"Logistic Regression is a classification Problem" Date:

LOGISTIC REGRESSION:

- In classification, the output 'y' is discrete value
- In logistic Regression, to develop an algorithm to determine what class a new input should fall into:

Example: Email: Spam/not Spam

PROBLEM WE DISCUSS → Binary Classification (0 or 1)

→ How to develop the classification algorithm?

→ If we use linear regression there exists a problem that if the number is greater than 1 or less than 0 so, in which category we will classify it.

→ Linear regression donot work on mixed data as well as Linear regression could not solve boundary problem.

→ Hypothesis can give value larger than 1 or less than 0

LOGISTIC REGRESSION (CLASSIFICATION)

HYPOTHESES REPRESENTATION:

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

$$h_\theta(x) = g(z)$$

So,

$$g(z) = g(\theta^T x)$$

Therefore,

$$h_\theta(x) = g(z) = g(\theta^T x) = \frac{1}{1+e^{-z}} = \frac{1}{1+e^{-\theta^T x}}$$

($\because g(z)$ = sigmoid)

$\because z$ is a real number

Where $z = \theta^T x$

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

$$\begin{bmatrix} \theta_0 \\ \theta_1 \end{bmatrix} \begin{bmatrix} 1 \\ x \end{bmatrix}$$

$$[\theta_0 \ \theta_1] \begin{bmatrix} 1 \\ x \end{bmatrix}$$

$\theta_0 + \theta_1 x = \theta^T x$

Date:

Hypotheses = $h_\theta(x) = \frac{1}{1+e^{-\theta^T x}}$

This is the Sigmoid function or the logistic function.

Question:

→ What value should be of $\theta^T x$?

→ It is not necessary that the value of $\theta^T x$ is always linear $(\theta_0 + \theta_1 x)$, it can be

$\theta^T x = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots$

or $\theta^T x = \theta_0 + \theta_1 x_1 + \theta_2 x_2^2 + \cdots$

→ What decides the value of z

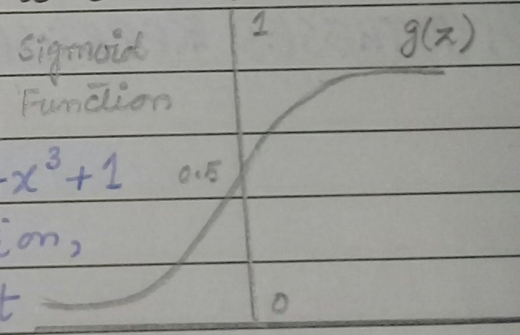
The value of z will be decided by Regularization

SIGMOID FUNCTION:

→ Crosses 0.5 at the origin, then flattens out

→ Asymptotes between 0 and 1

Asymptotes means: They to Reach or give Nearest value 0 or 1, but not exactly reach to 0 or 1 (decimal value)



→ $g(\theta_0, \theta_1)$ → Similar to $f(x) = x^2 + x^3 + 1$

→ In $f(x)$ x is a part of a function, similarly, in $g(\theta_0, \theta_1)$ θ is a part of a function.

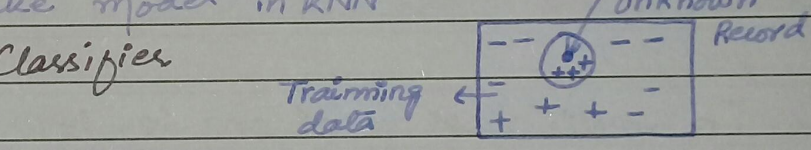
Date:

→ KNN for classification & Regression
→ Understanding How it works
→ Example
→ Python code

K-Nearest Neighbour used for both Linear & Classification
It is the lazy algorithm:
→ because when you work on linear regression

→ When you develop model their exists a plan or best line or a decission boundary behind the model
→ In KNN the interpretation of model of trainning data load into memory & when you give test data then it reads from initial, hitting training data again & again, we don't make model in KNN

Nearest Neighbour Classifier
Requires three things
→ The set of stored records
→ Distance Metric to compute distance b/w records.
→ The value of K, the number of nearest neighbours to retrieve



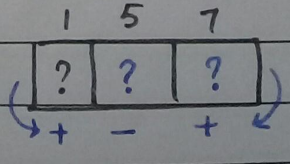
• ہر ایک کے ساتھ distance کرے گا کوئی Method لگائے گا۔

(K) → لسٹ کا نearest کاؤنٹ کرنا چاہتے ہو تم۔

1 test sy n distance agy

IF K = 3 the we pick 3 nearest distances (small)

ٹیسٹ ڈیٹا کے قریب ترین



Normally K is odd

Sign Similarity
Majority Voting } Classification

Date:

→ We are using training data again & again for all tests why it is called "Lazy algorithm."

→ For linear regression, if we apply KNN so, what happened?

- Calculate distance
- Store distance
- 5 Nearest Neighbour apply
- Take Mean

4	8	11	19	51
?	?	?	?	?

Value — Sum of all values test mean value

distance Metric → Kons a tool for Similarity

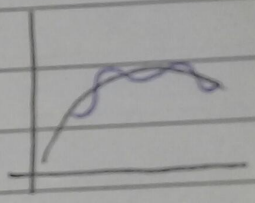
Sno	X_1	X_2	Y
1	2	5	3.9
2	6	3	4.1
3	7	4	2.7
4	1	1	1
	6	8	3 ?

→ cosine,

→ Kaisa lo k tie na ho

→ We will take mean if their is a tie

→ minimum distance with majority voting



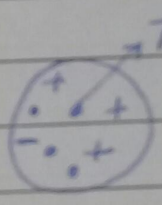
→ In multiclass problem so, how to decide target class?

→ majority voting

→ The minimum mean result

→ It is used for Partitioning

Measuring tool = Euclidean distance



The formula of KNN by Euclidean Distance

$$Dist(C_1, C_2) \sqrt{\sum_{i=1}^{n} (attri(C_1) - attri(C_2))^2}$$

Multi class Problem
We have 2 attributes

Date:

	Sweet	Crunch	Food Type
Grape	8	5	Fruit
Greenbean	3	7	Vegetable
Nuts	3	6	Protein
Orange	7	3	Fruit
Tomato	6	4	?

Maximum K = 4

Minimum K = 1

$Dist^{(1)}(C_T, C_1) = \sqrt{(6-8)^2 + (4-5)^2}$

$= \sqrt{5}$

$Dist^{(1)}(C_T, C_1) = 2.2$

$Dist^{(2)}(C_T, C_2) = \sqrt{(6-3)^2 + (4-7)^2}$

$\sqrt{18}$

$Dist^{(2)}(C_T, C_2) = 4.2$

$Dist^{(3)}(C_T, C_3) = \sqrt{(6-3)^2 + (4-6)^2}$

$= \sqrt{13}$

$Dist^{(3)}(C_T, C_3) = 3.6$

$Dist^{(4)}(C_T, C_4) = \sqrt{(6-7)^2 + (4-3)}$

$= \sqrt{2}$

$Dist^{(4)}(C_T, C_4) = 1.4$

K = 1

$\min_1 (Dist^{(1)}, Dist^{(2)}, Dist^{(3)}, Dist^{(4)})$

Now, $\underbrace{Dist^{(4)}}_{\min} \rightarrow$ Fruit

Prediction for Tomato

Fruit

Date:

FOR K=2

$Dist^{(1)}$ (F) , $Dist^{(3)}$ (P)

K= 4

$Dist^{(1)}$	$Dist^{(2)}$	$Dist^{(3)}$	$Dist^{(4)}$
F	P	P	F

→ We will took out mean & then, we classify

Outcome → will lies where

load data

im

Date:

LINEAR REGRESSION (IMPORTANT POINTS)

- Supervised Learning algorithm * Linear regression is not uncertain
- Works on Continuos values
- draws a linear line

COST FUNCTION FORMULA & GRAPH:

$$J(\theta_0, \theta_1) = \frac{1}{2M} \sum_{i=1}^{m} (h_\theta(x^i) - y^{(i)})^2 \quad \because h_\theta(x) = \theta_0 + \theta_1 x$$

Cost function is used to find θ only

GRADIENT DESCENT FORMULA & GRAPH:

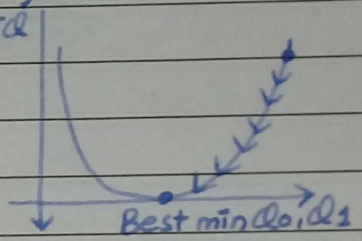
$$\theta_0 = \theta_0 - \frac{\alpha}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})$$

$$\theta_1 = \theta_1 - \frac{\alpha}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)} - y^{(i)}) * x^{(i)}$$

} FOR Linear Regression

General formula of Gradient Descent:

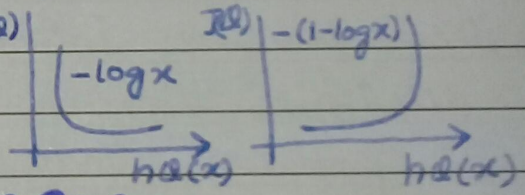
$$\theta_j = \theta_j - \alpha \frac{\partial J}{\partial} J(\theta)$$



Gradient Descent is used to find best minimum θ

LOGISTIC REGRESSION: (IMPORTANT POINTS)

→ Supervised learning algorithm
→ Works on discrete values
→ draws a boundary line b/w classes.



COST FUNCTION FORMULA & GRAPH:

$$J(\theta) = \frac{-1}{m} \left[\sum_{i=1}^{m} y^{(i)} \underbrace{\log h_\theta(x^{(i)})}_{\text{FOR 1}} + \underbrace{(1 - y^{(i)}) \log (1 - h_\theta(x^{(i)})}_{\text{FOR 0}} \right]$$

Cost function is used to find θ only

GRADIENT DESCENT FORMULA & GRAPH:

$$\theta_j = \theta_j - \alpha \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x_j^{(i)}$$

$$\because h_\theta(x) = \frac{1}{1 + e^{-\theta^T x}}$$

Date:

DECISSION TREE

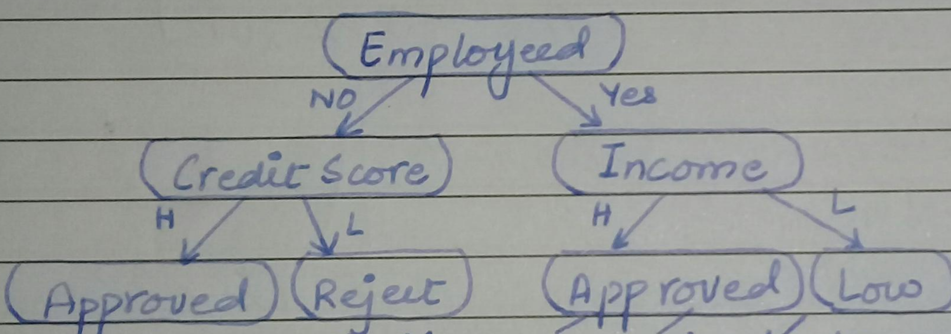
It is used for classification & regression.

~~I~~ It is mostly used for classification

→ Algorithm used for decission tree ID3 Heuristic algorithm.

→ A decission tree is a tree structured classifier

→ It has two types of Nodes
- decission nodes &
- Leaf nodes



Imp ہمیں یہ binary کلاسیفیکیشن کیوں نظر آ رہی ہے؟

Decission Tree Introduction

attribute → mean nodes

The popular attribute selection measure

→ Information gain (Pink بڑی ہو / Blue بڑی ہو) کی جاتی ہے یہاں یہ استعمال ہوتا ہے

→ Gini-index (Python & R uses Gini method)

Hum dono ko use krskty hain

Date:

Har attribute may ktni information Hai
Play is Categorical so, we solved it with information gain

→ Jis may sub sy ziyada information Hogi usko hum Parent Node banygay

→ Or yay nikalnay k liyay hum entropy use krygay i.e, +/-

→ Or recursively
Tree binary Tree hona zaruri nahi Hai

What is Pure?

اگر ہم کسی attributes کے against سارے '+'
یا سارے '-' ہوں تو وہ Pure ہے -
اگر tree کی attribute میں mix ہو تو ہم مزید برانچز بنانی پڑیں گی

attributes khtm hojyn or pure na ho to yay jo error ata

→ ID3 is used for making Tree

- What is the importance of node?

Decission Tree:

$$H(S) = -P(+)\log_2 P(+) - P(-)\log_2 P(-)$$

Entropy representation; P → Probability; $(+)$ → Yes; $(-)$ → NO

→ S.... Subset of training examples

→ P(+) or P(-) -- % of postive or negative examples in S

Cer

$$H(S) = -P(+)\log_2 P(+) - P(-)\log_2 P(-)$$

$$-3/6 \log_2 3/6 - 3/6$$

Date:

How we gain information Gain

→ How many items in pure sets

→

$$Gain(S, A) = H(S) - \sum_{V \in Values(A)} \frac{|S_V|}{|S|} H(S_V)$$

(H(S): main entropy)

→ V: possible values of A

→ S: set of examples (X)

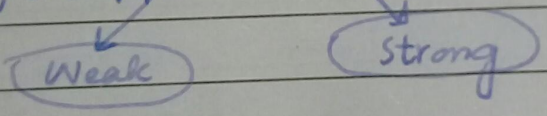
→ S_V: Subset where $X_A = V$

9 Yes / 5 No

$-\frac{9}{14}\log_2\frac{9}{14} - \frac{5}{14}\log_2$ (Wind)

5/14

$H(S) = 0.94$



Gain (S, Wind)

$= H(S) - 8/14\ H(S_{weak}) - 6/14\ H(S_{weak})$

$= 0.94 - 8/14 * 0.81 - 6/14 * 1.0$

$= 0.049$

First we find main Information Gain

Date:

$$-\frac{19}{4}\log_2\frac{9}{14}$$

Information gain & Entropy

Example To find Parent node among

Step # 01: Select parent node

i- Calculate IG

Step # 01: To find parent node among the provided nodes

i- to calculate IG for all nodes

ii- to calculate entropy

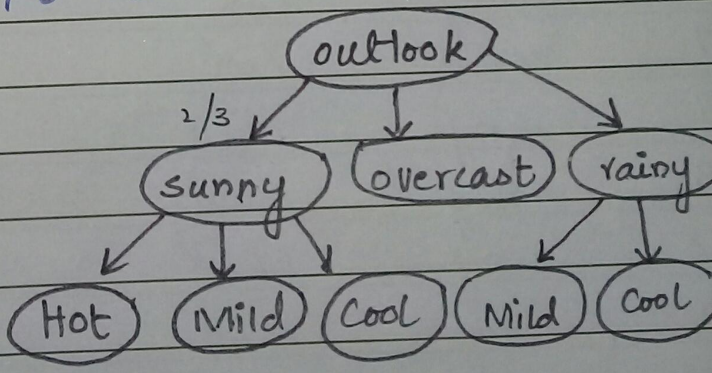
سب سے زیادہ بہتر جو ہو رہا ہے outlook کا ہو رہا ہے Split

$$H(S) = (3) = -\left[\frac{-9}{14}\log_2\frac{9}{14} + \frac{5}{14}\log\frac{5}{14}\right]$$

$$= 0.409 + 0.530$$

$$IHS(3) = 0.940$$

9/5 = 14

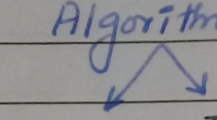


Imp

Date:

DECISSION TREE:

→ A "Decission tree" is a tree which where each node represents a "Feature (Attribute)", each link (Branch) represents a "Decision (Rule)" and each leaf represents an "Outcome."



- Gini Index
- Entropy function
- Information Gain

Sno	Outlook	Temperature	Humidity	Windy	Playtennis
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
3	Overcast	Hot	High	Weak	Yes
4	Rainy	Mild	High	Weak	Yes
5	Rainy	Cool	Normal	Weak	Yes
6	Rainy	Cool	Normal	Strong	No
7	Overcast	Cool	Normal	Strong	Yes
8	Sunny	Mild	High	Weak	No
9	Sunny	Cool	Normal	Weak	Yes
10	Rainy	Mild	Normal	Weak	Yes
11	Sunny	Mild	Normal	Strong	Yes
12	Overcast	Mild	High	Strong	Yes
13	Overcast	Hot	Normal	Weak	Yes
14	Rainy	Mild	High	Strong	No

→ If we have equal number of Positive & Negative then, Entropy = 0

→ If we have only Positive Example & only Negative Example then, Entropy = 0

Date:

Playtennis is a class attribute

→ Step 1: Create a Root Node?

• How to choose the attribute as the root Node?

The attribute that best classifies the training data; use this attribute as the root of the Tree.

→ Calculate **ENTROPY** (Amount of uncertainity in dataset)

$$\text{Entropy} = \frac{-P}{P+n}\log_2\left(\frac{P}{P+n}\right) - \frac{n}{P+n}\log_2\left(\frac{n}{P+n}\right)$$

→ Calculate **AVERAGE** INFORMATION:

$$I(\text{Attribute}) = \sum \frac{Pi+ni}{P+n}\,\text{Entropy}(A)$$

→ Calculate **INFORMATION GAIN**: (Difference in Entropy before & After spliliting dataset on Attribute (A))

$$\text{Gain} = \text{Entropy}(S) - I(\text{Attribute})$$

ID3 Algorithm Steps:

1- Compute Entropy for data set Entropy (S)

2- For every attribute or feature:

1- Calculate Entropy for all other values Entropy(A

2- Take average Information Entropy for the curre attribute

3- Calculate Gain for the current attribute

3- Pick the Highest gain attribute

4- Repeat until we get the desired tree.

Count Play Tennis Positive & Negative

Play Tennis	Yes	No	
	9	5	Total = 14

$$\text{Entropy} = \frac{-P}{P+n}\log_2\left(\frac{P}{P+n}\right) - \frac{n}{P+n}\log_2\left(\frac{n}{P+n}\right)$$

$$\text{Entropy} = \frac{-9}{9+5}\log_2\left(\frac{9}{9+5}\right) - \left(\frac{5}{9+5}\right)\log_2\left(\frac{5}{9+5}\right)$$

$$\text{Entropy} = \frac{-9}{14}\log_2\left(\frac{9}{14}\right) - \frac{5}{14}\log_2\left(\frac{5}{14}\right) = 0.940$$

→ For each attribute (let say Out look)

- Calculate Entropy for each values: 'Sunny', 'Rainy', ~~'Windy'~~ Overcast

Outlook	Yes	No	Total	
* Rainy	3	2	5	Out look
* Overcast	4	0	4	
* Sunny	2	3	5	

$$\text{Entropy of Rainy} = \frac{-3}{3+2}\log_2\left(\frac{3}{3+2}\right) - \frac{2}{3+2}\log_2\frac{2}{3+2}$$

Entropy of Rainy = 0.971

$$\text{Entropy of Sunny} = \frac{-2}{2+3}\log_2\left(\frac{2}{2+3}\right) - \left(\frac{3}{2+3}\right)\log_2\left(\frac{3}{2+3}\right)$$

Entropy of Sunny = 0.971

$$\text{Entropy of Overcast} = \frac{-4}{4+0}\log_2\left(\frac{4}{4+0}\right) - \left(\frac{0}{4+0}\right)\log_2\left(\frac{0}{4+0}\right)$$

Entropy of Overcast = 0

• Calculate Average Information Entropy:

$$I(\text{Outlook}) = \frac{P_{sunny}+n_{sunny}}{P+n}\text{Entropy}(\text{outlook}=\text{sunny}) + \frac{P_{rainy}+n_{rainy}}{P+n}\text{Entropy}(\text{outlook}=\text{rainy}) + \frac{P_{overcast}+n_{overcast}}{P+n}\text{Entropy}(\text{outlook}=\text{overcast})$$

$$I(\text{Outlook}) = \frac{2+3}{9+5}(0.971) + \frac{3+2}{9+5}(0.971) + \frac{4+0}{9+5}(0)$$

$$I(\text{outlook}) = 5.3944 + 0.3467 + 0$$

$$I(\text{outlook}) = 0.693$$

• Calculate Gain: attribute is outlook

$$\text{Gain} = \text{Entropy}(S) - I(\text{Attribute})$$

$$\text{Gain} = 0.940 - 0.693$$

$$\text{Gain} = 0.247$$

Sunny	Yes	No	Total	Entropy
* Hot	0	2	2	
* Mild	1	1	2	
* Cool	1	0	1	

$$\text{Entropy of Sunny \& Hot} = -\frac{0}{0+2}\log_2\left(\frac{0}{0+2}\right) - \left(\frac{2}{0+2}\right)\log_2\left(\frac{2}{0+2}\right)$$

Entropy of Sunny & Hot = 0

$$\text{Entropy of Sunny \& Mild} = -\frac{1}{1+1}\log_2\left(\frac{1}{1+1}\right) - \frac{1}{1+1}\log_2\left(\frac{1}{1+1}\right)$$

Entropy of Sunny & Mild = 1

Entropy of Sunny & Cool = $-1/1+0\log_2(1/0+1) - 0/1+0\log_2(0/0+1)$

Entropy of Sunny = 0

Date:

• Calculate Average Information Entropy:

$I(\text{Sunny}) = \frac{P_{Hot} + n_{Hot}}{P+n}\ \text{Entropy}(\text{sunny}=\text{Hot}) + \frac{P_{Mild} + n_{Mild}}{P+n}\ \text{Entropy}(\text{sunny}=\text{Mild}) + \frac{P_{Cool} + n_{Cool}}{P+n}\ \text{Entropy}(s=c)$

$I(\text{sunny}) = \frac{0+2}{5}(0) + \frac{1+1}{5}(1) + \frac{1+0}{5}(0)$

$I(\text{sunny}) = 0.4$

• Calculate Gain: attribute is Sunny

Gain = Entropy(S) − I(sunny)

Gain = 0.940 − 0.4

Gain = 0.54

Sunny and Humidity	Yes	No	Total	Entropy
High	0	3	3	
Normal	2	0	2	

Entropy of Sunny & Humidity (High) $= \frac{-P}{P+n}\log_2\frac{P}{P+n} - \frac{-n}{P+n}\log_2\frac{n}{P+n}$

Entropy of S&H=High $= \frac{-0}{0+3}\log_2\frac{0}{0+3} - \frac{3}{0+3}\log_2\frac{3}{0+3}$

Entropy of S&H=High = 0

Entropy of S&H=Normal $= \frac{-2}{2+0}\log_2\frac{2}{2+0} - \frac{0}{2+0}\log_2\frac{0}{2+0}$

Entropy of S&H=Normal = 0

Calculate Average Information Entropy:

$I(\text{Humidity}) = \frac{P_{Humidity} + n_{Humidity}}{P+n}\ \text{Entropy}(\text{Humidity}=\text{High}) + \frac{P_{normal} + n_{normal}}{P+n}\ \text{Entropy}(\text{Humidity}=\text{Normal}$

Date:

$$I(\text{Humidity}) = \frac{0+3}{5}(0) + \frac{2+0}{5}(0)$$

$$I(\text{Humidity}) = 0$$

- Calculate Gain: attribute is Humidity

$$\text{Gain} = \text{Entropy}(S) - I(\text{Humidity})$$

$$\text{Gain} = 0.940 - 0$$

$$\text{Gain} = 0.94$$

Sunny & Windy	Yes	No	Total Entropy
Weak	1	2	3
Strong	1	1	2

$$\text{Entropy of Sunny \& Windy = Weak} = \frac{-p}{p+n}\log_2\frac{p}{p+n} - \frac{n}{p+n}\log_2\frac{n}{p+n}$$

$$\text{Entropy of Sunny \& Windy = Weak} = \frac{-1}{1+2}\log_2\frac{1}{1+2} - \frac{2}{1+2}\log_2\frac{2}{1+2}$$

$$\text{Entropy of Sunny \& Windy = Weak} = 0.9182$$

$$\text{Entropy of Sunny \& Windy = Strong} = \frac{-1}{1+1}\log_2\frac{1}{1+1} - \frac{1}{1+1}\log_2\frac{1}{1+1}$$

$$\text{Entropy of Sunny \& Windy = Strong} = 1$$

Calculate Average Information Entropy:

$$I(\text{Windy}) = \frac{P(\text{Weak})+n\,\text{Weak}}{P+n}\,\text{Entropy}(\text{Windy} = \text{Weak}) + \frac{P(\text{strong})+n\,\text{strong}}{P+n}\,\text{Entropy}(\text{Windy} = \text{strong})$$

$$I(\text{Windy}) = \frac{1+2}{5}(0.9182) + \frac{1+1}{5}(1)$$

$$I(\text{Windy}) = 0.55092 + 0.4$$

$$I(\text{Windy}) = 0.9509$$

$error_{train}(h) < error_{train}(h')$

$error_{\times}(h) > error_{\times}(h')$

Date:

Calculate Information gain for Attribute (A)

$H(S) = 1$

var $A >= 5$

$$H(A >= 5) = -\frac{5}{5+7}\log_2\frac{5}{12} - \frac{7}{12}\log_2\frac{7}{12}$$

$H(A >= 5) = 0.9799$

$$H(A < 5) = -\frac{3}{4}\log_2\frac{3}{4} - \frac{1}{4}\log_2\frac{1}{4}$$

$H(A < 5) = 0.81128$

$$IG_1(A) = H(S) - \sum \frac{|S_v|}{|S|} H(S_v)$$

$$IG_1(A) = 1 - \left[\frac{12}{16} \times 0.9799 + \frac{4}{16} \times 0.81128\right]$$

$IG_1(A) = 0.0622$

Information gain of D

$$H(D >= 1.4) = -\frac{0}{7}\log_2\frac{0}{7} - \frac{7}{7}\log_2\frac{7}{7}$$

$H(D >= 1.4) = 0$

$$H(D < 7.4) = -\frac{8}{9}\log_2\frac{8}{9} - \frac{1}{9}\log_2\frac{1}{9}$$

$H(D < 7.4) = 0.5032$

$$IG_1(D) = 1 - \left[\frac{7}{16} \times 0 + \frac{9}{16} \times (0.5032)\right]$$

$IG_1(D) = 1 - (0 + 0.2830)$

$IG_1(D) = 0.717$

> = 4.2
<= 4.2

Date:

$H(S) = 1$

Information gain of B

var B >= 3

$$H(\text{var } B >= 3) = -\left[\frac{8}{12}\log_2\frac{8}{12} + \frac{4}{12}\log_2\frac{4}{12}\right]$$

$H(\text{var } B >= 3) = 0.9182$

Var B < 3

$$H(\text{var } B < 3) = -\left[\frac{0}{4}\log_2\frac{0}{4} - \frac{4}{4}\log_2\frac{4}{4}\right]$$

$H(\text{var } B < 3) = 0$

$$IG(B) = 1 - \frac{12}{16}(0.9182) - \frac{4}{16}(0)$$

$IG(B) = 0.7070$

Similarly $IG(C) = 0.54880$

$IG(D) = 0.41189$

Information gain of C

var C >= 4.2

$$H(\text{var } C >= 4.2) = -\left(\frac{0}{6}\log_2\frac{0}{6} + \frac{6}{6}\log_2\frac{6}{6}\right)$$

$H(\text{var } C >= 4.2) = 0$

$$H(\text{var } C <= 4.2) = -\left(\frac{8}{10}\log_2\frac{8}{10} + \frac{2}{10}\log_2\frac{2}{10}\right)$$

$H(\text{var } C <= 4.2) = 0.72192$

$$IG(C) = 1 - \left[\frac{6}{16} \times 0 + \frac{10}{16} \times 0.72192\right]$$

$IG(C) = 1 - 0.4512$

$IG(C) = 0.5488$

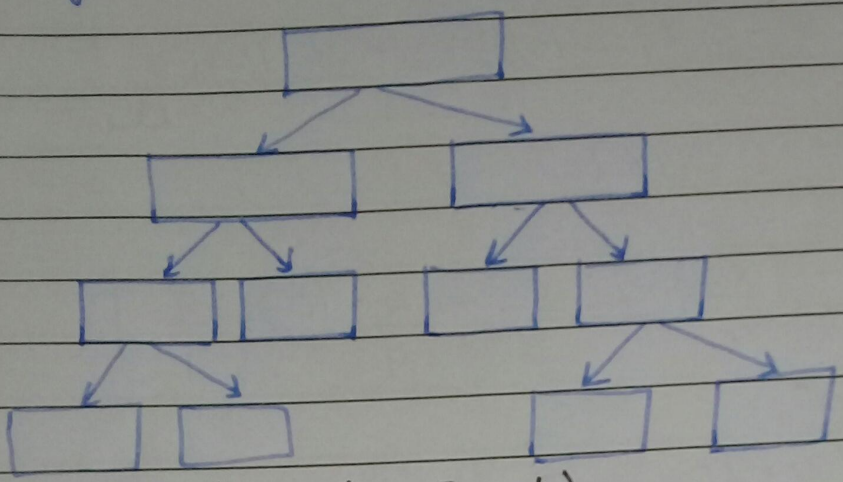
بہت سارے decission Tree استعمال

بہت ساری trees

Date:

RANDOM FOREST:

Building & Evaluating the random forest



1- Bagging (Random Forest)

2- Boosting (Adaboost, Extreme Gradient Boosting)

RSD → Row Selection Data

Row Sample with replacement →

We can re-use data which we copied

→ We use mean as an alternate of Majority Voting in case of continuous Values

→ Random forest is a technique like decission tree

→ ensemble method

بہت ساری trees کو ملا کر بناتے ہیں۔

→ Ensemble method used two techniques

1- Bagging (Bootstrap aggregation) {Random Forest}

2- Boosting (Ada Boost, Extreme Gradient Boosting}

We are going to disscuss bagging technique of Random Forest

Ensemble → means → fuse

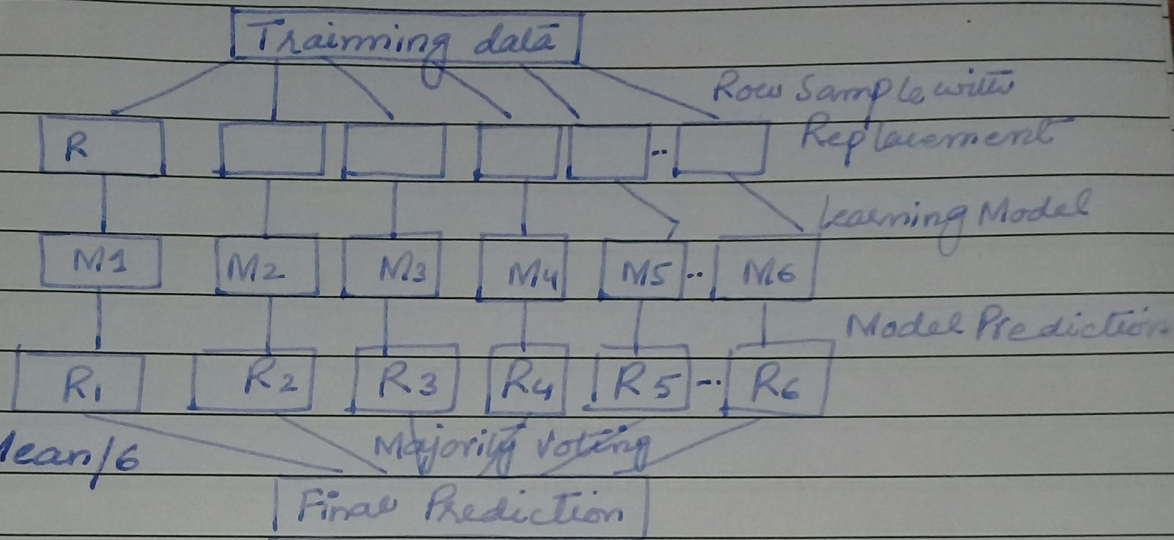
Test data {same} [illegible]

Bagging → تھیلا

Date:

Random Forest – Bagging

→



Bagging is used for both classification & Regression

$$\text{Sensitivity} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}$$

SE = 139/139+32 = 0.8128 = 81%

Cross Validation

Problem over come by cross Validation

The baioness & ambuigity is removed through it

- Holdout method (a single test took out)
- K fold CV
- Leave one out CV
- Bootstrap Method

	Has Heart Dissease	Does Not
Has Heart Disseas	True Positive 139	False Positive 20
Doesn't has HeartD	False Negative 32	True Negative 112

↓ Sensitivity ↓ Specificity

Sensitivity

$$\text{Specificity} = \frac{\text{True Negatives}}{\text{True Negatives} + \text{False Positives}}$$

SP = 112 / 112+20 = 0.8484 = 84%

Rows → corresponds to what was predicted

Columns → corresponds to Known truths

* True positives are patients that had heart disease that were also predict to have heart dissease.
* True Negatives are patients that did not have heart dissease & were predicted not to have heart dissease.
* [illegible]
* False Negative are when a patient has heart dissease but the prediction say they did not
* False Positive are patients that donot have heart dissease but the prediction says that they do

→ We can also calculate Sensitivity & Specificity through Confusion Metrix

	Troll 2	Gore Police	Cool as ice	
Troll 2	(12)	102	(93)	→ FP
Gore Police	112	23	77	TN
Cool as ice	(83)	92	17	

Sensitivity (TROLL 2)

For Troll 2: True Positive Troll 2 / True Positive Troll 2 + False Negatives Troll 2

$$S = \frac{12}{12 + 195} = 0.0579 = 5.79\%$$

Sensitivity (Gore Police)

$$Se = \frac{23}{23 + 194} = 0.11 = 11\%$$

Sensitivity (Cool as ice)

$$Se = \frac{93}{93 + 94} = 0.4973 = 49.73\%$$

Specificity (Troll 2)

For Troll 2:- True Negatives Troll 2 / True Negatives Troll 2 + False Positives Troll 2

$$SP = \frac{209}{209 + 195} = 0.5173 = 51.7\%$$

$$Sp = \frac{205}{205 + 189} = 0.52 = 52\%$$

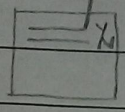
$$Se = \frac{17}{17 + 170} = 0.09$$

$$Sp = \frac{249}{249 + 175} = 0.59$$

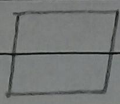
Date:

Clustering

→ Unsupervised learning Introduction:

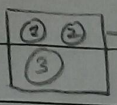


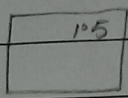
If label then called Supervised learning



If donot label then first clustered & then

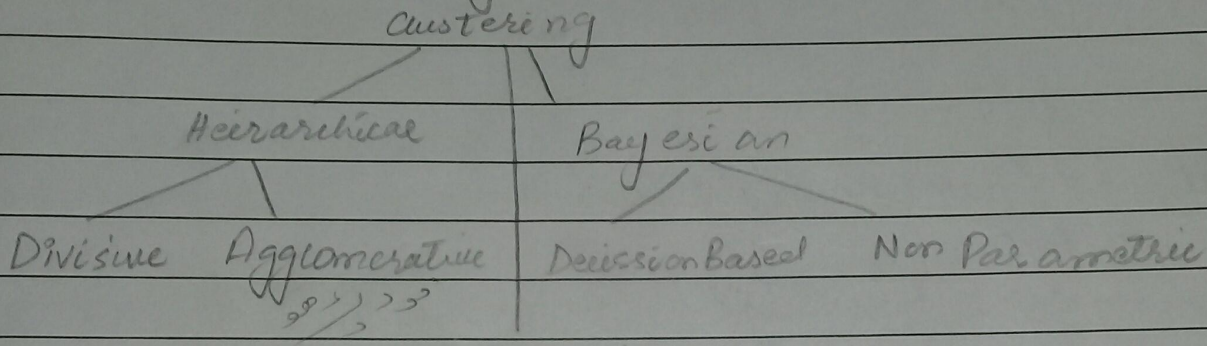
ML





Cluster = same thing

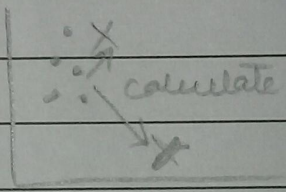
unlabelled → means target is not known



We are going to use centroid

Euclidean distance $d(x_i, x_j) = \sum_{k=1}^{d} (x_i^{(k)} - x_j^{(k)})^2$

Clustring → is not only Method



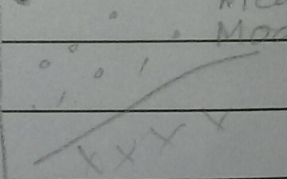
In centroid method Find:

→ Find Initial Seed (Randomly Picked)

→ Calculate distrance from initial seed 1 & seed 2

→ If the distance is less with S1 then belong to S1

When you got first cluster then you took out Mean/Mode & then New centroid will be found then the process repeat iteratively.



Calculate Sum of squared error

10/150 * 100 = 6.67 → 800

	A1 A2 ... An	TV
X1		1
X2		1
⋮		0
Xn		⋮

⇒ Model ⇒ ⇓ Y/No

Date:

correct 140/150 * 100 = 93.3

Variance matrix

$$CM = a^T \cdot a = \begin{bmatrix} 30 & 30 & 0 & -30 & -30 \\ 30 & 10 & 0 & -10 & -30 \\ -30 & 10 & 0 & 0 & 20 \end{bmatrix} \begin{bmatrix} 30 & 30 & -30 \\ 30 & 10 & 10 \\ 0 & 0 & 0 \\ -30 & -10 & 0 \\ -30 & -30 & 20 \end{bmatrix}$$

* Same accuracy will be obtain if we reduce data set.

$$CM = a^T \cdot a = \begin{bmatrix} 3600 & 2400 & -1200 \\ 2400 & 2000 & -1400 \\ -1300 & -1400 & 1400 \end{bmatrix} = \begin{bmatrix} MM & MP & ME \\ PM & PP & PE \\ EM & EP & EE \end{bmatrix}$$

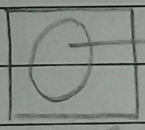
+ve: directly proportional

-ve: Inversely proportional

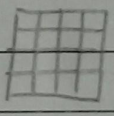
1- How to reduce the attributes?

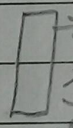
2- Whenever attributes will be reduced leads to effect the accuracy.

3- This also used as feature extraction



→ discrete format Transformation → Sampling → Quantization

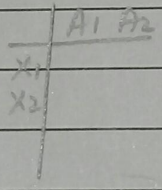


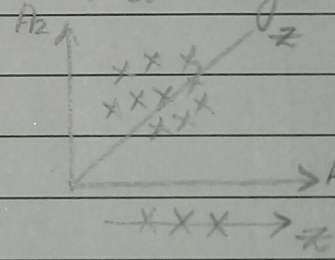


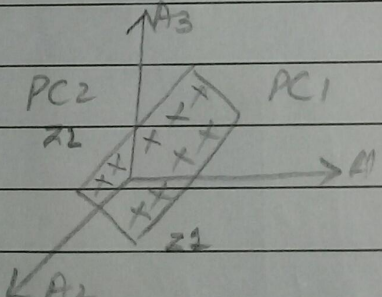
→ Feature Vector

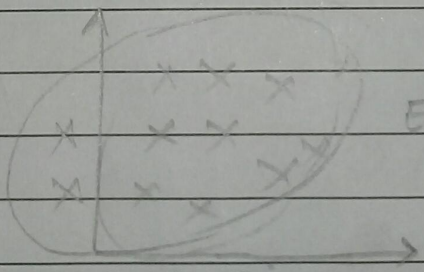
→ Feature extraction

→ dimention reduction

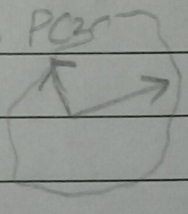




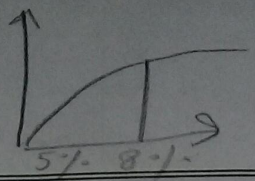




Cartisean space



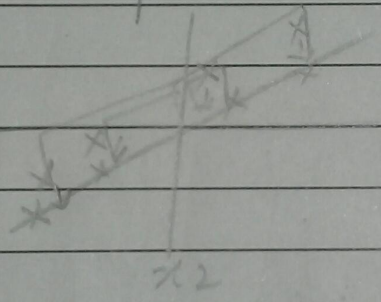
9000 X 5



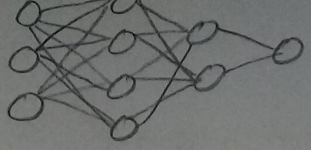
Date:

→ How can we move from one space to another?
→ The co-ordenates will be same after moving to other space?
Yes, the co-ordenates will remain same.
→ In those, 9000 which are important PC's?
→ How to pick PC's?
* Eigen space is a collection of principle components
* We used Principal Component Analysis for Entering into a new space.

PC1
Orthogonal → independant ⅀ 90° angle



Plot the attribute on the line at the angle of 90°
Then take distance for each 'X' to the origin
Then plot different PC's for finding best PC1
Highest Distance = PC1
Technical Word = Map Honay k bad jiska Variance ziada Hoga wo pick Hoga

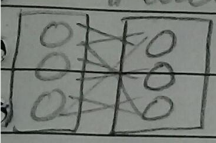


ML Lecture

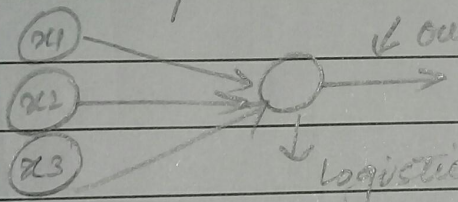
Date:

* Neural Networks learning it's own features
→ Can we get a one unit neural network to compute this goal

Input Y → Dendrite
Output Y → Axon
Nucleus
| Cell of Body



→ features recieving from a source
→ We generate layers
→ Neuron perform specific tasks
→ Preprocessing is done through the layers
→ Neuron is also called "Active Function"
* Feed input via input wires
* Logistic unit does computation
* Sends output down



input wires

Edges are called "Weight"

Date:

Covariance of Multi Dimensional Model

→ Covariance produces the relationship between two or more random variables.

Suppose

$$A = \begin{bmatrix} 90 & 80 & 40 \\ 90 & 60 & 80 \\ 60 & 50 & 70 \\ 30 & 40 & 70 \\ 30 & 20 & 90 \end{bmatrix}$$

Score of 3 subjects
S - students

$$\begin{bmatrix} \overline{300} & \overline{250} & \overline{350} \\ 300/5 & 250/5 & 350/5 \end{bmatrix}$$ Total size

$$\begin{bmatrix} 60 & 50 & 70 \end{bmatrix}$$ Mean score

To calculate the deviation matrix : a

$$a = A - [1]A \cdot \left(\frac{1}{5}\right)$$

unit matrix ~ 4 unit matrix

$$a = \begin{bmatrix} 90 & 80 & 40 \\ 90 & 60 & 80 \\ 60 & 50 & 70 \\ 30 & 40 & 70 \\ 30 & 20 & 90 \end{bmatrix} - \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} 90 & 80 & 40 \\ 90 & 60 & 80 \\ 60 & 50 & 70 \\ 30 & 40 & 70 \\ 30 & 20 & 90 \end{bmatrix} \cdot \left(\frac{1}{5}\right)$$

$$a = \begin{bmatrix} 90 & 80 & 40 \\ 90 & 60 & 80 \\ 60 & 50 & 70 \\ 30 & 40 & 70 \\ 30 & 20 & 90 \end{bmatrix} - \begin{bmatrix} 60 & 50 & 70 \\ 60 & 50 & 70 \\ 60 & 50 & 70 \\ 60 & 50 & 70 \\ 60 & 50 & 70 \end{bmatrix}$$

$$a = \begin{bmatrix} 30 & 30 & -30 \\ 30 & 10 & 10 \\ 0 & 0 & 0 \\ -30 & -10 & 0 \\ -30 & -30 & 20 \end{bmatrix}$$

-ve: Inversely Proportional

$$\frac{\sqrt{(x_i - \bar{x})^2 + (Y - \bar{Y})^2}}{n-1}$$

Date:

+ve directly proportional

Co-Variance Matrix

$$CM = a^T \cdot a = \begin{bmatrix} 30 & 30 & 0 & -30 & -30 \\ 30 & 10 & 0 & -10 & -30 \\ -30 & 10 & 0 & 0 & 20 \end{bmatrix} \begin{bmatrix} 30 & 30 & -30 \\ 30 & 10 & 10 \\ 0 & 0 & 0 \\ -30 & -10 & 0 \\ -30 & -30 & 20 \end{bmatrix}$$

$$CM = \begin{bmatrix} 3600 & 2400 & -1200 \\ 2400 & 2000 & -1400 \\ -1200 & -1400 & 1400 \end{bmatrix} = \begin{bmatrix} MM & MP & ME \\ PM & PP & PE \\ EM & EP & EE \end{bmatrix}$$

$$\begin{bmatrix} 30\times30 + 30*30 + 0\times0 + -30\times-30 + -30\times-30 \\ 30\times30 + 10\times10 + 0\times0 + -10\times-10 + -30\times-30 \end{bmatrix}$$