\}$

repeat till cluster center is same

Hierarchical Clustering:

① Agglomerative — starts with N grp

N cluster
find dist b/w each cluster
using Euclidean's distance
whichever is near combined
as 1 cluster
~~top to down~~
bottom up

② Divisive — start with 1 grp.

		A	B	C	D	E
C1	A	0	1	$\sqrt{2}$	$\sqrt{10}$	$\sqrt{20}$
	B	1	0	$\sqrt{5}$	$\sqrt{17}$	$\sqrt{29}$
C2	C	$\sqrt{2}$	$\sqrt{5}$	0	$\sqrt{8}$	$\sqrt{18}$
C3	D	$\sqrt{10}$	$\sqrt{17}$	$\sqrt{8}$	0	$\sqrt{2}$
C4	E	$\sqrt{20}$	$\sqrt{29}$	$\sqrt{18}$	$\sqrt{2}$	0

C1 C2 C3 C4

9+9

1+1

Points	x	x_2
A	1	1
B	1	0
C	0	2
D	2	4
E	3	5

points are smaller clubbed to 1 cluster

$AD = \sqrt{(2-1)^2 + (4-1)^2}$

$= \sqrt{1+3^2}$

$= \sqrt{10}$

$1+16$

$2^2+5^2 = \sqrt{29}$

Between A & B A is smallest consider it,

1.41 & 2.23 → smallest

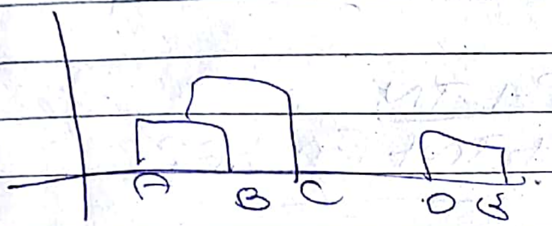
	A & B	C	D	E
A & B	0	1.41	3.16	4.47
C	1.41	0	2.82	4.24
D	3.16	2.82	0	1.41
E	4.47	4.24	1.41	0

→ same as table no comparing

Smallest value is 1.41

A BC → combined

DE - Combined



	A & B & C	D	E
A & B & C	0	2.82	4.24
D	2.82	0	1.41
E	4.24	1.41	0

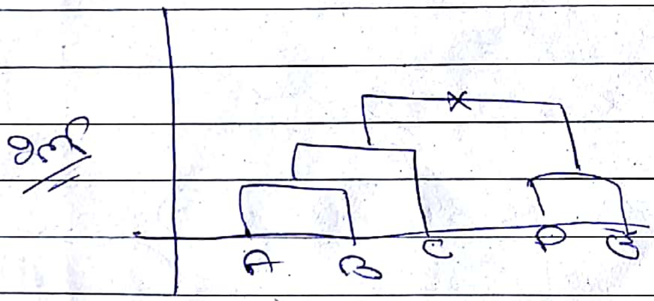
(A & B) & D = 3.16

C & D = 2.82

min = 2.82

	A & B & C	D & E
A & B & C	0	2.82
D & E	2.82	0

	A & B & C & D & E
A & B & C & D & E	0



Evaluation measure & Combining learners

→ to check accuracy of model

(1) Confusion matrix :-

Positive class - true positive (TP)

confusion matrix

Prediction

Class	Yes	No
Yes	TP	FN
No	FP	TN

FN → false negative

TN → true negative

FP → false positive

Accuracy = no of correctly classified classes / Total no of example

$$= \frac{TP + TN}{TP + TN + FN + FP}$$