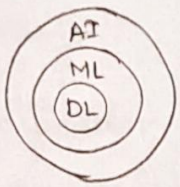
7/2/25

Machine Learning: ML is a branch of AI that enables computers to learn from data and make predictions without being explicitly programmed (without human involvement).

Relation between ML & AI



- Initially in ML the data is in structured form that is taken from DL where the data structure is random and bigger (unstructured).
- DL will combine with Neural network & convert to structured data to pass to ML.

- AI is a broader field that aims to creati inteligent machines.
- ML is a subset of AI that focuses on algorithms that learn from data and improve based on experience.
- DL is a specialized subset of ML using neural networks for image processing complex task, speech recognization.

Importance of ML:

- Automates taskes and dicision making.
- Improves efficiency and accuracy in various fields.
- Helps in discovering patterns from large amounts of data.

11/2/25

Realtime applications of ML

- Siri
- Gemini
- face regnisation on smart phones
- spam email in gmail
- virtual assistance
- product recommendations on e-commerce, movies.

Real time domains.

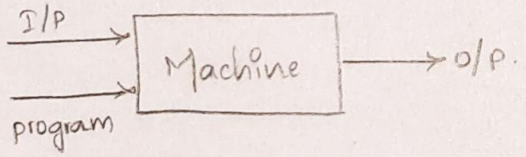
- Healthcare
- finance
- Retail
- self-driving cars
- cyber security.
- social media.

Course Objectives / Outcomes.

Objectives:

- To understand the basic threory underlining Machine learning.
- To be able to formulate ML problems corresponding different applications
- To understand a range of machines learning algorithms along with their strengths and weaknesses.
- To learn the applications of ML algorithms in solving problems of moderate complexity.
- To be familiarized with the challenges when ML algorithms are applied to unstructed dataset.

17/2/25



learning Pardigms

1) Supervised learning.
2) unsupervised learning
3) Reinforcement learning.

→ Supervised learning: we will train the data set where it contains labels along with samples and objects. passing labels means samples.

18/2/25

→ Machine + database + Model + Algorithm (or) models + program + identify the patterns + interface relationship among the patterns [extract patterns].

Classification:

If the possible outcomes of the function are predefined and discrete it is called classification.

Regression:

if the possible outcome values of the function are continuous real values. then it is called regression.

→ Unsupervised Learning: The algorithm is given an unlabeled dataset and is tasked with finding patterns, structures stretches or relationships with in the data

→ Reinforcement Learning: In this learning an agent learnes to interact with an environment to achieve a specific goal.

- The agent takes actions and receive rewards or panalties based on its actions. the goal is to learn a policy that maximize the cummulative reward overtime.

Step-1 Data Pre-processing.

i) Import the data

ii) clean the data

iii) Split into trainings and test sets

iv) feature scaling.

Step-2: Modelling.

→ Build the modul

→ Train the model

→ Make predictions

Step 3: Evaluation.

→ Calculate Performance metrics

→ Make a verdict.

21/2/25

Design a Learning System: (DLS).

data object, rows

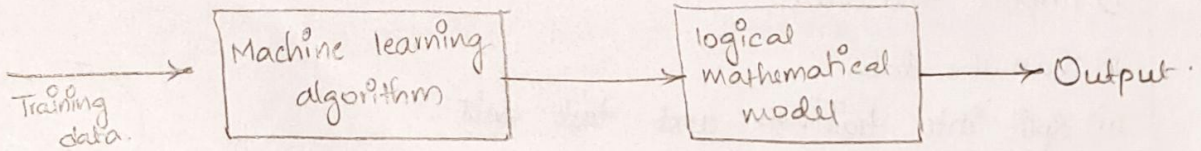
columns, features, attributes.

→ Importance of histograms, Box plot, scatter plot, heat maps.

purpose & In which content.

According to "Samuel Arthur", ML enables a machine to automatically learn from data and improve performance ~~and~~ from an experience and predict things without explicitly programmed.

In simple words, when we feed the training data to the machine learning algorithm, this algorithm will produce a mathematical model, and with the helps of the machematical model, the machine will make a prediction, and take a decision without being explicitly programmed.



more no. of samples.

Ex: In driver less car, the taining data is fed to algorithm like, how to drive car in high ways, busy roads, hill stations, narrow streets etc.. with factors like speed limit, parking, stop at signal etc.., After that a logical and mathematical model is created, on the basis of that the car will work according to the logical model.

Also the more data is fed, the more efficient output is produced.

Defination:

→ According to "Tom Mitchell" - "a computer program is said to be learning from experience (E), with respect to some task (T).

- Thus, the performance measure (P) is the performance at task (T), which is measured by 'P' and it improves with an experience (E)."

$E \propto P$.

25/2/25

Design a learning System.

Choosing Training Experience.

↓

Choosing Target function

↓

Choosing representation of Target function

↓

Choosing function approximation.

↓

final design.

Step-1: Choosing Training Experience / Data.

- which will fed to the machine learning algorithm.
- Algorithm must have a significant impact ~~of~~ on the success or failure of the model, based on training data.

① Direct / Indirect feedback
② Degree
③ Distribution of Samples.

} Parameters / substeps to be Considered.

① Direct/ Indirect feedback

Check whether the training experience provides, direct or indirect feedback to the algorithm based on choices of the performance system.

② Degree.

The degree of a training [algorithm] experiences refers to the extent upto which the learner can control the sequence of training.

③ Distribution of Samples:

Across which performace will be tested in this mode i.e, more diverse set of training experience can be the better performance.

⇒ Step-II Choosing the target function.

what type of knowledge is learned and how it is used by performance system.

Ex: Checkers Game.

Target function ⇒ $v(b)$

$b \rightarrow$ board state

legal moves set = B.

1. b is final state that is won, then $v(b) = 100$
2. loss, then $v(b) = -100$.
3. draw, $v(b) = 0$.
4. b is not a final state then, $v(b) = v(b')$.

⇒ Step-III Choosing the representation of target function.

for any board state, we calculate function 'c' as linear combination of following board features, i.e, $c(b)$ =

features:

$x_1 \rightarrow$ no. of black pieces on board.

$x_2 \rightarrow$ no. of red pieces on board.

$x_3 \rightarrow$ no. of black kings on board.

$x_4 \rightarrow$ no. of red kings on board.

$x_5 \rightarrow$ no. of black pieces threatened by red pieces.

$x_6 \rightarrow$ no. of red pieces threatened by black pieces.

$w \rightarrow$ weightage

$$\boxed{C(b) = w_0 + w_1x_1 + w_2x_2 + w_3x_3 + w_4x_4 + w_5x_5 + w_6x_6.}$$

28/2/25

Design Learning System

$\Rightarrow$ Step-IV Choosing a learning app. algorithm to learn target function

to learn target function (f) we need a set of training examples to describe a perticular board state and training value

$f = (\underset{(b)}{\text{board state}}, \text{train}(b))$

ordered pair $= (b, \text{train}(b))$

Case: Black won the game.

$x_1 = 3$; $x_2 = 0$; $x_3 = 1$; $x_4 = 0$; $x_5 = 0$; $x_6 = 0$.

$(\langle x_1 = 3, x_2 = 0, x_3 = 1, x_4 = 0, x_5 = 0, x_6 = 0 \rangle + 100)$

we need to do 2 steps in this phase.

1) Estimating the training values

In every step we consider the successor [depending on the next step of the opponent].

$$V\,\text{train}(b) \longleftarrow \hat{V}(\text{successor}(b)).$$

This represents the next board state [estimating that this move

will help or destroy opponent].

$V^{\wedge}$ → approximation.

2) <u>Adjusting the weights</u>.

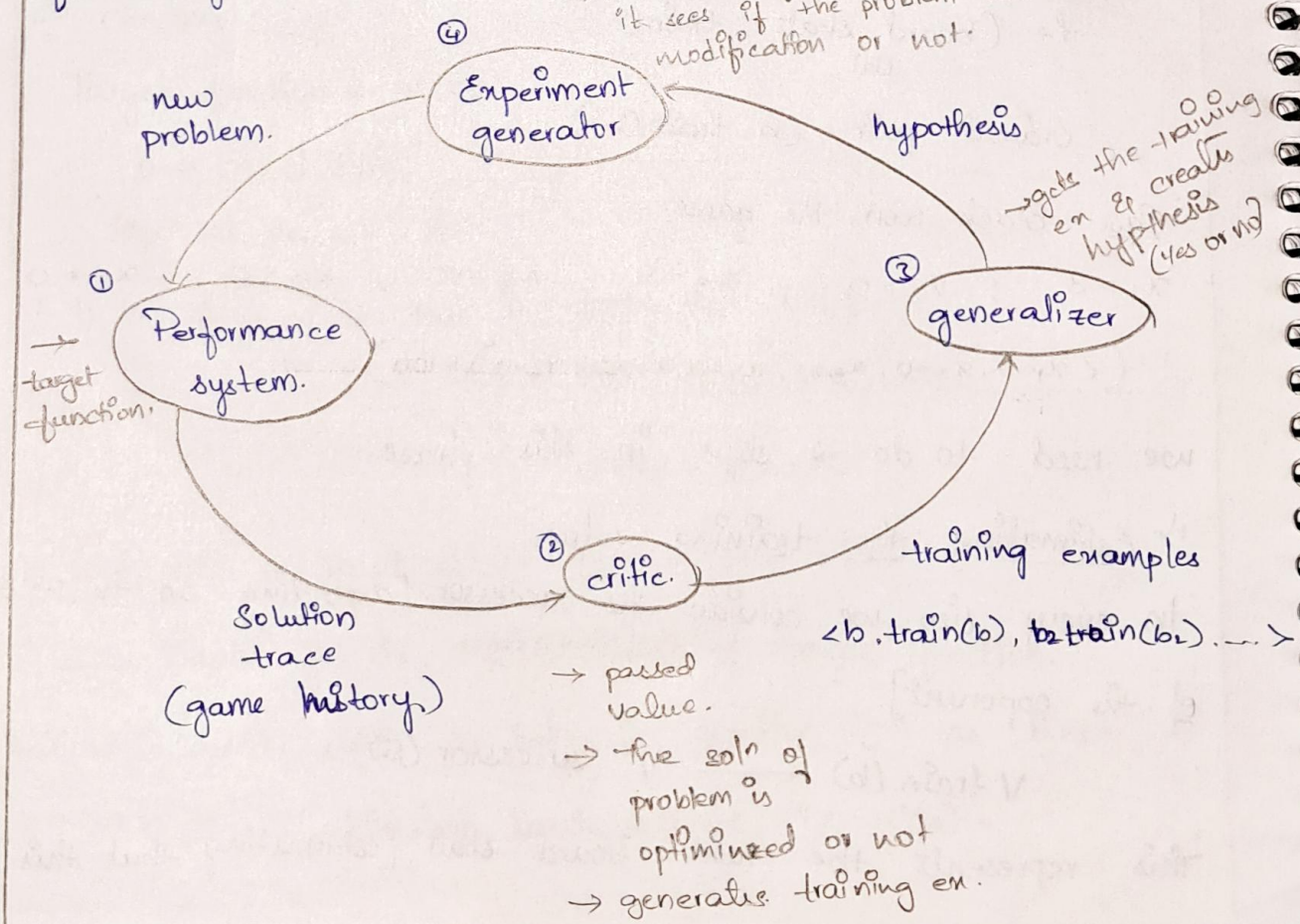
There are some algorithms to find weights of linear function by using LMS [least mean square] method used to assign weights based on which we calculate error.

$$\text{error} = (V(\text{train}(b)) - V^{n}(b))^2.$$

- If error = 0 ; no need to change weights.
- If error is +ve ; each weight is increased in proportion.
- If error is -ve ; each weight is decreased in proportion.

⇒ <u>Step-V</u> final design.

final design contains 4 modules.

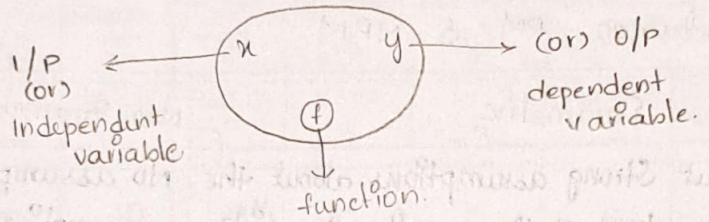


4/3/25

Parametric and Non-Parametric Machine learning models.

① Model Parameters:

Machine learning involves learning a function f that maps input variable ~~to~~ (x) to output variables (y) with the function configuration initially unknown $y = f(x)$.



② Importance of Model Parameter:

- A model parameter is a configuration variable that is internal to the model and who's value can be estimated from the given data.

i) they are required by the model when making the prediction.

ii) their values define the skill of the model on your problem

iii) they are estimated or learned from historical training data.

iv) they are often not set manually by the practioner.

v) they are often saved as the learned model.

③ Parametric model (PM).

Input parameters are to be passed to the model.

Parametric machine learning, a learning model that summerises data with a set of fixed size parameters. [Independent on the no. of instances of training.]

Parametri machine learning algorithms are which optimizes the function to a known form.

Non Parametric model. (NPM).

- Non Parametric machine learning algorithms are those which don't make any specific assumptions about the type of mapping function.
- They are prepared to choose any functional form from training data by not making assumptions.

(5) → Difference between PM & NPM

	Parametric	Non Parametric
1. Assumption about function form.	Strong assumptions about the form of the mapping function.	No assumptions / weak assumption about function form.
2. Model complexity.	Typically lower complexity. suitable simple problems.	higher complexity, can fit wide range of functions
3. Data requirements	requires less training data	requires more training data.
4 speed of training.	faster training time	slower training time.
5 Flexibility.	limited to specific form	capable of fitting various functional forms.
6. Generalization ability.	may struggle with complex relationships among the variables.	Can generalize well to ~~no~~ unseen data.
7. Examples.	- linear regression - logistic regression. - perceptron.	- Decision Trees - K-nearest neighbors (KNN) - Neural n/ws.

5/8/25

UNIT-2

MODELS

1) Regression: to predict the dependent variable based on independent variable (context of)
2) Simple linear regression model.

Diameter (In inches) x	Price (y)	Mean (x)	Mean (y)	Deviation (x)	Deviation (y)	Product of Deviation.	Sum of product of deviation	Square of Deviation of x.
8	10	$\frac{8+10+12}{2}$	$\frac{10+13+16}{2}$	8 - 10 = -2	10 - 13 = -3	-2 × -3 = 6	6 + 6 = ~~12~~	$(-2)^2$ = 4
10	13	10	13	10 - 10 = 0	13 - 13 = 0	0 × 0 = 0	12	0^2 = 0 4
12	16			12 - 10 = +2	16 - 13 = +3	+2 × 3 = 6	~~12~~	2^2 = 4.

1* → In regression model: establish a relationship b/w independent variable & dependent variable.

2* → to fore cast new observations

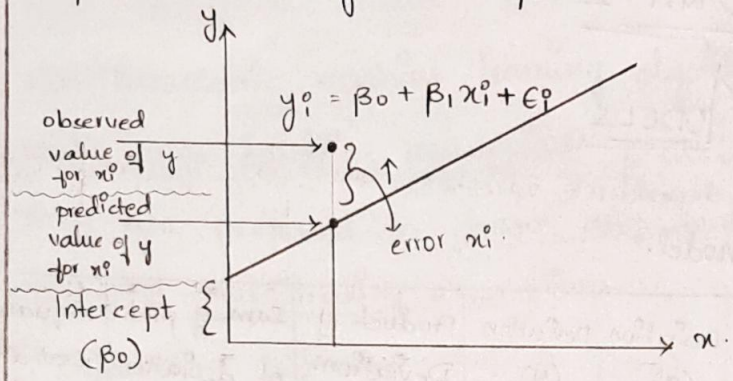
} Importance

Linear Regression:

- dependent variable means, the variable we wish to predict or expain.
- independent variable means, the variable used to explain the dependent variable.
- Regression Analysis is used to
  - ↳ predict the value of dependent variable based on the value of atleast one independent variable.
  - ↳ Explain the impact of changes in an independent variable on the dependent variable.

[Simple Linear Regression Model:] →

② Simple linear Regression Model:



error:
difference b/w original observed value − predicted value
[It must be +ve]

Ex: Predicting pizza prices.

Step-1 : data collection

Step-2 : Calculations

Step-3 : Prediction

Step-4 : Visualization

→ Original value − Predicted value

ϵ → Error.

$y = mx + c$

$y_i = \beta_0 + \beta_1 x_i + \epsilon_i$

β_0 → intercept, β_1 → slope.

Table ←

$$\text{Calculate } (\beta_1) = \frac{\text{Sum of product of deviation}}{\text{Sum of square of deviation for } x}$$
(slope)

Calculate (β_0) Intercept = mean of (y) − (β_1 * mean of (x))

$\Rightarrow \beta_1 = \frac{12}{4} = 3$

$\Rightarrow \beta_0 = 13 - (3 * 10) = -17$

$y = \beta_0 + \beta_1 x$

$= 3 + (-17)x.$

$= 3 - 17x.$

7/3/25

2. Least Square Method:

- Maximum likelyhood estimate by using least square method.
- least square method is used to reduce the error.

x	y	mean (x)	mean (y)	Deviation (x)	Deviation (y)	Product of Deviation	Sum of product of deviation.	Square of Deviation of x.
2	5			-4	-8	32	128	16
3	6			-3	-7	21		9
4	10			-2	-3	6		4
5	12	6	13.	-1	-1	1		1
6	11			0	-2	0		0
7	14			1	1	1		1
8	18			2	5	10		4
9	20			3	7	21		9
10	22			4	9	36		16

$$\beta_1 = \frac{\text{Sum of product of deviation}}{\text{Sum of square of deviation of } x}$$

$$= \frac{128}{60}$$

$$= 2.13$$

$$\beta_0 = \text{mean of } (y) - (\beta_1 * \text{mean of } (x))$$

$$= 13 - (2.13 * 6)$$

$$= 0.22$$

$$y = \beta_0 + \beta_1 x$$

$$= 0.22 + 2.13x$$

$y = 0.22 + 2.13x$

→ when $x = 2$

$y = 0.22 + 2.13(2)$

$= 4.48$

→ when $x = 3$

$y = 0.22 + 2.13(3)$

$= 6.61$

→ $x = 4 \Rightarrow 8.74$

→ $x = 5 \Rightarrow 10.87$

→ $x = 6 \Rightarrow 13$

→ $x = 7 \Rightarrow 15.13$

→ $x = 8 \Rightarrow 17.26$

→ $x = 9 \Rightarrow 19.39$

→ $x = 10 \Rightarrow 21.52$

Predicted y values.

3. Assessing the ~~value~~ Accuracy of the model.

1. Sum of Squared residuals errors (SSE)
2. Mean Squared error (MSE)
3. Root Mean Squared error (RMSE)

→ Squared residuals error means how far the actual data points are from the predicted ~~data~~ regression line.

- a smaller ~~SRE~~ SSE means a better fit, while a larger ~~SRE~~ SSE means the model is making bigger mistakes.

y Predicted y values

$\Rightarrow (5 - 4.48)^2 = 0.2704$

$\Rightarrow (6 - 6.61)^2 = 0.3721$

$\Rightarrow (10 - 8.74)^2 = 1.5876$

$\Rightarrow (12 - 10.87)^2 = 1.2769$

$\Rightarrow (11 - 13)^2 = 4$

$\Rightarrow (14 - 15.13)^2 = 1.2769$

$\Rightarrow (18 - 17.26)^2 = 0.5476$

$\Rightarrow (20 - 19.39)^2 = 0.3721$

$= (22 - 21.52)^2 = 0.2304$

9.934

$\frac{9.934}{9} = 1.103$ → mean squared error

$= \sqrt{1.103}$ → root mean squared error

$= 1.050$

Cont of Problem ①

$y = 3 - 17x$

$x = 8 \Rightarrow 3 - 17(8) = -133$

$x = 10 \Rightarrow 3 - 17(10) = -167$

$x = 12 \Rightarrow 3 - 17(12) = -201$

} Predicted y values.

Sum of Squared residuals errors.

$(10 - (-133))^2 = 20449$

$(13 - (-167))^2 = 32400$

$(16 - (-201))^2 = 47089$

$\Rightarrow 20449 + 32400 + 47089 = 99938$

$$\frac{99938}{3} = 33312.67$$

$$\sqrt{33312.67} = 182.52.$$

11/3/24

Multiple linear regression Model:

- involves multiple predictors or independent variables and one dependent variable.
- This is an extension of linear regression model.
- The multiple regression of 2 variables x_1 & x_2 is given as follows.

$$\boxed{y = f(x_1, x_2)}$$

$$\boxed{y = a_0 + a_1 x_1 + a_2 x_2} + \epsilon$$

In general this is given for 'n' independent variables.

$$\Rightarrow \boxed{y = f(x_1 x_2 \cdots x_n)}$$

$$\boxed{y = a_0 + a_1 x_1 + a_2 x_2 + a_3 x_3 + \cdots a_n x_n + \epsilon}$$

Here $x_1 x_2 x_3 \cdots x_n$ are independent variables or predictor variables

y is a dependent variable

$a_0, a_1, a_2 a_3 \cdots a_n$ are coefficients of the regression eqn.

ϵ is error.

Ex: Apply multiple regression for the values given in the table. where weekly sales along with sales for products x_1 & x_2

X_1 (Product x_1)	X_2 (Product x_2)	y (weekly sales)
1	4	1
2	5	6
3	8	8
4	2	12.

Use matrix approach for finding multiple regression eqn

$$y = a_0 + a_1 x_1 + a_2 x_2$$

(default)

$$\begin{bmatrix} 1 \\ 6 \\ 8 \\ 12 \end{bmatrix} = x \begin{array}{c} \begin{array}{ccc} a_0 & x_1 & x_2 \end{array} \\ \begin{bmatrix} 1 & 1 & 4 \\ 1 & 2 & 5 \\ 1 & 3 & 8 \\ 1 & 4 & 2 \end{bmatrix} \end{array} \qquad \hat{a} = \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix}$$

- The coefficient of multiple regression eq^n is given as

$$\hat{a} = \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix}$$

- The regression coefficients for multiple regression is calculated the same way as linear regression.

$$\boxed{\hat{a} = \left((X^T X)^{-1} X^T \right) y}$$

$$X^T \Rightarrow \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 2 & 3 & 4 \\ 4 & 5 & 8 & 2 \end{bmatrix}$$

$$X^T X \Rightarrow \begin{bmatrix} 4 & 10 & 19 \\ 10 & 30 & 46 \\ 19 & 46 & 109 \end{bmatrix}$$

$$(X^T X)^{-1} \Rightarrow \begin{bmatrix} 3.15 & -0.59 & -0.3 \\ -0.59 & 0.204 & 0.0163 \\ -0.3 & 0.0163 & 0.0546 \end{bmatrix} \times \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 2 & 3 & 4 \\ 4 & 5 & 8 & 2 \end{bmatrix}$$

$$(X^T X)^{-1} X^T \Rightarrow \begin{bmatrix} 1.36 & 0.47 & -1.02 & 0.19 \\ -0.32 & -0.11 & 0.13 & 0.24 \\ -0.06 & 0.002 & 0.18 & -0.12 \end{bmatrix} \times \begin{bmatrix} 1 \\ 6 \\ 8 \\ 12 \end{bmatrix}$$

$$\left((X^T X)^{-1} X^T\right) y \Rightarrow \begin{bmatrix} -1.69 \\ 3.48 \\ -0.05 \end{bmatrix} \Rightarrow \hat{a}$$

$\Rightarrow a_0 = -1.69$; $a_1 = 3.48$; $a_2 = -0.05$

$y = a_0 + a_1 x_1 + a_2 x_2$

$$\boxed{y = -1.69 + 3.48 x_1 - 0.05 x_2}$$

$x_1 = 1 \quad x_2 = 4$

$y = -1.69 + 3.48(1) - 0.05(4) = 1.59$

$x_1 = 2 \quad x_2 = 5$

$y = -1.69 + 3.48(2) - 0.05(5) = 5.02$

$x_1 = 3 \quad x_2 = 8$

$y = -1.69 + 3.48(3) - 0.05(8) = +8.35$

$x_1 = 4 \quad x_2 = 2$

$y = -1.69 + 3.48(4) - 0.05(2) = 12.13$

<u>Errors</u>

Predicted y

y

$1 - 1.59 = 0.59$

$6 - 5.02 = 0.98$

$8 - 8.35 = -0.35$

$12 - 12.13 = -0.13.$

1.09 (SRE)

$\frac{1.09}{4} = 0.2725$ (MSE)

$\sqrt{0.2725} = 0.523$ (RMSE)

12/3/25

(2) The data in the table relate grams plant dry weight (y) to present soil organic matter (x_1) and kilograms supplymental of nitrogen.

soil organic x_1	nitrogen x_2	dry plant (grams) weight. y
7	2.6	78.5
1	2.9	74.5
11	5.6	104.3
11	3.1	87.6
7	5.2	95.9
11	5.5	109.2
3	7.1	102.7

$y = a_0 + a_1 x_1 + a_2 x_2$

$\hat{a} = ((x^T x)^{-1} x^T) y$

$$\begin{bmatrix} 78.5 \\ 74.5 \\ 104.3 \\ 87.6 \\ 95.9 \\ 109.2 \\ 102.7 \end{bmatrix} = \begin{bmatrix} 1 & 7 & 2.6 \\ 1 & 1 & 2.9 \\ 1 & 11 & 5.6 \\ 1 & 11 & 3.1 \\ 1 & 7 & 5.2 \\ 1 & 11 & 5.5 \\ 1 & 3 & 7.1 \end{bmatrix} \Rightarrow x.$$

$$x^T = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 7 & 1 & 11 & 11 & 7 & 11 & 3 \\ 2.6 & 2.9 & 5.6 & 3.1 & 5.2 & 5.5 & 7.1 \end{bmatrix} \times \begin{bmatrix} 1 & 7 & 2.6 \\ 1 & 1 & 2.9 \\ 1 & 11 & 5.6 \\ 1 & 11 & 3.1 \\ 1 & 7 & 5.2 \\ 1 & 11 & 5.5 \\ 1 & 3 & 7.1 \end{bmatrix}$$

$x^T x =$

$$X^T X = \begin{bmatrix} 7 & 51 & 32 \\ 51 & 471 & 235 \\ 32 & 235 & 163.84 \end{bmatrix}$$

$$(X^T X)^{-1} = \begin{bmatrix} 1.7995 & -0.068 & -0.253 \\ -0.068 & 0.01 & -0.001 \\ -0.0253 & -0.001 & 0.057 \end{bmatrix} \times X^T$$

$$(X^T X)^{-1} X^T =$$

$$\begin{bmatrix} 0.6615 & 0.9969 & -0.3721 & 0.2608 & 0.003 & -0.3468 & -0.2035 \\ -0.0008 & -0.0616 & 0.036 & 0.0390 & -0.0035 & 0.0364 & -0.0459 \\ -0.1122 & -0.0887 & 0.0547 & -0.0879 & 0.0362 & 0.0490 & 0.1489 \end{bmatrix}$$

$$((X^T X)^{-1} X^T) y$$

$$\Rightarrow \begin{bmatrix} 51.769 \\ 1.485 \\ 6.706 \end{bmatrix}$$

$$y = 51.9 + 1.5x_1 + 6.5x_2$$

$$y = 51.9 + 1.5x_1 + 6.5x_2.$$

$x_1 = 7 \qquad x_2 = 2.6$

$y = 51.9 + 1.5(7) + 6.5(2.6) \Rightarrow 79.3$

$y = 51.9 + 1.5(1) + 6.5(2.9) \Rightarrow 72.25$

$y = 51.9 + 1.5(11) + 6.5(5.6) \Rightarrow 104.8$

$y = 51.9 + 1.5(11) + 6.5(3.1) \Rightarrow 88.55$

$y = 51.9 + 1.5(7) + 6.5(5.2) \Rightarrow 96.2$

$y = 51.9 + 1.5(11) + 6.5(5.5) \Rightarrow 104.15$

$y = 51.9 + 1.5(3) + 6.5(7.1) \Rightarrow 102.55$

<u>Errors</u>

$$\left.\begin{aligned}
78.5 - 79.3 &= -0.8 \\
74.5 - 72.25 &= 2.25 \\
104.3 - 104.8 &= -0.5 \\
87.6 - 88.55 &= -0.95 \\
95.9 - 96.2 &= -0.3 \\
109.2 - 104.15 &= 5.05 \\
102.7 - 102.55 &= (0.15)^2
\end{aligned}\right\} 32.47 \,//\, (SRE)$$

$$MSE = \frac{32.47}{7} = 4.638 \,//$$

$$RMSE = \sqrt{4.638} = 2.15 \,//$$

Attribute types

1. Nominal
2. Ordinal
3. Internal
4. Ratio.

Nature of Attribute

It contains names

Speciality of Attribute

→ To differentiate one with another

[mean, median, mode]

Central tendency: ←

Mode operation to replace missing values

Nominal attribute: Categorical, Qualitative

Ex: Categorises, states or name of things

haircolor, occupation, idnumbers, pincodes.

An important nominal attribute is binary which exists in only two states either 0 or 1

binary nomial.

Symmetric Asymmetric.

- A binary nomial / attribute is symmetric if both of its states are equally valuable and carry the same weight

Ex: Male and female.

- A binary attribute is assymmetric if both of its outputs / outcomes are not equally valuable.

18/3/25

- The values of nominal attributes are just different names i.e, they provide only enough information to distinguish from one to other.

Eg: Zip codes, employee ids, eye colors, genders.

Ordinal Attribute: → arranging in order,

- The value of an ordinal attribute provide enough information to order objects

- An ordinal attribute is an attribute with possible values that have a meaningful order or ranking among them, but the magnitude between successive values is not known.

Ex: hardness of minarals [values are good, better, best]

Grades, street numbers etc..

- The central tendency of ordinal attribute can be represented by its mode & its median, but the mean cannot be defined.

Interval Attributes: → calculate diff b/w values.

- for Interval Attributes the differences between values are meaningful. i.e, unit of measurement exists [+, −]
- These attributes are measured on a scale of equal size units.
- We can compare & quantify the values between values of interval attributes.

Ex: A temperature attribute.

Calender dates.

- The central tendency of Interval attribute can be represented by its mode, its median and its mean.
- There is a no true zero point.

Ratio Attribute: → +, −, * calculation is possible.

- for ratio attribute both differences and ratios are meaningful.
- ratio attribute is a numeric attribute with an inherent zero point.

Ex: A no. of votes attribute is a ratio attributes.

A no. of words in a sentences attribute is a ratio attribute.

- Central tendency of a ratio attribute can be represented by its mean, median and mode.

mean → symmetric data (or) no outliers.

Median → skewed data (or) data without liners.

mode → catagorical data, finding common values.

Categorical vs numeric.

Categorical (qualitative nature)	Numeric. (quantitative nature)
- nominal and ordinal attributes are collectively refered to a categorical or qualitative attributes	- Interval and ratio are collectively refered to a quantitative or numeric attributes
- Qualitative attributes such as emp Id, lack most of the properties of numbers.	- Quantitative attributes are represented by numbers and most of the properties of numbers
- Even if they are represented by no.s i.e, integers, they should be treated more like symbols.	- Quantitative attributes can be integer valued or continuous.
- Mean of values does not have any meaning.	- Numeric operations such as mean, standard deviation are meaningful.

19/3/25

Logistic Regression Model:

→ generates Continuous Values

Yes No.

1. What is the model
2. Importance / Purpose. → if the output is categorical, this method is used.
3. Equation evaluation of the model.
4. Problem.

→ max & min → (−∞ to ∞).

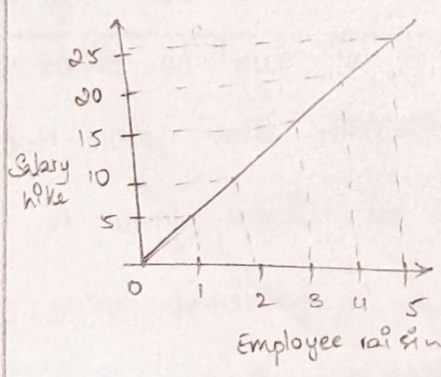
- Linear regression predicts the numerical responce but it is not suitable for predicting the categorical variables.
- When categorical variables are involved; it is called classification ~~variable~~ problem.

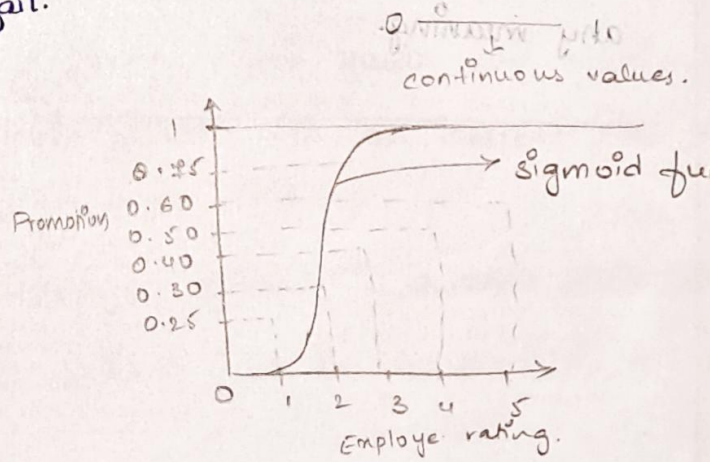
→ max & min → (0 to 1) Probability.

- logistic regression is suitable for binary classification problem.

Ex: Different context for logistic regression.

1) mail spam or not spam.
   ↳ categorical nature : Yes / No.
2) Student admited or not.
3) Student pass or fail.





- consider the above example, an organisation wants to determine an employee's salary increase based on their performance for this purpose a liner regression algorithm will help them to decide.
- Now what if the organisation wants to know whether an employee would get a promotion or not based on their performance.

graph-2 ↑

- The above graph based on threshould value, the organisation can decide [whether an employee] is promoted.

Equation of the Model.

To understant logistic regression, lets go over the odds of success.

$$\text{odds of success } (\theta) = \frac{\text{Probability of an event happening.}}{\text{Probability of an event not happening.}}$$

$$\theta = \frac{P}{1-P}$$

- The values of odds range between 0 to ∞

P happening: $\frac{1}{1-1} = \frac{1}{0} = \infty$ [Max].

P not happening: $\frac{0}{1-0} = \frac{0}{1} = 0$ [Min]

$$\text{if; } \underbrace{\frac{P}{1-P}}_{\text{range } 0 \text{ to } \infty} = \underbrace{\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \cdots \beta_n x_n}_{\text{range } -\infty \text{ to } \infty}$$

$$\text{then; } \boxed{\log\left(\frac{P(x)}{1-P(x)}\right) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots \beta_n x_n}$$

exponentiating both sides.

$$e^{\ln\left(\frac{P(x)}{1-P(x)}\right)} = e^{\beta_0 + \beta_1 x_1} \qquad \boxed{e^{\ln(x)} = x}$$

$$\boxed{\frac{P(x)}{1-P(x)} = e^{\beta_0 + \beta_1 x_1}}$$

let $y = e^{\beta_0 + \beta_1 x_1}$

$$\Rightarrow \frac{P(x)}{1-P(x)} = y.$$

$$P(x) = y(1-P(x))$$

$$P(x) = y - yP(x)$$

$$P(x) + y\,P(x) = y$$

$$P(x)\,(1+y) = y$$

$$\boxed{P(x) = \frac{y}{1+y}}$$

$$P(x) = \frac{e^{\beta_0 + \beta_1 x_1}}{1 + e^{\beta_0 + \beta_1 x_1}}$$

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

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

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

↳ sigmoid eqn.

25/3/25

① In the student dataset. the entrance marks based on the historic data ~~based~~ of those who are selected or not selected. based on the logistic regression, the values of learnt parameters ($\beta_0, \beta_1, \beta_2$) are $\beta_0 = 1$ & $\beta_1 = 8$. Assuming the marks of x student, $x = 60$ Threashold limit $= 0.50$

Compute the resultant class.

$\beta_0 = 1 \quad \beta_1 = 8$

$z = \beta_0 + \beta_1 x$

$z = 1 + 8 \times 60 = 481$

$$\frac{1}{1+e^{-z}} \Rightarrow \frac{1}{1+e^{-481}} = \frac{1}{1+1.271} = \frac{1}{2.271} = 0.44$$

Then it is observed that $0.44 < 0.50$

∴ The candidate with exam's 60 is not selected.

② Assume the model suggested by the optimizer for odds of passing the course is log (odds) = $-64 + 2 * \text{hours}$. The data set of pass / fail in an exam of 5 students is given in the table. Use logistic regression as classifier to answer the following questions

i) Calculate the probability for pass for the student who studied for 33 hours.

ii) At least how many hours student should study that makes he will pass the course with the probability of more than 95%.

Hours study	Pass (1)/fail (0)
29	0
15	0
32	1
28	1
39	1

i) $\log(\text{odds}) = \underset{\beta_0}{-64} + \underset{\beta_1}{2} * \text{hours}$

$$P(x) = \frac{1}{1+e^{-(\beta_0 + \beta_1 x + \beta_2 x + \cdots)}}$$

$$= \frac{1}{1+e^{-2}} = 0.88$$

$z = \beta_0 + \beta_1 x$

$= -64 + 2(33)$

$= 2$

11b $0.95 = \frac{1}{1+e^{-z}}$

$0.95(1+e^{-z}) = 1$

$0.95 + 0.95 \times e^{-z} = 1$

$0.95 \times e^{-z} = 1 - 0.95$

$e^{-z} = \frac{1-0.95}{0.95}$

$e^{-z} = \frac{0.05}{0.95}$

ln on both sides.

$\ln(e^{-z}) = \ln(0.0526)$

$-z = -2.945$

$z = 2.94$

$z = \beta_0 + \beta_1 x$

$2.94 = -64 + 2x$

$2.96 + 64 = 2x$.

$x = \frac{66.96}{2}$

$x = 33.47$

$\therefore$ Students who studied for atleast 33.47 hours will get a 95% on the exam.

Confussion Matrix

A confussion matrix is a table used to evaluate the performance of classification model. Based on a confussion matrix, we can calculate

1. accuracy of the model.
2. Precision of the model
3. Recall / sensitivity / True positive rate
4. f_1 - score / True negative / Specificity.
5. false positive rate
6. Miss classification.
7. prevalence.

	Predicted (NO)	Predicted (Yes).
Actual (NO)	45 (True negative)	5 (false positive).
Actual (Yes)	5 (FN)	95 (TP)

- Consider the confusion matrix given above for binary classifier. predicting the presence of a diesease. The classifier made a total of 150 predictions. Out of those the classifier predicted 'yes' for 100 times and 'no' for 50 times.

In reality 100 patients in the sample have disease. and 50 patients don't.

1. Accuracy: Over all, how often is the classifier correct.

$$\text{Accuracy} = \frac{TN + TP}{TN + TP + FP + FN} = \frac{45 + 95}{150} = 93.33\%.$$

2. Miss classification rate: overall, how often is it wrong.

$$\text{Miss classification} = \frac{FN + FP}{TN + TP + FP + FN} = \frac{5+5}{150} = 6.67\%.$$

3. True positive rate / Sensitivity / Recall:

When its actually 'yes', how often does it predict 'yes'.

$$\text{Recall} = \frac{TP}{\text{Actual yes}} = \frac{95}{100} = 95\%$$

4. False positive rate:

When its actually 'no', how often does it predict 'yes'

$$\text{False positive rate} = \frac{FP}{\text{Actual No}} = \frac{5}{50} = 10\%$$

5. True negative rate / Specificity: / f1-score:

When its actually 'no', how often does it predict 'no'

$$\text{True negative rate} = \frac{TN}{\text{Actual No}} = \frac{45}{50} = 90\%$$

[TN + FP]

6. Precision:

When it predicts 'yes', how often is it correct?

$$\text{Precision} = \frac{TP}{TP + FP} = \frac{95}{100} = 95\%$$

7. Prevalence:

How often does the 'yes' condition, actually occur in our sample?

$$\text{Prevalence} = \frac{TP + FN}{\text{Total samples}} = \frac{100}{150} = 66.67\%$$

[TN + FP + FN + TP]

Confusion matrix of COVID-19.

	Predicted (Yes) (Positive)	Predicted (No) (Negative).
Actual (Yes) covid +	TP 40	FN 10
Actual (No) Covid -	FP 5	TN 45

$$\text{Accuracy} = \frac{TP+TN}{\text{Total}} = \frac{85}{100} = 85\%.$$

$$\text{Miss classification} = \frac{FN+FP}{\text{Total}} = \frac{15}{100} = 15\%$$

$$\text{Recall} = \frac{TP}{\text{Actual Yes}} = \frac{40}{50} = 80\%.$$

$$\text{False positive rate} = \frac{FP}{\text{Actual Yes}} = \frac{5}{50} = 10\%.$$

$$\text{True negative rate} = \frac{TN}{\text{Actual no}} = \frac{45}{50} = 90\%.$$

$$\text{Precision} = \frac{TP}{TP+FP} = \frac{40}{45} = 89\%$$

$$\text{Prevalence} = \frac{TP+FN}{\text{Total}} = \frac{50}{100} = 50\%.$$

(a) Using the logistic regression model with 2 predictors for the riding mover example with $B_0 = -25.9382$; $B_1 = 0.1109$ — "Income" ; $B_2 = 0.9638$ — "lot_size".

Where B_1 & B_2 are the of the income & lot_size.

Using the logestic regression model with propability cut of 0.75, classify the following 6 customers as "owner", or "non owner" present the results in a classification matrix & find the accuracy of the model based on a given data set.

	Income	lot_size	ownership.	Predicted.
i)	60.0	18.4	owner	=> non-owner
ii)	64.8	21.6	owner	=> owner
iii)	84.0	17.6	non-owner	=> non-owner
iv)	59.4	16.0	non-owner	=> non-owner
v)	108.0	17.6	owner	=> owner
vi)	75.0	19.6	non-owner	=> owner

i)

$$P(x) = \frac{1}{1+e^{-(B_0+B_1x+B_2x)}}$$

$$= \frac{1}{1+e^{-1.550}}$$

$$= 0.175$$

$$Z = B_0 + B_1x + B_2x$$

$$= -25.9382 + 0.01109(60) + 0.9638(18.4)$$

$$= -1.550$$

$\because 0.175 < 0.75$

$\therefore$ non-owner.

ii) $P(x) = \frac{1}{1+e^{-2.0662}}$

$= 0.887$

$\because 0.887 > 0.75$

$\therefore$ owner.

$z = \beta_0 + \beta_1 x + \beta_2 x$

$= -25.9382 + 0.1109(64.8)$

$+ 0.9638(21.6)$

$= 2.0662$

iii) $P(x) = \frac{1}{1+e^{-0.34028}}$

$= 0.5842$

$\therefore$ non-owner.

$z = -25.9382 + 0.1109(84.0)$

$+ 0.9638(17.6)$

$= 0.34028$

iv) $P(x) = \frac{1}{1+e^{-(-3.92994)}}$

$= 0.019$

$\therefore$ non-owner.

$z = -25.9382 + 0.1109(59.4)$

$+ 0.9638(16.0)$

$= -3.92994$

v) $P(x) = \frac{1}{1+e^{-3.00188}}$

$= 0.952$

$\therefore$ Owner

$z = -25.9382 + 0.1109(108.0)$

$+ 0.9638(17.6)$

$= 3.00188$

vi) $P(x) = \frac{1}{1+e^{-1.26978}}$

$= 0.7807$

$\therefore$ Owner.

$z = -25.9382 + 0.1109(75.0)$

$+ 0.9638(19.6)$

$= 1.26978$

	Actual class	Predicted class.
1.	owner	non-owner
2.	owner	owner
3.	non-owner	non-owner
4.	non-owner	non-owner
5.	owner	owner
6.	non-owner	owner.

	Predicted (owner)	Predicted (non-owner)
Actual (owner)	2 (TP)	1 (FN)
Actual (non-owner)	1 (FP)	2 (TN)

⇒ Accuracy. $= \frac{TN+TP}{Total} = \frac{4}{6} = \cancel{70\%}$ 66.33%

⇒ f1-Score $= \frac{TN}{\text{Actual no}} = \frac{2}{3} = 66.33\%$

⇒ Recall $= \frac{TP}{\text{Actual yes}} = \frac{2}{3} = 66.33\%$

⇒ Precision. $= \frac{TP}{TP+FP} = \frac{2}{2+1} = \frac{2}{3} = 66.33\%$.

26/3/25

K- Nearest Neighbors (KNN) Machine learning algorithm.

1) KNN - without weighted distance
   - ① discrete or categorical
   - ② Continuous or Numerical.
2) KNN - with weighted distance.
   - ③ discrete
   - ④ Continuous.

- The most basic instance - base method is K- nearest neighbour algorithm.
- The algorithm assumes all instances correspond to point in the end dimentional space R^n.
- The neatest neighbour of a instance are defined in terms of Standard Euclidean distance.
- The arbitiary instance 'x' be describe by the feature vectors / attributes

$$X \rightarrow (a_1(X), a_2(X), a_3(X), \ldots)$$

- let us assume x_i, x_j with two instances with attributes

$$x_i = a_1, a_2, a_3, \ldots a_n$$

$$x_j = a_1, a_2, a_3, \ldots a_n$$

- The distance between 2 instances is defined as (x_i, x_j) ⇒

$$(x_i, x_j) = d(x_i, x_j) = \sqrt{(a(x_i) - a_1(x_j))^2 + (a_2(x_i) - a_2(x_j))^2 \ldots}$$

- in the nearest neighbour algorithm the target function maybe either discrete value or real value.

Ans.

Q.

Height a_1	weight a_2	Target variable.	Distance.
150	50	Medium	8.06
155	55	Medium	2.21 — ①
160	60	large	6.71 — ③
161	59	large	6.41 — ②
158	65	large	11.05
x_q 157	54	large ?	

$$\text{distance} = \sqrt{(a_1(x_i) - (a_1(x_j)))^2 + (a_2(x_i) - a_2(x_j)) - a_2(x_j))^2}$$

③ $= \sqrt{(150-157)^2 + (50-54)^2}$

② $= \sqrt{49+16}$

$= \sqrt{65} = 8.06$

① , ② , ③ are the closest neighbours for 157 , 54

any no of neighbours can be considered [usually given]

$K = 3$ [Considered neighbours].

[closest neighbours for given values]

$$\hat{f}(x_q) \longleftarrow \arg\max \sum_{i=1}^{K} \delta(v, f(x_i)).$$

discrete function [only 2 outputs]

$v \Rightarrow$ medium/large.

$f(x_i) \Rightarrow$ neighbour's Target

where $\delta(a,b) = 1$ if $a = b$

where $\delta(a,b) = 0$ if $a \neq b$.

considering ①'s Target [medium]

$\hat{f}(x_q) = \delta(\text{medium}, \text{medium}) + \delta(\text{medium}, \text{large}) + \delta(\text{medium}, \text{large})$

$= 1 + 0 + 0 = 1$

considering ②'s & ③'s Target [large]

$\hat{f}(x_q) = \delta(\text{large}, \text{medium}) + \delta(\text{large}, \text{large}) + \delta(\text{large}, \text{large})$

$= 0 + 1 + 1 = 2.$

$\therefore$ Max value is 2 $\therefore \hat{f}(x_q) = 2 \Rightarrow$ large.

$\therefore$ Target variable for 157 , 54 is large. ✓

Q.

Training Instance	x_1	x_2	output	Distance.
I_1	7	7	0	4 — ③
I_2	7	4	0	5
I_3	3	4	1	3 — ①
I_4	1	4	1	3.60 — ②
	3	7	1 //	

$\Rightarrow \sqrt{(x_2-x_1)^2+(y_2-y_1)^2}$

$K = 3 \qquad t_1 = (3,7) \Rightarrow$ o/p ?

$K = 3$

① 's Output $\Rightarrow 1$

$\hat{f}(x_q) = \delta(1,1) + \delta(1,1) + \delta(1,0)$

$= 1 + 1 + 0 = 2.$

Output $\Rightarrow 0.$

$\hat{f}(x_q) = \delta(0,1) + \delta(0,1) + \delta(0,0)$

$= 0 + 0 + 1 = 1$

Max argument $= 2 \Rightarrow$ Output is 1

$\therefore$ Output of $t_1(3,7) = 1$

Q. [Same as question ① Dataset with same distances] X.

Height	weight	Target variable.	Distance.
150	50	1.5	8.06
155	55	1.2	2.21 — ①
160	60	1.8	6.71 — ③
161	59	2.1	6.40 — ②
158	67	1.7	16.03
157	54	?	

$K = 3.$

$\hat{f}(x_q) = \frac{1.2 + 2.1 + 1.8}{3}$

$= 1.7$

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

$\Downarrow$

Continuous function

28/3/25

Using weighted (KNN).

Height	weight	Target	Distance	weight = $\frac{1}{\text{distance}}$
150	50	Medium	8.06	0.12
155	55	Medium	2.24 — 1	0.45
160	60	Large	6.71 — 3	0.15
161	59	Large	6.40 — 2	0.16
158	65	Large	11.05	0.09
157	54	Large		

$$\hat{f}(x_q) = \arg\max_{v \in V} \sum_{i=1}^{k} w_i \, \delta(v, f(x_i))$$

⇒ Discrete

$= 0.45 \times \delta(\text{medium}, \text{medium}) + 0.16 \times \delta(\text{medium}, \text{large}) + 0.15 \times \delta(\text{medium}, \text{large})$

medium $= 0.45 \times 1 + 0.16 \times 0 + 0.15 \times 0 = \underline{\underline{0.45}}$

large ⇒ $0.45 \times \delta(\text{large}, \text{medium}) + 0.16 \times \delta(\text{large}, \text{large}) + 0.15 \times \delta(\text{large}, \text{Large})$

$= 0.45 \times 0 + 0.16 \times 1 + 0.15 \times 1 = \underline{0.31}$

Height	weight	Target	distance	weight = $\frac{1}{\text{distance}}$
150	50	1.5	8.06	0.12
155	55	1.2	2.24 — 1	0.45
160	60	1.8	6.71 — 3	0.15
161	59	2.1	6.40 — 2	0.16
158	65	1.7	11.05	0.09
157	54	?		

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

⇓

Continuous.

$$\hat{f}(x_q) = \frac{0.45 \times 1.2 + 0.16 \times 2.1 + 0.15 \times 1.8}{0.45 + 0.15 + 0.16}$$

$= 1.50.$

Q. A restaurant 'A' sells burger with optional flavours; pepper, ginger, chilly. Every day this week we tried a burger A to E and kept a record of which you like using Hamming distance. Show how the ~~3nnn~~ 3nn classifier with majority voting would classify. pepper = F, ginger = T, chilly = T. liked = ?

Sol:

	Pepper	ginger	chilly	liked	Hamming distance
A	T	T	T	F	1+0+0=1 — 3
B	T	F	F	T	1+1+1=3
C	F	T	T	F	0+0+0=0 — 1
D	F	T	F	T	0+0+1=1 — 2
E	T	F	F	T	1+1+1=3
Sol	F	T	T		

Hamming Distance:

~~Here~~ Using Hamming distance formula,

- if value of x_1 is blue and x_2 is also blue then the hamming distance b/w x_1 & x_2 is 0.
- if value of x_1 is blue & x_2 is red then the value of hamming distance is 1.

$$\hat{f}(x_q) = \delta(T,F) + \delta(T,T) + \delta(T,F)$$

[T] $= 0 + 1 + 0 = 1$

$$\hat{f}(x_q) = \delta(F,F) + \delta(F,T) + \delta(F,F)$$

[F] $= 1 + 0 + 1 = 2$

upbill here for print.

SUPPORT VECTOR MACHINE LEARNING.

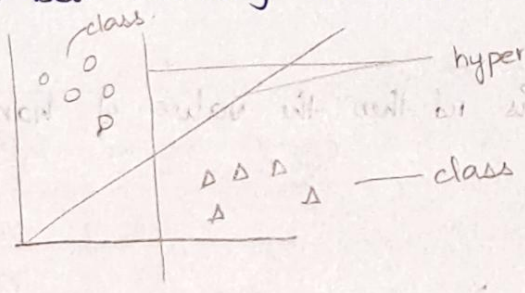
(SVM).

labelled data.

SVM is one of the most popular supervised learning algorithm. which is used for classification as well as regression.

Target variable nature is discrete continuous

- The goal of the SVM algorithm is to create the best line or decision boundary, that can segrigate n dimentional space into classes, so that we can easily classify the new data point in correct calogery in the future.
- The best decision boundary is called a hyperplane.
- There can be multiple lines or multiple decision boundaries to segrigate the classes in n dimentional space, but we need to findout the best decision boundary that helps to classify the data points. this best boundary is known as hyperplane of SVM.



2/4/25

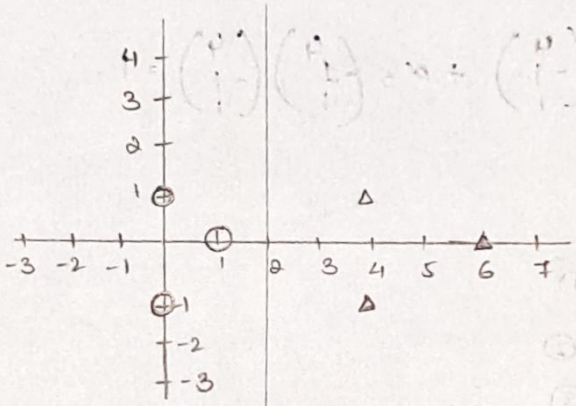
Linear SVM is used for linearly separable data. which mean, if a dataset can be classified into 2 classes by using a single strike line. Then such data is termed as linearly separable data and classifier is called as linearly SVM classifier.

Non-linear SVM is used for non linearly separable data. which means, if a dataset can't be classified using a straight line, then such data is termed as non-linear data & the

classifier is called as non-linear SVM classifier.

→ The data points or vectors that the closest to the hyperplane and which effect the position of ~~vector~~ hyper plane are termed as support vector. These vectors support the hyperplane hence called as support vector.

Q. Points (4,1) (4,-1) and (6,0) belongs to +ve class. → △
points (1,0) (0,1) and (0,-1) belongs to -ve class — ○.
Draw an optimal hyperplane to classify the points.



Linear Ex

It can be observed the support vectors are (1,0) (4,1) (4,-1)

$$S_1 = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \qquad S_2 = \begin{pmatrix} 4 \\ 1 \end{pmatrix} \qquad S_3 = \begin{pmatrix} 4 \\ -1 \end{pmatrix}$$

The augmented vectors can be obtained by adding the bias given as follows.

$$\hat{S}_1 = \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} \qquad \hat{S}_2 = \begin{pmatrix} 4 \\ 1 \\ 1 \end{pmatrix} \qquad \hat{S}_3 = \begin{pmatrix} 4 \\ -1 \\ 1 \end{pmatrix}$$

From these a set of 3 equations can be obtained based 3 support vectors as follows.

$$\alpha_1 \hat{S}_1 \hat{S}_1 + \alpha_2 \hat{S}_2 \hat{S}_1 + \alpha_3 \hat{S}_3 \hat{S}_1 = -1 \quad \rightarrow \hat{S}_1 \text{ belongs to -ve class}$$

$$\alpha_1 \hat{S}_1 \hat{S}_2 + \alpha_2 \hat{S}_2 \hat{S}_2 + \alpha_3 \hat{S}_3 \hat{S}_2 = +1$$

$$\alpha_1 \hat{S}_1 \hat{S}_3 + \alpha_2 \hat{S}_2 \hat{S}_3 + \alpha_3 \hat{S}_3 \hat{S}_3 = +1$$

} $\hat{S}_2$ & $\hat{S}_3$ belongs to +ve class

$$\alpha_1 \begin{pmatrix}1\\0\\1\end{pmatrix}\begin{pmatrix}1\\0\\1\end{pmatrix} + \alpha_2 \begin{pmatrix}4\\1\\1\end{pmatrix}\begin{pmatrix}1\\0\\1\end{pmatrix} + \alpha_3 \begin{pmatrix}4\\-1\\1\end{pmatrix}\begin{pmatrix}1\\0\\1\end{pmatrix} = -1$$

$$2\alpha_1 + 5\alpha_2 + 5\alpha_3 = -1$$

$$\alpha_1 \begin{pmatrix}1\\0\\1\end{pmatrix}\begin{pmatrix}4\\1\\1\end{pmatrix} + \alpha_2 \begin{pmatrix}4\\1\\1\end{pmatrix}\begin{pmatrix}4\\1\\1\end{pmatrix} + \alpha_3 \begin{pmatrix}4\\-1\\1\end{pmatrix}\begin{pmatrix}4\\1\\1\end{pmatrix} = 1$$

$$5\alpha_1 + 18\alpha_2 + 16\alpha_3 = 1$$

$$\alpha_1 \begin{pmatrix}1\\0\\1\end{pmatrix}\begin{pmatrix}4\\-1\\1\end{pmatrix} + \alpha_2 \begin{pmatrix}4\\1\\1\end{pmatrix}\begin{pmatrix}4\\-1\\1\end{pmatrix} + \alpha_3 \begin{pmatrix}4\\1\\1\end{pmatrix}\begin{pmatrix}4\\-1\\1\end{pmatrix} = 1$$

$$5\alpha_1 + 16\alpha_2 + 18\alpha_3 = 1$$

$$2\alpha_1 + 5\alpha_2 + 5\alpha_3 = -1 \quad \text{—— (1)}$$

$$5\alpha_1 + 18\alpha_2 + 16\alpha_3 = 1 \quad \text{—— (2)}$$

$$5\alpha_1 + 16\alpha_2 + 18\alpha_3 = 1 \quad \text{—— (3)}$$

$$\alpha_1 = -3 \qquad \alpha_2 = +1 \qquad \alpha_3 = 0$$

The optimal hyperplane is given as

$$\boxed{W = \sum_{i=1}^{i=3} \alpha_i * \hat{S}_i}$$

$$= -3 \times \begin{pmatrix}1\\0\\1\end{pmatrix} + 1 \times \begin{pmatrix}4\\1\\1\end{pmatrix} + 0 * \begin{pmatrix}4\\-1\\1\end{pmatrix}$$

$$= \begin{pmatrix}-3\\0\\-3\end{pmatrix} + \begin{pmatrix}4\\1\\1\end{pmatrix} + \begin{pmatrix}0\\0\\0\end{pmatrix}$$

$$= \begin{pmatrix}1\\1\\-2\end{pmatrix}$$

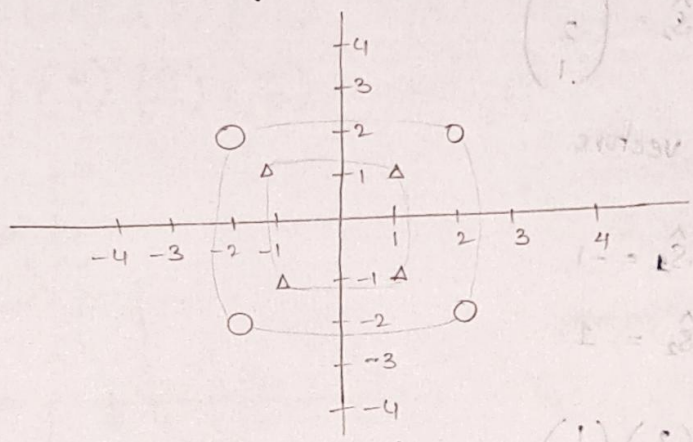
4/4/25

Non-linear Ex

Suppose we are given positive label points $\left\{\begin{pmatrix}2\\2\end{pmatrix}\begin{pmatrix}2\\-2\end{pmatrix}\begin{pmatrix}-2\\-2\end{pmatrix}\begin{pmatrix}-2\\2\end{pmatrix}\right\}$ +ve [O] labels.

Suppose we are given negative label points $\left\{\begin{pmatrix}1\\1\end{pmatrix}\begin{pmatrix}1\\-1\end{pmatrix}\begin{pmatrix}-1\\-1\end{pmatrix}\begin{pmatrix}-1\\1\end{pmatrix}\right\}$ -ve labels. [Δ]

Draw an optimal hyper plane.



in the above ploting points it is obvious no such hyper plane exist in input space. Therefore we must use a non-linear SVM that is we need to convert data from one feature space to another

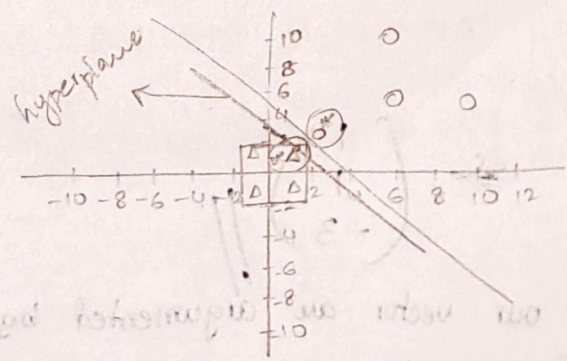
$$\begin{bmatrix}x_1\\x_2\end{bmatrix} = \begin{cases} 4 - x_2 + |x_1 - x_2| & \text{if } \sqrt{x_1^2+x_2^2} > 2 \\ 4 - x_1 + |x_1 - x_2| & \text{if } \sqrt{x_1^2+x_2^2} < 2 \\ \begin{bmatrix}x_1\\x_2\end{bmatrix} & \end{cases}$$

Positive lebeded points

$\begin{pmatrix}2\\2\end{pmatrix}\begin{pmatrix}10\\6\end{pmatrix}\begin{pmatrix}6\\6\end{pmatrix}\begin{pmatrix}6\\10\end{pmatrix}$

Negative labeled points

$\begin{pmatrix}1\\1\end{pmatrix}\begin{pmatrix}1\\-1\end{pmatrix}\begin{pmatrix}-1\\-1\end{pmatrix}\begin{pmatrix}-1\\1\end{pmatrix}$.



Support vectors

$$s_1 = \begin{pmatrix} 1 \\ 1 \end{pmatrix} \qquad s_2 = \begin{pmatrix} 2 \\ 2 \end{pmatrix}$$

augmented values

$$\hat{s}_1 = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \qquad \hat{s}_2 = \begin{pmatrix} 2 \\ 2 \\ 1 \end{pmatrix}$$

eq^{n}s for augmented vectors.

$$\alpha_1 \hat{s}_1 \hat{s}_1 + \alpha_2 \hat{s}_2 \hat{s}_1 = -1$$

$$\alpha_1 \hat{s}_1 \hat{s}_2 + \alpha_2 \hat{s}_2 \hat{s}_2 = 1$$

$$\alpha_1 \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}\begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} + \alpha_2 \begin{pmatrix} 2 \\ 2 \\ 1 \end{pmatrix}\begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} = -1 \Rightarrow 3\alpha_1 + 5\alpha_2 = -1$$

$$\alpha_2 \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}\begin{pmatrix} 2 \\ 2 \\ 1 \end{pmatrix} + \alpha_2 \begin{pmatrix} 2 \\ 2 \\ 1 \end{pmatrix}\begin{pmatrix} 2 \\ 2 \\ 1 \end{pmatrix} = 1 \Rightarrow 5\alpha_1 + 9\alpha_2 = 1$$

$$\alpha_1 = -7 \quad ; \quad \alpha_2 = 4.$$

The optimal hyperplane

$$\boxed{w = \sum_{i=1}^{n} \alpha_i \hat{s}_i}$$

$$\Rightarrow w = \sum_{i=1}^{2} \alpha_i \hat{s}_i$$

$$\Rightarrow \alpha_1 \hat{s}_1 + \alpha_2 \hat{s}_2$$

$$-7\begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} + 4\begin{pmatrix} 2 \\ 2 \\ 1 \end{pmatrix} \Rightarrow \begin{pmatrix} 1 \\ 1 \\ -3 \end{pmatrix} \Rightarrow$$ at (1, 1) with intercept of (3, 3).

finally remembering that our vector are argumented by the bias we can equate our last vector $\bar{w}$, hyperplane off set 'b' and write them separately

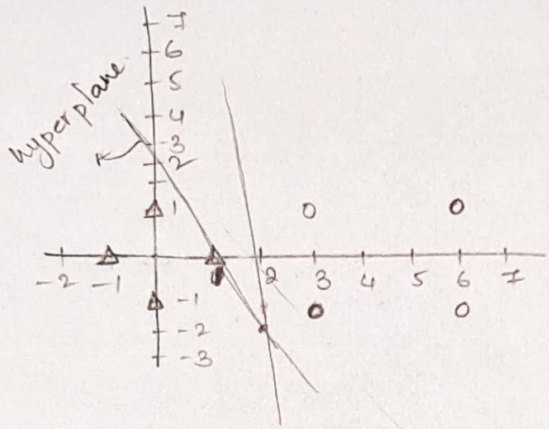
The hyperplane eq^n is $\boxed{y = wx + b}$

Linear SVM

Q) Find out the optimal hyperplane for the +ve labeled & -ve labeled by using linear SVM.

$\begin{pmatrix}3\\1\end{pmatrix} \begin{pmatrix}3\\-1\end{pmatrix} \begin{pmatrix}6\\1\end{pmatrix} \begin{pmatrix}6\\-1\end{pmatrix}$ +ve. [o]

$\begin{pmatrix}1\\0\end{pmatrix} \begin{pmatrix}0\\1\end{pmatrix} \begin{pmatrix}0\\-1\end{pmatrix} \begin{pmatrix}-1\\0\end{pmatrix}$ -ve. [Δ]



Support vectors:

$s_1 = \begin{pmatrix}1\\0\end{pmatrix}$ $s_2 = \begin{pmatrix}3\\1\end{pmatrix}$ $s_3 = \begin{pmatrix}3\\-1\end{pmatrix}$

augmented values

$\hat{s}_1 = \begin{pmatrix}1\\0\\1\end{pmatrix}$ $\hat{s}_2 = \begin{pmatrix}3\\1\\1\end{pmatrix}$ $\hat{s}_3 = \begin{pmatrix}3\\-1\\1\end{pmatrix}$

eq^n for augmented vectors

$\alpha_1 \hat{s}_1 \hat{s}_1 + \alpha_2 \hat{s}_2 \hat{s}_1 + \alpha_3 \hat{s}_3 \hat{s}_1 = -1$

$\alpha_1 \hat{s}_1 \hat{s}_2 + \alpha_2 \hat{s}_2 \hat{s}_2 + \alpha_3 \hat{s}_3 \hat{s}_2 = 1$

$\alpha_1 \hat{s}_1 \hat{s}_3 + \alpha_2 \hat{s}_2 \hat{s}_3 + \alpha_3 \hat{s}_3 \hat{s}_3 = 1$

$\Rightarrow \alpha_1 \begin{pmatrix}1\\0\\1\end{pmatrix}\begin{pmatrix}1\\0\\1\end{pmatrix} + \alpha_2 \begin{pmatrix}3\\1\\1\end{pmatrix}\begin{pmatrix}1\\0\\1\end{pmatrix} + \alpha_3 \begin{pmatrix}3\\-1\\1\end{pmatrix}\begin{pmatrix}1\\0\\1\end{pmatrix} = -1 \Rightarrow 2\alpha_1 + 4\alpha_2 + 4\alpha_3 = -1$

$\Rightarrow \alpha_1 \begin{pmatrix}1\\0\\1\end{pmatrix}\begin{pmatrix}3\\1\\1\end{pmatrix} + \alpha_2 \begin{pmatrix}3\\1\\1\end{pmatrix}\begin{pmatrix}3\\1\\1\end{pmatrix} + \alpha_3 \begin{pmatrix}3\\-1\\1\end{pmatrix}\begin{pmatrix}3\\1\\1\end{pmatrix} = 1 \Rightarrow 4\alpha_1 + 11\alpha_2 + 9\alpha_3 = 1$

$\Rightarrow \alpha_1 \begin{pmatrix}1\\0\\1\end{pmatrix}\begin{pmatrix}3\\-1\\1\end{pmatrix} + \alpha_2 \begin{pmatrix}3\\1\\1\end{pmatrix}\begin{pmatrix}3\\-1\\1\end{pmatrix} + \alpha_3 \begin{pmatrix}3\\-1\\1\end{pmatrix}\begin{pmatrix}3\\-1\\1\end{pmatrix} = 1 \Rightarrow 4\alpha_1 + 9\alpha_2 + 11\alpha_3 = 1$

$\alpha_1 = -3.5$ $\alpha_2 = 0.75$ $\alpha_3 = 0.75$

$\boxed{\bar{w} = \sum_{i=1}^{3} \alpha_i \hat{s}_i} \Rightarrow -3.5 \begin{pmatrix}1\\0\\1\end{pmatrix} + 0.75 \begin{pmatrix}3\\1\\1\end{pmatrix} + 0.75 \begin{pmatrix}3\\-1\\1\end{pmatrix}$

$= \begin{pmatrix}1\\0\\-2\end{pmatrix}$

$$= \sum_{d \in D} (t_d - o_d) \frac{\partial}{\partial w_i}$$

$$\boxed{\frac{\partial E}{\partial w_i} = \sum_{d \in D} (t_d - o_d)(-x_{id})}$$

$$\Delta w_i = -\eta \frac{\partial E}{\partial w_i}$$

$$\boxed{\Delta w_i = \eta \sum_{d \in D} (t_d - o_d)(x_{id})}$$

$$\boxed{w_i \leftarrow w_i + \Delta w_i}$$

$$= \sum_{d \in D} (t_d - o_d) \frac{\partial w_i}{\partial w_i}$$

$$\boxed{\frac{\partial E}{\partial w_i} = \sum_{d \in D} (t_d - o_d)(-x_{id})}$$

$$\Delta w_i = -\eta \frac{\partial E}{\partial w_i}$$

$$\boxed{\Delta w_i = \eta \sum_{d \in D} (t_d - o_d)(x_{id})}$$

$$\boxed{w_i \leftarrow w_i + \Delta w_i}$$

15/4/25

Ensemble Machine Learning:

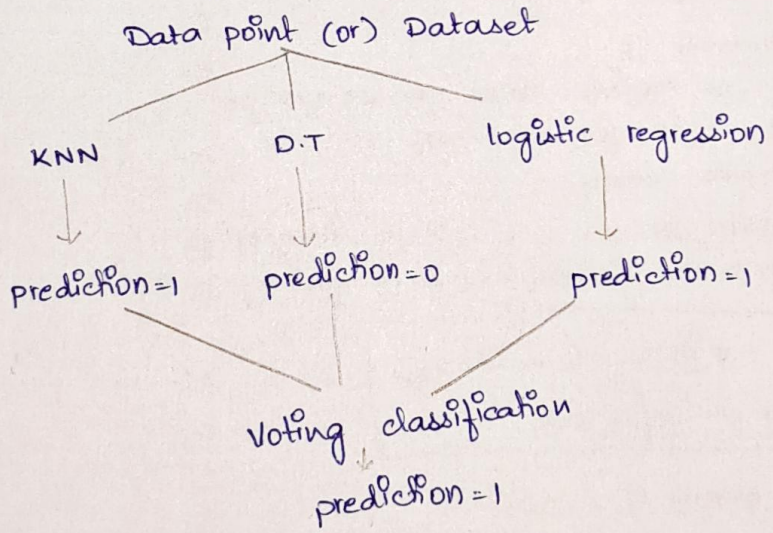
Ensemble learning is a supervised learning technique to improve overall performance of the model by combining the predictions from multiple models.

Methods

1. Voting
2. Bagging / Bootstrap aggregation.
3. Random forest
4. Boosting
5. Blending. / Stacked generalization

1. Voting:

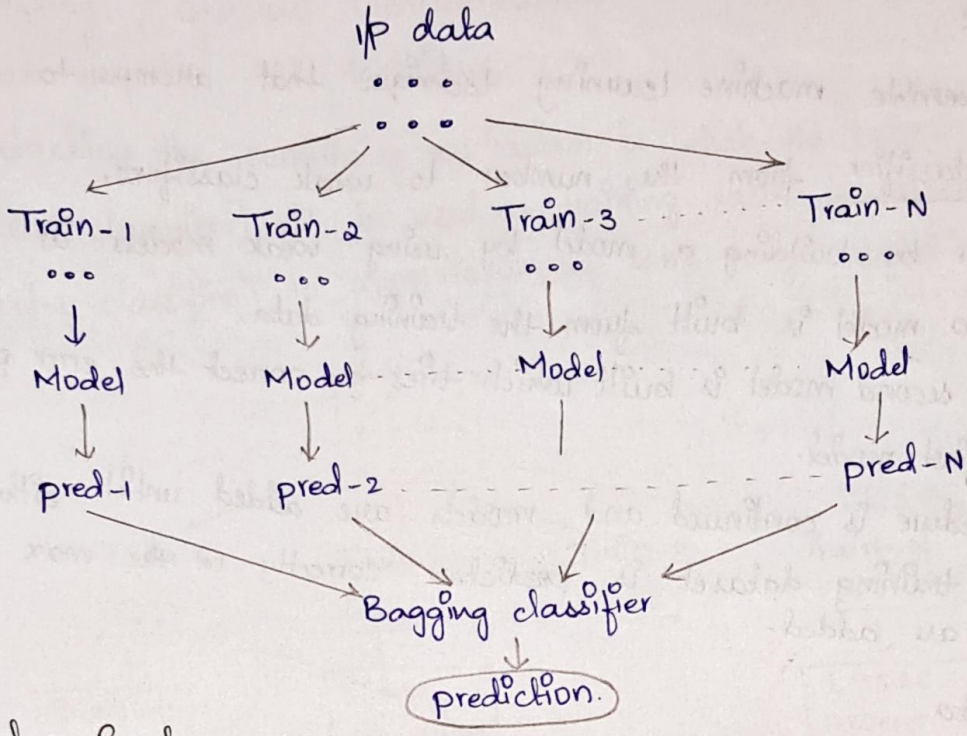
Voting is a ensemble machine learning algorithm that involves making a prediction. i.e., average [regression] (or) the sum (classification). of multiple machine learning models.



2. Bagging / Bootstrap aggregation.

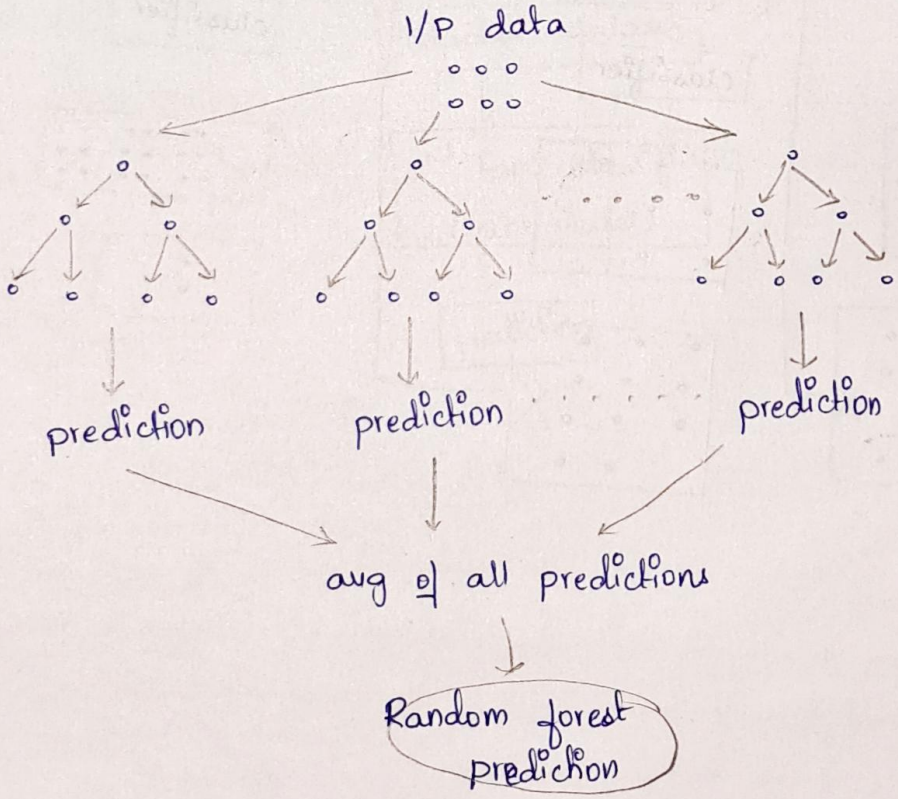
is a ensemble machine technique designed to improve stability & accuracy of ML algorithms like classification & regression.

- It decreases the variance ~~&~~ ~~health~~ helps to avoid overfitting
- It is usually applied to decision tree methods
- Bagging is a special case of model averaging approach.



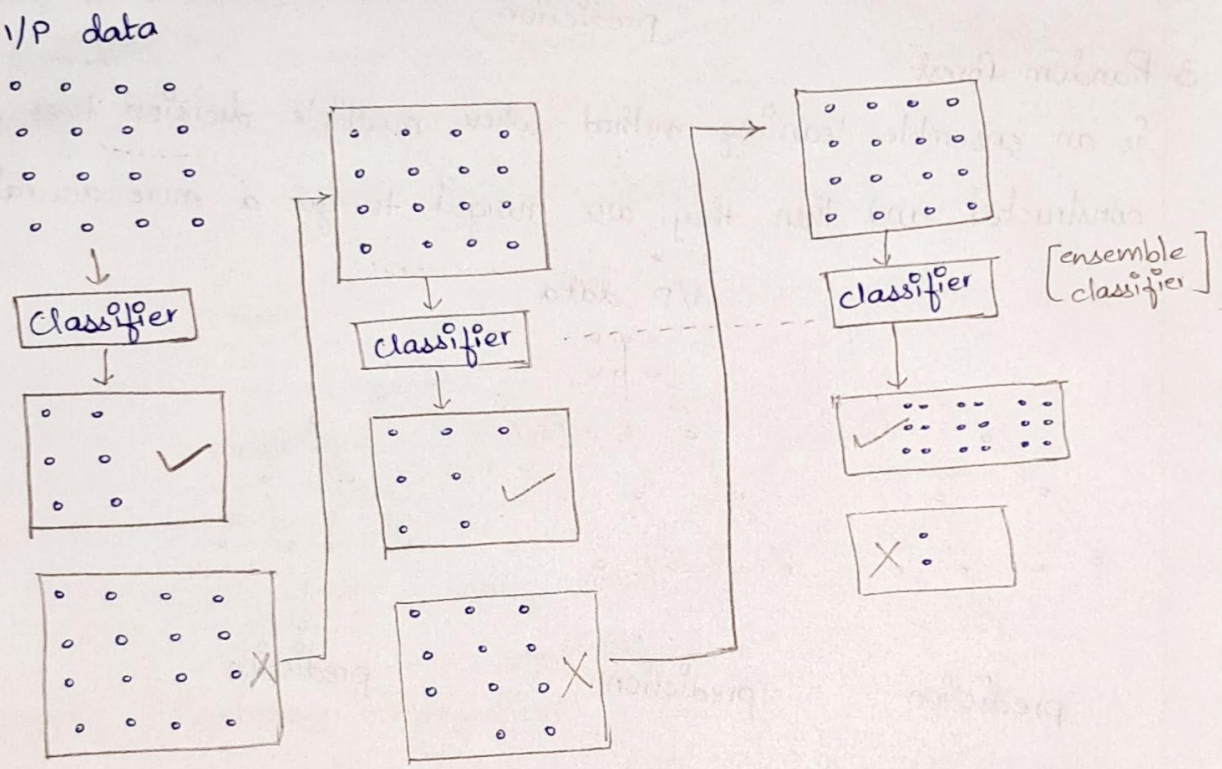
3. Random forest

is an ensemble learning method where multiple decision trees are constructed and then, they are merged to get a more acurate prediction



4. Boosting:

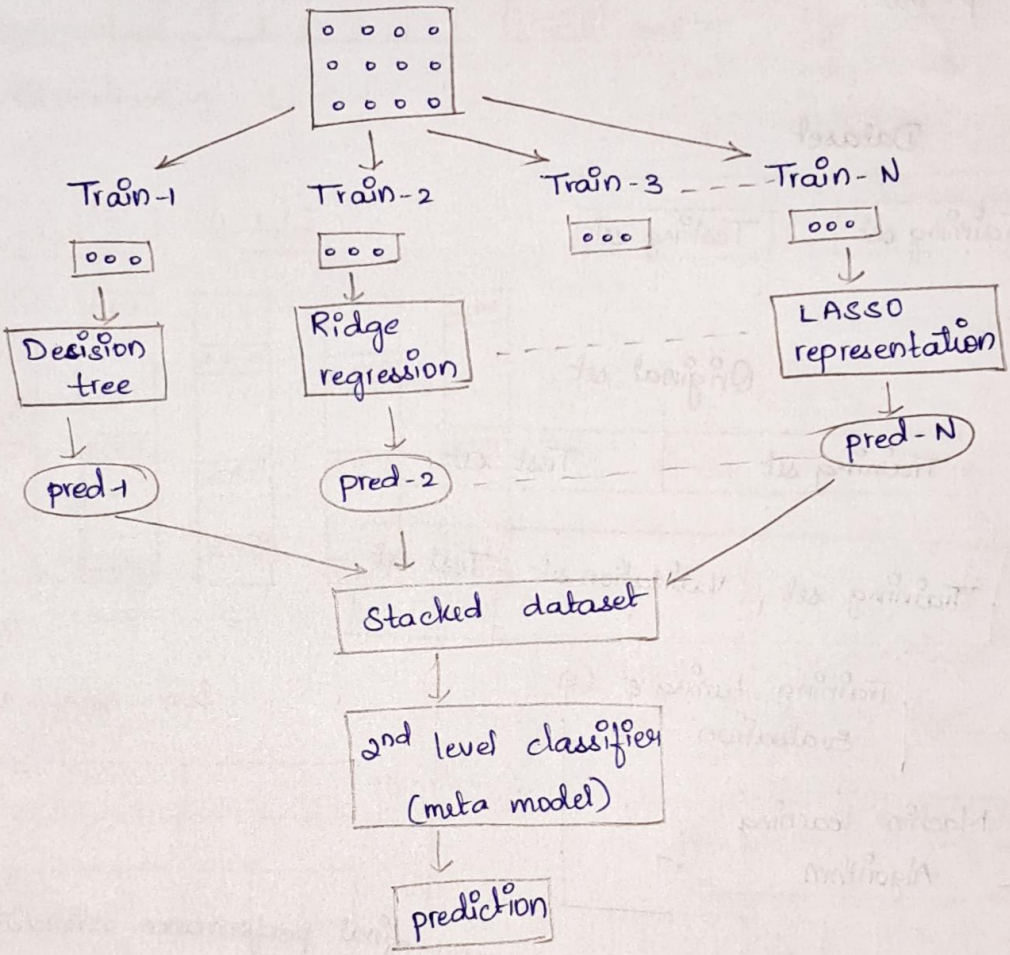
- is an ensemble machine learning technique that attemps to build a <u>strong classifier</u> from the number to weak classifers.
- It is done by building a model by using weak models in series
- firstly, a model is built from the training data.
- Then the second model is built which tries to correct the error present in the first model.
- This procedure is continued and models are added until either the complete training dataset is predicted correctly or the max no. of models are added.



5 Blending: / Stacked Generalization

is an ensemble ML technique

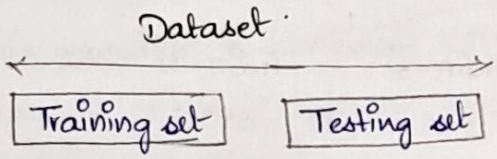
In stacking the combining mechanism is that the output of the classifiers [level-0 classifiers] will be used as training data for another classifier [level-1 classifier] to approximate the same target function.

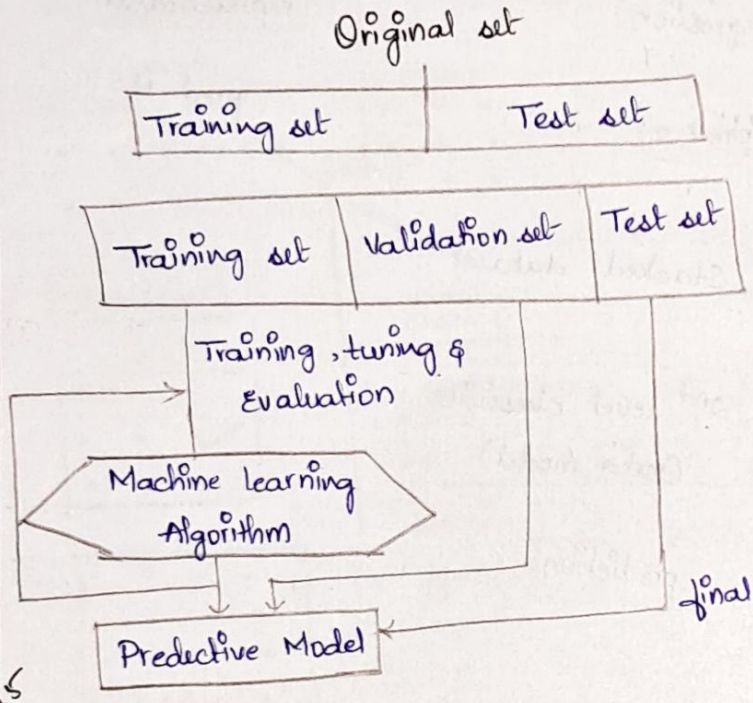


Cross Validation.

1) K-folds
2) Stratified K-folds
3) leave-one-out
4) leave-p-out.

①





final performance estimate.

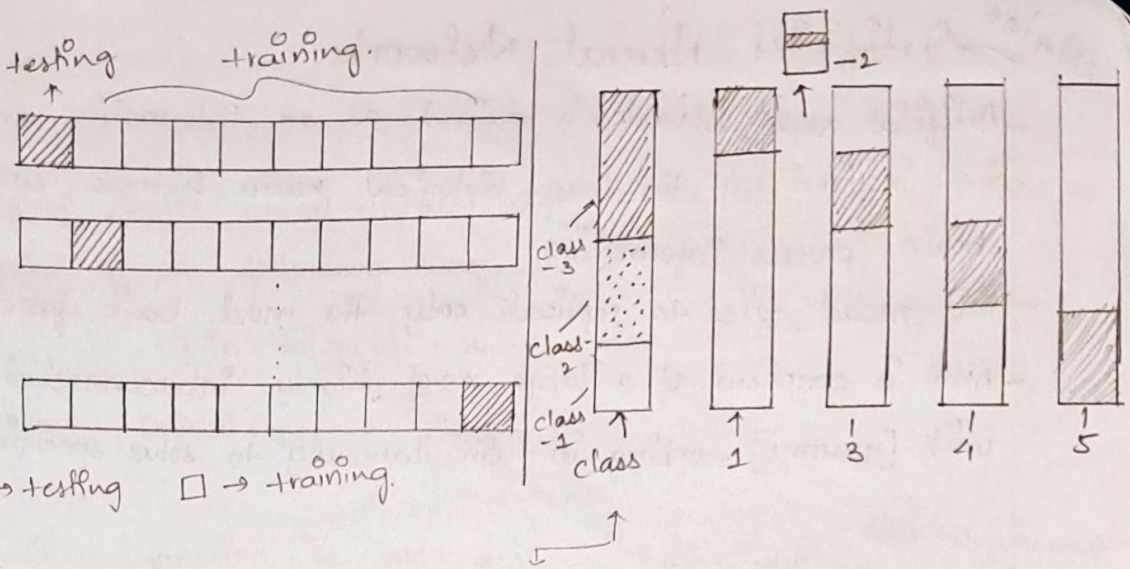
Kahoot apps games.

16/4/25

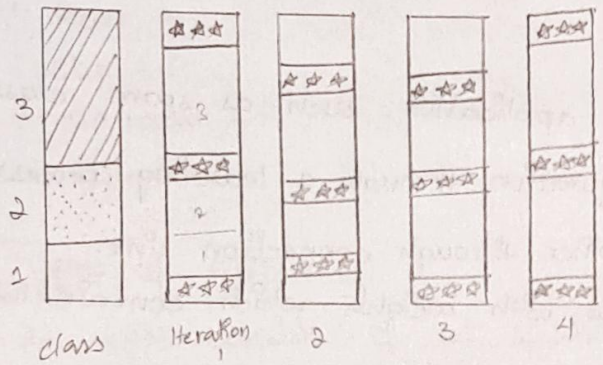
Cross Validation.

The idea of cross validation arises because of the problems with train test model or train test validation model. It basically wants to guarantee that the score of the model does not depend on the way we pick the train & test set.

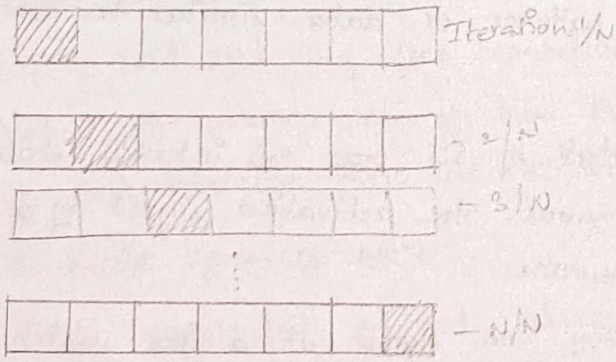
1) **K-folds**



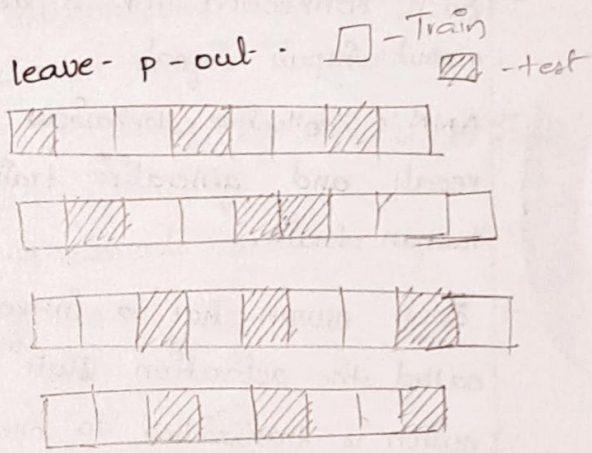
2) **Stratified K-fold**



3) **leave-one-out.**



4) **leave-p-out.**



22/4/25

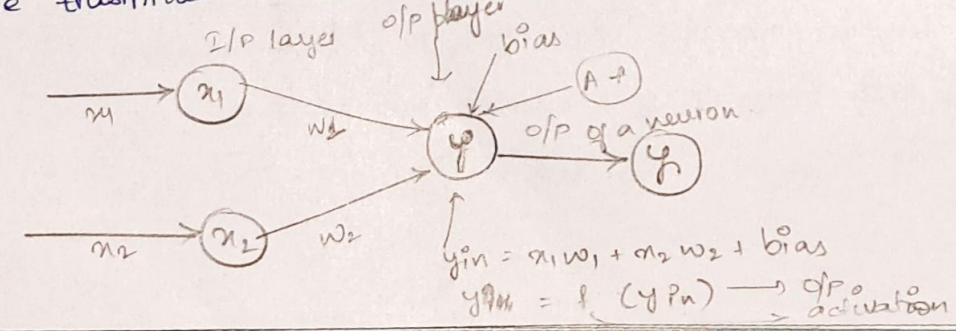
Artificial Neural Networks.

- Artificial neural network is defined as an information processing model i.e., inspired by the way biological neuro systems such as the brain, process information
- This model tries to replicate only the most basic functions of brain
- ANN is composed of a large no. of highly interconnected processing units [neurons] working in simultaneously to solve specific problems.

Neurons
Connection links
weights
Outputs
Activation function.

23/4/25

- ANN is configured for specific application, such as spam classification, face regnization, pattern configuration through a learning process.
- each neuron is connected ~~with~~ by other through connection link.
- Each connection link is associated with weights which contain information about input signal.
- ANN's collective behaviour is charactorized by their ability to learn, recall and generalize training patterns or data. similar to that of human brain.
- Each neuron has a internal state of its own. this internal state is called the activation ~~state~~ level of neuron. the activation signal of a neuron is transmitted to other neurons
- Remember, a neuron can send only one signal at a time, which can be trasmitted to other neurons.



- Input neurons x_1 & x_2 are connected to the output neuron y over a weighted interconnection links (w_1, w_2). for.
- For the above simple neural network architecture, the net input has to be calculated in the following way, where

$$y_{in} = x_1 w_1 + x_2 w_2 + bias.$$ & Activation function.

$$Output(y) = f(y_{in})$$

activation function

The activation function is not fixed one, based on our requirement we are using different activation function.

1. Relu A.F
2. Step A.P
3. sin wave A.F
4. Sigmoid.

Types of ANN's:

- An ANN consists of set of highly interconnected processing elements (neurons) such that each processing element output is found to be connected through weights to the other processing elements or to itself [delay lead and lag-free connections are allowed].
- Hence, the arrangements of these processing elements and the geometry of their interconnections are essential for ANN.
- 5 basic types of ANN.

1) Single layer-fed forward network.
2) Multi layer fed forward network
3) Single node with its own feedback.
4) Single layer recurrent networks.
5) multi layer recurrent networks.

25/4/25

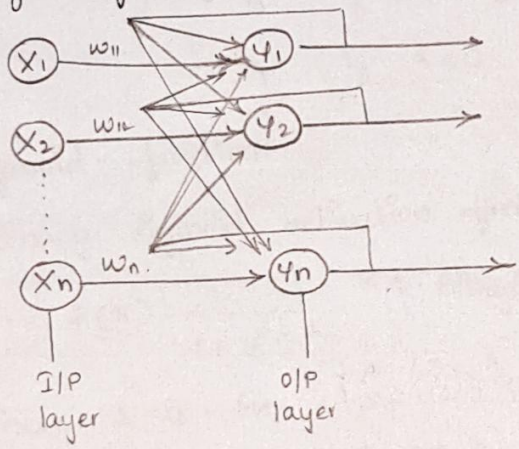
Basic Components of ANN.

1. Connections.
2. Learning.
3. Activation functions.

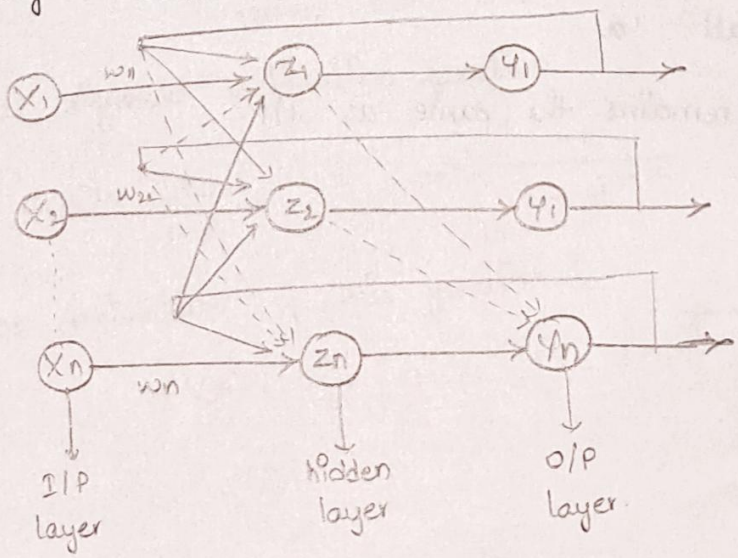
(1) Connections:

1. Single layer fed forward network
2. Multi layer fed forward network
3. Single node with its own feedback.
4. Single layer recurrent neural network
5. Multi layer recurrent neural network.

4. Single layer recurrent neural network.



5. Multi layer recurrent neural network.



② Learning :-

The main property of an ANN is its capability to learn. Learning/training is a process by means of which a neural network adapts itself to a stimulate by making proper parameter adjustments, resulting in the prediction of desired responce.

There are 2 kinds of learning in ANN.

1) Parameter learning:
It updates the connecting weights in a neural network.

2) Structure learning:
It focuses on change in network structure. (which includes the no. of processing elements) as well as their connection types).

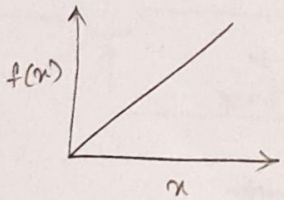
③ Activation function:

1. Identity function.
2. Binary step function.
3. Bipolar step function.
4. Ramp function.
5. Sigmoid function. — Binary sigmoid A.f / Bipolar sigmoid A.f

1. Identity function.

$f(x) = x$ for all 'x'.

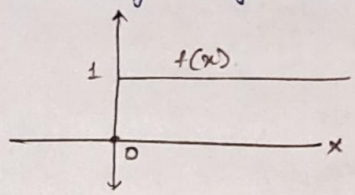
The o/p here remains the same as i/p.



2. Binary step function.

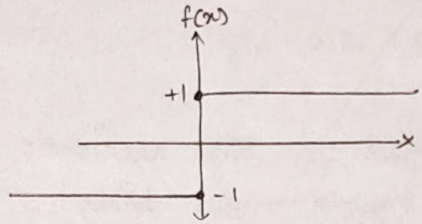
This function is most widely used in single layer network to convert net input into a binary.

$$f(x) = \begin{cases} 1 & \text{if } x \geq 0 \\ 0 & \text{if } x < 0. \end{cases}$$



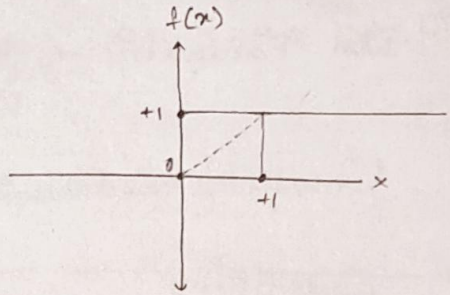
3. Bipolar step function.

$$f(x) = \begin{cases} 1 & \text{if } x \geq 0 \\ -1 & \text{if } x < 0 \end{cases}$$



4. Ramp function.

$$f(x) = \begin{cases} 1 & \text{if } x > 1 \\ x & \text{if } a \leq x \leq 1 \\ 0 & \text{if } x < 0. \end{cases}$$



26/4/25

5. Sigmoid function.

1) Binary sigmoid activation function.

$$f(x) = \frac{1}{1 + e^{-\lambda x}}$$

here λ is the steepness parameter. the ~~div~~ derivation of the function is $f'(x) = \lambda f(x)\,[1 - f(x)]$

range b/w 0 to 1.

2) Bipolar sigmoid activation function

$$f(x) = \frac{2}{1 + e^{-\lambda x}} - 1 \Rightarrow \frac{1 - e^{-\lambda x}}{1 + e^{-\lambda x}}$$

the derivation of this function is

$$f'(x) = \frac{\lambda}{2}\,[1 + f(x)]\,[1 - f(x)].$$

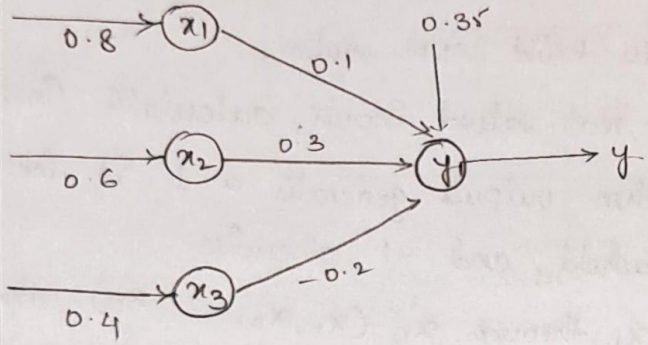
Parametric → high varience
low bias

Non Parametric → low varience
high bias.

Underfitting models → less accuracy in training & test

Overfitting " → better " in training & less
accuracy in testing

Q.



$y_{in} = x_1 w_1 + x_2 w_2 + x_3 w_3$

$y = f(y_{in}) = \frac{1}{1+e^{-y_{in}}}$

y_{in} = Net I/P of the all the neuro.

$y_{in} = 0.35 + 0.8 * 0.1 + 0.6 * 0.3 + 0.4 * (-0.2)$

$= 0.53.$

~~BS = 0.6294~~

$\frac{1}{1+e^{-0.53}} \Rightarrow \frac{1}{1+0.5886} = \frac{1}{1.5886} = 0.6294. \Rightarrow$ Binary sigmoid

$\frac{2}{1+e^{-0.53}} \Rightarrow 0.2539 \Rightarrow$ Bipolar sigmoid.

SSE $\rightarrow \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$

$\hat{y}_i = B_0 + B_1 x_i$

$\sum_{i=1}^{n} [y_i - (B_0 + B_1 x_i)]^2$

$\sum_{i=1}^{n} [y_i^2 + (B_0 + B_1 x_i)^2 - 2y_i (B_0 + B_1 x_i)]$

$\sum_{i=1}^{n} [y_i^2 + B_0^2 + B_1^2 x_i^2 + 2 B_0 B_1 x_i - 2y_i B_0 - 2y_i B_1 x_i]$

$\sum_{i=1}^{n} y_i^2 + \sum_{i=1}^{n} B_0^2 + \sum_{i=1}^{n} B_1^2 x_i^2 + \sum_{i=1}^{n} 2 B_0 B_1 x_i - \sum_{i=1}^{n} 2y_i B_0 - \sum_{i=1}^{n} 2y_i B_1 x_i$

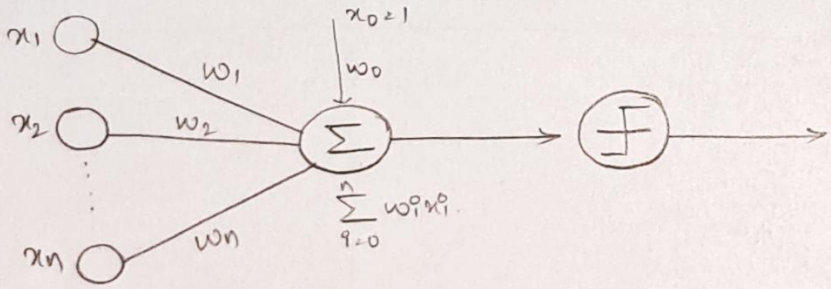
28/4/25

Preception learning Algorithm:

- A perceptron unit is used to build ANN system.
- A perceptron takes a vector of real valued inputs, calculates linear combination of these inputs, then outputs generates a '1' if the result is greater than some threashold, and '-1' otherwise.
- More precisely given inputs x_1 through x_n $(x_1, x_2, \ldots x_n)$, the output is $o(x_1 \times w_1 + x_2 w_2 + \cdots x_n w_n)$ computed by the perceptron is equal to 1 or -1.

$$o(x_1, x_2, \ldots x_n) = \begin{cases} 1 & \text{if } w_0 + w_1 x_1 + w_2 x_2 \ldots w_n x_n > 0 \\ -1 & \text{if otherwise.} \end{cases}$$

where each w_i is a real valued constant or weight. that determines the contribution of input x_i to the perceptron output.



$$o/p = \begin{cases} 1 & \text{if } \sum_{i=0}^{n} w_i x_i > 0 \\ -1 & \text{otherwise.} \end{cases}$$

One way to learn an ~~ap~~ acceptable weight vector is to begin with random weights, then iteratively apply the perceptron to each training example, modifing the perceptron weights whenever it misclassifies an example.

The process is repeated, iterating through the training examples as many times as needed. until the perceptron classifies all training examples correctly.

Weights are modified at each step according to the perceptron training rule, which revises the weight 'w_i' associated with input 'x_i'.

According to the rule.

$$\boxed{\begin{aligned} w_i &\leftarrow w_i + \Delta w_i \\ \text{where } \Delta w_i &= \eta (t - o)\, x_i \end{aligned}}$$

target o/p

i/p associated with weight

actual o/p getting.

learning rate

learning rate (η) is a small value that is set to a small value by default (like 0.1 / point value).

$$\boxed{\Delta w_i = \eta \times \text{error} \times x_i}$$

Q. Implement 'and' gate by using perceptron training rules with the following parameters.

A	B	A∩B
0	0	0
0	1	0
1	0	0
1	1	1

$w_1 = 1.2$

$w_2 = 0.6$

$\eta = 0.5$

Threshold = 1.

1st Sample: $w_1 = 1.2$, $x_1 = 0$. $w_2 = 0.6$, $x_2 = 0$. Target = 0.

$w_i \cdot x_i = x_1 w_1 + x_2 w_2 = 0 * 1.2 + 0 * 0.6 = 0$. O/P = 0.

This is not greater that threashold of 1 ∴ The o/p = 0.

2nd Sample: $w_1 = 1.2$, $x_1 = 0$. $w_2 = 0.6$, $x_2 = 1$. Target = 0.

$\Rightarrow 1.2 * 0 + 0.6 * 1 = 0.6 < 1 \Rightarrow$ O/P = 0.

This is not greater than threashold of 1 ∴ O/P = 0.

3rd Sample: $x_1 = 1$. $x_2 = 0$.

$\Rightarrow 1.2 * 1 + 0.6 * 1 = 1.2 > 1$ (Threashold).

∴ O/P = 1 but target = 0. $\Rightarrow$ Violates property

20/1/24

$$w_1 = w_1 + \Delta w_1$$

$$w_2 = w_2 + \Delta w_2.$$

$$\Delta w_1 = \boxed{0.5 * (1-0) * 1 = -0.5} \qquad \Delta w_2 = \boxed{0.5 * (1-0) * 0 = 0}$$

$$w_1 = 1.2 - 0.5 = 0.7 \qquad w_2 = 0.6 - 0 = 0.6$$

$\therefore$ $w_1 = 0.7$; $w_2 = 0.6$ are the new weights.

<u>1st sample</u>: $x_1 = 0$ $x_2 = 0$ target $= 0$

$w_1 = 0.7$ $w_2 = 0.6$

$w_i x_i = 0.7 * 0 + 0.6 * 0 = 0 < 1$

$\therefore$ O/P $= 0$. ✓

<u>2nd Sample</u>: $x_1 = 0$ $x_2 = 1$ target $= 0$

$w_i x_i = 0.7 * 0 + 0.6 * 1 = 0.6 < 1$

$\therefore$ O/P $= 0$ ✓

<u>3rd Sample</u>: $x_1 = 1$ $x_2 = 0$

$w_i x_i = 0.7 * 1 + 0.6 * 0 = 0.7 < 1$ target $= 0$

$\therefore$ O/P $= 0$ ✓

<u>4th Sample</u>: $x_1 = 1$ $x_2 = 1$ target $= 1$.

$w_i x_i = 0.7 * 1 + 0.6 * 1 = 1.3 > 1$

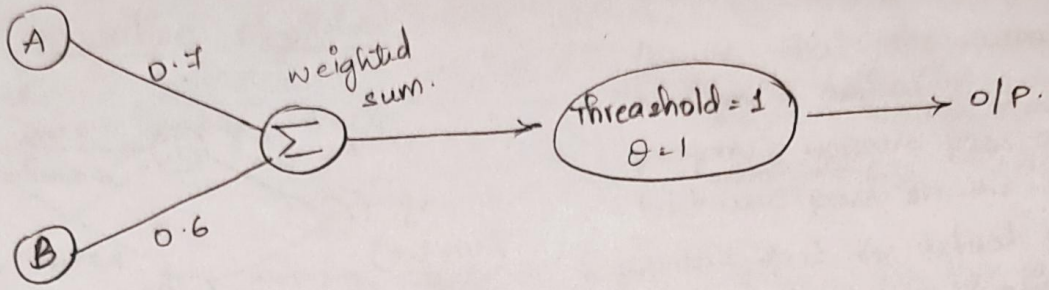
$\therefore$ O/P $= 1$ ✓ & target $= 1$

$\therefore$ Properties accepted.

$\therefore$ weights of the perceptron are

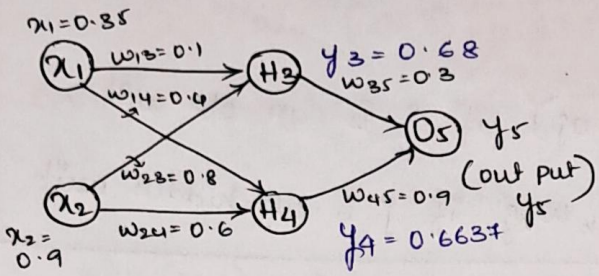
$$\underbrace{w_1 = 0.7 \quad \& \quad w_2 = 0.6}_{\text{Model parameters.}}$$

29/4/25



30/4/25

Back Propagation Algorithm.



Assume that the neurah have a sigmoid activation function. Perform a forward pass (calculate error) and a backward pass (update weights) on the network. assume that the actual o/p of y is 0.5. the activation function & learning rate is '1'. perform another forward pass.

o/p of H3.

$= x_1 * w_{13} + x_2 * w_{23}$

$= (0.35 * 0.1) + (0.9 * 0.8) = 0.755.$

$$o/p = \frac{1}{1+e^{-x}} = \frac{1}{1+e^{-0.755}} = 0.68.$$

$y_3 = 0.68$

$= x_2 * 0.6 + 0.35 * 0.4 = 0.8$

$$y_4 = o/p = \frac{1}{1+e^{-0.68}} \Rightarrow y_4 = 0.6637$$

$O_5 = 0.68 * 0.3 + 0.6637 * 0.9 = 0.801$

$$y_5 = \frac{1}{1+e^{-0.801}} = 0.69 \Rightarrow \text{Network total o/p.}$$

Error $\Rightarrow y_{total} - y_5 = 0.5 - 0.69$

$= -0.19$

1st forward pass completed.

Each weight changed by

$$\Delta w_{ji} = \eta \delta_j o_i .$$

$$\delta_j = o_j (1 - o_j)(t_j - o_j) \text{ if } j \text{ is an o/p unit}$$

$$\delta_j = o_j (1 - o_j) \sum_k \delta_k w_{kj} \text{ if } j \text{ is an hidden unit}$$

t_j - current target output for unit j

δ - error measure for unit j.

$$\delta_5 = y_5 (1 - y_5)(y_{target} - y_5)$$

$$= 0.69 (1 - 0.69)(0.5 - 0.69)$$

$$\delta_5 = -0.0406.$$

$$\delta_3 = y_3 (1 - y_3) w_{35} \times \delta_5$$

$$= 0.68 * (1 - 0.68) * (0.3 * (-0.0406))$$

$$\delta_3 = -0.00265$$

$$\delta_4 = y_4 (1 - y_4)(w_{45} * \delta_5)$$

$$= 0.6637 (1 - 0.6635)(0.9 * (-0.0406))$$

$$\delta_4 = -0.0082.$$

<u>Compute new weights.</u>

$$\Delta w_{ji} = \eta \delta_j o_i$$

$$\Delta w_{45} = \eta \delta_5 \cdot y_4$$

$$= 1 * (-0.0406) * (0.6637)$$

$$\Delta w_{45} = -0.0269$$

$w_{45} = w_{45} + \Delta w_{45} = 0.9 + (-0.0269)$

w_{45} (new) $= 0.8731$

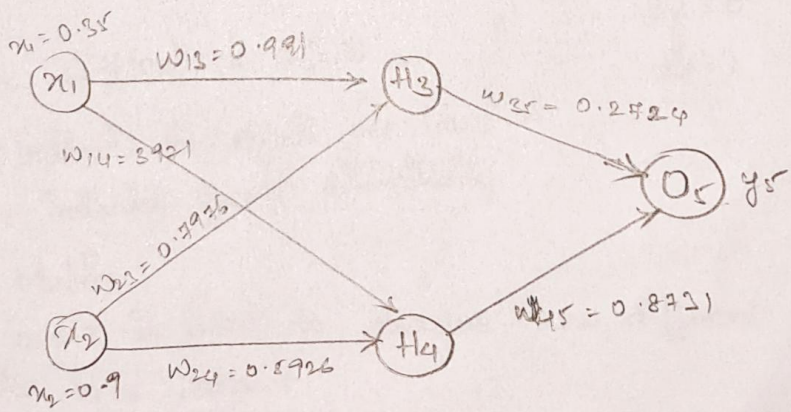
$\Delta w_{14} = \eta \delta_4 x_i = 1 * (-0.0082) * (0.35)$

$\Delta w_{14} = -0.00287$ $\quad$ w_{14} new $= 0.3971$

w_{14} (new) $= 0.4 + (-0.00287) = 0.3971$

i	j	w_{ij}	δ_i	x_i	η	updated weights w_{ij}
4	5	0.9	-0.0406	0.6637	1	0.8731
1	4	0.4	-0.0082	0.35	1	0.3971
3	5					0.2724
1	3					0.0991
2	3					0.7976
2	4					0.5926.

updated graph:



y_5 with new weights

$y_5 = 0.6820$

Error $= 0.5 - 0.6820$

$= -0.182$

3/6/25

UNIT-IV

CLUSTURING.

1. Similarity.
2. Dissimilarity.

Clustering: Groups similar data objects together. This helps in identifying patterns such as customer segments.

Outlier Analysis: Identifies objects that are significally different from the majority of the data, potentially ~~is~~ indicating errors or interesting anomalies.

Nearest Neighbour Classification:

Similarity Measures:

- Quantifies how alike two objects are.
- Value of 0 indicates no similarity.
- Higher values indicate greater similarity, with 1 typically representing complete similarity.

Dissimilarity.

- Quantifies how different 2 objects are.
- Value of 0 indicates the objects are identical.
- Higher value indicates greater dissimilarity.

$dis = 0 \quad x = y$

$dis = 1 \quad x \neq y.$

Dissimilarity Matrix:

A dissimilarity matrix is used to measure how different objects are from one another in a dataset.

- $n \times n$ matrix.
- Stores dissimilarities $d(i,j)$ between objects i and j.

Properties:

- $d(i,j)$ is non-negative and close to 0 when objects are similar.
- $d(i,j) = 0$.
- $d(i,j) = d(j,i)$ [symmetric matrix]. elements below diagonal = elements above diagonal

Calculation of Dissimilarity for Nominal Attributes.

Nominal attributes

Step-1: Compare Attributes.

$$d(i,j) = \frac{\text{No. of mismatches}}{\text{Total attributes}}$$

Step-2: Pair wise dissimilarity.

A vs B

Attribute	A	B	match?
Gender	M	F	X
City	Delhi	Mumbai	X
color	red	blue.	X

Mismatches = 3 total attributes = 3 $\Rightarrow 3/3 = 1$

Person	Gender	City	favorite Color.
A	M	Delhi	Red
B	F	Mumbai	Blue
C.	M	Mumbai	Red.

$A|B = 1$

$B|C = 0.667$

$C|A = 0.33$.

$$\begin{array}{c|ccc} & A & B & C \\ \hline A & 0 & 1.00 & 0.33 \\ B & 1.00 & 0 & 0.67 \\ C & 0.33 & 0.67 & 0 \end{array}$$

Ordinal Attributes.

Person	Satisfaction	Education level.
A	Neutral	graduate
B	Very satisfied	post graduate
C.	dissatisfied.	secondary.

Step-1: Assign Ranks.

{Very dissatisfied = 1 ; dissatisfied = 2 ; Neutral = 3 ; satisfied = 4 ; v. satisfied = 5}

{Primary = 1 ; Secondary = 2 ; graduate = 3 ; post graduate = 4}.

	Satisfaction	Education.
A	3	3
B	5	4
C	2	2.

Step-2: Normalize the ranks.

z = rank-1 / max rank-1

Person	satisfaction	Education.
A	(3-1)/4 = 0.5	(3-1)/3 = 0.67
B	(5-1)/4 = 1.0	(4-1)/3 = 1.00
C.	(2-1)/4 = 0.25	(2-1)/3 = 0.33

Step-3: Calculate dissimilarity. [e.g., Manhattan distance].

$$d(i,j) = |z_1(i) - z_1(j)| + |z_2(i) - z_2(j)|$$

A vs B: $|0.5 - 1.0| + |0.67 - 1.0| = 0.5 + 0.33 = 0.83$

A vs C: $|0.5 - 0.25| + |0.67 - 0.33| = 0.25 + 0.34 = 0.59$

B vs C: $|1.0 - 0.25| + |1.0 - 0.33| = 0.75 + 0.67 = 1.42$.

	A	B	C
A	0	0.83	0.59
B	0.83	0	1.42
C.	0.59	1.42	0

4/6/25

Symmetric Binary Attributes.

Both values 0 & 1 are equally important.
Use simple matching coefficient:

$d(i,j) = f_{01} + f_{10} / f_{00} + f_{11} + f_{01} + f_{10}$.

Object	Snacks	Drinks
A	1	0
B	1	1
C	0	1

A vs C : (1, 0) vs (0, 1)

$d(A,C) = 2/2 = 1$

B vs C : (1, 1) vs (0, 1)

$d(B,C) = 1/2 = 0.5$

A vs B : 1 mismatch
1 match.

$d(A,B) = 1/2 = 0.5$

$$\begin{array}{c|ccc} & A & B & C \\ \hline A & 0 & 0.5 & 1 \\ B & 0.5 & 0 & 0.5 \\ C & 1 & 0.5 & 0 \end{array}$$

Attribute Type	Method	formula / Rule.
Nominal	matching	mismatches / total.
Symmetric Binary	matching.	$(f_{10} + f_{01})$ / total.
Asymmetric Binary	Jaccard coefficient	$\frac{(f_{10} + f_{01})}{(f_{11} + f_{10} + f_{01})}$

Example Data for Numeric Attributes.

Object	height	weight
A	160	55
B	165	65
C	170	60.

1. Euclidean distance [Dissimilarity].

$d(A,B) = \sqrt{(160-165)^2 + (55-65)^2} = \sqrt{25+100} = \sqrt{125} \approx 11.18$

$d(A,C) = \sqrt{(160-170)^2 + (55-60)^2} = \sqrt{100+25} = \sqrt{125} \approx 11.18$

$d(B,C) = \sqrt{(165-170)^2 + (65-60)^2} = \sqrt{25+25} = \sqrt{50} \approx 7.07$

Dissimilarity Matrix. [Euclidean].

$$\begin{array}{c|ccc} & A & B & C \\ \hline A & 0 & 11.18 & 11.18 \\ B & 11.18 & 0 & 7.07 \\ C & 11.18 & 7.07 & 0 \end{array}$$

---

1. K-means clustering Algorithm.

Q. Suppose the datapoints are clustered into 3 types [3 clusters].

* $A_1(2, 10)$; $A_2(2, 5)$; $A_3(8, 4)$.
* $B_1(5, 8)$; $B_2(7, 5)$; $B_3(6, 4)$.

$C_1(1, 2)$; $C_2(4, 9)$,

Initially I am assuming A_1, B_1, C_1 as the center of each cluster successfully.

			C1 (2,10)	C2 (5,8)	C3 (1,2)	Cluster.
A_1	2	10	0	3.61	8.06	1
A_2	2	5	5	4.24	3.16	3
A_3	8	4	8.44	5.00	7.28	2
B_1	5	8	3.61	0.00	7.21	2
B_2	7	5	7.07	3.61	6.71	2
B_3	6	4	7.21	4.12	5.39	2
C_1	1	2	8.06	7.21	0.00	3
C_2	4	9	2.24	1.41	7.62	2

(2,10) distance (2,10)

(2,5) distance (2,10)

⋮

nearest to cluster. [Minimum value].

finding new centers.

C_1 (2,10)

C_2 (6,6) → Addition of points in that cluster / total points belonging.

C_3 (1.5, 3.5)

6/6/25

			New Cl_1 (2,10)	Cl_2 (6,6)	Cl_3 (1.5,3.5)	Old Cluster	New cluster.
A_1	2	10	0	5.6	6.51	1	1
A_2	2	5	5	4.12	1.58	3	3
A_3	8	4	8.48	2.8	6.51	2	2
B_1	5	8	3.6	2.23	5.70	2	2
B_2	7	5	7.07	1.41	5.70	2	2
B_3	6	4	7.21	2	4.52	2	2
C_1	1	2	8.06	6.40	1.58	3	3
C_2	4	9	2.23	3.6	6.04	2	1

Cl_1 (3, 9.5)

Cl_2 (6.5, 5.25)

Cl_3 (1.5, 3.5).

			New Cl_1 (3, 9.5)	New Cl_2 (6.5, 5.25)	New Cl_3 (1.5, 3.5)	old Cluster	New cluster.
A_1	2	10	1.12	6.54	6.52	1	1
A_2	2	5	4.6	4.50	1.58	3	3
A_3	8	4	7.43	1.95	6.52	2	2
B_1	5	8	2.5	3.13	5.70	2	1
B_2	7	5	6.02	0.56	5.70	2	2
B_3	6	4	6.26	1.35	4.53	2	2
C_1	1	2	7.76	6.39	1.58	3	3
C_2	4	9	1.12	4.51	6.04	1	1

Cl_1 (3.66, 9)

Cl_2 (7, 4.33)

Cl_3 (1.5, 3.5)

			New Cl1 (3.66, 9)	New Cl2 (7, 4.33)	New Cl3 (1.5, 3.5)	Old Cluster	New Cluster.
A_1	2	10	1.93	7.56	6.52	1	1
A_2	2	5	4.33	5.04	1.58	3	3
A_3	8	4	6.62	1.05	6.52	2	2
B_1	5	8	1.67	4.18	5.70	1	1
B_2	7	5	5.21	0.67	5.70	2	2
B_3	6	4	5.52	1.05	4.53	2	2
C_1	1	2	7.49	6.44	1.58	3	3
C_2.	4	9	0.33	5.55.	6.04	1	1

Cluster - 1

(2,10) (5,8) (4,9)

Cluster - 2

(8,4) (7,5) (6,4)

Cluster - 3

(2,5) (1,2).

Q. Use K-means to cluster the following data into 2 groups.

{2, 4, 10, 12, 3, 20, 30, 11, 25}.

- Initial cluster centroid are $M_1 = 4$ & $M_2 = 11$.

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

Initial centroids $M_1 = 4$, $M_2 = 11$.

Data points	Distance to M1	M2	Cluster	~~New Cluster~~.
2	2	9	1	
4	0	7	1	
10	6	1	2	
12	8	1	2	
3	1	8	1	
20	16	9	2	
30	26	19	2	
11	7	0	2	
25	21	14	2.	

M1 : 4 $\quad C_1 = \{2, 4, 3\}$

M2 : 11 $\quad C_2 = \{10, 12, 20, 30, 11, 25\}$

New cluster centroids :

M1 : 3

M2 : 18

DataPoints	Distance to M1 : 3	Distance to M2 : 18	Cluster	New cluster.
2	1	16	1	1
4	1	14	1	1
10	7	8	2	1
12	9	6	2	2
3	0	15	1	1
20	17	2	2	2
30	27	12	2	2
11	8	7	2	2
25	22	7	2	2

M1 : 3 $\quad C_1 = \{2, 4, 10, 3\}$

M2 : 18 $\quad C_2 = \{12, 20, 30, 11, 25\}$

New centroids

M1 : 4.75

M2 : 19.6

Current centroids :

M1 : 4.75

M2 : 19.6

DataPoints	Distance to M1 : 4.75	Distance to M2 : 19.6	Cluster	New Cluster
2	2.75	17.6	1	1
4	0.75	15.6	1	1
10	5.25	9.6	1	1
12	7.25	7.6	2	1
3	1.75	16.6	1	1
20	15.25	0.4	2	2
30	25.25	10.4	2	2
11	6.25	8.6	2	1
25	20.25	5.4	2	2

Unsupervised learning → unlabeled data.

C1 = {2, 4, 10, 11, 12, 3}.

C2 = {20, 30, 25}.

New centroids:

M1 : 7

M2 : 25

Data Points	Distance M1 : 7	Distance M2 : 25	Cluster	New Cluster
2	5	23	1	1
4	3	21	1	1
10	3	15	1	1
12	5	13	1	1
3	4	22	1	1
20	13	5	2	2
30	23	5	2	2
11	4	14	1	1
25	18	0	2	2.

final cluster

C1 = {2, 4, 10, 11, 12, 3}

C2 = {20, 30, 25}.

Hiearchical Clustering / Agglomerative clustering :

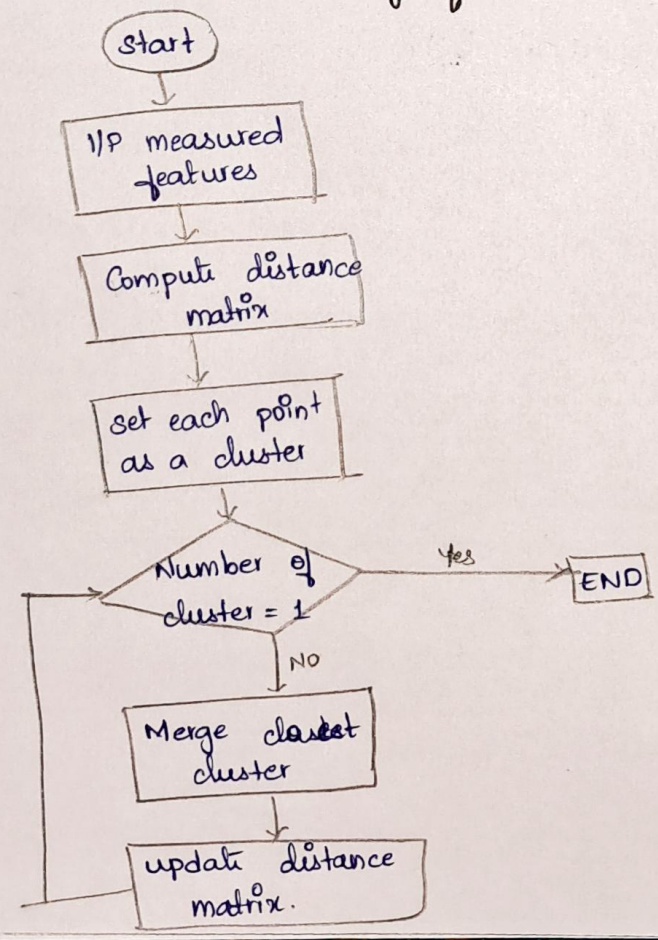
① Agglomerative Hierarchical clustering

- Minimum Distance [single linkage Technique]
- Maximum Distance [Complete linkage Techniques]
- Average Distance [Average linkage Techniques]
- Centroid linkage and ward's method

② Divisive Hierachical clustering.

- Agglomerative Hierarchical clustering follows the bottom-up approach.
- Initially each data point is considered as a singleton cluster and then succesively we merge the data points which are close to each other.
- The process is repeated untill all clusters have been merged into a single cluster that contains all data.
- This clustering algorithm does not require us to prespecify the no. of clusters.

Agglomerative Hierarchical clustering flowchart.



Implementation of Agglomerative Hierarchical clustering using single linkage clustering.

	a	b	c	d	e.
a	0	9	3	6	11
b	9	0	7	5	10
c	3	7	0	9	2
d	6	5	9	0	8
e	11	10	2	8	0

- The single linkage clustering uses the "minimum distance formula".

⇒

	a	b	(c,e)	d
a	0	9	?	6
b	9	0	?	5
(c,e)	?	?	0	?
d	6	5	?	0

min ⇒ 2
club (c,e)

Cl_2 : {a}, {b} {c,e} {d}.

- $d(\{c,e\},\{a\}) = \min\{d(c,a), d(e,a)\} = \min\{3,11\} = 3$
- $d(\{c,e\},\{b\}) = \min\{d(c,b), d(e,b)\} = \min\{7,10\} = 7$
- $d(\{c,e\},\{d\}) = \min\{d(c,d), d(e,d)\} = \min\{9,8\} = 8$

⇒

	a	b	(c,e)	d
a	0	9	3	6
b	9	0	7	5
(c,e)	3	7	0	8
d	6	5	8	0

min ⇒ 3
club (a,c,e)

⇒ Cl_3 = {a,c,e}, {b}, {d}

	(a,c,e)	b	d
(a,c,e)	0	?	?
b			
d			

$d(\{a,c,e\},\{b\}) = \min\{d(a,b), d(c,d), d(e,b)\}$

$- d(\{a,c,e\},\{b\}) = \{9,7,10\} = 7$

$- d(\{a,c,e\},\{d\}) = \{d(a,d)\ d(c,d)\ d(e,d)\} = \{6,9,8\} = 6.$

	(a,c,e)	b	d
(a,c,e)	0	7	6
b	7	0	5
d	e	5	0

$C_4 = \{a,c,e\}\ \{b,d\}$

	(a,c,e)	(b,d)
(a,c,e)	0	?
(b,d)	?	0

Min = 5

club (b,d).

$d(\{a,c,e\},\{b,d\})$

	(a,c,e)	(b,d)
(a,c,e)	0	6
(b,d)	6	0

Implementation of agglomerative hierarchical clustering using complete linkage.

	a	b	c	d	e
a	0	9	3	6	11
b	9	0	7	5	10
c	3	7	0	9	2
d	6	5	9	0	8
e	11	10	2	8	0

10/6/25 Naive Bayes Classifier.

→ import pandas as pd
from sklearn.preprocessing import LabelEncoder
from sklearn.naive_bayes import CategoricalNB
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, classification_report.

BBN

→ # load tennis.csv dataset
data = pd.read_csv ('tennis.csv')
print (data.columns)
print (data.head())
Index (['outlook', 'temp' .....)
// dataset [1-4].

BB
10/06/2025.

→ data.keys()
Index (['outlook', 'temp', ...)

p(play=0)

→ le = LabelEncoder()
data_encoded = data.apply
data.head()

	outlook	temp	humidity	windy	play
0	2	1	0	0	0
1	2	1	0	1	0
2	0	1	0	0	1
3	1	2	0	0	1
4	1	0	1	0	1

$$p(C_i|x) = \frac{p(x/C_i)\, p(C_i)}{p(x)}$$

prior

→ data.info()
- - - - - - -

→ x = data_encoded.drop('play', axis=1)
y = data_encoded ['play']
print (x.shape)
print (y.shape)
(14,4)
(14,).

x = (outlook = 2, temp = 1)

p((C0|x)

p(Ci/x)

→ x_train, x_test, y_train, y_test = train_test_split (x, y, test_size=0.2, random_state=42)

→ model = CategoricalNB()
model.fit (x_train, y_train)
CategoricalNB.

→ y_pred = model.predict (x_test)
print ("Accuracy :", accuracy_score (y_test, y_pred)).

Accuracy : 0.3333....

→ print ("classification Report: \n", classification_report (y_test, y_pred)).

→ model = CategoricalNB()
Scores = cross_val_score (model, x, y, cv=5)

→ model.fit (x, y)
y_pred = model.predict(x)

→ from sklearn.model_selection import cross_val_score.
scores = cross_val_score (model, x, y, cv=5)
print ("Cross-validation scores : ", scores)
print ("Average Accuracy :", scores.mean())

Cross-validation scores : [0.666...7 1.. 0.66...7 0. 0.5]
Average Accuracy : 0.5666...7

→ import matplotlib.pyplot as plt
plt.plot (range (1,6), scores, marker='o', linestyle='-', color='blue')
plt.title ('Naive Bayes 5-fold Cross Validation Accuracy')
plt.xlabel ('fold')
plt.ylabel ('Accuracy')
plt.ylim (0,1)
plt.grid (True)
plt.show()

→ model.fit(X, y)

```
print("features mappings for 'outlook': ", dict(zip(data['outlook'],
                                                    data_encoded['outlook'])))
new_sample = pd.DataFrame([[2,0,1,0]], columns = ['outlook', 'temp',
                                                  'humidity', 'windy'])
prediction = model.predict(new_sample)
print("Predicted class (encoded): ", prediction)
print("Predicted class (label): ", data['play'].unique()[prediction[0]])
```