Unit - II

Supervised Learning

→ Supervised Learning is the types of ML in which machines are trained using 'labelled' training data, and basis of that data m/c predict the output.

→ Labelled data means some i/p data is already tagged with the correct output.

→ In SL, the training data provided to the machines work as the supervisor that teaches the machines to predict the output correctly.

→ SL is a process of providing i/p data and output data to the machine learning model.

→ The aim of SL algorithm is to find a mapping function to map the i/p variable (x) with the o/p variable (Y)

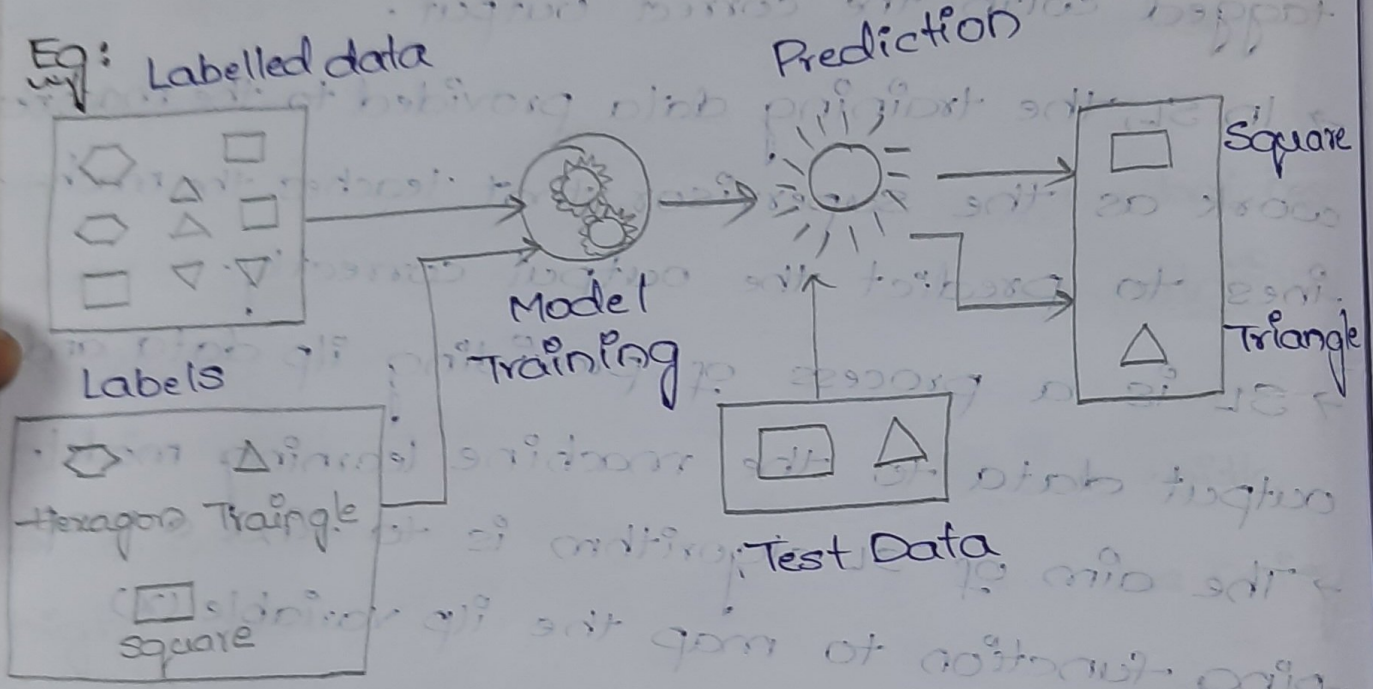
→ In real world SL is used for Risk Assessment, Image classification, Fraud Detection, Spam filtering etc.,

How Supervised learning works?

→ In SL models are trained using labelled data set, where models learns about each type of data.

→ once, training process is completed the model is tested on the basis of test data & then it predicts the output.

Eg: Labelled data



→ If the given shape has 4 sides, & all are equal, then it will be labelled as a square.

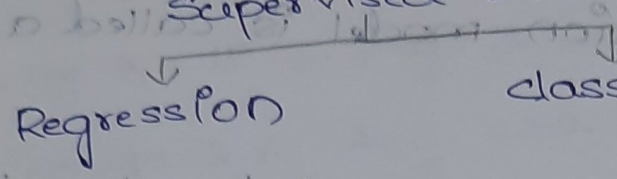
→ If the given shape has 3 sides, then it will be labelled as a Triangle.

→ If the given shape has six equal sides then it will be labelled as Hexagon.

→ Now, after training, we test our model using test set

→ The MLC is already trained on all types of shapes, when it finds new shape, it classifies the shape on the basis of no. of sides and predicts the output.

Types of Supervised Learning Algorithms:



Regression Algorithms:

→ These are used if there is a relationship b/w the i/p & o/p variable.

→ It is used for the prediction of continuous variables such as Weather forecasting, Market Trends etc.,

→ Some of the popular Regression Alg are

a) Linear Regression
b) Non-linear Regression
c) Regression Trees etc.,

i) Simple Linear Regression ✓
ii) Multiple Regression ✓
iii) Polynomial Regression
iv) Logistic Regression ✓

Simple Linear Regression:

→ Assume that there is only one independent variable x, If the relationship blw x & y (dependent or dep variable) is modeled by the relation, $y = a + bx$

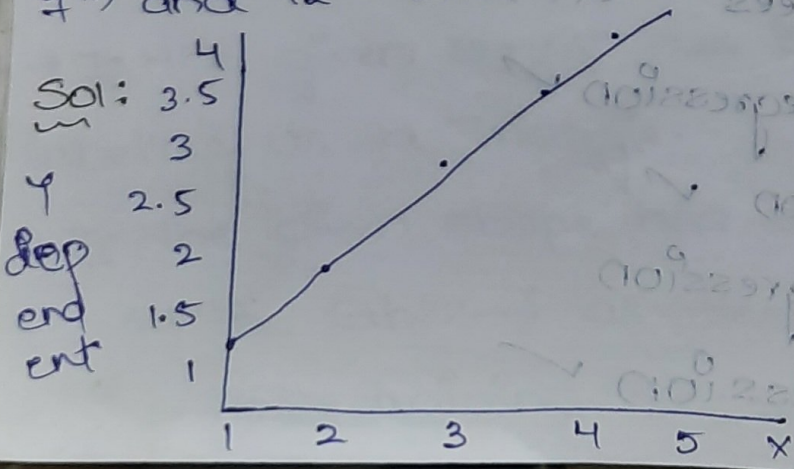
then the regression model is called a Linear Regression Model.

Ex: Let us consider an example where the five week's sales data (in thousands) is shown

x_i (Week)	y_i (Sales in thousands)
1	1.2
2	1.8
3	2.6
4	3.2
5	3.8

Apply Linear Regression technique to predict the 7th and 12th week sales.

Sol:



$y = a_0 + a_1 * x + e$ (e = error)

a_0 = intercept

a_1 = coefficient of independent variable (x)

where

$$a_1 = \frac{(\overline{xy}) - (\overline{x})(\overline{y})}{\overline{x^2} - \overline{x}^2}$$

$$a_0 = \overline{y} - a_1 * \overline{x}$$

Sum of $x = 15$, Avg $= 3$

Sum of $y = 12.6$, Avg $= 2.52$

	x_i	y_i	x_i^2	$x_i * y_i$
	1	1.2	1	1.2
	2	1.8	4	3.6
	3	2.6	9	7.8
	4	3.2	16	12.8
	5	3.8	25	19
Sum	15	12.6	55/5	44.4/5
Average	3	2.52	$\overline{x_2} = 11$	$\overline{xy} = 8.88$

$\overline{x} = 3$, $\overline{y} = 2.52$, $\overline{x^2} = 11$, $\overline{xy} = 8.88$

$$a_1 = \frac{(\overline{xy}) - (\overline{x})(\overline{y})}{\overline{x^2} - \overline{x}^2} = \frac{(8.88) - (3 * (2.52))}{11 - 3^2}$$

$$= 0.66$$

$$a_0 = \bar{y} - a_1 * \bar{x}$$

$$= 2.52 - 0.66 * 3$$

$$= 0.54$$

Linear Regression equation is

$$y = a_0 + a_1 * x$$

$$y = 0.54 + 0.66 * x$$

The predicted 7th week Sale (when $x = 7$) is,

$$y = 0.54 + 0.66 * 7$$

$$= 5.16$$

The predicted 12th week Sale (when $x = 12$) is,

$$y = 0.54 + 0.66 * 12$$

$$= 8.46$$

Multiple linear Regression:

→ In linear regression model we have one dependent and one independent variable.

→ Multiple regression model involves multiple predictors or independent variables and one dependent variable.

→ this is an extension of the linear regression problem.

→ The multiple regression of two variables x_1 and x_2 is given as follows:

$$y = f(x_1, x_2)$$

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

a_0, a_1, a_2 = coefficients of multiple linear Regression

→ In general, this is given for 'n' independent variables as:

$$y = f(x_1, x_2 \cdots x_n)$$

$$y = a_0 + a_1 x_1 + a_2 x_2 + \cdots + a_n x_n + \epsilon$$

→ here $x_1, x_2 \cdots x_n$ are predictor variables, y is the dependent variable, $(a_0, a_1, a_2 \cdots a_n)$ are the co-efficients of the regression equation and ϵ is the error term.

Ex:

x_1 (independent variable) product1 Sales	x_2 product2 sales	Y (dependent variable) Weekly sales
1	4	1
2	5	6
3	8	8
4	2	12

→ Here the matrices for Y and x are given as follows: Bias

$$x = \begin{Bmatrix} 1 & 1 & 4 \\ 1 & 2 & 5 \\ 1 & 3 & 8 \\ 1 & 4 & 2 \end{Bmatrix} \text{ and } Y = \begin{Bmatrix} 1 \\ 6 \\ 8 \\ 12 \end{Bmatrix}$$

→ The coefficient of the multiple regression equation is given as

$$a = \begin{Bmatrix} a_0 \\ a_1 \\ a_2 \end{Bmatrix} \begin{matrix} \leftarrow \text{intercept} \\ \leftarrow \text{independent}(x_1) \\ \leftarrow \text{independent}(x_2) \end{matrix}$$

→ The regression coefficient for multiple regression is calculated the same way as linear regression:

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

$$x^T x = \begin{Bmatrix} 1 & 1 & 1 & 1 \\ 1 & 2 & 3 & 4 \\ 4 & 5 & 8 & 2 \end{Bmatrix}_{3\times 4} \begin{Bmatrix} 1 & 4 \\ 1 & 5 \\ 1 & 8 \\ 1 & 2 \end{Bmatrix}_{4\times 3}$$

$$= \begin{Bmatrix} 4 & 10 & 19 \\ 10 & 30 & 46 \\ 19 & 46 & 109 \end{Bmatrix}_{3\times 3}$$

$$\begin{bmatrix} 1*1+1*1+1*1+1*1 & 1\times 1+1*2*1*3+4\times 4 & 1*4+1*5+1*8+1* \\ 1*1+2*1+3*1+4*1 & 1*1+2*1+3*3+4\times 4 & 1\times 4+2\times 5+3\times 8+ \\ 4\times 1+5\times 1+8\times 1+2\times 1 & 4\times 1+5\times 2+8\times 3+2\times 4 & 4\times 4+5\times 5+8\times 8+ \end{bmatrix}$$

$$(x^T x)^{-1} = \begin{Bmatrix} 4 & 10 & 19 \\ 10 & 30 & 46 \\ 19 & 46 & 109 \end{Bmatrix}^{-1} =$$

$$= \begin{Bmatrix} 3.15 & -0.59 & -0.30 \\ -0.59 & 0.20 & 0.016 \\ -0.30 & 0.016 & 0.054 \end{Bmatrix}$$

$$(X^T X)^{-1} X^T = \begin{Bmatrix} 3.15 & -0.59 & -0.30 \\ -0.59 & 0.20 & 0.016 \\ -0.30 & 0.016 & 0.054 \end{Bmatrix} \times \begin{Bmatrix} 1 & 1 & 1 & 1 \\ 1 & 2 & 3 & 4 \\ 4 & 5 & 8 & 2 \end{Bmatrix}$$

$$= \begin{Bmatrix} 1.36 & 0.47 & -1.02 & 0.19 \\ -0.32 & -0.098 & 0.155 & 0.26 \\ -0.065 & 0.005 & 0.185 & -0.125 \end{Bmatrix}$$

→ $\hat{a} = ((X^T X)^{-1} X^T) Y$

$$= \begin{Bmatrix} 1.36 & 0.47 & -1.02 & 0.19 \\ -0.32 & -0.098 & 0.155 & 0.26 \\ -0.065 & 0.005 & 0.185 & -0.125 \end{Bmatrix} \times \begin{Bmatrix} 1 \\ 6 \\ 8 \\ 12 \end{Bmatrix}$$

$$= \begin{Bmatrix} -1.69 \\ 3.48 \\ -0.05 \end{Bmatrix} \begin{matrix} a_0 \\ a_1 \\ a_2 \end{matrix}$$

→ Multiple linear regression equation

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

$$y = -1.69 + 3.48 x_1 - 0.05 x_2$$

Logistic Regression:

→ Linear regression predicts the numerical response but is not suitable for predicting the categorical variables.

→ When categorical variables are involved, it is called classification problem.

→ Logistic regression is suitable for binary classification problem.

→ For example, the following scenarios are instances of predicting categorical variables.

1. Is the mail spam or not spam?
The answer is Yes or no.
Thus, categorical dependent variable is a binary response of Yes or No.

2. If the student should be admitted or not is based on entrance exam marks.
Here, categorical variable response is admitted or not.

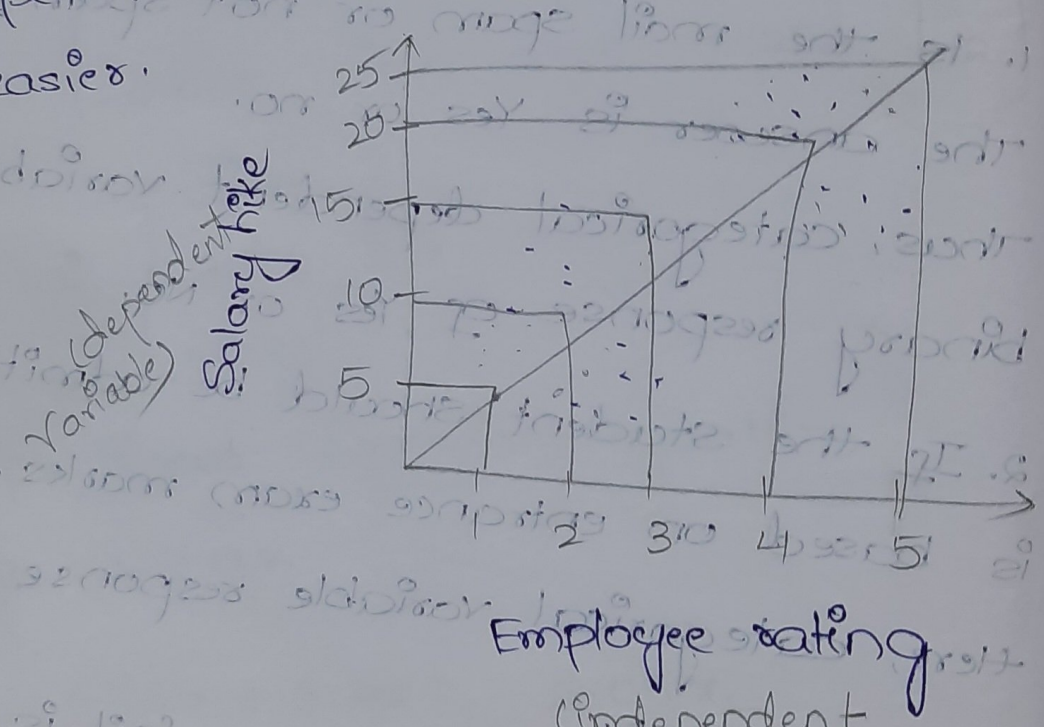
3. The student being pass or fail is based on marks secured.

How does the Logistic Regression Algorithm work?

Ex: An organization wants to determine an employee's salary increase based on their performance.

→ For this purpose, a linear regression alg will help them decide.

→ plotting a regression line by considering the employee's performance as the independent variable, and the salary increase as the dependent variable will make their task easier.

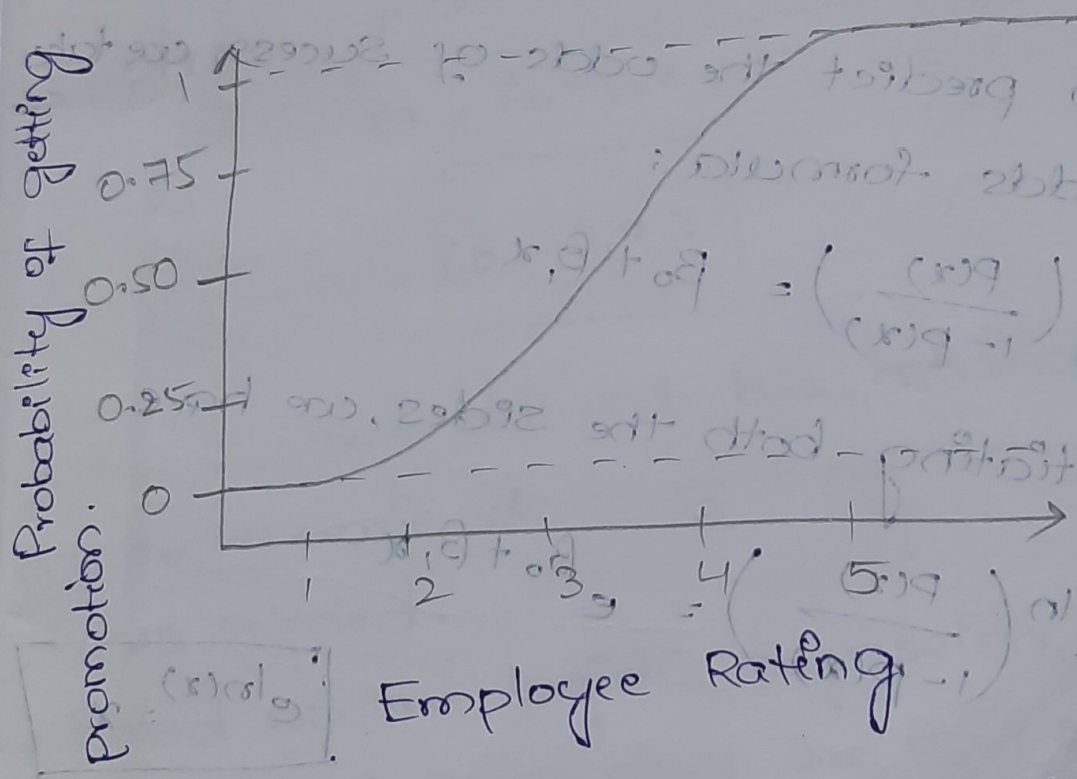


→ Now, what if the organization wants to know whether an employee would get a promotion or not based on their performance?

→ The above linear graph won't be suitable in this case.

→ As such, we clip the line at zero and one, and convert it into a sigmoid curve (s curve)

→ Based on the threshold values, the organization can decide whether an employee will get a salary increase or not.



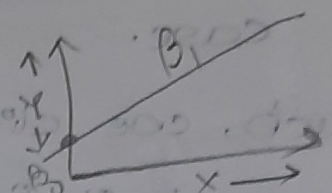
→ The values b/w '0' & '1' are called 'threshold values'.

if th=0.5

To understand logistic regression, let's go over the odds of success.

$$\text{odds}(\Theta) = \frac{\text{Probability of an event happening}}{\text{probability of an event not happening}}$$

$$\text{odds}(\Theta) = \frac{p}{1-p}$$



→ The values of odds range from zero to ∞ and the values of probability lies b/w 0 & 1.

→ consider the equation of a straight line:

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

→ Now to predict the odds of success, we take log on odds formula:

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

→ Exponentiating both the sides, we have

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

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

$e^{\ln(x)} = x$

↓

Formula.

Let $Y = e^{\beta_0 + \beta_1 x}$

then $\frac{P(x)}{1-P(x)} = Y$

$\rightarrow P(x) = Y(1-P(x))$

$\rightarrow P(x) = Y - Y(P(x))$

$\rightarrow P(x) + Y(P(x)) = Y$

$\rightarrow P(x)(1+Y) = Y$

$\rightarrow P(x) = \frac{Y}{1+Y}$

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

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

Ex:

→ The student dataset has entrance mark based on the historic data of those who are selected or not selected.

→ Based on the logistic regression, the values of the learnt parameters are $\beta_0 = -1$ & $\beta_1 = 8$

→ Assuming marks of $x = 60$, compute the resultant class.

$$p(x) = \frac{1}{1+e^{-(\beta_0+\beta_1 x)}} \qquad (\beta_0+\beta_1 x) = (1+8\times 60) = 48)$$

$$= \frac{1}{1+e^{-481}} = 0.44$$

→ If we assume the threshold value as 0.5, then it is observed that $0.44 < 0.5$, therefore the candidate with marks 60 is not selected.

Ex:

Hours study	pass (1) / fail (0)
29	0
15	0
33	1
28	1
39	1

The dataset of pass or fail in an exam of 5 students is given in the table.

use Logistic Regression as classifier to answer the following questions.

1. calculate the probability of pass for student who studied 33 hours.
2. At least how many hours, student should study that makes he will pass the course with the Probability of more than 95%.

Assume the model suggested by the optimizer for odds of passing the course is

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

→ We use Sigmoid function in logistic regrsin

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

1. Calculate the probability of pass for the student who studied 33 hours.

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

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

$$z = -64 + 2 * 33$$

$$= -64 + 66 = 2$$

→ $$p = \frac{1}{1+e^{-2}} = 0.88$$

→ That is, if student studies 33 hours, then there is 88% chance that the student will pass the exam.

2. Atleast how many hours student should study that makes he will pass the course with the probability of more than 95%:

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

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

$$\Rightarrow 0.95 * e^{-z} = 1 - 0.95$$

$$\Rightarrow e^{-z} = \frac{0.05}{0.95} \Rightarrow 0.0526,$$

Apply Natural log functions

formula

$$\boxed{\ln(e^x) = x}$$

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

$$\Rightarrow -z = \ln(0.0526) = -2.94$$

$$\boxed{z = 2.94}$$

$$\log(\text{odds}) = z = -64 + 2 * \text{hours}$$

$$\Rightarrow 2.94 = -64 + 2 * \text{hours}$$

$$\Rightarrow 2 * \text{hours} = 2.94 + 64$$

$$\Rightarrow 2 * \text{hours} = 66.94$$

$$\Rightarrow \text{hours} = \frac{66.94}{2} = 33.47 \text{ hours.}$$

The student should study atleast 33.47 hours so that he will pass the exam with more than 95% probability.

→ Linear Regression predicts the numerical response but is not suitable for predicting the categorical variables.

→ When categorial variables are involved, it is called classification problem.

→ Logistic regression is suitable for binary classification problem.

→ Here, the output is often a categorical variable.

Applications of Logistic Regression:

→ Is the mail spam or not spam? The answer is Yes or No. Thus, categorical dependent variable is a binary response of Yes or no.

→ If the student should be admitted or not is based on entrance examination marks. Here, categorical variable response is admitted or not.

Advantages of Linear Logistic Regression:

→ Logistic Regression performs better when the data is linearly seperable.

→ It does not require too many computational resources as it's highly interpretable.

→ There is no problem scaling the input features. It does not require tuning.

→ It is easy to implement and train a model using logistic regression.

→ It gives a measure of how relevant a predictor (coefficient size) is, and its direction of association (positive or negative).

Linear Regression	Logistic Regression
1. used to solve regression problems.	used to solve classification problems.
2. The response variables are continuous in nature	The response variable is categorical in nature.
3. It helps estimate the dependent variable when there is a change in the independent variable.	It helps to calculate the possibility of a particular event taking place.
4. It is a straight line.	It is an S-curve (S = sigmoid).

Classification - Bayes theorem:

→ Bayes theorem is the cornerstone of Bayesian learning methods because it provides a way to calculate the posterior probability P(h|D) from
i) the prior probability P(h),
ii) Probability over the data set P(D) and
iii) current probability P(D|h).

$$P(h|D) = \frac{P(D|h)\,P(h)}{P(D)}$$

Maximum A Posteriori (MAP) hypothesis:

→ The learner considers some set of candidate hypothesis H and is intrested in finding the most probable hypothesis $h \in H$ given the observed data D (or at least one of the maximally probable if there are several.

→ Any such maximally probable hypothesis is called MAP hypothesis.

→ We can determine the MAP hypothesis by using Bayes theorem to calculate the posterior probability of each candidate hypothesis.

→ More precisely, we will say that h_{MAP} is a MAP hypothesis provided

$$h_{MAP} = \underset{h \in H}{\operatorname{argmax}}\ P(h|D)$$

$$= \underset{h \in H}{\operatorname{argmax}}\ \frac{P(D|h)\,P(h)}{P(D)}$$

$$\boxed{h_{MAP} = \underset{h \in H}{\operatorname{argmax}}\ P(D|h)\,P(h)}$$

Maximum Likelihood and Least-squared error hypothesis:

→ Many learning approaches such as neural network learning, linear regression & polynomial curve fitting try to learn a continuous-valued target function.

→ under certain assumptions any learning algorithm that minimizes the squared error btw the o/p hyp predictions & the training data will o/p a Maximum Likelihood hypothesis.

→ The significance of this result is that it provides a Bayesian justification (under certain assumptions) for many neural n/w & other curve fitting methods that attempt to minimize the sum of squared errors over the training data.

→ In order to find the Max likihood hypothesis in Bayesian learning for continuous valued target function, we start the max likihood hypothesis definition but using lower case p to refer to the probability density function.

$$h_{ML} = \underset{h \in H}{\operatorname{argmax}}\ p(D|h)$$

→ We assume a fixed set of training instances $(x_1 \ldots x_m)$ and therefore consider the data D to be the corresponding sequence of target values

$$D = (d_1 \ldots d_m)$$

→ Here, we can write $p(D|h)$ as the product of the various $p(d_i|h)$

$$h_{ML} = \underset{h \in H}{\operatorname{argmax}} \prod_{i=1}^{m} p(d_i|h)$$

Naive Bayes classifier: Algorithm:

→ Bayes theorem is the cornerstone of Naive Bayes classifier because it provides a way to calculate the posterior probability $P(h|D)$, from the prior probability $P(h)$, together with $P(D)$ and $P(D|h)$

$$P(h|D) = \frac{P(D|h)\,P(h)}{P(D)}$$

$$h_{MAP} \equiv \underset{h \in H}{\operatorname{argmax}}\ P(h|D)$$

$$\equiv \underset{h \in H}{\operatorname{argmax}}\ \frac{P(D|h)\,P(h)}{P(D)}$$

$$= \underset{h \in H}{\operatorname{argmax}}\ P(D|h)\,P(h)$$

→ The Bayesian approach to classifying the new instance is to assign the most probable target value, V_{MAP} given the attribute values $\langle a_1, a_2 \cdots a_n \rangle$ that describe the instance.

$$V_{MAP} = \underset{v_j \in V}{\operatorname{argmax}}\ P(v_j | a_1, a_2 \cdots a_n)$$

→ We can use Bayes theorem to rewrite this expression as

$$V_{MAP} = \underset{v_j \in V}{\operatorname{argmax}} \frac{P(a_1, a_2 \ldots a_n | v_j) P(v_j)}{P(a_1, a_2 \ldots a_n)}$$

$$= \underset{v_j \in V}{\operatorname{argmax}} P(a_1, a_2 \ldots a_n | v_j) P(v_j)$$

Naive Bayes classifier:

$$V_{NB} = \underset{v_j \in V}{\operatorname{argmax}} P(v_j) \prod_i P(a_i | v_j)$$

Ex:

Day	Outlook	Temp	Humidity	Wind	Play Tennis
D1	Sunny	Hot	High	Weak	NO
D2	Sunny	Hot	High	strong	No
D3	overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	cool	Normal	Weak	Yes
D6	Rain	cool	Normal	Strong	No
D7	overcast	cool	Normal	strong	Yes
D8	Sunny	Mild	High	Weak	NO
D9	Sunny	cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	strong	Yes
D12	overcast	Mild	High	strong	Yes
D13	overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	strong	No

→ Neco instance to be classified,

<outlook = sunny, temp = cool, Humidity

Humidity = High, Wind = strong>

→ Naive Bayes classifier

$$V_{NB} = \underset{v_j \in V}{\operatorname{argmax}}\ P(v_j) \prod_i P(a_i | v_j)$$

P(yes | neco instance) = P(yes) * P(sunny | yes) *
P(cool | yes) * P(high | yes) *
P(strong | yes)

V_{NB}(Yes)

P(no | neco instance) = P(no) * P(sunny | no) *
P(sunny | no) * P(cool | no) *
P(high | no) * P(strong | no)

V_{NB}(NO)

$P(yes) = \frac{9}{14}$ and $P(no) = \frac{5}{14}$

Outlook	Yes	NO
sunny	2|9	3|5
overcast	4|9	0|5
Rainy	3|9	3|5

Humidity	Yes	NO
High	3|9	4|5
Normal	6|9	1|5

Temp	Yes	NO
Hot	2/9	3/5
Mild	4/9	2/5
cool	3/9	2/5

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

P(yes/neco instance) = 0.0053

P(no/neco instance) = 0.0206

As P(no/neco instance) is more, Hence, the neco example is classified as 'NO' in this case.

Naive Bayes classifier - Example 2:

→ Estimate conditional probabilities of each attributes { color, legs, height, smelly } for the species classes:

{ M, H } using the data given in the table.

→ using these probabilities estimate the probability values for the neco instance - (color = Green, legs = 2, Height = Tall and Smelly = NO).

$$V_{NB}^{(Yes)} = \frac{V_{NB}(Yes)}{V_{NB}(Yes) + V_{NB}(NO)} = 0.205$$

$$V_{NB}^{(NO)} = \frac{V_{NB}(NO)}{V_{NB}(Yes) + V_{NB}(NO)} = 0.795$$

NO	color	legs	Height	Smelly	Species
1	White	3	Short	Yes	M
2	Green	2	Tall	NO	M
3	Green	3	short	Yes	M
4	White	3	short	Yes	M
5	Green	2	short	NO	H
6	White	2	tall	NO	H
7	White	2	Tall	NO	H
8	White	2	short	Yes	H

New instance = (color = Green, legs = 2, Height = Tall and smelly = NO)

$P(M) = \frac{4}{8} = 0.5$

$P(H) = \frac{4}{8} = 0.5$

color	M	H
White	2/4	3/4
Green	2/4	1/4

Legs	M	H
2	1/4	4/4
3	3/4	0/4

Height	M	H
Tall	3/4	2/4
Short	1/4	2/4

Smelly	M	H
Yes	3/4	1/4
NO	1/4	3/4

P(M/Neco Instance) = P(M) * P(color = Green/M)
* P(legs = 2/M) * P(Height = tall/M) * P(smelly = no/M)

$\Rightarrow P(M/\text{Neco Instance}) = 0.5 * \frac{3}{4} * \frac{1}{4} * \frac{3}{4} * \frac{1}{4}$

$= 0.0117$

P(H/Neco Instance) = P(H) * P(color = Green/H)
* P(legs = 2/H) * P(Height = tall/H) * P(smelly = no/H)

$\rightarrow P(H/\text{Neco Instance}) = 0.5 * \frac{1}{4} * \frac{4}{4} * \frac{2}{4} * \frac{3}{4}$

$= 0.047$

P(H/Neco Instance) > P(M/Neco Instance)

Hence, the neco instance belongs to species H.

Example 3:

S.no	color	Type	Origin	stolen
1	Red	sports	Domestic	Yes
2	Red	sports	Domestic	NO
3	Red	sports	Domestic	Yes
4	Yellow	sports	Domestic	NO
5	Yellow	sports	imported	Yes
6	Yellow	SUV	imported	NO
7	Yellow	SUV	imported	Yes
8	Yellow	SUV	Domestic	NO
9	Red	SUV	imported	NO
10	Red	sports	imported	Yes

New instance = (Red, SUV, Domestic)

$P(Yes) = \frac{5}{10} = 0.5$; $P(NO) = \frac{5}{10} = 0.5$

Color	Yes	NO
Red	3/5	2/5
Yellow	2/5	3/5

Type	Yes	NO
sports	4/5	2/5
SUV	1/5	3/5

origin	Yes	No
Domestic	2/5	3/5
imported	3/5	2/5

$P(Y|n1) = P(Yes) * P(color=Red|Yes) * P(type=SUV|Yes) * P(or=Dom|Yes)$

$= \frac{5}{10} * \frac{3}{5} * \frac{1}{5} * \frac{2}{5} = \frac{3}{125} = 0.024$

$P(N|n1) = P(NO) * P(Color=Red|NO) * P(type=SUV|NO) * P(or=D|NO)$

$= \frac{5}{10} * \frac{2}{5} * \frac{3}{5} * \frac{3}{5} = \frac{9}{125} = 0.072$

NO

Example 3:

S.no	color	Type	Origin	stolen
1	Red	sports	Domestic	Yes
2	Red	sports	Domestic	NO
3	Red	sports	Domestic	Yes
4	Yellow	sports	Domestic	NO
5	Yellow	sports	imported	Yes
6	Yellow	SUV	imported	NO
7	Yellow	SUV	imported	Yes
8	Yellow	SUV	Domestic	NO
9	Red	SUV	imported	NO
10	Red	sports	imported	Yes.

New instance = (Red, SUV, Domestic)

$P(Yes) = \frac{5}{10} = 0.5$; $P(NO) = \frac{5}{10} = 0.5$

Color	Yes	NO
Red	3/5	2/5
Yellow	2/5	3/5

Type	Yes	NO
sports	4/5	2/5
SUV	1/5	3/5

Origin	Yes	No
Domestic	2/5	3/5
imported	3/5	2/5

$P(Y|n) = P(Yes) * P(Colr=Red|Yes) * P(type=SUV|Yes) * P(Or=Domestic)$

$= \frac{5}{10} * \frac{3}{5} * \frac{1}{5} * \frac{2}{5} = \frac{3}{125} = 0.024$

$P(N|n) = P(NO) * P(Colr=Red|NO) * P(type=SUV|NO) * P(Or=Dom|NO)$

$= \frac{5}{10} * \frac{2}{5} * \frac{3}{5} * \frac{3}{5} = \frac{9}{125} = 0.072$ NO

K-Nearest Neighbour Classifier:

→ KNN is based on feature similarity, we can do classification (is done using distance measure of new instance) using KNN classifier.

Def: KNN is one of the simplest supervised ML Algorithm mostly used for classification.

→ It is also called as Lazy learning.

Example 1: Given data Query = x (Constant) — nearest neighbour

x = (maths = 6, CS = 8) & K = 3

CS - computer science

Classification = pass / Fail

Maths	CS	Result
4	3	F
6	7	P
7	8	P
5	5	F
8	8	P

Euclidean Distance (d)

$$d = \sqrt{(x_{o_1} - x_{A_1})^2 + (x_{o_2} - x_{A_2})^2}$$

o → observed value

a → actual value.

K=3

1) Calculate $d_1 = \sqrt{(6-4)^2 + (8-3)^2}$

$= \sqrt{2^2 + 5^2} = \sqrt{29} = 5.38$

2) Calculate $d_2 = \sqrt{(6-6)^2 - (8-7)^2} = \sqrt{0+1} = 1$

3) $d_3 = \sqrt{(6-7)^2 + (8-8)^2} = \sqrt{1+0} = 1$

4) $d_4 = \sqrt{(6-5)^2 + (8-5)^2} = \sqrt{1+9} = \sqrt{10} = 3.16$

5) $d_5 = \sqrt{(6-8)^2 + (8-8)^2} = \sqrt{4+0} = \sqrt{4} = 2$

Compare with 3 neighbours.

d_2, d_3, d_5 = Pass

Instance-based learning Ex:

value can be predicted based on the previous examples by calculating the distance measure.

Ex:

S.NO	Height	Weight	Target
1	150	50	Medium
2	155	55	Medium
3	160	60	Large
4	161	59	Large
5	158	65	Large
6	157	54	?

Given K=3, $d_1 = 8.06$, $d_2 = 2.24$, $d_3 = 6.71$

$d_4 = 6.40$, $d_5 = 11.05$ d_6 = Large

→ Ex 2 KNN

Height (CM)	Weight (kg)	class
167	51	underweight
182	62	Normal
176	69	Normal
173	64	Normal
172	65	Normal
174	56	underweight
169	58	Normal
173	57	Normal
170	55	Normal
170 (x_2)	57 (y_2)	?

x_2, y_2 - fixed

First find the nearest neighbour.

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

$$d_1 = \sqrt{(170-167)^2 + (57-51)^2}$$

$$d_2 = \sqrt{(170-182)^2 + (57-62)^2}$$

$d_1 = 6.7$; $d_2 = 13$; $d_3 = 13.4$; $d_5 = 7.6$;

$d_6 = 8.2$; $d_7 = 4.1$; $d_8 = 1.4$; $d_9 = 2$; $d_8 = 3$

→ Sort based on distance.

If $k=1$ then it is Normal.

if $k=2$, then Normal.

KNN-3:

The Restaurent A sells burger with optional flavours pepper, Ginger and chilly.

Everyday this week you have tried a burger(A to E) & kept a record of which you liked.

using thamming distance, show how the 3NN classifier with majority voting would classify.

{pepper: false, ginger = true, chilly: true}

Day	pepper	Ginger	chilly	liked	Distance
(2) A	True	True	True	False	1+0+0 = 1
B	True	False	False	True	1+1+1 = 3
(1) C	False	True	True	False	0+0+0=0
(3) D	False	True	False	True	0+0+1=1
E	True	False	False	True	1+1+1=3

→ In Hamming distance, if the values are same that is x_1 is same as x_2 then distance is 0 (hence) or 1.

Ex: if x_1 = blue; x_2 = blue then distance blw x_1 & x_2 = 0

if x_1 = blue; x_2 = red then x_1 & x_2 = 1

→ From this 3, we have to find new lable.

Majority is False.

So, new example is classified as False in this case.

Support Vector Machine : Introduction :

→ SVM is one of the most popular supervised learning algorithms, which is used for classification as well as Regression problems.

→ primarily, it is used for classification problems in ML.

→ The goal of the SVM algorithm is to create the best line (or) decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future.

→ The best decision boundary is called Hyperplane

How to draw Hyperplane ?

→ The dimensions of the hyperplane depend on the features present in the dataset, which means if there are 2 features, then hyperplane will be a straight line.

→ If there are 3 features, then hyperplane will be a 2-dimensional plane.

→ We always create a hyperplane that has a maximum margin, which means the maximum distance between the data points.

SVM can be two types:

1. Linear SVM
2. Non-linear SVM

1. Linear SVM: Linear SVM is used for linearly seperable data, which means if a dataset can be classified in to two classes by using a single straight line, then such data is termed as linearly seperable data, and classifier is used called as Linear SVM classifier.

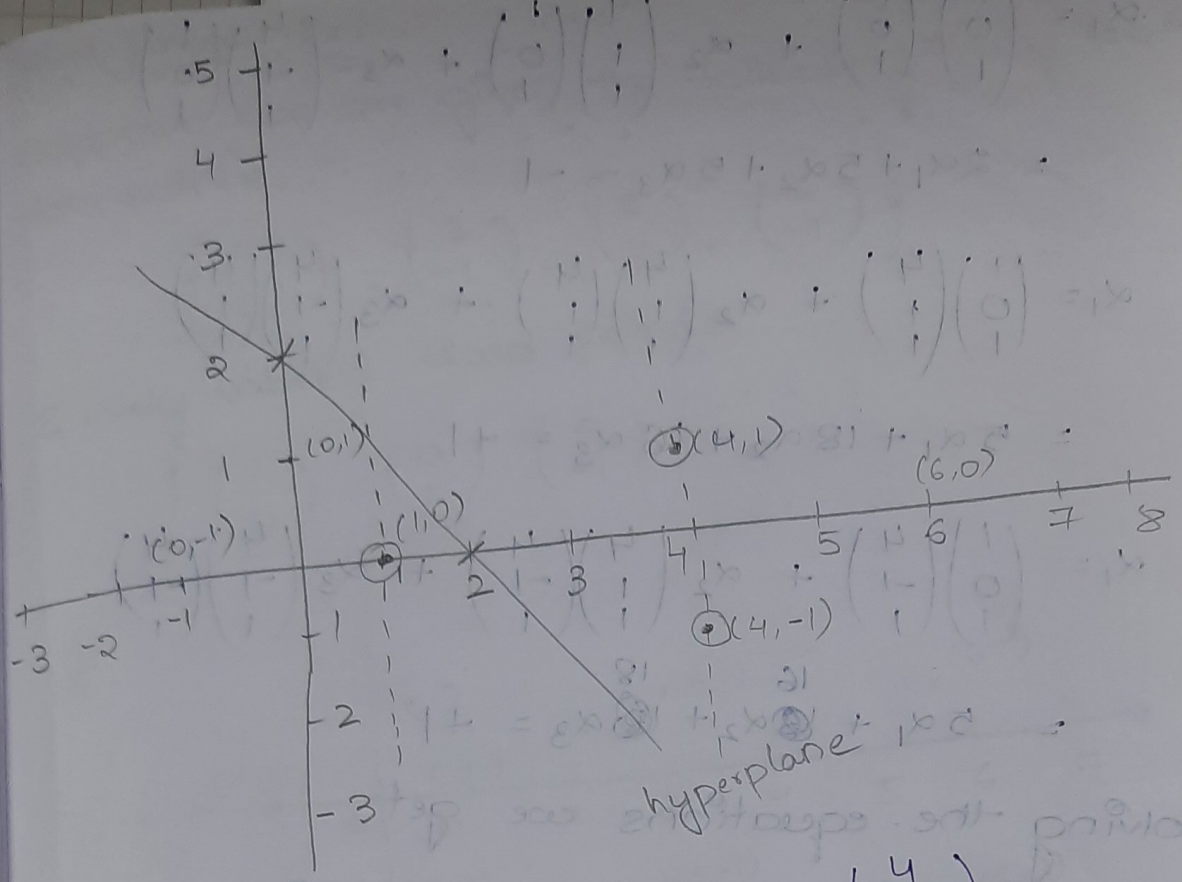
2. Non-linear SVM: It is used for Non-linearly seperated data, which means if a dataset cannot be classified by using a straight line then such data is termed as Non-linear data and classifier used is called as Non-linear SVM classifier.

SVM - Example:

→ points (4,1), (4,-1) and (6,0) belong to class +ve

→ points (1,0), (0,1) & (0,-1) belong to -ve.

→ Draw an optimal hyperplane to classify the points.



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

→ The augumented vector can be obtained by adding the bias given as follows:

$$\bar{S}_1 = \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} ; \quad \bar{S}_2 = \begin{pmatrix} 4 \\ 1 \\ 1 \end{pmatrix} ; \quad \bar{S}_3 = \begin{pmatrix} 4 \\ -1 \\ 1 \end{pmatrix}$$

Set of 3 equations can be obtaind based on SVM

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

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

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

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

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

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

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

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

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

Solving the equations we get

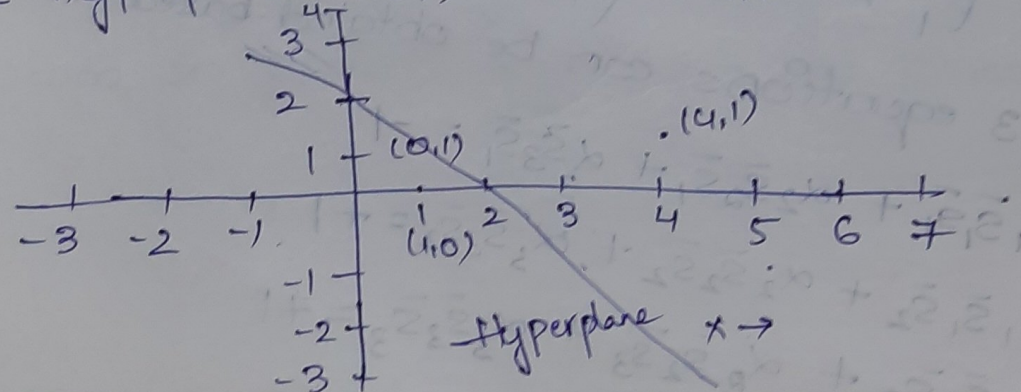
$$\alpha_1 = -3 \; ; \; \alpha_2 = +1 \; ; \; \alpha_3 = 0.$$

The optimal hyperplane is given as

$$w = \sum_{1}^{3} \alpha_i * \bar{s}_i$$

$$= -3 * \begin{pmatrix}1\\0\\1\end{pmatrix} + 1 * \begin{pmatrix}4\\1\\1\end{pmatrix} + 0 * \begin{pmatrix}4\\-1\\1\end{pmatrix} = \begin{pmatrix}1\\1\\-2\end{pmatrix}$$

The hyperplane is (1,1) with an offset -2.



0½-parallel
01-horizontal
11-45°

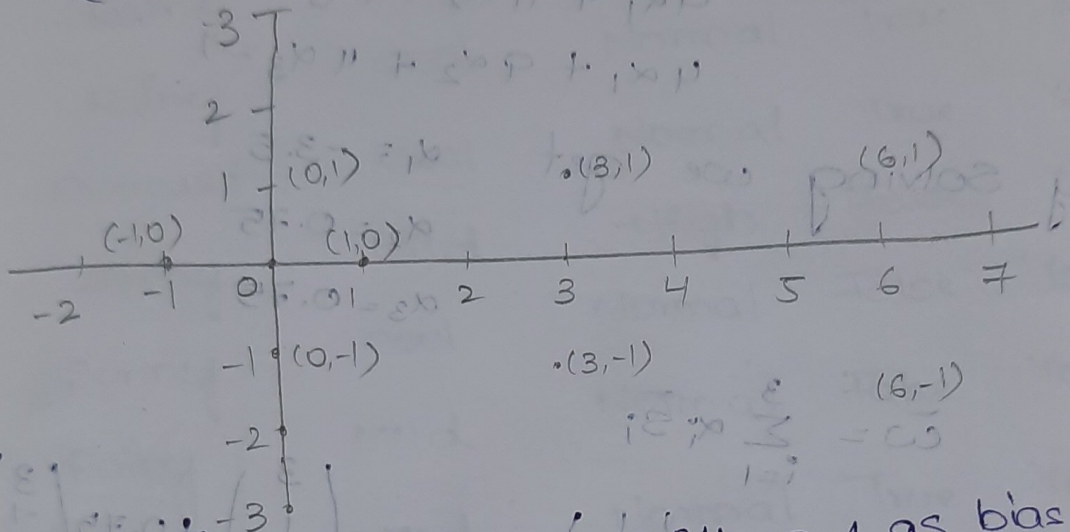
SVM-Linear-Ex:

Suppose we are given the following +vely labeled data points;

$$\left\{\begin{pmatrix}3\\1\end{pmatrix}, \begin{pmatrix}3\\-1\end{pmatrix}, \begin{pmatrix}6\\1\end{pmatrix}, \begin{pmatrix}6\\-1\end{pmatrix}\right\}$$

& -vely labeled data points

$$\left\{\begin{pmatrix}1\\0\end{pmatrix} \begin{pmatrix}0\\1\end{pmatrix} \begin{pmatrix}0\\-1\end{pmatrix} \begin{pmatrix}-1\\0\end{pmatrix}\right\}$$



Each vector is augmented with a 1 as bias i/p.

So, $s_1 = \begin{pmatrix}1\\0\end{pmatrix}$ then $\bar{s}_1 = \begin{pmatrix}1\\0\\1\end{pmatrix}$

$s_2 = \begin{pmatrix}3\\1\end{pmatrix}$ then $\bar{s}_2 = \begin{pmatrix}3\\1\\1\end{pmatrix}$ & $s_3 = \begin{pmatrix}3\\-1\end{pmatrix}$ then $\bar{s}_3 = \begin{pmatrix}3\\-1\\1\end{pmatrix}$

3 equations:

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

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

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

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

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

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

by solving $2\alpha_1 + 4\alpha_2 + 4\alpha_3 = -1$

$4\alpha_1 + 11\alpha_2 + 9\alpha_3 = 1$

$4\alpha_1 + 9\alpha_2 + 11\alpha_3 = 1$

by solving we get $\alpha_1 = -3.5$

$\alpha_2 = 0.75$

$\alpha_3 = 0.75$

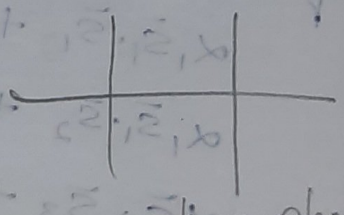
$$\bar{w} = \sum_{i=1}^{3} \alpha_i \bar{s}_i$$

$$= -3.5 \begin{pmatrix}1\\0\\1\end{pmatrix} + 0.75 \begin{pmatrix}3\\1\\1\end{pmatrix} + 0.75 \begin{pmatrix}3\\-1\\1\end{pmatrix}$$

Finally, remembering that our vectors are augmented with a bias.

→ We equate the last entry in $\bar{w}$ as the hyperplane offset b and write the seperating hyperplane equation $y = wx + b$

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



Hyperplane

CART (Classification & Regression Techniques)

Ex:

Outlook	Temp	Humidity	Windy	play
Sunny	Hot	High	False	NO
Sunny	Hot	High	True	NO
Overcast	Hot	High	False	Yes
Rainy	Mild	High	False	Yes
Rainy	cool	Normal	False	Yes
Rainy	cool	Normal	True	NO
Overcast	cool	Normal	True	Yes
Sunny	Mild	High	False	NO
Sunny	cool	Normal	False	Yes
Rainy	Mild	Normal	False	Yes
Sunny	Mild	Normal	True	Yes
Overcast	Mild	High	True	Yes
Overcast	Hot	Normal	False	Yes
Rainy	Mild	High	True	NO

→ For the given play Tennis Data set apply the Decision Tree algorithm & find the optimal decision tree.

→ Also predict class label for the following example

Sunny	Hot	Normal	True	?

Outlook:

Overcast	4	Yes	4
		NO	0
Sunny	5	Yes	2
		NO	3
Rainy	5	Yes	3
		NO	2

Attribute	Rules	Error	Total Error
Outlook	Sunny → NO	2/5	4/14
	Overcast → Yes	0/4	
	Rainy → Yes	2/5	

Temp:

Hot	4	Yes	2
		NO	2
Mild	6	Yes	4
		NO	2
Cold	4	Yes	3
		NO	1

Attribute	Rules	Error	Total Error
Temp	Hot → NO	2/4	5/14
	Mild → Yes	2/6	
	Cool → Yes	1/4	

Humidity:

High	7	Yes	3
		NO	4
Normal	7	Yes	6
		NO	1

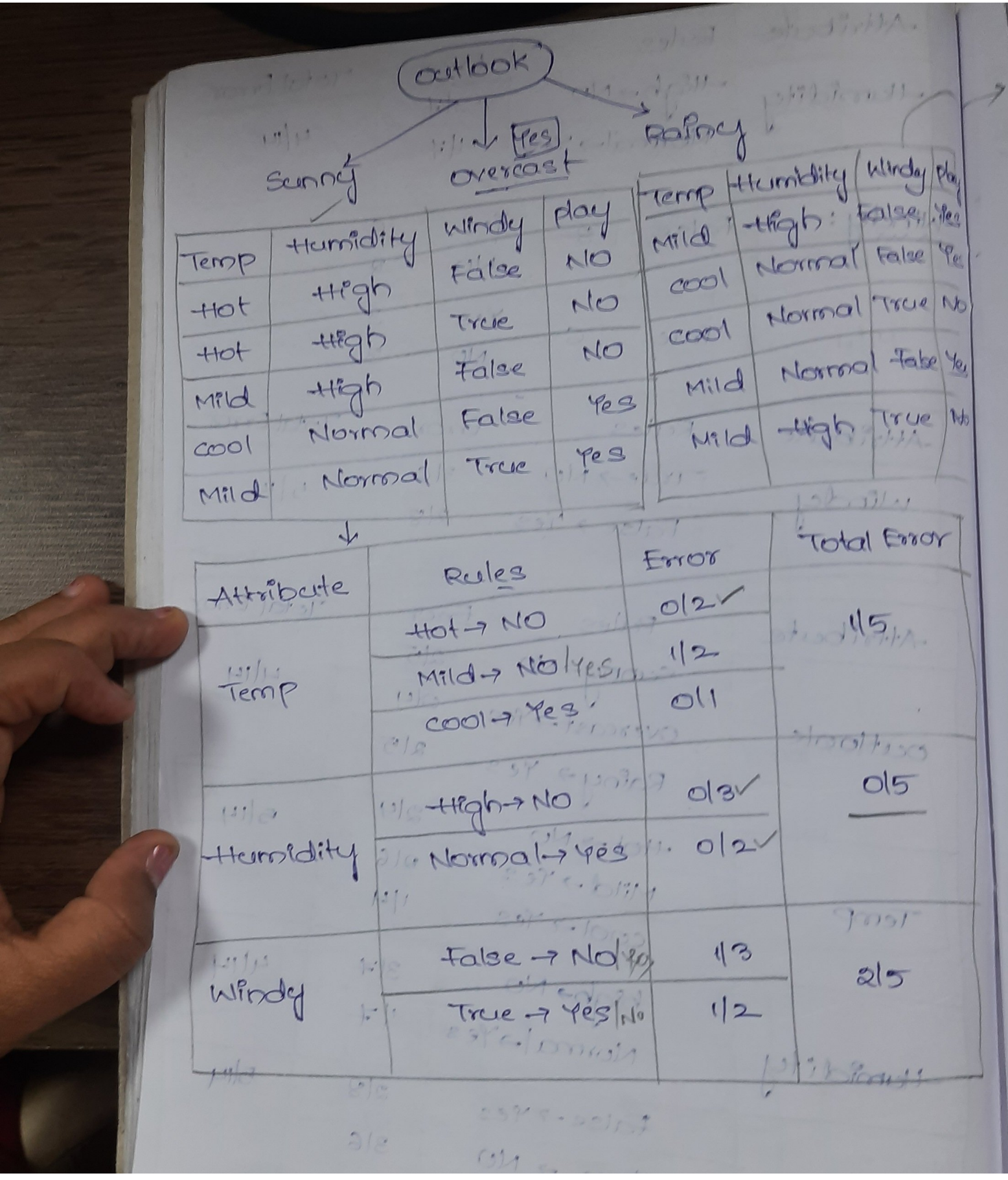
Attribute	Rules	Error	Total Error
Humidity	High → NO	3/7	4/14
	Normal → Yes	1/7	

Windy:

False	8	Yes	6
		NO	2
True	6	Yes	3
		NO	3

Attribute	Rules	Error	Total Error
Windy	True → NO Yes	3/6	5/14
	False → Yes	2/8	

Attribute	Rules	Error	Total Error
outlook	Sunny → NO	2/5	4/14
	overcast → Yes	0/4	
	Rainy → Yes	2/5	
Temp	Hot → NO	2/4	5/14
	Mild → Yes	2/6	
	cool → Yes	1/4	
Humidity	high → NO	3/7	4/14
	Normal → Yes	1/7	
Windy	False → Yes	2/8	5/14
	True → NO	3/6	



Temp	Humidity	Windy	play
Hot	High	False	NO
Hot	High	True	NO
Mild	High	False	NO
cool	Normal	False	Yes
Mild	Normal	True	Yes

Temp	Humidity	Windy	play
Mild	High	False	Yes
cool	Normal	False	Yes
cool	Normal	True	No
Mild	Normal	False	Yes
Mild	High	True	No

Attribute	Rules	Error	Total Error
Temp	Hot → NO	0/2 ✓	4/5
	Mild → No/Yes	1/2	
	cool → Yes	0/1	
Humidity	High → NO	0/3 ✓	0/5
	Normal → Yes	0/2 ✓	
Windy	False → No/Yes	1/3	2/5
	True → Yes/No	1/2	

Attribute	Rules	Error	Total Error
Temp	Mild → Yes	1/3	2/5
	cool → Yes/No	1/2	
Humidity	High → No/Yes	1/2	2/5
	Normal → Yes	1/3	
Windy	False → Yes	0/3	0/5
	True → No	0/2