Unit-3

Cluster Analysis

BASIC CONCEPT of clustering

- Cluster → nothing @ just grouping similar data

Clustering → It refer to group the entire data into similar form & other group data will always diff

[Example
→ E-commerce site decide it entire customer into 3 cluster
1 cluster : "Discount lovers"
2 cluster : "Premium Buyer"
3 cluster : "one-time buyer"
] Here site cluster into 3 customer

"Measures of similarity"

- Euclidean Distance

$$d(P,Q) = \sqrt{\sum (P_i - Q_i)^2}$$

- Manhattan Distance

$$d(P,Q) = \sum |P_i - Q_i|$$

- Cosine Similarity

$$\text{Similarity}(x,y) = \frac{x \cdot y}{\|x\| \|y\|}$$

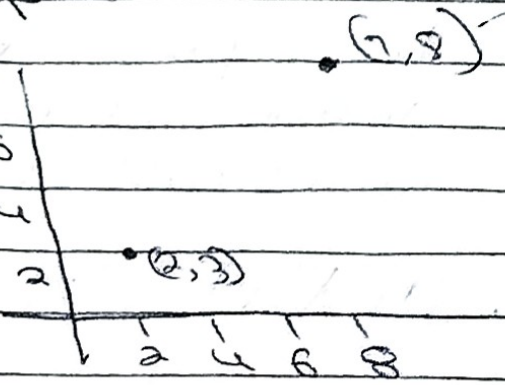
- Jaccard similarity

$$\text{Jaccard}(A,B) = \left| \frac{A \cap B}{A \cup B} \right|$$

Euclidean distance

↳ let understand by example

let a food app;
customer address point (2,3)
Restaurant 11 point (7,8)



so by Euclidean distance we find shortest

So the shortest distance, $d = \sqrt{\Sigma (x_i - y_i)^2}$

$= d = \sqrt{(7-2)^2 + (8-3)^2} = \sqrt{50} = \boxed{7.07}$

Other type question euclidean distance

↳ if we have four diff. cluster so one new person come with (x,y) so which cluster is suitable for new one

Real Example → A mall decide his customer on the Basis of Age(X) and income(Y) so new customer come with age 28 & 42K income so tell which cluster is good

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

	age	Inc
C1	25	40
C2	30	45
C3	45	80
C4	50	85

clusters

⇒ Cnew = (28, 42)

⇒ distance from C1

$d = \sqrt{(28-25)^2 + (42-40)^2} = 3.6$

distance from C2

$d = \sqrt{(28-30)^2 + (42-45)} = 3.6$

distance from C3

$d = \sqrt{(28-45)^2 + (42-80)^2} = 41.6$

distance from C4

$d = 48.3$

so C1 & C2 cluster is good for Cnew

Manhattan Distance

(yen euclidean se alag h yah seedha straight line me distance ni krke yaha block me hota h)

let understand by example

let suppose you're in grid-type city and your location is (2,3) & your friend (7,8) so shortest straight horizontal distance?

$d(X,Y) = |x_1 - x| + |y_1 - y|$

$= |7-2| + (8-3) = 5+5 = 10$ block

Actual use :-

can service actual road distance ni Rahne kelyuse krti ha (jaha city grid system ho vaha)

Real life example

let suppose you are in restaurant origin point and you need to go diff location and the city is grid so find all distance

Restaurant location R(2,3)

customer location
C1 (5,7)
C2 (8,8)
C3 (8,2)

distance b/w Res to C1

$|5-2| + |7-3| = 7$

distance b/w Res to C2

$|8-2| + (8-3) = 11$

dis b/w Res to C3

$|8-2| + |2-3| = 12$

(हम किसी भी 2 पॉइंट की दूरी को निकाल सकते हैं जैसे रेस्टोरेंट से C2 जाना है सीधा तो 6 unit लगेंगी)

Page No.
Date

Ques from PYQ :-

=) Given two object $x = (22, 1, 42, 10)$ and $y = (20, 0, 36, 8)$

(1) Euclidean distance

$$d = \sqrt{(20-22)^2 + (0-1)^2 + (36-42)^2 + (8-10)^2}$$

$$d = \sqrt{(2)^2 + (1)^2 + (6)^2 + (2)^2} = \sqrt{4+1+36+4} = \sqrt{45}$$

(2) Manhattan Distance

$$d = |20-22| + |0-1| + |36-42| + |8-10|$$

$$= 2 + 1 + 6 + 2 = 11$$

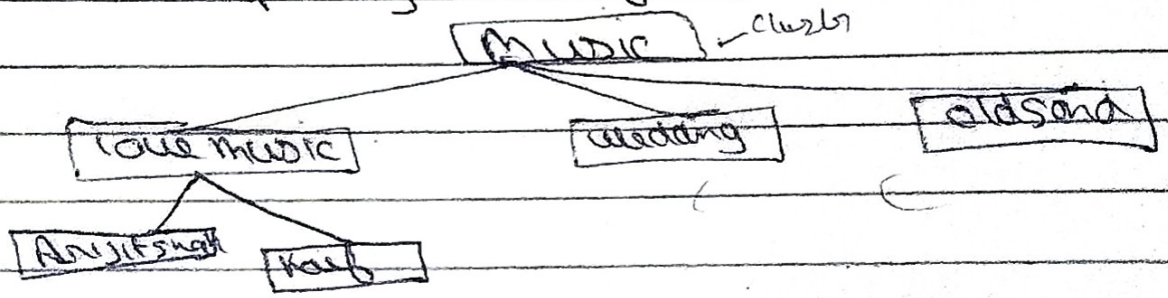
Type of clusters

① Partitioning method =>
=> These method divide the dataset into K partition (as cluster) where each partition represent a cluster
Real life eg:- customer devide into diff group (cluster)
cluster
K-Mean^ => Assign each point to the nearest cluster centroid and iteratively update centroid
K-Medoids (PAM - Partitioning Around Medoids):
Similar to K-Mean but choose actual data point as centroids, making it robust to outliers

② Hierarchical Methods
=> Hierarchical clustering is unsupervised technique where entire data are arranged in hierarchical or tree structure (dendrogram)

each node represent a cluster and leaf of node is data

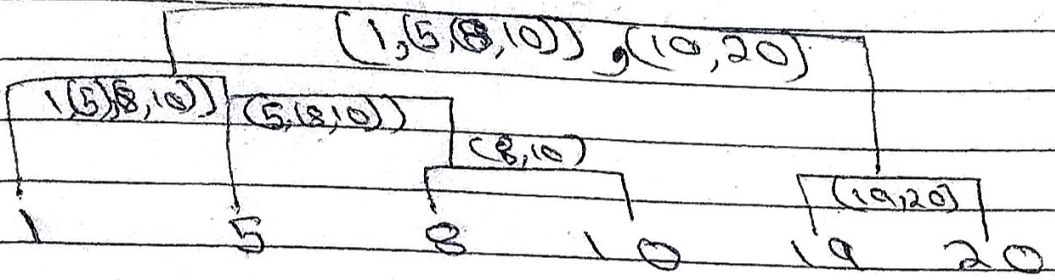
Example by tree of hierarchical method (Divisive)



Agglomerative Algorithm

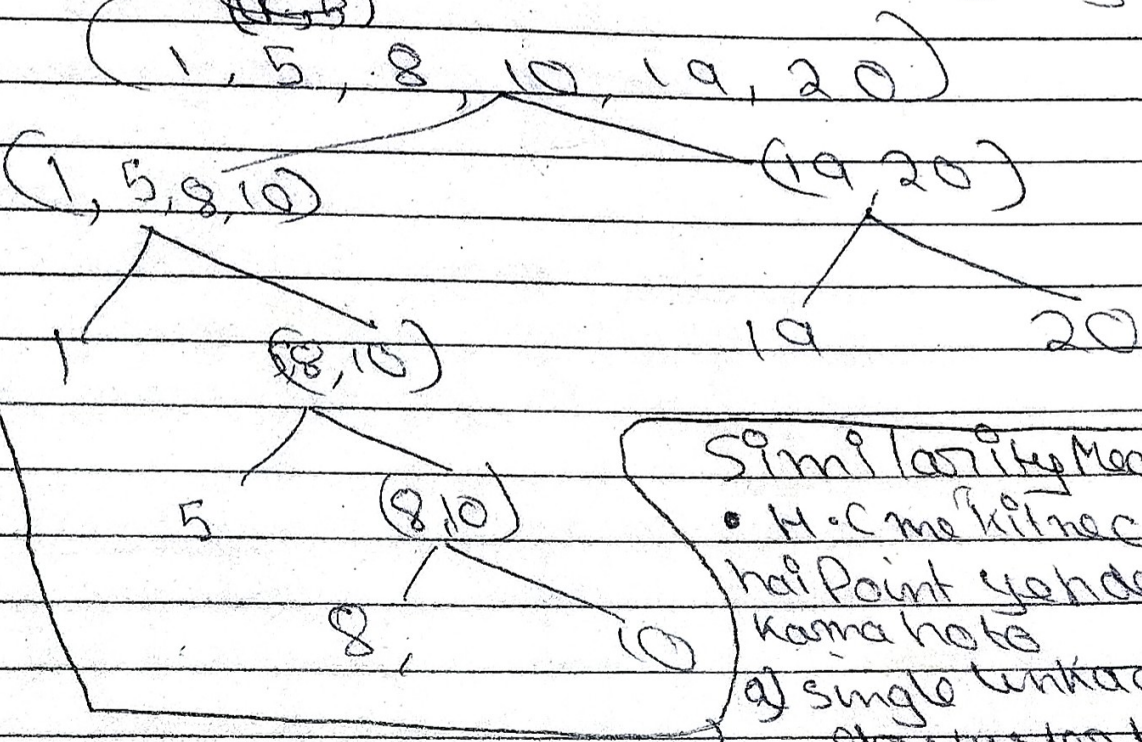
Let understand by example,
in spotify user create playlish of love song
so user 1 spotify suggest love song like to cluster

=> It start from Bottom to top



divisive

=> It start from top to bottom initially Big cluster



Similarity Measure
- H.C me kitne close" hai Point yeh decide karna hota
a) single linkage: do cluster kitne member kadise
b) complete linkage

3) Density based method
= Cluster are formed based on the density of data point
key algorithm
DBSCAN → Density Based Spatial clustering of Application with Noise) : finds clusters of arbitrary shape & identifies noise

Page No.
Date

Ques on K-mean

customer	Income (X)	Spending score (1-100)
✓ C1	2	10
✓ C2	4	12
✓ C3	5	25
✓ C4	10	30
✓ C5	11	35

Cluster 1
C1, C5

Cluster 2
C2, C3, C4

let
C1
C1

from C1
$C3 = \sqrt{(5-2)^2+(25-10)^2}$
$= \sqrt{(3)^2+(15)^2} = \sqrt{234} = 15.29$

C3 from C2 $= \sqrt{(5-4)^2+(25-12)^2}$
$\sqrt{(1)^2+(13)^2} = \sqrt{170} = 8$

So C3 from C2 is short so C3 goin C2
now centroid point of cluster2 $\Rightarrow \frac{4+5}{2}, \frac{25+12}{2} = 4.5, 18.5$

C4 from C1 $= \sqrt{(10-2)^2+(30-10)^2}$
$\Rightarrow \sqrt{(8)^2+(20)^2} = \sqrt{464} = 21.54$

C4 from C2 $= \sqrt{(10-4.5)^2+(30-18.5)^2}$
$\sqrt{(5.5)^2+(11.5)^2} = \sqrt{162.5} = 12.74$

So C4 from C2 is short so C4 go in C2
now centroid point of cluster2 $= \frac{4.5+10}{2}, \frac{18.5+30}{2}$
$= 7.25, 24.25$

C5 from C1
$\sqrt{(11-2)^2+(35-10)^2}$
$= \sqrt{(9)^2+(25)^2}$
$= \sqrt{81+625}$
$= \sqrt{706} = 26.570$

C5 from C2
$\sqrt{(11-7.25)^2+(35-24.25)^2}$
$\sqrt{(3.75)^2+(10.75)^2}$
$= \sqrt{(14.0625)+(115.5625)}$
$= 29.625$

So C5 from C1 is short so C5 is go to C1
now cluster point of C1 $= \frac{11+2}{2}, \frac{35+10}{2}$
$= 6.5, 22.5$

measure for cluster validation
↳ It mean that cluster (group) we make are they really meaningful or not

Type of cluster validation

(1) Internal validation
= cluster ka evaluate karte ha dataset ke andar ke hi info se

- Internal validation check
1) Ek How much cluster point closely & how much cluster different
Eg → let suppose we have 100 people dataset where height weight and you make 3-cluster and run K-mean
now internal validation say cluster 1 people height weight variation (good data) if cluster 1 & cluster 2 ek dusre se alag hoga to (good seperate

- External validation
⇒ let you have student dataset of percentage you make 3 cluster weak, Avg, Strong but you have actual label that is achieve you now compare the actual label with cluster label
Rand index → tell how much cluster label is close the actual label

Relative validation
1) do make K=3 and K=4 and now check which is more useful

Page No.
Date 6 oct 25

DATA MINING →

Unit → 4

Association Rule mining: Transaction data-set, frequent itemset, support measure, rule generation, confidence of association rule, Apriori algorithm, Apriori principle

- Transaction data-set → A Transaction Data-set is a collection of transaction where each transaction represent a set of item bought by a customer at a specific time

How transaction Data-set look like

Transaction ID	Item Bought	Each Row → one customer shopping basket
T_1	Bread, Milk	
T_2	Bread, Diaper, Beer, Egg	Each "Item Bought" = The set of Product Purchased Together
T_3	Milk, Diaper, Beer, Coke	
T_4	Bread, Milk, Diaper, Beer	
T_5	Bread, Milk, Diaper, Coke	

This entire table ⇒ Transaction Data-set

Real life example/use

⇒ This transaction Data set have hidden feature which are useful for mall, company find best use full Pattern from transaction data set

Frequent itemset → is a group of item that appear together frequently in a transaction data

In short items are bought together many time, they form a Frequent itemset

Support → Mean how frequently an item (or item) appear in the entire data set
↳ out of all transaction, in how many transaction does this item appear

$$\text{Support} = \frac{\text{no. of transaction containing } X}{\text{no. of total transaction}}$$

From our previous data set we find support

Item	Appearin	support (count/5)	support %
Bread	4	4/5	80%
Milk	4	4/5	80%
Diaper	4	4/5	80%
Beer	3	3/5	60%
cola	2	2/5	40%
Egg	1	1/5	20%

What is Support Threshold (minimum support)?
when we analyzed big data, we can't consider every small item we only want frequent item that appear many time so we decide a minimum Support threshold [If support 7/60% we call it frequent
so after apply minimum support the frequent is Bread, Milk, Diaper Beer 7/60%

Support for itemset (Two item Together)

Itemset	Appear in	support	support %
{Bread, Milk}	T1, T4, T5	3/5 = 0.6	60%
{Milk, Diaper}	T3, T4, T5	3/5 = 0.6	60%
{Diaper, Beer}	T2, T3, T4	3/5 = 0.6	60%
{Bread, Diaper}	T2, T4, T5	3/5 = 0.6	60%
{Bread, Beer}	T2, T4	2/5 = 0.4	40%

why we not make cola, egg like comb. coz they are already less than 60%.

Interpretation → all two item set appear ~~more~~ together 60% of all Transaction except one combination that's {Bread, Beer}

Rule Generation

→ Rule Generation mean creating if-then rule from the frequent itemsets we discovered
In short: tells how one item is related to another from other frequent itemsets.

- {Bread, Milk}
- {Milk, Diaper}
- {Diaper, Beer}
- {Bread, Diaper}

For itemset {Bread, Milk} we can generate the following rule

(1) Bread → Milk — If a person buys Bread, they are likely to buy Milk.
~~Milk → Bread~~
(2) Milk → Bread — If a person buys Milk they are likely to buy bread

kuch nahi a Rule Generation me bus hame combination ko if then me likhna hai

Confidence measure / confidence of association

Confidence tell us how often the rule has been found true

eg -> when a customer buy A good so how much time customer by B good

$$\text{Confidence}(A \to B) = \frac{\text{support}(A \cup B)}{\text{support}(A)}$$

let take a example

eg:- {Bread} -> {milk}

meaning "if a customer buy bread, they are also likely to buy milk"

A = Bread
B = Milk

$$\text{Confidence}(A \to B) = \frac{\text{support}(A \cup B)}{\text{support}(A)}$$

Calculate support value

support(Bread) -> How many transaction have bread

T_1, T_2, T_4, T_5 -> 4 transaction

$$= \frac{4}{5} = 0.8$$

support(Bread ∪ Milk) -> How much transaction

$T_1, T_2, T_4, T_5 = 4$

$$= \frac{4}{8} = 0.8$$

$$\text{Confidence} = (\text{Bread} \to \text{Milk}) = \frac{0.8}{0.80}$$

$$= 100\%$$

$$\text{confidence}(\text{Milk} \to \text{Diaper}) = \frac{s(M \cup D)}{s(M)}$$

$$= \frac{0.3}{0.40} = 0.75$$

Apriori Algorithm

⇒ Apriori Algorithm is used to find frequent itemsets (group of item that often appear together in transaction)

A shopkeeper have 5 transaction as cond

T_1 = Tea, Sugar, Milk ; T_2 = Tea, Sugar ; T_3 = Tea, Milk
T_4 = Sugar, Milk , T_5 = Tea, Sugar, Milk

Goal - To find out which item are frequently bought together and make strong association like if a customer buy Tea they are likely to buy sugar

ROUGH Page.

Step by step of apriori Algorithm

let take minimum support = 60%.

That mean

Minimum count ⇒ 60% of 5 = 3 transaction

So, any item or combination that appear in at least 3 transaction is consider frequent

Step-2 find Frequent 1-itemset (L1)

Item	Count	Support	Frequent	
Tea	4	80%	✓	all above the minimum
sugar	4	80%	✓	
milk	4	80%	✓	

L_1 = {Tea}, {Sugar}, {Milk}

Step-3 Generate candidate 2-item

Candidate	count	support	Frequent
{Tea, Sugar}	3	60%	✓
{Tea, milk}	3	60%	✓
{Sugar, milk}	3	6%	✓

*All 3 pair are frequent since each appear in >/ 3 transaction

L2 = {Tea, Sugar}; {Tea, milk}; {Sugar, milk}

Step 4 Generate Candidate 3-item

Candidate	count	support	Frequent
{Tea, Sugar, milk}	2	40%	X

Step 5 Final frequent itemset

L1 => {Tea}; {Sugar}; {milk}

L2 => {Tea, Sugar}; {Tea, milk}; {Sugar, milk}

Step 6 generate association rule

1) {Tea} → {Sugar} 2) {Sugar} → {Tea}

(3) {Tea} → {milk} (4) {milk} → {Tea}

(5) {Sugar} → {milk} 6 {milk} → {Sugar}

Step 7 Calculate Confidence

{Tea} → {Sugar}

confidence(Tea → Sugar) = S(T ∪ S) / S(S) = 0.6 / 0.8 = 0.75

Confidence = 0.75

mean → whenever customer buy Tea there is a 75% chance they also buy Sugar

Step 8: Real-life Business use (Optional)
Shopkeeper learn from this:
- Customer who buy Tea also buy sugar (75%)
- Customer who buy Milk also buy sugar 60%

So he can:
- keep Tea, Sugar, Milk together on the shelf
- offer combo discounts:
  "Buy Tea & get ₹5 off on Sugar"

This increase sale and convenience

Apriori Property

All subsets of frequent itemset must also be frequent

Ex → If {Tea, Sugar, Milk} was frequent then {Tea, sugar}, {Sugar, Milk} and {Tea, Milk} must also be frequent

DATA MINING
Unit => 5
Classification

Classification=> Naive Bayes classifier, Nearest Neighbour classifier, decision tree overfitting, confusion matrix, evaluation metrics and model evaluation

=> Classification-> divide the entire data into different categories based on some feature using trained model

ONE Real life example
trained model make a model for email where two cat. spam and not spam so we give some keyword like if email contain win a phone now/ cong. you won 1000 this type message so it will go in spam else non-spam

In Short
↳ Classification => predicting a category label (not a no) using data pattern

Regression no numeric output data h eg (house price)
classification ne categorical output data h eg predicting house type

Naive Bayes classifier

=> It's based on Bayes' Theorem

$$P(class/Data) = \frac{P(\frac{Data}{class}) \times P(class)}{P(DATA)}$$

If $P(\frac{spam}{email}) > P(\frac{not\ spam}{email})$, so email is spam

What "Naive" mean

=> completelely independent, which mean two data/email aint similar

Real-life Example

Problem :- Predict which email is spam which is not spam on the basis of two word -> ["offer" & "win"]

Training Data

Email	"offer" word	"win"	class [spam / non spam]
E1	Yes	Yes	Spam
E2	Yes	No	Spam
E3	No	Yes	Spam
E4	Yes	No	Not Spam
E5	No	No	Not Spam

$P(spam) = \frac{3}{5}$, $P(not\ spam) = \frac{2}{5}$

Step 2

Conditional Probabilities from Data:

Condition

P(offer = Yes | Spam) = 2/3

P(win = Yes | Spam) = 2/3

P(offer = Yes | Not Spam) = 1/2

P(win = Yes | Not Spam) = 0/2 = 0

Step-3 new email com your win and existing both so

For spam

$P(\text{spam} \mid \text{offer, win}) \propto P(\text{offer} \mid \text{spam}) \times P(\text{win} \mid \text{spam}) \times P(\text{spam})$

$= \frac{2}{3} \times \frac{2}{3} \times \frac{3}{5} = \frac{4}{15} = 0.266$

for not spam

$P(\text{not spam} \mid \text{offer, win}) \propto P(\text{offer} \mid \text{not spam}) \times P(\text{win} \mid \text{not spam}) \times P(\text{not spam})$

$\Rightarrow \frac{1}{2} \times 0 \times \frac{2}{5} = 0$

P(Spam) > P(not spam)

So mail is spam.

another new email we offer you a car

for spam =

$P(\text{spam} \mid \text{offer}) \propto = P(\text{offer} \mid \text{spam}) \times P(\text{spam}) = \frac{2}{3} \times \frac{3}{5}$

$P(\text{not spam} \mid \text{offer}) \propto = P(\text{offer} \mid \text{not spam}) \times P(\text{Not Spam})$

$\Rightarrow \frac{1}{2} \times \frac{2}{5} = 20\% \quad 40\%$

So mail is spam

Nearest Neighbour classifier (K-NN)

⇒ Take the decision by watch neighbour data

e.g:- If 10 people and your classmate people take a cricket jar (3-4 people) so by (K-NN) you will also say i am cricket jar.

Real life example

Customer	Like Shoes	Like Jeans	Class
A	✓	✓	Fashion lover
B	✓	✗	Fashion lover
C	✗	✓	Fashion lover
D	✗	✗	Non Fashion lover

Now Customer come E which are like shoes & not like jeans so E is nearest to B so E will put into Class Fashion lover where B

If numeric data so we used Euclidean

Customer	Income	Credit Score	Loan Allowed
A	25	300	No
B	50	600	Yes
C	45	580	Yes
D	20	200	No

new customer
E
Income = 40
Score Credit = 590

Distance calculate for E(40, 590)

Distance(E, B) = 14.14

Distance(E, C) = 11.18 — Most nearest so E come in C group

Distance(E, A) = 290

Distance(E, D) = 391

decision tree overfitting

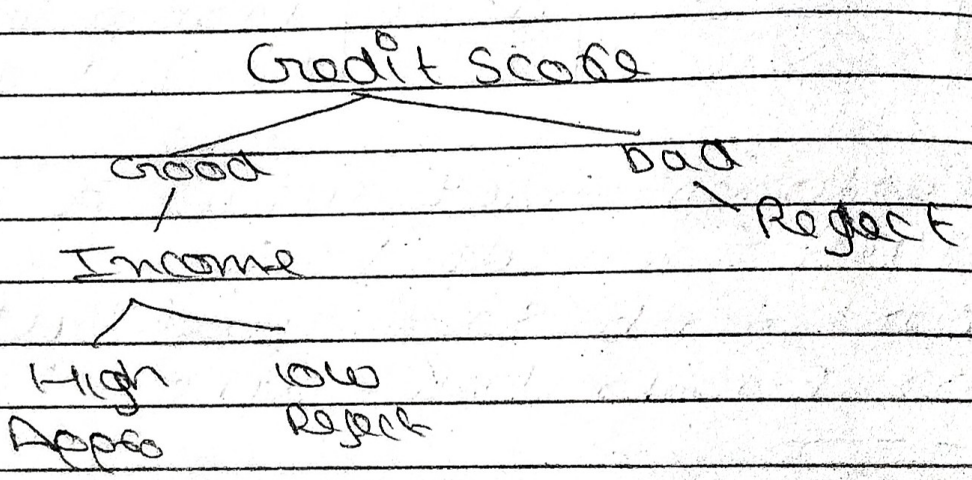
Decision Tree classifier

- Decision Tree is a supervised learning algorithm which do both classification as well as prediction

How Tree structure looks?

- Root node → Started point
- Branches → conditions
- Leaf/Nodes → Final decision/output

Let make a loan approval system for bank we have Income (High/low), Credit score (Good/Bad) Age (old/young)



Confusion Matrix

Confusion Matrix is a table where we see how much prediction are correct and how much wrong. They provide deep insight. They tell us which type of mistake model done

Real life example

We make a machine learning model who predict which customer is good for loan approval

So, possible result

- Yes
- No

Model prediction & actual result both have 4 possible combination

Confusion matrix Structure

	Predicted Yes	Prediction No
Actual Yes	True Positive	False Negative
Actual No	False Positive	True Negative

Real life meaning of TP, FP, FN, TN

	meaning	loan Example
True Positive	Model ne correctly "Yes" bola	Actual loan-worthy person ko model ne allow kara
True Negative	Model ne correctly "no" bola	Actual risky person ko model ne correct kare
False positive	Model ne galat "Yes" bola	Risky person ko model ne galti se approve karay
False negative	Model ne galti se no bola	loan worthy person ko model ne galat reject kar

Actual Yes	45 (Tr. Posit)	15 (False-ve)	60
actual No	5 (False+ve)	35 (T. -ve)	

Yeh na actual 60

Example

We have 100 people data where 60 people actually loan-worthy & 40 is non-loan worthy

Model ne predict Kiya:

- 50 Ko approve kiya 50 Ko reject

	Predict Yes	Predict No
Actual Yes (60) →	45 (TP)	15 (FN)
Actual No (40) →	10 (FP) (5)	30 (TN)

Now calculate Metric

Matric	Formula	Meaning
Accuracy	(TP+TN) / Total = $\frac{75}{100}$ = 75%	model overall sahi kitni baar
Precision (-ve predict)	$\frac{TP}{TP+FP}$ = $\frac{45}{45+10}$ = 81.8%	Jine Ko approve kiya unme se kitne sachme loan worthy
~~True ratio~~ Tre-ute	$\frac{Tn}{Tn+FP}$	
Recall (Sensitivity)	$\frac{TP}{(TP+FN)}$ = $\frac{45}{45+15}$ = 75%	Jitne sachme loan worthy th unme se kitne ko model ne approve kiya
F1 Score	~~2 x Precision x~~ $\frac{2 \times Precision \times Recall}{(Precision + Recall)}$ = $\frac{2 \times (81.8\% \times 75\%)}{81.8\% + 75\%}$ ⇒ 78%	Balance b/w Precision & Recall

Yeh sab bhi imp hai

consider the class ka email

200 [80 spam] are u P col

no. 60

100 are not spam — no more machine learning 20

Example you build a machine model to predict whether person has diabetes or not

Yes → diabetes perso

No → non-diabetes

100 patient

② 80 application Actual 50 → loan-worthy

30 → non loan worthy

ML = 55 loan-worthy

25 loan non-loan

confusion matrix

	ML Yes Prediction yes 55	Prediction No 25	
Actual yes 50	45	5	50
Actual No 30	10	20	30

kuch bhi bana sakta h

$$\frac{2 \times P \times R}{P + R} \rightarrow 2 \times Tn$$

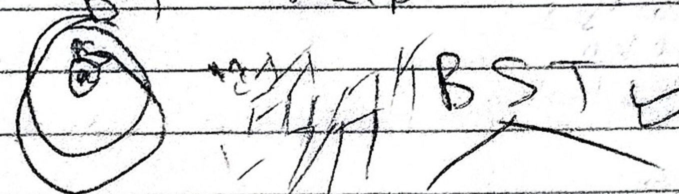
RR BST [illegible]

H.o.w.

Bf = L − R — Balanced

2 − 2 = 0 — 6 % coo. D ata

B.T Help



BST — log n

Left me small, Rme Bad

AC = 100

APM = 80

(20)

−1, 0, +1 ✓ AVL

Bf = (height of LST − h of RST)

−1, 0, 1 & no limit

Bal Rea

Evaluation Metrics →

Q Yes → A bank want to predict whether a customer should get a loan or not (Yes/No) Using an ML model

actual yes → 12 Prediction Yes = 11
actual No → 8 Prediction No = 9

Confusion Matrix

	Pred. Yes 11	Pred. No 9	
A. Yes 12	10 TP	2 FN	12
A. No 8	1 FP	7 TN	8
			20

Evaluation Matrix

Accuracy = $\frac{TP+TN}{TP+TN+FP+FN} = \frac{9+7}{20} = \frac{17}{20} = 85\%$

(ML kitne sahi)

Precision $= \frac{TP}{TP+FP} = \frac{10}{11} \times 100 = 90.90\%$

Yes bola me model kitna sahi ka

Recall $= \frac{TP}{TP+FN} \Rightarrow \frac{10}{10+2} = \frac{10}{12} \times 100 = 83\%$

Jo not no actual yes me se kitne ko sahi me yes kaha

F1-Score

$$\frac{2 \times \text{Precision} \times \text{Recall}}{(\text{Pre} + \text{Re})} = \frac{2 \times 90.90\% \times 83\%}{90.90\% + 83\%}$$

$$\frac{1.5099}{1.6634} = 0.90710$$

$$= 90\%$$

Q-1 Comparision b/w K mean and DBscan

=> K-mean is centroid based algorithm in K mean Data are divided into K no cluster

DB scan Density Based algorithm
DBscan have two thing / parameter
ε (epsilon) -> maximum distance to consider neighbor & minpoint minimum no of neighbor to form a dense region
-> where dense area so cluster

Q-2 How to determin optimal no of cluster
=> Aim -> To Identify how much K are naturally present if K less so cluster vague & mixed if K more so unnecessary splitting
most common method
Elbow Method
-> Here we try diff value of K eg K 1 to 10
then calculate within cluster sum (for every K cluster) square sum
WCSS = Σ (of K = 1 to 10)^2
then make WCSS & K graph

K	WCSS
1	1000
2	500
3	250
4	230
5	220

-> less distance