Data Science :-

Data Science is study of data to extract meaningful insights for business.

→ It is a multidisciplinary approach that combines fields of mathematics, statistics, artificial intelligence & Computer Engineering to analyze large amount of data.

→ Data Science is about data gathering, analysis and decision making.

→ Data Science is about finding patterns in data through analysis & make future predictions.

→ Data Science combines statistical tools, methods, & technology to generate meaning from data

Why Data Science :-

→ Because we have too many data such as money, reviews, customer data & people working etc.

→ We want to make it clear & easy to understand.

→ Data analysis let people make better decision either faster & better.

History of Data Science —

→ The roots of data science can be traced back to 1960s & 1970s when CS & Statistics began to intersect.

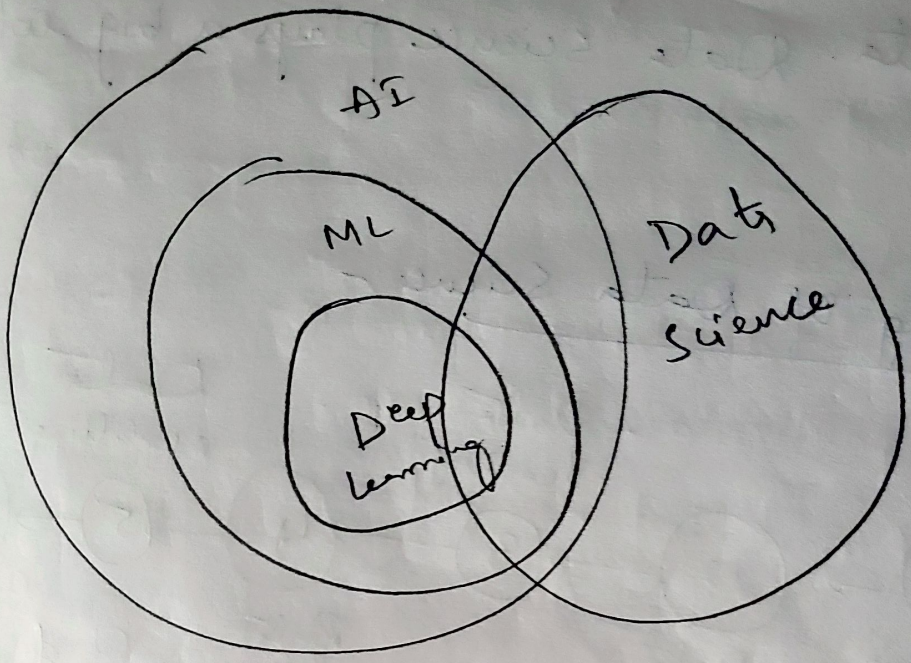
→ In 1962, John Tukey described a field he called "data analysis", which resembles modern data science.

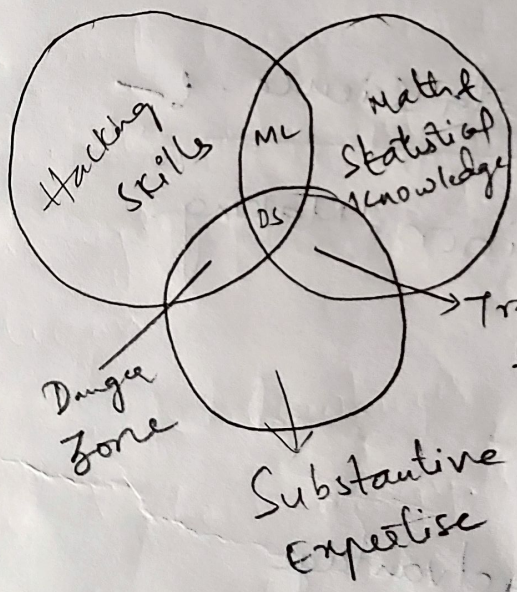
→ In 1985, in a lecture given to Chinese Academy of Sciences in Beijing, C.F. Jeff Wu used the term "data science" for the first time as an alternative name for statistics.

→ while William S. Cleveland is credited with data science as separate discipline in modern era.

→ Peter Naur proposed it as a replacement for the term "Computer Science. Hence the word data science was invented by Peter Naur.

Venn diagram for AI





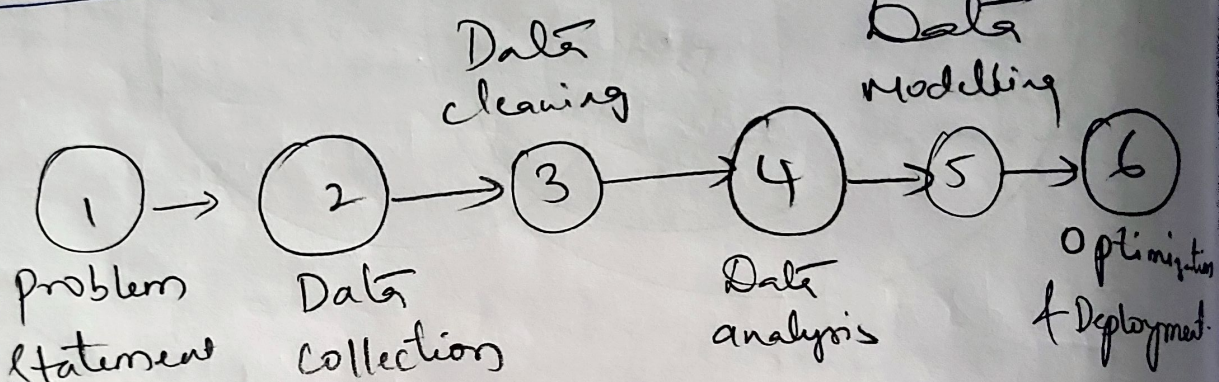
Venn diagram for data science

⇒ In a world of data space where Organizations deal with petabyte and exabyte of data, the era of Bigdata emerged.

→ It was a great challenge & concern for industries for storage until 2010, But now framework like Hadoop & other solved them.

→ And, then the focus shifted to processing of data. Data science plays a big role here.

Working of Data Science



Advantages of Data Science :-

1. Improved Decision-Making
2. Cost effective
3. Innovation
4. Competitive advantage
5. Time Saving

Role Of Data Scientist :-

→ Data Scientists help organizations understand and handle data, and address complex problems using knowledge from a range of technology niches.

Tools for Data science :-

1. R
2. Python
3. SQL
4. Hadoop.
5. Tableau.
6. Weka.

Big Data & Data Science Hype :-

Big Data

→ Huge data is a term used to describe a gigantic amount of both organized and unstructured knowledge that is so immense using traditional database and programming methods is impossible to handle.

Types of Big data :-

→ There are three types of data behind Big Data

1. Structured
2. Semi-structured &
3. Unstructured.

Structured

↳ It mean data that can be processed, Stored and retrieved in a fixed format.

Ex:- Employee details in organization.

Unstructured:-

↳ It alludes to information that doesnot have a particular structure.

ex:- Email, text, image & video.

Semi-Structured:-

↳ It is a Combination of both Structured and unstructured

ex:- XML files.

→ Big Data refers to vast volumes of data generated at high velocity from a variety of sources.

→ The data is characterized by three V's.

1. volume
2. velocity
3. variety.

1. Volume :- Big data involves large datasets that are too complex for traditional data processing tools to handle. The data can range from terabytes to petabytes of information

2. Velocity :- Big Data is generated in real-time or near-real time, requiring fast processing to extract meaningful insights.

3. variety :- The data comes in multiple form. like. Structured, semi-structured & unstructured data.

Difference between Big Data & Data science :-

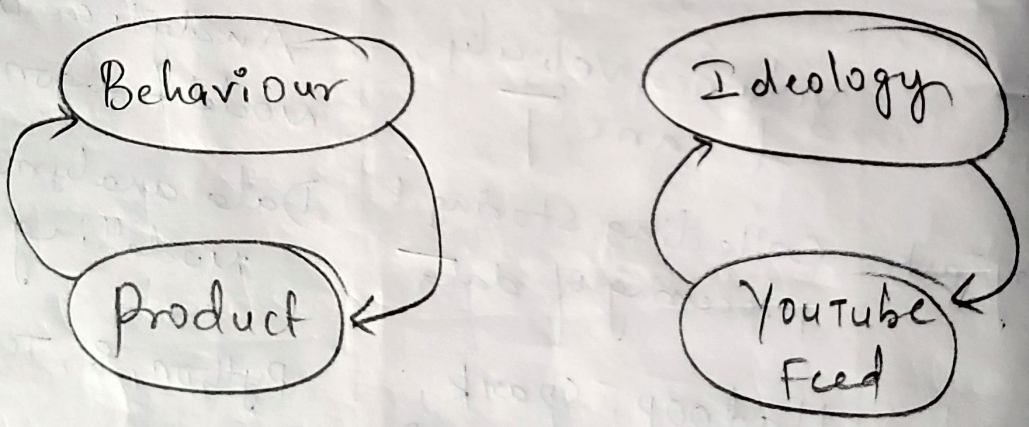
Aspect	Big Data	Data science
Definition	Handling & processing vast amount of data	Extracting insights & knowledge from data
objective	Efficient Storage, Processing & management.	Analyzing data to inform decision & predict trends.
Focus	volumes, velocity, & variety	Analytical method model & algorithm.
Primary Task	Collecting, storing & processing of data	Data analysis & modelling.
Tools	Hadoop, Spark, NoSQL (MongoDB)	Python, R, Tensorflow
Data types	Structured, semi & unstructured.	Processed & clean data.
Skill set.	Data Engineering & distributed computing.	Statistical analysis & ML programming
Typical Roles.	Data Engineers & Big Data Analysts	Data Scientists, ML Engineers.

Why Now :-

→ There is availability of massive amount of data from many aspects of our lives. There is abundance of inexpensive computing power.

→ All our activities online such as shopping, communication, news, search records etc are tracked.

→ These massiveness of data makes it very interesting as well as challenging. Often data itself is building blocks of data products — Amazon Recommendation Systems, Facebook friend Recommendations etc.



→ Behavioural Data of people changes product. The product changes the behavior of people.

→ Large Scale of data processing, increased memory & bandwidth & cultural acceptance of technology in our lives make it possible unlike a decade ago.

Getting the past Hype—

→ Data science enables companies not only to understand data from multiple sources but also to enhance decision making.

⇒ As a result, data science is widely used in almost every industry, including health care, finance, marketing, banking etc.

→ Around all the hype, in other word, there is a truth that it is something new. but, it's a fragile (easily broken & damaged).

Data classification—

→ Data classification is process of organizing data into categories for its most effective and efficient use.

→ Data classification in Data science refers to process that tags and categorize any kind of data so, that it can be better understood and analysed.

→ Datafication is a process of "taking all aspect of life and turning them into data".

→ Datafication is an interesting Concept and led us to Consider its importance with respect to people's intentions about sharing their own data.

→ We are being datafied, or rather our actions are, when we "like" something on Oxaline, we are intending to be datafied

→ when we walk around in a store, or even on the street, we are being datafied Completely unintentional way via, CC, Google glasses etc.

Current Landscape of Datascience :—

→ It is a new field, focused on the process and Systems that enables the organization to extract Knowledge or Insight from data & translate it into actions.

→ which emphasizes Data Science as an applied area capable of generating results in organization of different Categories & sizes.

Introduction to R

→ R Language Stands out as a powerful tool in modern era of statistical computing & data analysis.

→ R programming is a leading tool for machine learning, Statistics and data analysis, allowing for easy creation of objects, functions and packages.

→ Designed by Ross Ihaka & Robert Gentleman at university of Auckland, New Zealand

→ R is platform Independent and open-source

R-features:-

1. Comprehensive Statistical Analysis.
2. Advanced Data visualization (ggplot2 & lattice)
3. Extensive package & libraries.
4. Open Source & Free.
5. platform Independence.
6. Integration with other languages (C, C++, Python, Java)

7. powerful Data Handling & storage:
   ↳ efficiently handle & store data supporting various types like Vector, Matrix, data.frame & lists.

8. Robust community & support.

9. Interactive Development Environment (IDE)
   ↳ R-studio.

Variables and Datatypes in R :-

Variables :- Rules
   ↳ '_' & '.' → are allowed
   ↳ always start with alphabets
   ↳ No special characters

Contains Constants predefined.

Pi — pi

Letters — a, b, c ... x, y, z.

LETTERS → A, B, C, ... X, Y, Z.

Months in a year → month.name,
month.abb

Basic datatypes

datatypes	values
1) Logical	True/false
2) Integer	Set of all intigers, Z.
3) Numeric	Set of all real numbers
4) Complex	Set of Complex numbers
5) character	"a", "b", "c" ---- "x", "y", "z", "@", "#", "$" --- etc.

1⇒ To find data type of object → typeof()

2⇒ verify if object is of a certain datatype.

ex:- is.data_type(object)

is.integer().

3) Convert the datatype of object to another.

as.datatype(object)

ex:- as.logical()

ex)

typeof(1)
↳ double

typeof("22-1-2000")
↳ character

is.character("21-11-2000")
↳ True

is.character(as.Date("22-11-2000"))
↳ false.

as.complex(2) → 2+0i

as.numeric("a") → NA.

Basic Objects

1. Vector → ordered collection of same data type.
2. List → ordered collection of objects
3. Dataframe → Generic tabular object.

Vectors

ex: x = c(2.3, 4.5, 6.7, 8.9)
print(x).

List:- ↳ could be vector, matrix, complex vectors, character arrays, function & so, on.

List ex:

ID = c(1, 2, 3, 4)
emp.name = c("Man", "Rag", "sha", "Din")
num.emp = 4.
emp.list = list(ID, emp.name, num.emp)
print(emp.list).

[[1]]
[1] - 1 2 3 4
[[2]]
[1] "man" "rag" "sha" "din"

Accessing list comp :

emp.list = list("Id" = ID, "Names" = emp.name, "Total staff" = num.emp)
print(emp.list$Names).
o/p:- [1] "Man" "Rag" "sha" "Din"

print (emp.list[1])
↳ [1] 1, 2, 3, 4

print (emp.list[2])
↳ $Names → Man Rag Sha Din

print (emp.list[[1]][1])
↳ 1.

print (emp.list[[2]][1])
↳ Man.

Manipulating Lists.

emp.list["Total stuff."]=5
emp.list[[2]][5]="abc".
emp.list[[1]][5]=5.
print (emp.list).

Concatination of lists

↳ c function : c(list1, list2).

emp.ages= list("ages"= c(23, 24, 54, 30, 32))
emp.list = c(emp.list, emp.ages).
print (emp.list)

Data frames:-

→ Data frame are generic data objects of R, used to store tabular data.

ex:

```
vec1 = c(1,2,3)
vec2 = c("R", "scilab", "Java")
vec3 = c("for prototyping", "for prototypes", "for scaleup")
df <- data.frame(vec1, vec2, vec3)
print(df)
```

↳ function and pass each vector as argument to function

Op:-

Vec1	vec2	vec3
	R	for prototyping
1	scilab	for prototyping
2	Java	for scaleup
3		

```
r1 <- str(df)
```

↓ function to retrieve structure of df

print(r1)

→ It can be created by importing data from file.

```
newDF = read.table(path = "path of file")
```

↓ location of file

→ A separator can also used to distinguish between entries. Default separator is space, ' '

en: `newDF = read.table(file = "path", sep)`

↓ tab/etc.

Accessing rows & columns :-

df[val1, val2] → rows. val1 & col-val2.

ex:- print(df[1:2]) → accessing 1st & second row

↳ O/P:-

	vec1	vec2	vec3
		R	For prototyping
1	1	Scilab	1
2	2		

print(df[, 1:2]) → accessing 1st & 2nd column

↳ O/P:-

	vec1	vec2
		R
1	1	Scilab
2	2	
3	3	Java.

print(df[1:2]) → accessing 1st & 2nd column.

↳ O/P:-

	vec1	vec2
		R
1	1	Scilab
2	2	
3	3	Java

df$vec1
1 2 3

df['vec2']
↳ R, Java C

Subset() :-

↳ It extracts subset of data based on conditions.

Ex:-

```
pd = data.frame("Name" = c("senthil", "abc", "xyz", "rr"),
"month" = c("Jan", "Feb", "Jan", "Feb"),
"BS" = c(141.2, 139.3, 135.2, 160.1),
"Bp" = c(90, 78, 80, 81)).
pd2 = Subset(pd, Name == "senthil" | BS > 150)
```

print(pd2)

Name	Month	Bs	Bp
Senthil	Jan	141.2	90
pqr	Feb	160.1	81.

Editing Dataframes ←

print(df) → O/P:-

df[[2]][2] = "R".

print(df) → O/P:-

	vec1	vec2	vec3
1		R	For prototyping
2		R	for prototyping
3		Java	For scaleup

→ A dataframe can also be edited using edit command

→ create an instance of dataframe and use edit command to open a table editor, changes can be made manually.

Ex:- myTable = data.frame()

myTable = edit(MyTable).

Adding extra rows & columns :—

- Extra row by "rbind"
  Extra col by "cbind"

ex:-
```
df = rbind(df, data.frame(vec1 = 4,
vec2 = "c", vec3 = "forscaleup"))
print(df)
df4 = cbind(df, vec4 = c(10, 20, 30, 40))
print("adding extra col")
print(df4)
```

→ nrow(df) no of row
ncol(df) " col

Delete row & columns :—

```
df2 = df[-3, -1]
print(df2)
```

→ '-' sign before value
& befor ',' for rows &
after ',' for col's.

o/p:-

	vec3	vec4
vec2		
R	Forfootoyuty	10
Scilab	4	20
C	Forscaleup	40

Conditional deletion :—

```
df(3) = df[, !names(df) %in% c("vec3")]
```

→ means no to row/col's which statisfy the cond'n.

o/p:-

	vec1	vec2	vec3
1	1	R	10
2	2	Scilab	20
3	3	Java	30
4	4	C	40

~~Statistical Inference~~

```
df(4) = df[!df$vec1 == 3, ]
print(df4)
```

↳ O/P :-

vec1	vec2	vec3	vec4
1	R	For prototyping	10
2	Scilab	4	20
4	C	For scaleup	40

Manipulating rows - the factor issue :-

→ when character columns are created in data.frame, they become factors

→ A factor variable are those where the character column is split into categories or factor levels.

ex:- df[3,1] = 3.1
df[3,3] = "Others"
print(df)

O/P :-

↳ warning message:
invalid factor level, NA generated

vec1	vec2	vec3
1.0	R	For Prototyping
2.0	Scilab	For Prototyping
3.1	Java	<NA>

Resolving factor issues:-

```
vec1 = c(1,2,3)
vec2 = c("R", "scilab", "Java")
vec3 = c("For prototyping", "for prototyping",
         "for scaleup")
df = data.frame(vec1, vec2, vec3, stringAsFactors = F)
df[3,3] = "other"
print(df)
```

↳ O/P :-

	vec1	vec2	vec3
1	1	R	For prototyping
2	2	scilab	For prototyping
3	3	Java	Others.

Control Structures :-

if-else →

```
x = 6
if (cond^n)
x > 7
{ x = x+1
}
else if (cond^n)
x > 5
{
x = x+2
}
else
{
x = x+3.
}
```

Sequence function.

↳ One Components of for loop.

→ seq (from, to, by, length).

→ Create equi-spaced points b/w from to

seq(from, to, by, length)
↓ ↓ ↓ ↓
Start end increment/ No. of element
decrement required

ex:- seq(from=1, to=10, by=3)

o/p :- 1, 4, 7, 10

seq(from=1, to=10, length=4)

1 4 7 10.

seq(from=1, to=10, by=4)

1 5 9. seq(from=1, to=10, length.out=5)

= 1.00 3.25 5.50 7.75 10.00

for loop. ↳ equally spaced.

for (iter in sequence)
↳ list/vector)
{ // statements
}

ex: n=5
sum=0
for(i in seq(1,n,1))
{ sum = sum+i
print(c(i,sum))
}

o/p

loop variable	sum
1	1
2	3
3	6
4	10
5	15

for loop with break

```
n=100
sum=0
for(i in seq(1,n,1))
{
  sum=sum+i
  print(c(i,sum))
  if(sum>15)
  {
    break
  }
}
```

o/p:-

loop	sum	cond^n
1	1	F
2	3	F
3	6	F
4	10	F
5	15	F
6	21	T.

while loop (is used whenever you want to execute stmt until specific cond^n is violated).

```
sum=0
i=0
fin_sum=15
while(sum<fin_sum)
{
  i=i+1
  sum=sum+i
  print(c(i,sum))
}
```

1	3	T
2	6	T
3	10	T
4	15	F

Data visualization of R :-

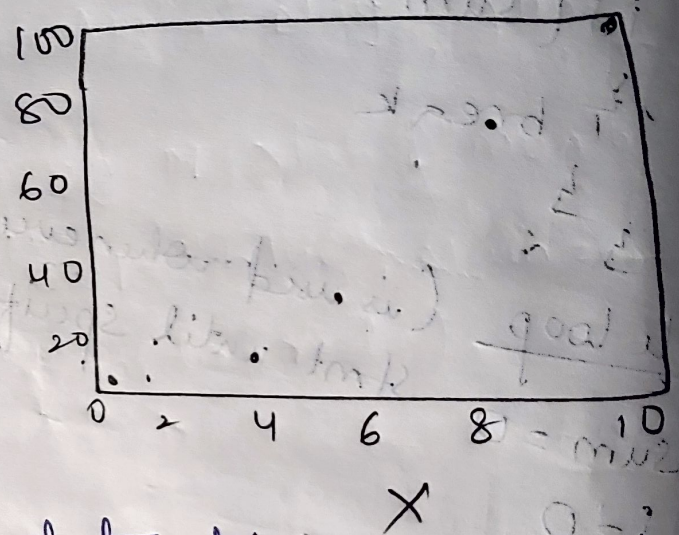
Basic graphics

→ Scatter
→ Line
→ Bar

Scatter plot :-

R-code

```
X = 1:10
Y = X^2
plot(Y)
```



→ Let us Illustrate dataset for scatter plot
dataset 'mtcars'

A dataframe with 32 observations on 11 variables

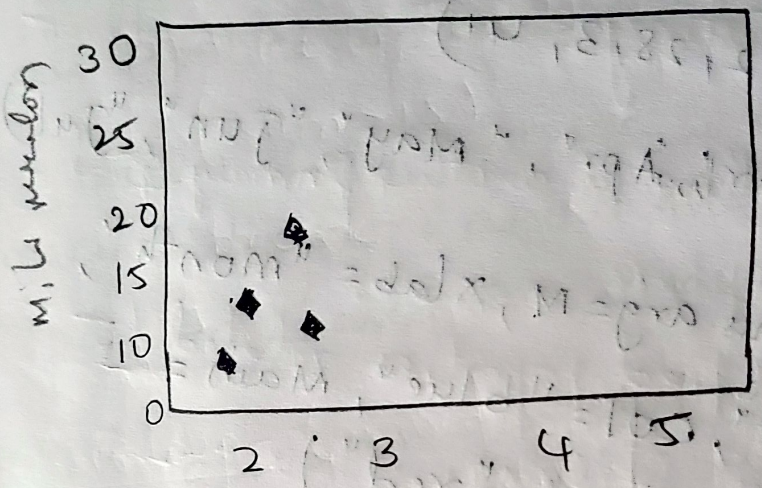
11 variables :-

mpg → miles/gallon
cyl → No of cylinders
disp → Displacement
wt → wt (1000lbs)
drat → Rear axle ratio
hp → Gross horse power
vs → V/s
qsec → 1/4 miletime

```
plot(mtcars$wt, mtcars$mpg,
  main = "Scatterplot Example", Xlab = "Car wt",
  Ylab = "Miles per Gallon", pch = 19)
```

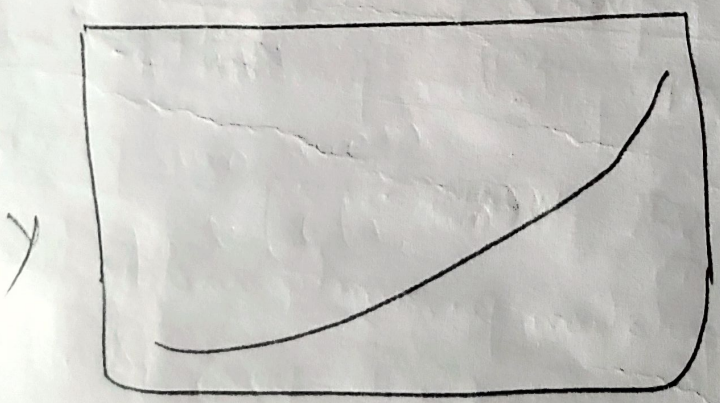
↓ corresponds to diff shapes for points

Scatterplot Eg:-



Lineplot :-

```
X = 1:10
Y = X^2
plot(X, Y, type = 'l')
```



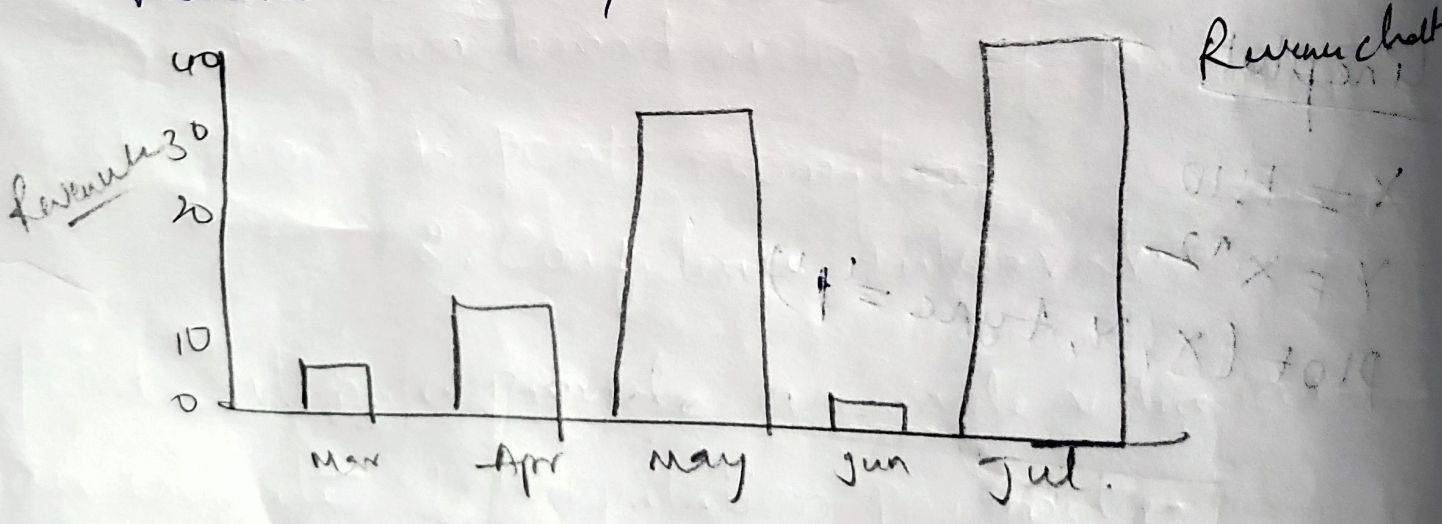
Bar plot

Syntax

barplot(H, names.arg, xlab, ylab, main, names.arg, col)

R-code

```
H <- c(7, 12, 28, 3, 41)
M <- c("mar", "Apr", "May", "Jun", "Jul")
barplot(H, names.arg=M, xlab="months",
ylab="Revenue", col="blue", Main=
"Revenue chart", border="red")
```



Statistical Inference :-

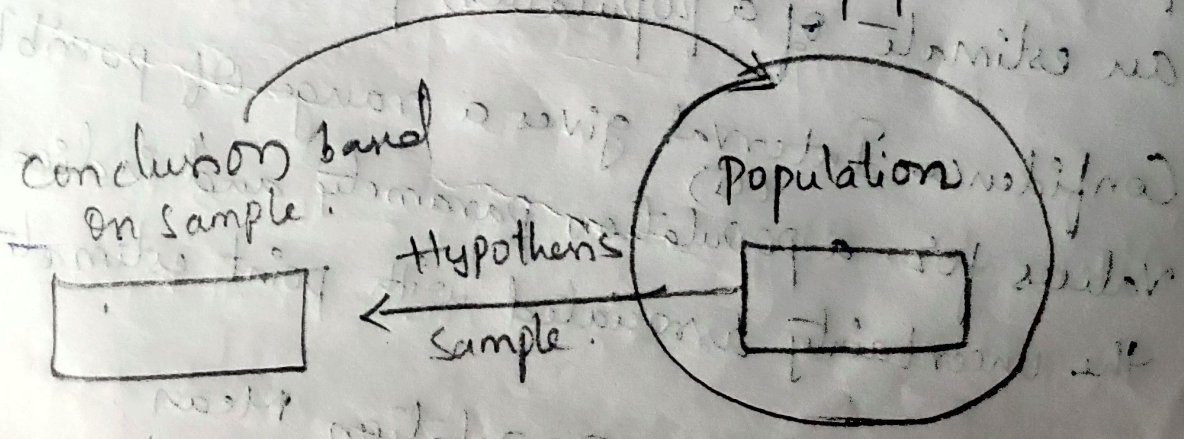
→ Statistical Inference is the process of analysing the result and making Conclusion from data subject to random Variation.

→ Hypothesis testing and Confidence intervals are application of statistical Inference.

→ Statistical inference is a method of making decisions about the parameters of a population, based on random sampling.

→ It is based on probability theory and probability distribution. It involves is making assumptions about the population and the sample using statistical models to analyse the data.

Branches of ~~Statistical Inference~~ :-



Branches of Statistical Inference :—

There are two main branches of statistical Inference

1. parameter Estimation
2. Hypothesis Testing.

Estimate :- is specific observed numerical value used to estimate an unknown population parameter

Parameter Estimation is another primary goal of Statistical inference.

→ They are quantified properties related to population you are studying.

There are two broad methods of parameter Estimation
1. point estimation
2. Confidence Interval.

Point estimate provide a single value as an estimate of a population parameter.

Confidence Interval (CI) gives a range of possible values for a population parameter and indicate the uncertainty associated with point estimate

Ex:- Estimating population Mean.

```
# Generate a sample dataset
Set.seed(123)
heights <- rnorm(50, mean=165, sd=15)
```

Calculate the Sample mean & standard deviation.

mean_height <- mean (heights).

sd_height <- sd (heights)

n <- length (heights).

→ Construct a 95% (CI) :-

We can calculate 95% Confidence Interval for mean height using t.test() function.

$$\bar{x} \pm z \cdot \frac{\sigma}{\sqrt{n}}$$

A random Sample of $n = 50$ males showed a mean average daily products equal to 756 grams with a Standard deviation of 35 grams. find 95% and 99% Confidence interval for population average μ ?

$\bar{x} = 756$ grams. $n = 50$, $\sigma = 35$. $z_{(95\%)} = 1.96$

$$I.E = 756 \pm 1.96 \cdot \frac{35}{\sqrt{50}}$$

$$746.30 \leq \mu \leq 765.70.$$

$z_{(99\%)} = 2.576$

$$756 \pm 2.576 \cdot \frac{35}{\sqrt{50}}$$

$$743.23 \leq \mu \leq 768.77.$$

Hypothesis Testing :-

→ An Hypothesis is an assumption or idea Specifically a statistical claim about an unknown parameter.

→ Hypothesis testing is a statistical method that is used to make a statistical decision using experimental data.

→ It evaluates two mutually exclusive statements about population to determine whether the statement is best supported by sample data.

Null hypothesis (H_0) :- It is a basic assumption made based on the problem knowledge

Ex:- A company's mean production is 50 unit/day

$H_0 : \mu = 50$

Alternative Hypothesis (H_1) :- The alternate Hypothesis is contraverse to null hypothesis

Ex:- A company's production is not equal to 50 units/day

$H_1 : \mu \neq 50$.

Procedure:-

1. Identify the parameter of interest, which you wish to test.
2. Construct the null & alternative hypothesis
3. Compute a test statistic which is a func^n of the sample set of observation.
4. Derive the distribution of test statistic under the null hypothesis assumption.
5. Choose a test criterion against which the test statistic is compared to reject/not reject the null hypothesis

→ No hypothesis test is perfect. There are inherent error since it is based on observation which are random.

→ The performance of a hypothesis test depend on

1. Extent of variability in data.
2. No of observation (sample size)
3. Test statistics
4. Test criterion (threshold)

Two sided & one-sided test

Two sided test

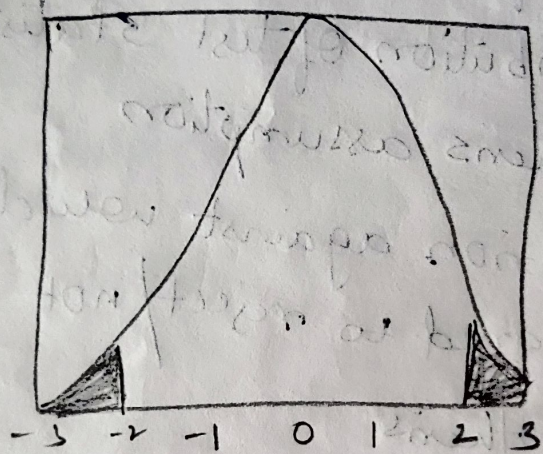
- $H_0: \mu = 0$
- $H_1: \mu \neq 0$

one sided test

$H_0: \mu = 0$

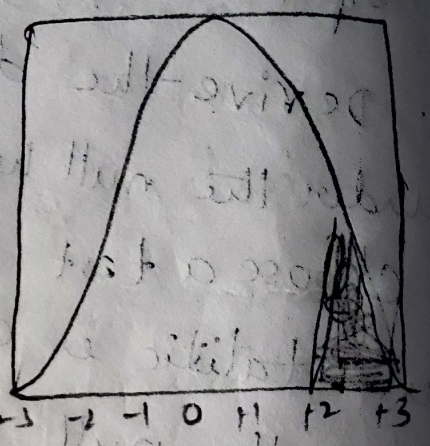
$H_1: \mu > 0$

- Test statistic standard normal Rv z.



if $z \leq -2$ or $z \geq 2$

Reject H_0



if $z \geq 1.5$

Reject H_0

Errors in Hypothesis Testing:-

Two types of error (Type I & Type II)

Decision ⇒ Truth ↓	H_0 is not reject	H_0 is rejected
H_0 is true	Correct Decision $Pr = 1-\alpha$	Type I error $Pr = \alpha$
H_1 is true	Type II error $Pr = \beta$	Correct decision $Pr = 1-\beta$

Test for Difference in mean:-

Two groups of teachers of similar capabilities are trained by two methods A and B. Is Method B more effective than method A?

10 teachers in each group.

Group 1 $\bar{x}_1 = 70$, $s_1 = 3.3665$.

Group 2 $\bar{x}_1 = 74$, $s_1 = 5.3955$.

Null Hypothesis $= \mu_1 - \mu_2 = 0$

Alternative hypothesis: $\mu_1 - \mu_2 < 0$.

Testing statistic.

$$t_1 = \frac{\bar{x}_1 - \bar{x}_2}{\sqrt{\frac{s_p^2}{N_1} + \frac{s_p^2}{N_2}}} \sim t_{N_1+N_2-2}$$

↳ total - 2 parameter

$10+10-2$

$= 18$ ↳ degree

$$s_p = \frac{(N_1-1)s_1^2 + (N_2-1)s_2^2}{N_1+N_2-2}$$

$$s_p = \frac{(10-1)(3.665)^2 + (10-1)(5.39)^2}{10+10-2}$$

$$s_p = \frac{120.56 + 261.46}{18}$$

$$s_p = 21.22$$

$$\frac{70 - 74}{\sqrt{\frac{(21.2)^2}{10} + \frac{(21.2)^2}{10}}} \approx$$

$$= \frac{-4}{\sqrt{\frac{44.9}{10} + \frac{44.9}{10}}}$$

$$t = \frac{-4}{\sqrt{2.11 + 2.11}} = \frac{-4}{2.11}$$

$$t = \sim 1.89$$

Critical value $\alpha = 0.05$ is -1.73

Decision :- Reject Null Hypothesis

(Method B is better)

→ Depending upon the type of data availability & size, different types of hypothesis testing are used to determine whether null hypothesis can be rejected / not.

$$Z = \frac{\bar{x} - \mu}{\frac{\sigma}{\sqrt{n}}}$$

$\bar{x}$ is simple mean

μ is population mean

σ — Standard deviation

n — size of sample.

* $t = \frac{\bar{x} - \mu}{s/\sqrt{n}}$, s is sample standard deviation

* $\chi^2 = \sum \frac{(O_i - E_i)^2}{E_i}$, O_i is Observed value, E_i is expected value.

Z-test

- used for a large sample size ($n \geq 30$).
- used to determine there is a diff b/w the population mean & sample mean

One sample: $z = \frac{\bar{x} - \mu}{\frac{\sigma}{\sqrt{n}}}$

Two sample $z = \frac{(x_1 - x_2) - (\mu_1 - \mu_2)}{\sqrt{\frac{\sigma_1^2}{n_1} + \frac{\sigma_2^2}{n_2}}}$

T-test:-

- used for sample size is small ($n < 30$).
- used to compare the sample mean & population mean.

One sample: $t = \frac{\bar{x} - \mu}{\frac{s}{\sqrt{n}}}$

Two sample $t = \frac{(\bar{x}_1 - \bar{x}_2) - (\mu_1 - \mu_2)}{\sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}}$

→ The mean average wt of men is greater than 100kg with a standard deviation of 15 kgs. 30 mens are chosen with avg wt of 112.5kg using hypothesis testing, check if there is enough evidence to support the researcher's claim. If C.I is 95%.

Sol $H_0 : \mu = 100$

$H_1 : \mu > 100$.

As this one-tail → $\alpha = 100\% - 95\%$

$= 5\%$

Critical value $= 1 - \alpha$

$= 1 - 5\%$

$= 0.95$

Using normal distribution table $z = 1.645$ for 0.95

Z test

$$z = \frac{\bar{x} - \mu}{\frac{\sigma}{\sqrt{n}}} \Rightarrow \mu = 100, \bar{x} = 112.5$$

$n = 30, \sigma = 15$

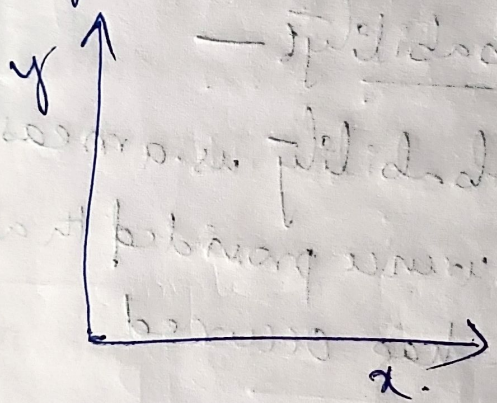
$$z = \frac{112.5 - 100}{\frac{15}{\sqrt{30}}} = 4.56$$

As $4.56 > 1.645$ ∴ Reject null Hypothesis

Statistical Modelling :-

→ Statistical Modeling is a process used to understand, interpret and predict data. It involves the creation of mathematical representation of the relationship b/w different variables in dataset

* Let us assume two variable x & y, to define the functional relationship b/w them



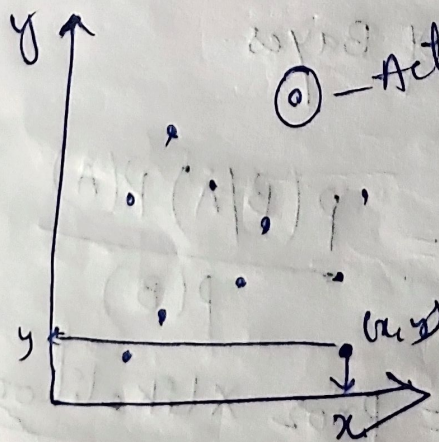
$y = f(x)$

$y = 3 + 2x$

$x = 2$

$y = 7$

→ Process of finding approximate value of y depending on x.



⊙ – Actual value of y given x.

Predictive Models → used when continuous values

× ? → categorical value

Probability Distribution :—

→ Probability is the chance that something will happen.

→ ex:- Toss a coin & rolling a dice

$$P = \frac{\text{No. of way it can happen}}{\text{Total no of outcomes}}$$

Conditional probability :—

→ Conditional probability is a measure of likelihood of occurence provided that another occurrence has occured.

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

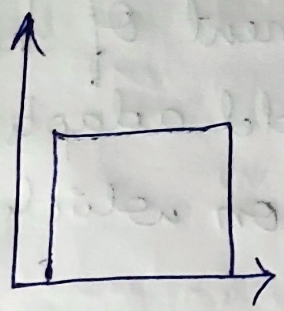
• In Data science i.e., Naive Bayes depends on theorem of Bayes.

Bayes theorem

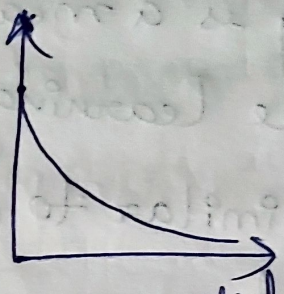
$$P(A/B) = \frac{P(B/A)\,P(A)}{P(B)}$$

$$\text{Posterior} = \frac{\text{prior} \times \text{likelihood}}{\text{evidence}}$$

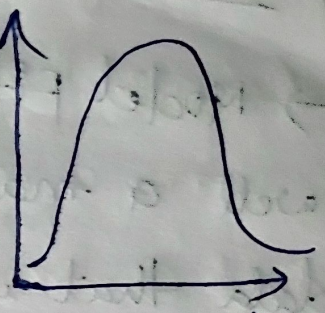
Continuous :-



uniform

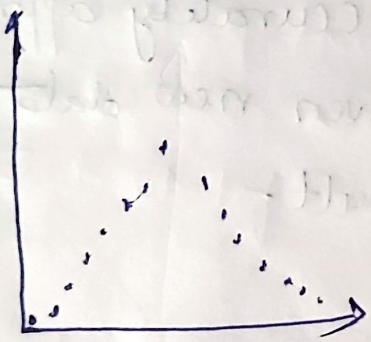


Exponential

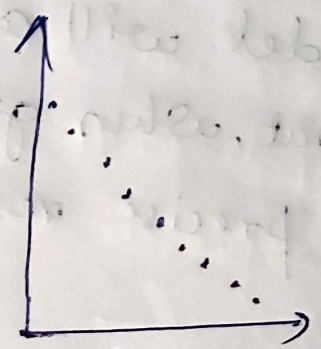


normal

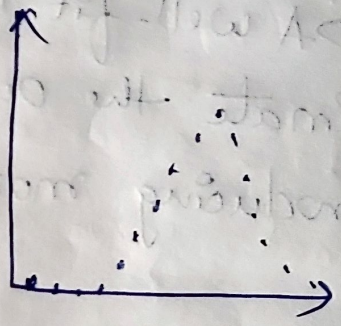
Discrete :-



Binomial



Geometric



Hypergeometric

General Binomial probability formula

$$P(k/n) = \frac{n!}{k!(n-k)!} P^{k}(1-P)^{n-k}$$

Normal distribution formula

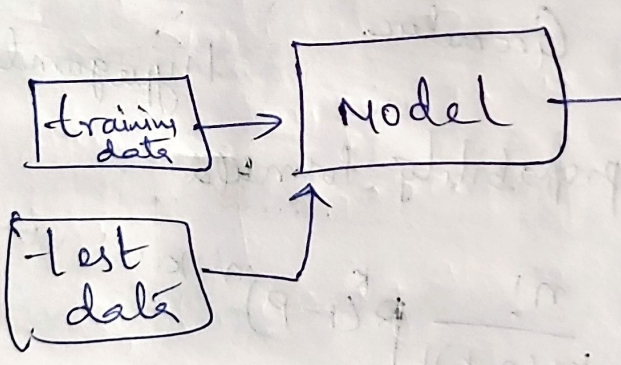
$$Z = \frac{\lambda - \mu}{\sigma}$$

Fitting a model :-

→ Model Fitting is a measurement of how well a machine learning model adapts to data that is similar to data on which it was trained.

→ The fitting process is generally built-in to models and is automatic.

→ A well-fit model will accurately approximate the output, when given new data, producing more precise result.



Underfitting

→ underfitting occurs when a model oversimplifies the data and fails to capture enough information on the relationship within it

→ This is frequently a result of insufficient

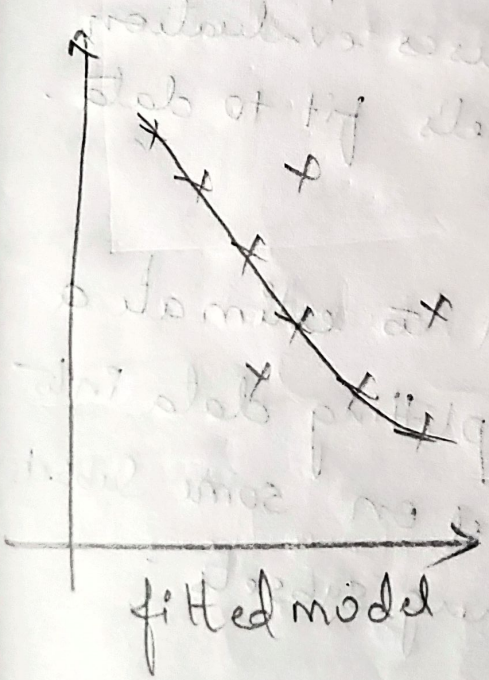
model training time.

Overfitting :-

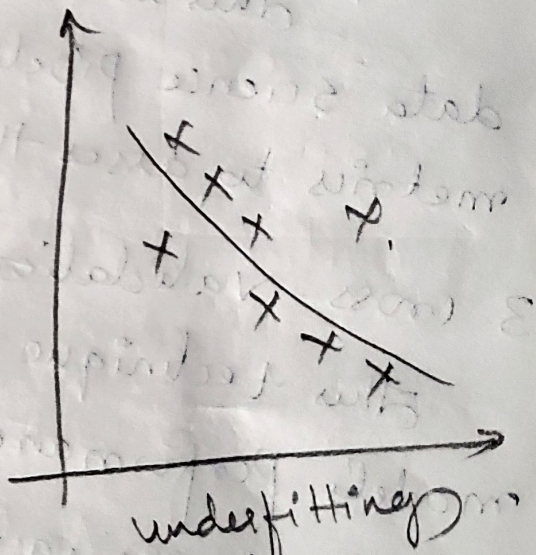
→ Overfitting is the opposite of underfitting

→ It occur when a model is overly sensitive to the data.

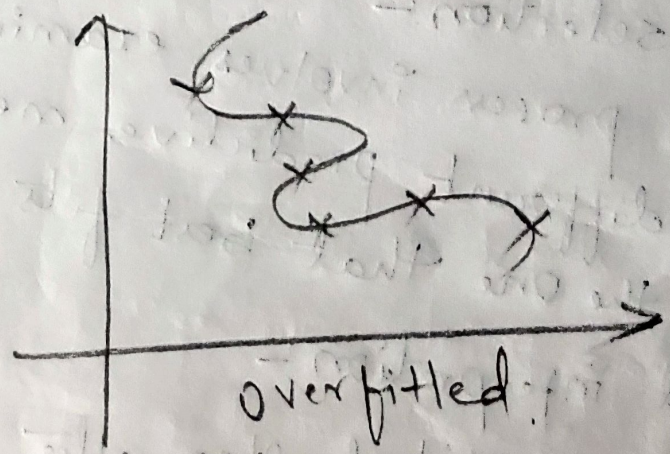
→ Overfitting is generally a result of Overtraining on training dataset.



fitted model



underfitting



overfitted

→ Model fitting is a process in data science that involves several steps to ensure a model accurately represents data and can make reliable predictions.

→ There are some steps which includes

1. data collection

2. Model evaluation :-
This is an important part of data science pipeline. It uses evaluation metrics to asses the model's fit to data.

3. cross-validation :-
This technique is used to estimate a model's performance by splitting data into subsets, training the model on some subsets & validating it on remaining subsets.

4. Model selection :-
This process involves examining the merits of different predictive method to identify the one that best fits the data

5. Model interpretion :-
It is a vital step after model

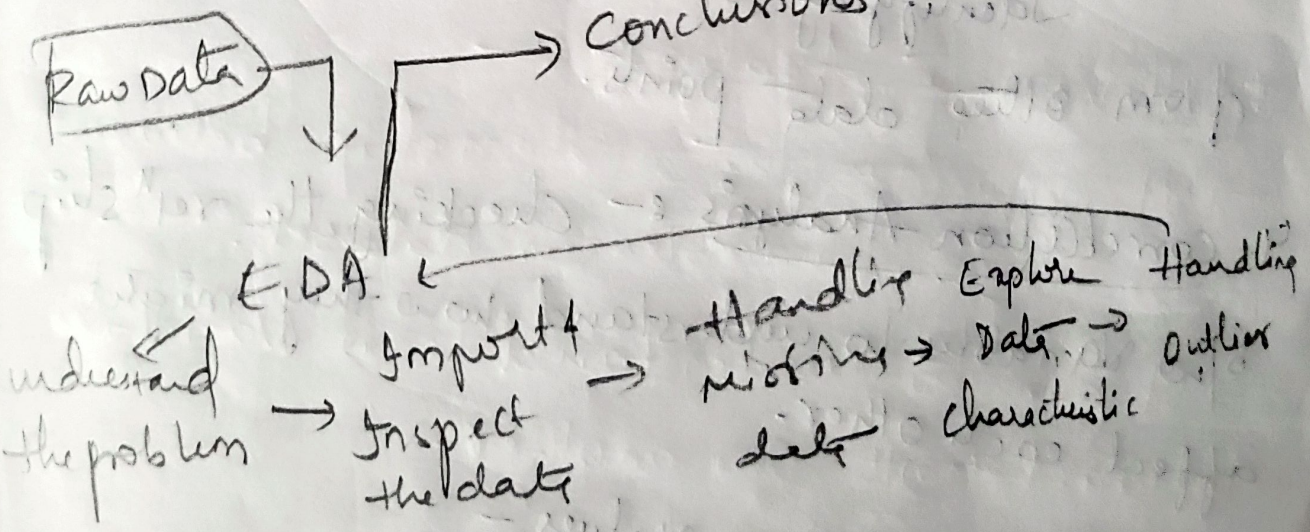
fitting.

Exploratory Data Analysis :-

→ Exploratory Data Analysis (EDA) is a crucial initial step in data science project.

→ It involves analyzing and visualizing data to understand its key characteristics, uncover patterns & identify relationship b/w. variables.

→ EDA is normally carried out as a preliminary step before undertaking extra formal statistical analyses & modeling.



Key Aspects of EDA Include:-

1. **Distribution of data** :- Examining the distribution of data points to understand their range, central tendency (mean, median) & dispersion (variance, S.D).

2. **Graphical Representation** :-
Utilizing charts, such as histograms, boxplots, scatter plots and bar charts to visualize rel^{n}ship within data & distribution of variables.

3. **Outlier Detection** :-
Identifying unusual values that deviates from other data points.

4. **Correlation Analysis** :- Checking the rel^{n}ship b/w variables to understand how they might affect each other.

5. **Handling missing values** :-
Detecting & deciding how to address missing data points, whether by imputation or removal, depending on their impact & amount of missing data.

6. Summary Statistics :-
calculating key statistics that provide insight into data trends.

7. Testing Assumptions :-
Many statistical test and models assume the data meet certain condition. EDA helps to verify these assumptions.

Types Of EDA :-

→ Depending on the number of columns we are analysing we can divide EDA into 3 types univariate, Bivariate and multivariate.

1. Univariate Analysis :-

→ It focuses on a single variable to understand its internal structure.
→ It is primarily concerned with describing the data and finding patterns existing in a single feature
Common techniques include...
1. Histogram : used to visualize distribution of variable
2. Box plots
3. Bar charts -

Bivariate Analysis :-

→ It involves exploring the connection between 2 variables.

→ It enables find associations, correlations, & dependencies b/w pairs of variables.

→ It is crucial form of EDA that examines the relationship between 2 variables.

Techniques used are

1. scatter plot.
2. Correlation coefficient.
3. Cross-tabulation
4. Line Graphs.

Multivariate Analysis :-

→ It examines the relationship between 2 or more variables in dataset.

→ It aims to understand how variables interact with one another.

Techniques used are-

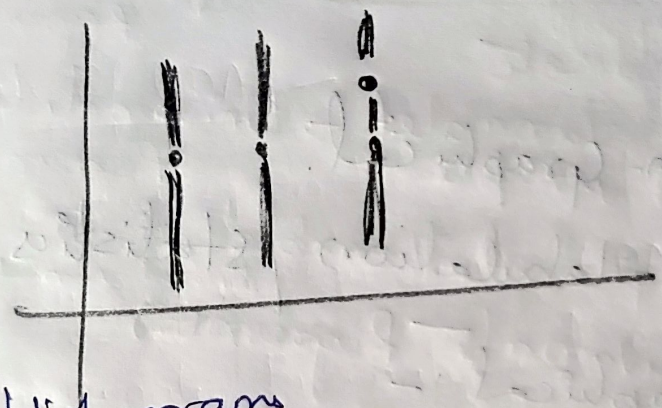
1. pair plots
2. principal Component Analysis.

1. Univariate Non-Graphical.

↳ find out the details of population data & some specific parameters of statistics central Tendency (Mean, median, mode). Range & S.D.

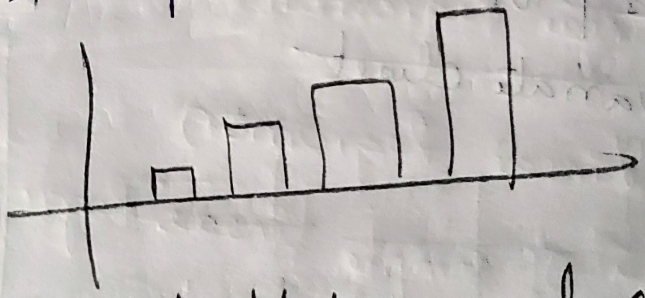
2. Univariate Graphical

1. Steam-and-leaf plot.

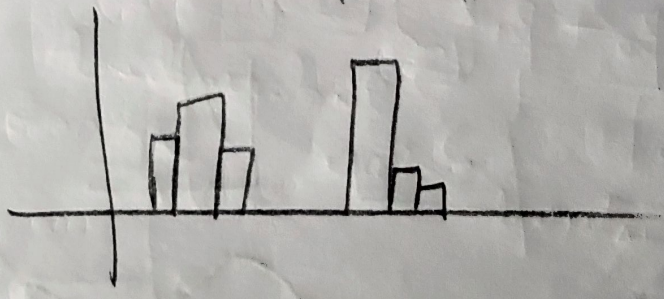


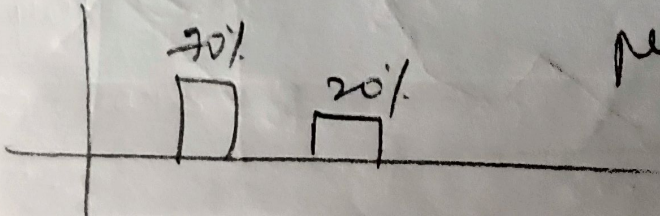
2. Histograms

↳ Simple Bar charts



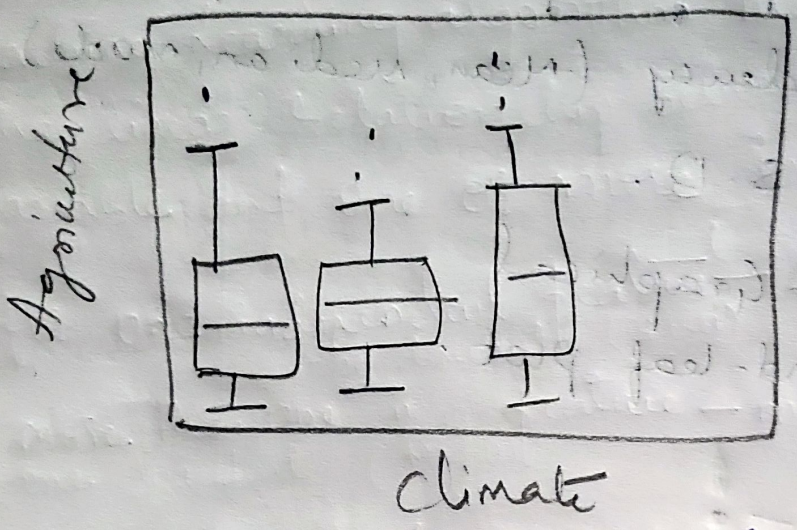
Multi/Grouped charts





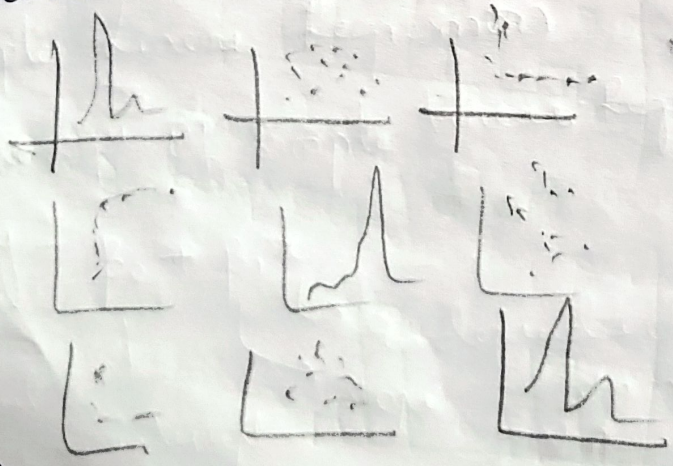
Percentage Bar charts

Boxplots :- used to display the distribution of quantitative value in data

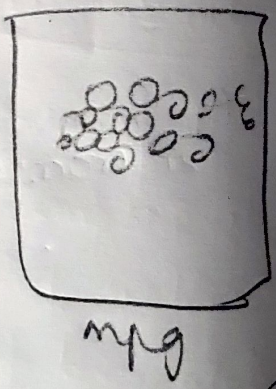


3. Multivariate Non-Graphical.
   ↳ cross-tabulation / statistics

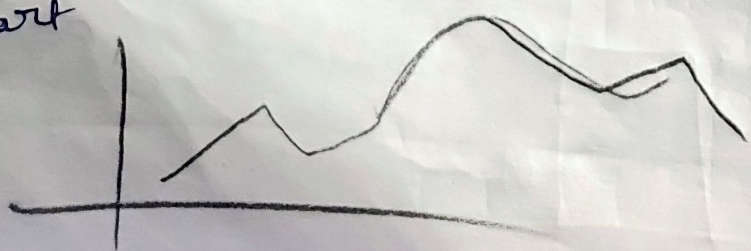
4. Multivariate Graphical :-
   1. Scatter plot.
   2. Multivariate chart.



Bubble chart



3 Run chart



UNIT-II

Three Basic Machine Learning Algorithm

→ Machine Learning is a branch of AI that focus on development of algorithm and statistical models that can learn from and make decisions/ prediction on data.

→ Supervised Learning is a type of machine learning where the algorithm learns from labelled data.

→ Labeled data means that dataset whose target value is already known.

→ Supervised Learning has two types.

1. Classification :- It predicts the class of dataset based on the independent i/p value.
2. Regression :- It predicts the continuous o/p value/ variable based on independent o/p variable.

eg:- prediction of house prices based on different parameter.

- Linear Regression :-

→ Linear Regression is a type of Supervised Machine Learning algorithm that Computes the linear relationship between the dependent variable and one or more independent variable.

→ for one independent feature — simple linear Regression. (SISO)
one dependent

more than one feature — multiple linear Regression. (MISO)
one dependent

→ One dependent variable → Univariate linear Regression
one independent variable

more than one dependent variable, it is known as Multivariate Regression.
independent variable

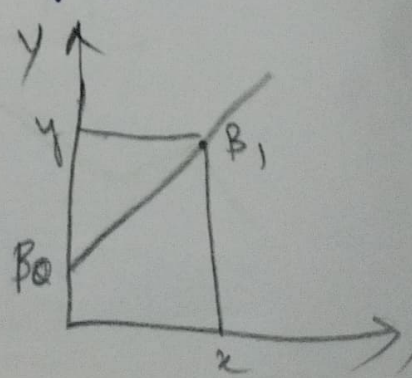
Types of Linear Regression

1. Simple Linear Regression

The eqn of simple linear Regression is

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

- y → dependent variable
- β_0 → intercept
- β_1 → slope
- x → independent variable



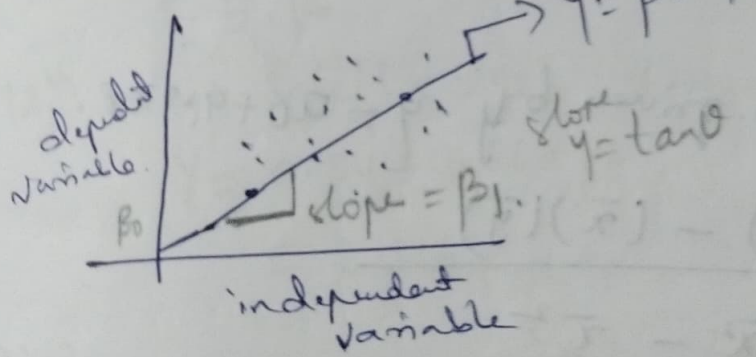
2. Multiple Linear Regression

→ The eqn of multiple linear Regression is

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 \cdots \beta_n x_n.$$

→ The Goal of the algorithm is to find the best fit line eqn that can predict the value based on the independent variable.

→ Linear Regression is a quiet and simple statistical regression technique for predictive analysis in ML.

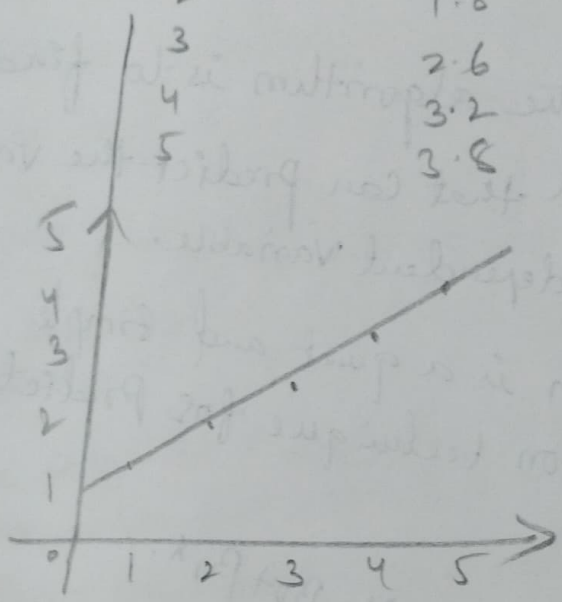


* Best fit line is a line that best fits the given scatter plot. Mathematically we can obtain the best-fit line by minimizing the Residual sum of squares (RSS).

Solved example

ex:-1

x_1 (week)	y_1 (sales in thousand)
1	1.2
2	1.8
3	2.6
4	3.2
5	3.8



Linear eqn is given by $y = a_0 + a_1 x + e$.

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

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

x	y	x_1^2	x * y
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	44.4
Avg = 3	2.52	$\overline{x^2} = 11$	$\overline{xy} = 8.88$

$$a_1 = \frac{8.88 - 3 * 2.52}{11 - 9}$$

$a_1 = 0.66$

$a_0 = 2.52 - 0.66 * 3$

$= 0.54$

$y = a_0 + a_1 x$

$y = 0.54 + 0.66x$

The predicted 7th week sale $(x = 7)$

$y = 0.54 + 0.66 * 7$

$= 5.16$

12th week sale $(x = 12)$

$y = 0.54 + 0.66 * 12$

$= 8.46$

ex:- 2

x (Salary)	y_i Expenditure
5	25
1	5
2	7
1	8
3	?

$\bar{x} = 2.25$ $\quad \overline{x^2} = 7.75$

$\bar{y} = 11.25$ $\quad \overline{xy} = 38$

$$a_1 = \frac{\overline{(xy)} - \bar{x} * \bar{y}}{\overline{x^2} - \bar{x}^2} = \frac{38 - 2.23 * 11.25}{7.75 - (2.25)^2}$$

$$= 4.72$$

$$a_0 = \bar{y} - a_1 \bar{x} = 11.25 - 4.72 * 2.25$$

$$= 0.62$$

Regression eq^n

$$y = 0.62 + 4.72 * x$$

3 lakh

$$y = 0.62 + 4.72 * 3$$

$$y = 14.78$$

Why linear Regression is important:-

1. Simplicity & interpretability
2. prediction
3. widespread applicability.

Evaluation Metrics:-

1. Coefficient of Determination or R-squared(R^2)
2. Root Mean Squared Error (RSME) and Residual Standard Error (RSE)

1. Coefficient Of Determination or R^2

→ R^2 is a number that explains the amount of variation that is captured by developed model.

→ It always ranges between 0 & 1.

→ The higher the value of R^2, the better the model fits the data.

$$R^2 = 1 - (RSS / TSS).$$

$$RSS = \sum_{i=1}^{n} (y_i - b_0 - b_1 x_i)^2$$

↓

diff b/w expected & actual O/P.

$$TSS = \sum (y_i - \bar{y}_i)^2$$

↓

Sum of errors of data pts from mean of response variable.

$R^2 = 0 \rightarrow$ poor fit

$R^2 = 1 -$ indicate good fit.

RSME :-

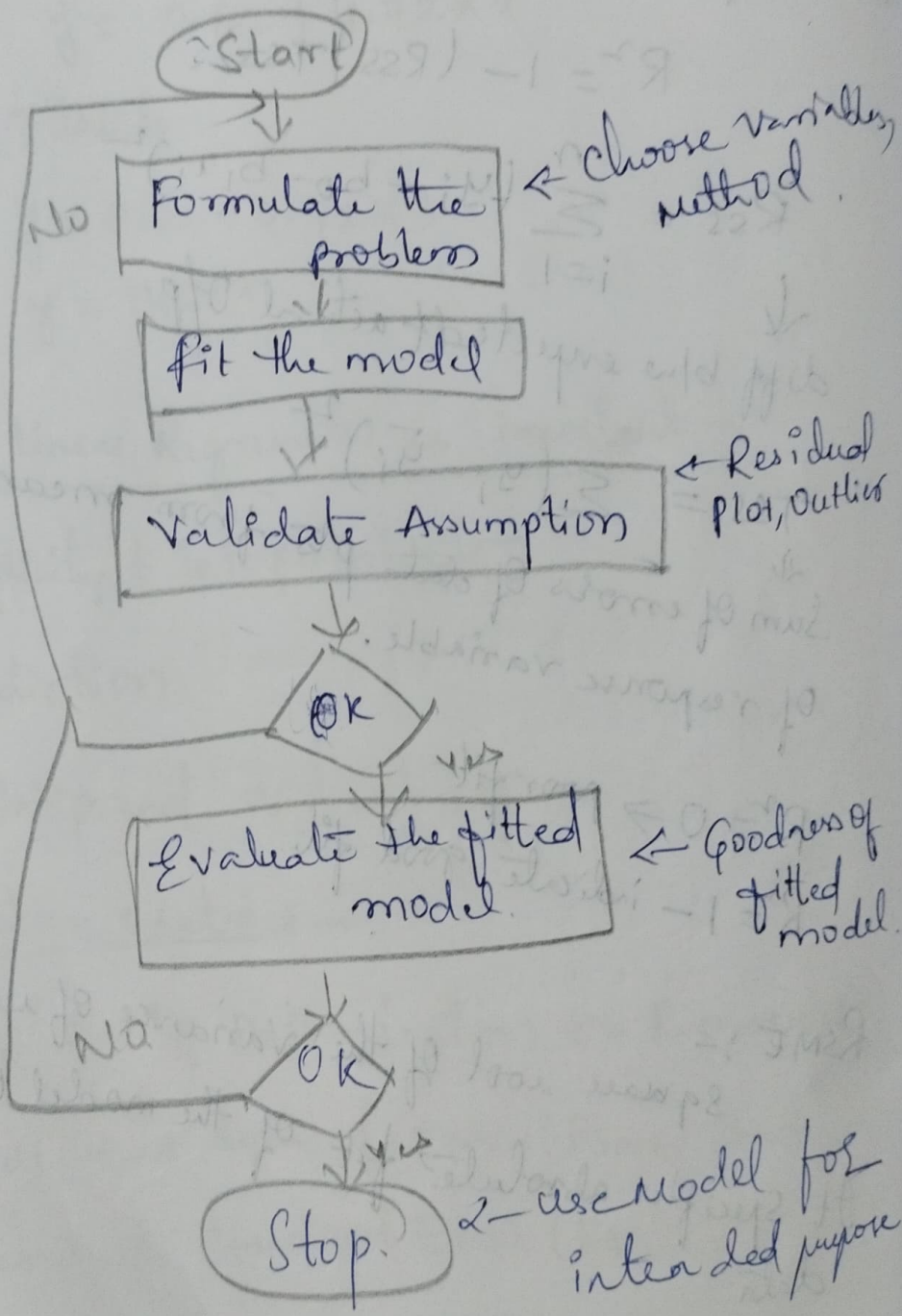
Square root of the variance of residual.

It specifies absolute fit of the model to data.

$$RMSE = \sqrt{\frac{RSS}{n}} = \sqrt{\sum_{i=1}^{n} \frac{(y_i^{Ac} - y_i^{pr})^2}{n}}$$

$$RSE = \sqrt{\frac{RSS}{df}} = \sqrt{\sum_{i=1}^{n} \frac{(y_i^{A} - y^{Proj})}{(n-2)}}$$

Working of Regression process :-



Implementation of Linear Regression Model in R

→ Let us Consider a dataset called ~~read~~ bonds.txt with 2 Variables Bidprice & Coupon Rate.

→ To load the dataset in R we use a function called read.delim

Syntax:- read.delim(filename, row.names=1)

Ex:- ⇒ bonds ← read.delim("bonds.txt", row.names=1)

→ To view the loaded dataset in tabular format we use view function.

⇒ view(bonds)

→ To Know the structure of the dataset we use str function

⇒ Str(bonds)

To Calculate the Summary / display the Summary of the dataset we use

⇒ summary(bonds)

that displays

minimum value, Median, Mean, 1st Quartile, 3rd Quartile and maximum value of the Variable in dataset.

To draw the scatter plot of the given data set we use plot function

plot(bonds$couponRate, bonds$bidprice, main = "Bidprice Vs CouponRate", ylab = "Bidprice", xlab = "CouponRate").

for getting the ~~linear~~ best-fit line model (line) that join the points in scatter plot we use abline(bonds) function in R.

→ To ~~evaluate~~ fit the dataset using linear regression model:

Syntax: lm(formula, data)

lm(dependent variable ~ independent variable)

Ex: bondsmod <- lm(bonds$Bidprice ~ bonds$couponRate)

$y = a_0 + a_1 x$

→ To Summary the linear Regression model we use Summary function again to get the intercept and slope value of linear Regression

Ex:- Summary(bondsmod).

O/P Call

lm(bonds$Bidprice ~ bonds$couponRate)

Residual

Min	1Q	Median	3Q	Max
-8.247	2.470	-0.838	2.550	10.525

	Estimated	std. deviation	t. value
intercept	74.7866	2.8267	
slope	3.0661	0.3068	

RSE = 4.175

Multiple R^2 = 0.7576 $\bar{R}^2$ (Adjusted) = 0.7441

F-statistics = 99.87 ...

Outliers :- points which do not conform to pattern in bulk of data.

Multiple Linear Regression

→ Multiple Linear Regression is an extension of Simple linear Regression as it take more than one independent variable and one dependent variable.

Ex:- house price

Eqn for Multiple Linear Regression.

$$y = b_0 + b_1 x_1 + b_2 x_2 + b_3 x_3 \cdots b_n x_n$$

Ex:-

x_1 product1 Sales	x_2 product2 Sales	y weekly sales
1	4	1
2	5	6
3	8	8
4	2	12

Matrices for X and x are given as

$$X = \begin{bmatrix} 1 & 1 & 4 \\ 1 & 2 & 5 \\ 1 & 3 & 8 \\ 1 & 4 & 2 \end{bmatrix} \quad \& \quad Y = \begin{bmatrix} 1 \\ 6 \\ 8 \\ 12 \end{bmatrix}$$

$a_0, a_1, a_2 \cdots$
$\downarrow x_1 \quad \downarrow x_2$

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

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

$$= \begin{bmatrix} 4 & 10 & 19 \\ 10 & 30 & 46 \\ 19 & 46 & 101 \end{bmatrix}^{-1}$$

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

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

$$= \begin{bmatrix} 0.05 & 0.47 & -1.02 & 0.99 \\ -0.32 & -0.098 & 0.155 & 0.26 \\ -0.065 & 0.005 & 0.185 & -0.125 \end{bmatrix} \times y \begin{bmatrix} 1 \\ 6 \\ 8 \\ 12 \end{bmatrix}$$

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

$$\begin{array}{r} -1.69 \\ -0.25 \\ \hline -1.94 \end{array}$$

$a_0 = -1.69$

$a_1 = 3.48$

$a_2 = -0.05$

$y = a_0 + a_1 x_1 + a_2 x_2$

$y = -1.69 + 3.48 \times x_1 + (0.05) \times x_2$

$= -1.69 + 6.96 - 0.25$

$= 6.56 - 1.94 = 5.00$

$$\begin{array}{r} -1.69 \\ 0.25 \\ \hline 1.94 \end{array}$$

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2$$

$$a_0 = \bar{y} - \beta_1 \bar{x}_1 - \beta_2 \bar{x}_2$$

$$\beta_1 = \frac{(\Sigma x_2^2)(\Sigma x_1 y) - (\Sigma x_1 x_2)(\Sigma x_2 y)}{(\Sigma x_1^2)(\Sigma x_2^2) - (\Sigma x_1 x_2)^2}$$

$$\beta_2 = \frac{(\Sigma x_1^2)(\Sigma x_2 y) - (\Sigma x_1 x_2)(\Sigma x_1 y)}{(\Sigma x_1^2)(\Sigma x_2^2) - (\Sigma x_1 x_2)^2}$$

x_1	x_2	y
1	4	4
2	5	6
3	8	8
4	2	12
Σ = 10	19	27
	4	4

$$\Sigma x_1^2 = \frac{(\Sigma x_1)^2}{n}$$

$x_1 y$	$x_2 y$	$x_1 x_2$	x_1^2	x_2^2
4	4	4	1	16
12	30	10	4	25
18	64	24	9	64
48	24	8	16	4
$\frac{79}{4}$	$\frac{122}{4}$	$\frac{46}{4}$	$\frac{30}{4}$	$\frac{109}{4}$

$$\Sigma x_1^2 = \frac{(\Sigma x_1)^2}{n}, \quad \Sigma x_1 y = \frac{\Sigma x_1 \Sigma y}{n}$$

$$\Sigma x_1 x_2 = \frac{\Sigma x_1 \Sigma x_2}{n}$$

$$\frac{\left(\frac{109}{4}\right)^2 \times \frac{79}{4} - \left(\frac{46}{4}\right) \times \frac{122}{4}}{\left(\frac{30}{4}\right)^2 \left(\frac{109}{4}\right)^2 - \left(\frac{46}{4}\right)^2} =$$

$$\frac{(10.9)^2 \times 79 - 4.6 \times 12}{(30)^2 (10.9)^2 - (4.6)^2}$$

$$\frac{11881 \times 79 - 5612}{900 \times 109 -}$$

$$\frac{938599 - 5612}{98100 - 211} = \frac{932987}{95984}$$

$$= 9.720$$

Ridge Regression :- (linear)

→ Ridge Regression is a type of linear Regression that include a regularization term to penalize large co-efficients.

→ Multicollinearity occurs when two or more predictor variable (independent) in a regression Model are highly co-related to unreliable & unstable estimate of regression.

→ This algorithm/model helps to prevent overfitting especially when their is multicollinearity.

$$eq^n :- \sum_{i=1}^{n} (y_i - \bar{y}_i)^2 + \lambda \sum_{j=1}^{p} \beta_j^2$$

λ is a parameter that Control strength of penalty.

Polynomial Regression :-

↳ used to model non-linear Relationship between the dependent variable & independent variable.

$$y = \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3 + \cdots \beta_n x^n + \epsilon$$

Support Vector Regression:- (Non-linear)

↳ that is based on the Support Vector Machine (SVM) algorithm.

→ It works by finding a hyperplane that minimize the sum of squared residual between the predicted and actual values.

Decision Tree Regression:- (non-linear)

↳ It is a type of regression algorithm that build a decision tree to predict target value.

Random Forest Regression:- (non-linear)

→ Random Forest Regression is an ensemble method that combines multiple decision trees to predict target value.

↳ Ensemble Methods are type of machine learning algorithm that combines multiple models to improve performance of overall model.

Lasso Regression (Least-Absolute shrinkage & selection operator) (linear)

↳ It another type of linear Regression that is used to prevent overfitting.

↳ It does by adding penalty to loss function that forces model to use some weights & to set other to zero

K-Means

→ K-Means Clustering is an unsupervised Machine learning algorithm, which groups the unlabeled dataset into different clusters.

→ unsupervised learning is the process of teaching a Computer to use unlabeled, unclassified data & enabling the algorithm to operate on data without supervision.

Types Of unsupervised learning

1. Clustering :—
   ↳ goal is to divide data into groups such that data points in same cluster are more similar to each other than to points in other cluster.

   K-Means —
   Hierarchical clustering — Builds clusters by merging small ones
   DBScan (Density based spatial clustering of Application with Noise) — clusters based on density

2. Dimensionality Reduction
   ↳ goal is to Reduce no of features in dataset while preserving essential structure.
   1. PCA (principal Component Analysis)
   2. t-SNE (t-distributed Stochastic Neighbor Embedding)

3. Association:-

↳ goal is to find rules or relnship b/w different variable in dataset.

1. Apriori Algorithm.
2. Eclat Algorithm.

K-Means:-

() It assign data points to one of the k-clusters depending on their distance from the center of clusters.

→ It starts by randomly assigning the cluster centroid in space.

→ Then each data point assign to one of the cluster based on its distance from centroid of cluster.

→ After assigning each point to one cluster, new centroid of cluster are calculated & assigned.

→ The process runs iteratively until it finds good clusters.

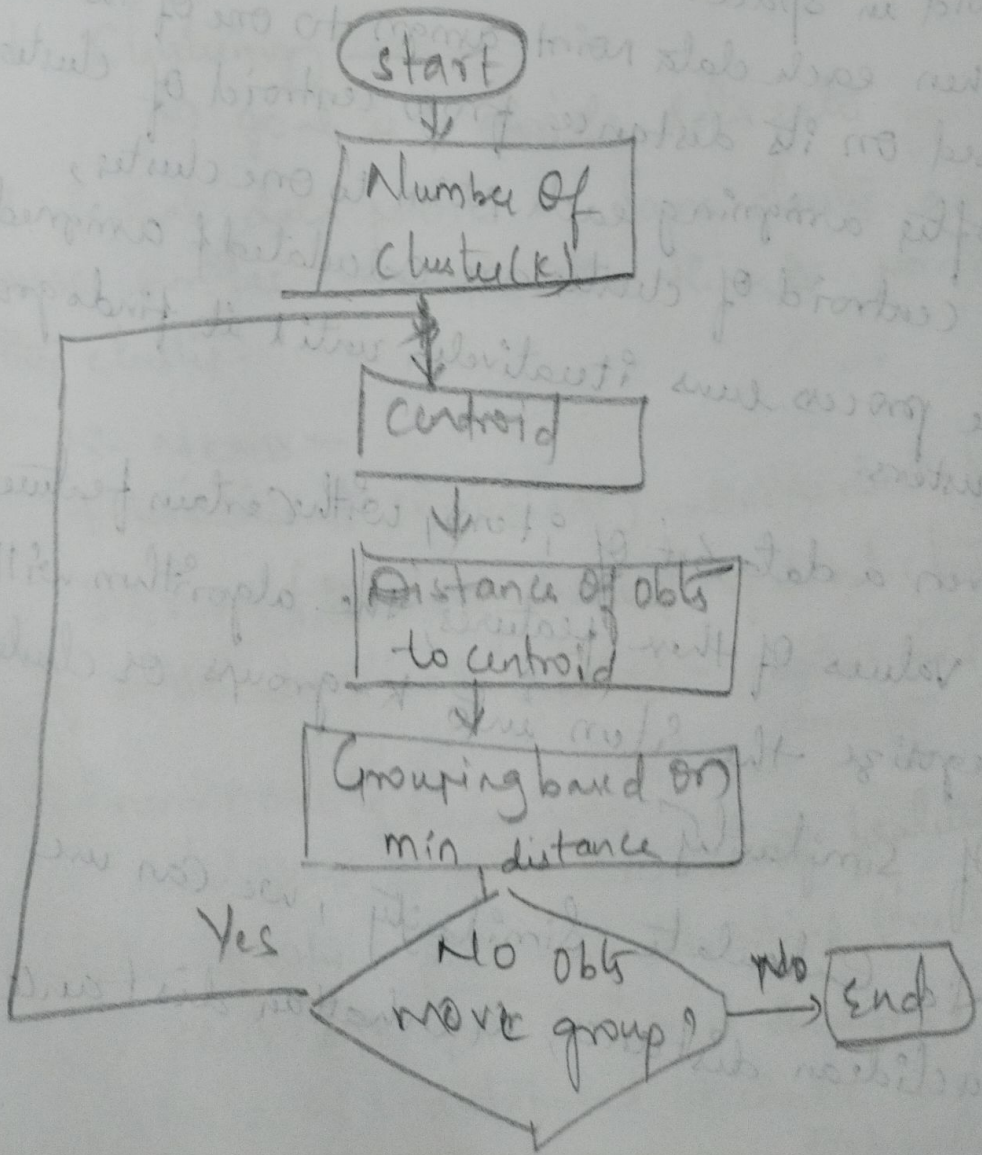
→ Given a data set of items, with certain features, and values of these features, the algorithm will categorize the item into k-groups or clusters of similarity.

→ We calculate similarity, we can use Euclidean distance, Manhattan distance,

Hamming distance, Cosine distance

Algorithm

1. Choose K number of random data points as initial centroids.
2. Repeat till cluster centers stabilize
   a. Allocate each point in D to nearest of Kth centroids.
   b. Compute Centroid for cluster using all points in cluster.



Advantages :-

1. K-means is simple, easy to understand and easy to implement.
2. It is also efficient, in which the time taken to cluster rises linearly with the number of data points.
3. No other clustering algorithm performs better than K-means.

Disadvantage :-

1. The user need to specify an initial value of k.
2. The process of finding the cluster may not converge.
3. It is not suitable for discovering clusters that are not hyper ellipsoids or hyper spheres).

Example :

x	y
2	4
2	6
5	6
4	7
8	3
6	6
5	2
5	7
6	3
4	4

2-

x.

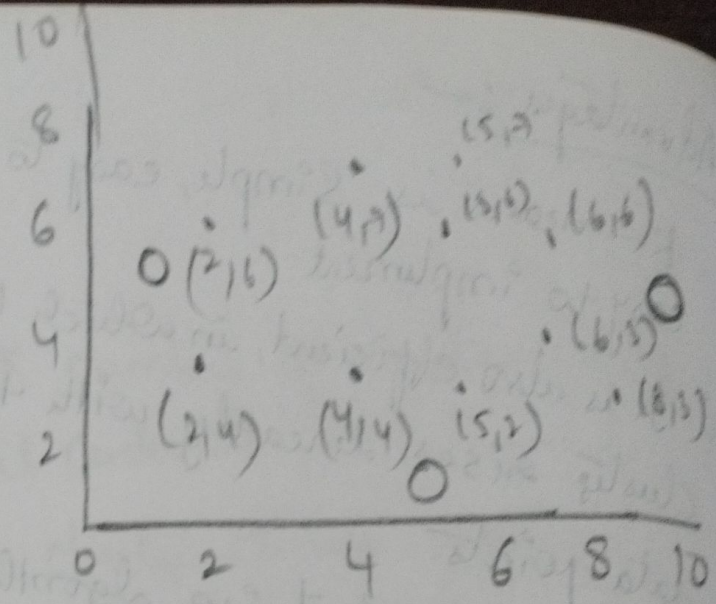
2, 4, 10, 12, 3,

20, 30, 11, 25

$M_1 = 4, M_2 = 11$

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

Cent = $\frac{\text{Sum}}{\text{no}}$



Iteration - 1

$C_1 - (1,5)$

$C_2 - (4,1)$

$C_3 - (8,4)$

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

x	y	(1,5)	(4,1)	(8,4)	cluster
2	4	1.41	3.61	6.00	C_1
2	6	1.41	5.39	6.32	C_1
5	6	4.12	5.10	3.61	C_3
4	7	3.61	6.00	5.00	C_1
8	3	7.28	4.47	1.00	C_3
6	6	5.10	5.39	2.83	C_3
5	2	5.00	1.41	3.61	C_2
5	7	4.47	6.08	4.24	C_3
6	3	5.30	2.83	2.24	C_3
4	4	3.16	3.00	4.00	C_2

$C_1 = \frac{2+2+4}{3}, \frac{4+6+7}{3}$

$= \frac{8}{3}, \frac{17}{3}$

$C_1 = (2.66, 5.66)$

$C_2 = (4.5, 3)$

$C_3 = (6, 5)$

Iteration – 2

X	Y	(2.66, 5.66)	(4.5, 3)	(6, 5)	Cluster
2	4	1.79	2.69	4.12	C_1
2	6	0.74	3.91	4.12	C_1
5	6	2.36	3.04	1.41	C_3
4	7	1.90	4.03	2.83	C_1
8	3	5.97	3.5	2.83	C_3
6	6	3.36	3.35	1	C_3
5	2	4.34	1.12	3.16	C_2
5	7	2.70	4.03	2.24	C_3
6	3	4.27	1.5	2 move from C3 to C2	C_2
4	4	2.13	1.12	2.24	C_2

Repeat this process until there is no

Movement

$C_1 - (2.66, 5.66)$

$C_2 - (5, 3)$

$C_3 - (6, 5.5)$

Implementing in R

```
Kmeans(x, Centers, iter.max=10, nstart=1,
       algorithm=c("Hartigan-Wong", "Lloyd",
                   "Forgy", "MacQueen"))
```

Where x is a dataset, which should be a numeric matrix or data frame

Centers specifies the no of clusters (k) to create or initial cluster centers.

iter.max : maximum no of iteration allowed default is 10.

nstart : no of random sets of initial cluster centers. default is 1.

algorith : Specifies the algorithm for clustering

1. Hartigan-Wong (default) : Standard algorithm proposed by Hartigan & Wong
2. Forgy :- Another version of k-means
3. Lloyd :- Standard k-mean algorithm
4. MacQueen :- A variation of k-mean algorithm

Ex:-

```
Kmean(x, centers=3, iter.max=10, nstart=1,
      algorithm="Haritgan-Wong").
```

K-Nearest Neighbours (K-NN)

⇒ K-NN is one of the simplest Machine learning algorithm based on supervised learning.

→ KNN uses the similarity between the new case and available case & put the new data into the category that is most similar to available.

→ KNN is non-parametric algorithm, which means it does not make any assumptions.

Working :—

1. Select the number 'K' of neighbour.
2. Calculate the Euclidean distance of K number of neighbour.
3. Take K nearest neighbor as per the calculated Euclidean distance. $(\sqrt{(x_2-x_1)^2+(y_2-y_1)^2})$
4. Assign the new data pts to that category for which the no of neighbour is maximum.

How to Select the value of K :—

→ There is no particular way to determine the value of K.

2. A very low value of k can be noisy/inaccurate.

3. large value of 'k' is good, but it may find some difficulties.

4. Always use an odd number for value of k.

Eg:-

S.NO	Maths	CS	Result
1	4	3	F
2	6	7	P
3	7	8	P
4	5	5	F
5	8	8	P
6	6	8	?

K = 3

1 $\sqrt{(6-4)^2+(8-3)^2} = \sqrt{2^2+5^2}$

$= \sqrt{4+25} = \sqrt{29} = 5.25$

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

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

4 $\sqrt{(6-5)^2+(8-5)^2} = \sqrt{1^2+3^2} = \sqrt{1+9}$

$= \sqrt{10} = 3.14$

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

~~Excuse 2, 3, 5~~ 1, 1, 2, 3.14, 5 . .

P P P F F

K=3

∴ 6, 8 fall under pass category.

Advantage :-

→ Can be applied to the data of any distribution
→ Very simple & intutive
→ Good classification if no of sample is large enough.
→ No training is required before classification

Disadvantage :-

→ Depends on 'k' value
→ Test storage is Computationally expensive
→ No training stage.

Filtering Spam and Naive Bayes :-

Naive Bayes is a supervised learning algorithm which is based on Bayes theorem.

→ It is mainly used in text classification that include a high-dimensional training dataset.

→ It is a probabilistic classifier, which means it predicts on the basis of probability of an object.

Bayes' Theorem :-

→ It is also called as Bayes Rule / Bayes law which is used to determine the probability of hypothesis with prior knowledge.

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

- $P(B|A)$ → likelyhood probability
- $P(A)$ → prior probability
- $P(B)$ → Marginal
- $P(A|B)$ → posterior probability

Working Of Naive Bayes :-

→ Convert the given data into frequency table
→ Generate Likelihood table by finding the probabilities of given features.
→ Use Bayes theorem to Calculate the posterior probability.

Ex:-

S.NO	Color	Legs	Height	Smelly	Species
1	white	3	short	yes	M
2	Green	2	Tall	NO	M
3	4	3	short	yes	M
4	white	3	"	yes	M
5	Green	2	"	NO	H
6	white	2	Tall	NO	H
7	white	2	4	4	H
8	white	2	short	yes	H
9.	Green	3	Tall	NO	?

Frequency Table

Color	(4) M	(4) H
Green	2/4	3/4
white	2/4	1/4

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

Heights	M	H	Smelly	M	H
Short	3/4	2/4	Yes	3/4	1/4
Tall	1/4	2/4	NO	1/4	3/4

$$P(M) = P(M) * P\left(\frac{\text{Green}}{M}\right) * P\left(\frac{3}{M}\right) * P\left(\frac{\text{Tall}}{M}\right) * P\left(\frac{\text{NO}}{M}\right)$$

$$= \frac{4}{8} \times \frac{2}{4} \times \frac{3}{4} * 1/4 * 1/4$$

$$= \frac{3}{4 \times 4 \times 4 \times 4} = \frac{3}{256}$$

$$= 0.0117$$

$$P(H) = P(H) * P\left(\frac{\text{Green}}{H}\right) * P\left(\frac{3}{H}\right) * P\left(\frac{\text{Tall}}{H}\right) * P\left(\frac{\text{NO}}{H}\right)$$

$$= \frac{4}{8} * \frac{3}{4} * 0 * \frac{2}{4} * 3/4$$

$$= 0.$$

$P(M) > P(H)$

$0.0117 > 0$

∴ The labeled value for Green, 3, Tall, NO is M.

Advantage :-

→ It is one of the fast & easy algorithm to predict a class

→ It can be used for Binary as well as multiclass classifications.

→ It is most popular choice for text classification problem.

Spam filtering :-

→ A spam filter is a program used to detect unwanted and virus infected emails and prevent those message from getting to a user's inbox.

→ Internet service providers (ISPs), free online email services use email spam filtering tools to minimize the risk of distributing spam.

Ex :- Microsoft Hotmail (Keywords in Subject)

→ Naive Bayes is widely used for spam filtering because it can classify emails based on probability of certain words appearing in spam ~~word~~ versus non-spam (ham) emails.

Ex:-

Step 1:- Training the model

→ The Naive Bays classifier is used/trained on a dataset of labeled emails, where each email is marked as either "spam" or "ham".

Spam Emails:

1. "Win a free iphone now"
2. "Congratulations, You have won a lottery".

Ham Emails:-

1. "Meeting scheduled for tomorrow".
2. "Let's catch up over lunch".

Step 2:- Calculate Word probabilities.

Probability of Spam:

$$p(\text{spam}) = \frac{\text{Total spam email}}{\text{Total email}}$$

$$= \frac{2}{4} = 0.5$$

Probability of ham

$$p(\text{ham}) = \frac{\text{Total ham email}}{\text{Total email}}$$

$$= \frac{2}{4} = 0.5$$

Next Calculate likelihood of individual words

→ Let's focus on "win", "free", "lunch", "meeting"

$P\left(\frac{win}{spam}\right) = \frac{2}{7}$ $\quad P\left(\frac{free}{spam}\right) = \frac{1}{7}$

$P\left(\frac{win}{ham}\right) = 0/8$ $\quad P\left(\frac{free}{ham}\right) = 0/8$

$P\left(\frac{lunch}{spam}\right) = 0/7$ $\quad P\left(\frac{meeting}{spam}\right) = 0/7$

$P\left(\frac{lunch}{ham}\right) = 1/8$ $\quad P\left(\frac{meeting}{ham}\right) = 1/8$

Step 3:- clarify the ~~probability of spam~~ New email.

"win a free lunch"

1. Calculate probability of spam.

$$P\left(\frac{Spam}{Email}\right) = P(Spam) * P\left(\frac{win}{spam}\right) * P\left(\frac{Free}{spam}\right) * P\left(\frac{lunch}{spam}\right) / P(Email)$$

win a free lunch

$$= 0.5 * \frac{2}{11} * \frac{1}{11} * \frac{0}{11} = 0.$$

$$P\left(\frac{ham}{email}\right) = P(ham) * P\left(\frac{win}{ham}\right) * P\left(\frac{free}{ham}\right) * P\left(\frac{lunch}{ham}\right) / P(Email)$$

$$= 0.5 \times \frac{0}{9} * \frac{0}{9} * \frac{1}{9} = 0.$$

→ Since probabilities are simplified here and both result is zero, In real implementation Smoothing techniques are applied to avoid zero Probability. → The probability of which is max, we consider that.

Why Naive Bayes works well in Spam filtering:

1. Fast & efficient
2. Continous Learning
3. Efficient Calculation
4. Scalable :- work for large dataset & quickly adapts to new patterns.

Data Wrangling :-

→ Data, in its raw form, often contains errors, is incomplete or is not in readily usable format.

→ Data Wrangling process transforms this raw data into more usable form.

→ Data Wrangling also known as data Munging, is the process of cleaning, transforming and organizing raw data into a format

Suitable for analysis.

→ This process is a crucial step in data science because raw data is often messy, unstructured and contains errors that can impact the quality of data derived from it.

→ There are 5 steps to data Wrangling.

1. Discovery :- Explore and understand the data to identify its sources, structure and potential issues.
(or) Data Collection

2. Data cleaning :- This step involves fixing errors, correcting inconsistencies and filling in missing values

(i) Handling Missing values: Filling the missing data using mean, median and mode or more advanced techniques like interpolation or predictive models

(ii) Removing duplicates :- Identifying and removing duplicate records to avoid twist result.

(iii) Correcting Errors :- Fixing errors in data, such as incorrect data format or outliers.

3. Data Transformation :- Converting data into usable format.

(i) Normalization and scaling :- Adjusting the

scale of numerical data, so that features contribute equally to the analysis.

(ii) Feature Engineering :- Creating new features from existing data to improve model performance

(iii) Encoding Categorical Variable :- Converting Categorical data into numerical data form using some techniques.

4. Data Integration :- Combing data from different sources to create unified dataset. This require joining tables or merging data from multiple file. ($\bowtie$)

5. Data Structuring :- Organizing data into a structured format, such as arranging data into a tabular form suitable for analysis.

6. Data Validation :- Ensuring that the transformed data meets the required quality standard by checking its accuracy, consistency and completeness.

7. Data Enrichment :- Enhancing the dataset with additional information, such as incorporating from external data sources to provide for analysis.

Importance of Data Wrangling:-

1. Improve Data Quality :- Raw data is often messy, & contain missing values, duplicate, incorrect format. Wrangling helps to clean & correct these issues.

2. Save Time :- Automating data wrangling steps helps in time saving during data analysis phase; allowing data scientist to focus more on modeling & interpreting results.

3. Enhances Data Decision-Making :- Clean & well-structured data provide more accurate insights, which are crucial for decision-making.

4. Improve Analysis :- properly wrangled data is structured & is easier to analyze, leading to more accurate results & insights.

Tools :-

Tools like Excel, OpenRefine etc are popular for powerful data transformation capabilities.

Data visualization :-

↳ using Matplotlib, Seaborn, or Tableau can help in identifying patterns, trends & anomalies.

Programming language :-

↳ R, SQL, Python (Panda, Numpy libraries)

Web Scrapping :-

Web Scrapping involves extracting data from websites and API's and tools play a crucial role in this process.

API's for Web Scrapping :-

API (Application program Interface) are designed to allow application to Interact with each other.

→ In Web Scrapping, API's can often provide direct access to structured data without the need to scrape HTML content.

1. Twitter API: Used for extracting tweets, user profile and Twitter data.
2. Google Maps API: provide access to location-based data such as address, geographical co-ordinates & business information.
3. IMDb API: Used for fetching movie, TV shows and Celebrity information.
4. YouTube Data API: Allow access to YouTube content like videos, channel, comments and playlists.

5. GitHub API :- fetches data related to repositories user, commits, and code search results.

6. OpenWeatherMap API : provides weather data for different location worldwide, including current weather, forecasts & historical data.

Tools for Web Scrapping :-

1. Beautiful Soup :-

Description :- A python library used for paring HTML and XML documents. It helps to navigate, search and modify the parse tree easily.

Strength :- Easy to use for beginners, excellent small to medium sized scraping tasks.

Limitation :- Slower than some alternatives when handling large dataset.

2. Scrapy :-

Description :- A powerful open-source python framework specifically designed for web scraping and crawing.

→ It is used to extract the data from website & Process it as per the user's requirement.

Strength:- Highly efficient, Can handle large-scale Scraping project, built-in Support for handling requests, and managing data pipelines.

Limitation:- Has a steeper learning curve Compared to simple libraries.

3. Selenium:-

Description:- A tool that automates web browser, allowing you to scrape data from dynamic & Java-rendered pages.

Strength:- Excellent for interacting with Complex web pages, automating form submission & navigating through web pages.

Limitation:- More Complex to setup

Slower Compared to headless Scrapping tools since it requires a browser.

4. Puppeteer:-

Description:- A Node.js library that provides a high-level API to control chrome browser programmatically.

Strength:- Perfect for Scrapping dynamic websites, taking screenshots & automating browsing tasks.

Limitation:- More Complex to set-up Compared to Phython based tool. requires JS knowledge.

5. Requests + lxml :-

Description :- A combination of python's requests library (HTTP requests) and lxml (for parsing XML & HTML).

Strength :- Fast & Efficient, great for simple scraping tasks where you need to extract data from static pages.

Limitations :- Requires manual handling of data extraction logic, not idea for ~~logic~~ large scale or complex scraping.

6. Playwright :-

Description :- A modern web automation tool that support multiple browsers (chrome, firefox, webkit). It can interact with web browser similar to puppeteer but support more browsers.

Strength :- Reliable for scraping more modern web application & can handle complex navigation scenario.

Limitation :- Simply/slightly more complex than Selenium, needs knowledge of Java script (JS) & Python.

7. Octoparse :-

Description :- A no-code web scraping tool that provides a user-friendly interface for scraping without writing code.

Strength :- Suitable for non-programmers, handle dynamic web-pages & offers cloud-based Scraping.

Limitations :- less flexible than custom-coded solution, with limitation on free usage.

8. parsehub :-

Description :- A visual Scraping tool that allow users to collect data from Complex websites including those using AJAX

Strength :- No Coding Skill required, handles dynamic, Content, easy Setup.

Limitation :- Limited free-tier, less Control compared to coding solution.

9. WebHarvy :-

Description :- A point-and-click web scraper that automates data extraction from websites

with no coding required.

Strength :- User-friendly, Support multiple data extraction types including text, images & URL.

Limitation :- Not as customized as code-based solution, Commercial license required for advanced features.

10. Apify :-

Description :- A cloud platform for web scraping & automation that Support JS/Python based scrapers.

Strength :- Scalable, can handle heavy loads, Supports both headless & full browser scraping.

Limitation :- paid service with free-tier limitation.

Ex:- Spam filtering

Spam
Send us your password
review us
Send us your account
" your password.

Not-Spam
password review
send us your review

Q
1.) review us now
review account

P(words)	spam	not-spam
send	3	1
us	3	1
your	3	1
password	2	1
review	1	2
account	1	0
	13	6

P(spam) = 4/6
P(ham) = 2/6

Review us = x
$P(S/x) = P(S) \times P(x/S)$
$P(\frac{ns}{x}) = P(ns) \times P(\frac{x}{ns})$

Challenges Of Data Wrangling

1. Scalability & performance
2. Handling unstructured & semi-structured Data
3. privacy & security concerns (personal Information)
4. Evolving Data sources & formats
5. Missing Data
6. Data Integration is complex

UNIT-III

Feature Generation and Feature Selection:-

Feature Generation (Feature Engineering):-

→ It involves creating new features from raw data that can be better represent the underlying pattern relevant to predictive models.

→ The goal is to transform or extract features that can improve the model's ability to make accurate predictions.

Feature selection:-

→ It aims to reduce the dimensionality of dataset by selecting the most important and relevant features.

→ This step is crucial for preventing overfitting, improving model performance, reducing training time & enhancing interpretability.

Benefits of Feature Generation and Feature Selection:

1. Improved Model Accuracy
2. Reduced Overfitting
3. Efficiency
4. Interpretability

Motivating application :- User (customer) retention

↳ ability to keep hold onto something overtime

→ Suppose an app called Netflix or prime video charges a monthly subscription fee, with revenue increasing with more users.

→ However, only 10% of new users returns after 1st month.

→ To boost revenue, there are 2 options.

1. increase the retention rate of existing users
2. Acquire new users.

→ Generally, retaining existing customer is cheaper than acquiring new one.

→ Focusing on Retention, a model has to built to predict if a new customer will return next ~~month~~ month based on their behaviour this month.

→ The data science team, including game designers, software engineers, statistician & marketing experts, collaborates to identify relevant features.

Here are some example:-

- → No of days user visited in first month
- → Amount of time until second visit
- → No of points on day
- → Total no of points in first month
- → Did user fill out Netflix/any app profile.
- → Age & gender of user
- → Screen size of device (Mobile, laptop, TV)

$$P(\text{retention}) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_n x_n)}}$$

Customer retention is a crucial focus in data science because it directly impacts a company's revenue, customer lifetime value (CLV), and overall business success. Data science leverages large volumes of customer data to predict, understand, and improve retention rates through various methods. Here's how customer retention is used in data science:

1. **Predictive Modeling for Churn Prediction**
Data science uses predictive models to identify customers at risk of churning (leaving) by analyzing historical data and identifying patterns associated with past churners.

Techniques Used: Logistic regression, decision trees, random forests, gradient boosting machines (GBM), neural networks, and support vector machines (SVM).
Example: A telecom company might use historical usage data, call drop rates, billing information, and customer service interactions to predict which users are likely to leave.

2. **Segmentation and Clustering**
Customer segmentation groups users based on shared characteristics, behaviors, or needs, allowing companies to tailor retention strategies to different segments.
Techniques Used: K-means clustering, hierarchical clustering, DBSCAN, and Gaussian mixture models.
Example: An e-commerce platform segments customers into groups such as frequent buyers, occasional shoppers, or discount-seekers to target them with personalized offers.

3. **Personalization and Recommendation Systems**
Data science powers recommendation engines that enhance user experience by suggesting relevant products, services, or content, keeping users engaged.
Techniques Used: Collaborative filtering, content-based filtering, matrix factorization, and deep learning approaches like neural collaborative filtering.
Example: Netflix uses a recommendation system to suggest shows based on viewing history, thus increasing user engagement and retention.

4. **Sentiment Analysis and Feedback Analysis**
Sentiment analysis examines customer feedback, reviews, and social media mentions to gauge satisfaction levels and identify potential issues before they lead to churn.
Techniques Used: Natural language processing (NLP), text mining, sentiment analysis algorithms, and topic modeling.
Example: A retailer analyzes social media comments and product reviews to detect negative sentiments that might indicate dissatisfaction and proactively address customer complaints.

5. **Customer Lifetime Value (CLV) Prediction**
CLV models predict the total value a customer will bring to the business over their lifetime, guiding companies in how much to invest in retaining them.
Techniques Used: Regression models, survival analysis, Markov chains, and machine learning models tailored to estimate long-term value.
Example: An insurance company uses CLV predictions to determine which customers to target with premium offers or loyalty incentives.

6. A/B Testing and Experimentation
Data science uses A/B testing to evaluate the impact of retention strategies, such as changes in pricing, new features, or marketing campaigns, by comparing user engagement across different groups.
Techniques Used: Statistical hypothesis testing, Bayesian analysis, and multivariate testing.
Example: An online service tests different subscription models (monthly vs. annual) to see which improves retention rates.

7. Time Series Analysis for Retention Patterns
Time series analysis helps identify trends, seasonality, and patterns in customer behavior over time, allowing businesses to forecast and adjust strategies accordingly.

Techniques Used: ARIMA models, exponential smoothing, LSTM (Long Short-Term Memory) networks, and Prophet.
Example: A fitness app uses time series analysis to predict when user engagement might drop (e.g., post-holiday season) and proactively launches re-engagement campaigns.

8. Anomaly Detection for Unusual Behavior
Anomaly detection models identify unusual patterns in user behavior that might indicate potential churn, such as a sudden drop in activity or changes in spending habits.
Techniques Used: Isolation forests, autoencoders, statistical outliers, and clustering-based approaches.
Example: A gaming company detects when frequent players suddenly stop logging in, triggering personalized outreach efforts to re-engage them.

9. Customer Journey Analysis
Data science tracks and analyzes the entire customer journey to understand key touchpoints that influence retention, helping businesses optimize the user experience.
Techniques Used: Path analysis, funnel analysis, sequence mining, and Markov models.
Example: An airline analyzes booking, check-in, and flight experiences to find and fix pain points that might drive customers away.

10. Dynamic Pricing Models
Dynamic pricing models adjust prices based on customer data, demand fluctuations, and user behavior to retain price-sensitive customers without sacrificing revenue.
Techniques Used: Reinforcement learning, optimization algorithms, and regression analysis.
Example: Ride-sharing services use dynamic pricing to balance demand and supply while keeping regular customers satisfied with fair pricing strategies.

Benefits of Using Data Science for Customer Retention
Proactive Interventions: Anticipate churn and engage customers with timely and personalized offers or support.
Enhanced Personalization: Deliver tailored experiences that resonate with individual users, increasing satisfaction and loyalty.

Resource Optimization: Focus retention efforts on high-value customers, maximizing ROI on marketing and customer service efforts.
Data-Driven Decision Making: Use insights from data science to refine strategies, measure impact, and continuously improve retention tactics.
Leveraging data science in customer retention helps businesses stay competitive by turning data into actionable insights, driving more targeted and effective retention strategies.

Why User Retention Matters
Higher Profitability: Retained customers tend to spend more over time and are more likely to refer others, lowering acquisition costs.
Brand Loyalty: Consistent retention efforts build stronger relationships, leading to long-term brand loyalty and advocacy.
Data Insights: Engaged users provide valuable data that can help improve products and services, further driving innovation.
Competitive Advantage: A high retention rate signals to the market that your business provides lasting value, setting you apart from competitors.

Feature Generation :-

Brain Storming

Brain Storming is a Creative thinking technique for coming up with new ideas and solving problems.
→ Teams use this ideation method to encourage new ways to thinking and collectively generate solutions.
→ Brainstorming allows & encourage free thinking for all ideas to be voiced without judgment, innovative environment.

Benefits of Brain Storming

1. Encourage Creativity
2. Fosters collaboration and team building
3. Generates Innovatives, revolutionary idea.
4. Establishes different perspectives.

5. Introduces many ideas quickly.

→ In data science, brain storming plays a crucial role in problem formulation, feature selection, model design & strategy development.

Key Principles of Brainstorming:—

1. Defer Judgement:- In this all Ideas are accepted, and no one criticizes or dismisses an idea.

2. Focus on Quality:- The goal is to generate as many ideas as possible. More idea may lead to greating chance of finding effective solution.

3. Stay focused on Topic:- While thinking freely, its important to maintain focus on the core problem or challenge

Types of Brainstorming:—

1. Group
2. Individual
3. Silent Brainstorming (Brainwriting)
   ↳ Ideas are written down rather than spoken out.

Steps:-

1. Define the problem
2. Set Ground Rule
3. Choose a Mentor
4. Generate Ideas
5. Record All Idea
6. Evaluate Ideas later.

Apply various brainstorming techniques and domain expertise to generate a comprehensive set of features for predicting customer retention, explaining each step in detail.

Brainstorming Techniques and Domain Expertise for Generating a Comprehensive Set of Features to Predict Customer Retention:

We'll employ several brainstorming techniques, such as Mind Mapping, SCAMPER, 5 Whys, and SWOT Analysis, alongside domain expertise in customer behavior, data science, and machine learning, to create a rich set of features for predicting customer retention.

1. Mind Mapping: Visualizing Connections

Step:

We start by creating a central idea (customer retention) and radiate potential influencing factors around it, such as behavioral data, demographics, engagement, external factors, business model, and satisfaction levels. Each category spawns multiple related sub-features.

Features Generated:

Behavioral Data:

Time Since Last Interaction: Track the time gap since the last customer interaction or purchase.

Usage Frequency: Analyze how often customers use the product/service.

Transaction Patterns: Monitor purchase frequency, size, and product type preferences.

Engagement:

Email Engagement: Monitor open rates, click-through rates, and responses to email campaigns.

Customer Support Interactions: Track the number and sentiment of customer support requests.

Social Media Activity: Check customer mentions, likes, comments, or direct engagement on social platforms.

Satisfaction:

Net Promoter Score (NPS): Collect and monitor customer satisfaction levels based on feedback.

CSAT (Customer Satisfaction Score): Measure post-service satisfaction to gauge loyalty.

Survey Results: Analyze feedback from user surveys for insights into customer happiness.

2. SCAMPER: Modifying Existing Concepts

Step:

SCAMPER helps us Substitute, Combine, Adapt, Modify, Put to another use, Eliminate, and Rearrange features. We take standard retention metrics and innovate around them by rethinking how they are used or measured.

Features Generated:

Substitute:

Instead of only relying on basic demographic data, use psychographic profiling (e.g., customer values, lifestyle choices).

Combine:

Merge product usage data with customer support ticket trends to create a unified "customer frustration index" that tracks whether usage correlates with increased support requests.

Adapt:

Use machine learning algorithms designed for other industries (like fraud detection models) to analyze suspicious behavior that may precede churn.

Modify:

Modify typical customer segmentation by adding new dimensions, such as brand sentiment scores derived from external data sources like reviews or third-party feedback.

Eliminate:

Eliminate unnecessary complex algorithms and prioritize simpler, explainable models that sales teams can better understand to predict churn likelihood.

Rearrange:

Rearrange traditional cohorts, such as focusing less on large customer segments and more on micro-cohorts, with highly specific behavior profiles (e.g., "heavy users, high-engagement, but negative social sentiment").

3. 5 Whys: Root Cause Analysis for Churn

Step:

Ask "Why?" repeatedly to drill down into the underlying reasons behind customer churn. This will help us identify potential predictive signals and uncover less-obvious factors leading to retention or churn.

Features Generated:

Why did customers stop engaging?

Product Complexity Index: Measure how difficult it is for users to navigate features. High complexity may be predictive of frustration.

Why did support tickets increase?

Support Response Time: Longer response times could correlate with churn, as slow service may frustrate customers.

Why did feedback scores drop?

Competitor Pricing Fluctuations: Price-sensitive customers may be drawn away by more affordable alternatives.

Why did engagement decline despite positive feedback?

Feature Fatigue: Too many features being released rapidly may overwhelm customers, leading to dissatisfaction.

4. SWOT Analysis: Exploring Strengths, Weaknesses, Opportunities, and Threats

Step:

Applying SWOT to customer retention allows us to explore the strengths and weaknesses in the current model, opportunities for new features, and potential threats from external factors.

Features Generated:

Strengths:

Customer Lifetime Value (CLV) Analysis: Identify high-value customers and predict retention based on the profitability of their relationship with the company.

Weaknesses:

Lagging Satisfaction Feedback: Customer feedback (like NPS) often comes after issues have already escalated. Introduce Proactive Satisfaction Indicators like monitoring "exit clicks" on websites or premature session endings as early indicators of discontent.

Opportunities:

Cross-Sell/Upsell History: Customers who are frequently upsold may show high engagement and therefore higher retention likelihood.

Community Involvement Metrics: Use metrics from community boards or user-generated content platforms to assess customer loyalty through brand advocacy.

Threats:

Competitor Market Entrances: Sudden dips in engagement could be a reaction to new competitors. Implement Competitive Pricing Alerts, where customer behavior is compared against fluctuations in competitor offerings.

5. Domain Expertise: Data Science and Customer Behavior

Step:

Leverage knowledge of machine learning, data science, and behavioral psychology to identify predictive features that align with real-world customer behaviors and patterns.

Features Generated:

RFM (Recency, Frequency, Monetary) Scoring:

Develop a customer value model based on how recently, frequently, and how much a customer has spent.

Customer Journey Analysis:

Break down the customer journey into key milestones (onboarding, first purchase, second purchase, renewal) and measure how far customers progress. Customers that don't reach key milestones are flagged as at-risk.

Churn Drivers from External Data:

Integrate external datasets such as economic indicators (e.g., local unemployment rates, economic downturns) to predict macro-level impacts on customer retention.

Sentiment Analysis of Customer Communications:

Apply sentiment analysis to all customer communications (emails, chats, reviews) and assign a Customer Sentiment Score to predict the likelihood of retention or churn based on positive/negative interactions.

Social Proof & Referral Influence:

If customers have referred others or been referred by someone, track how social connections influence retention rates.

Final Comprehensive Set of Features:

Behavioral:

Usage frequency, transaction patterns, time since last interaction, support ticket sentiment.

Engagement:

Email engagement, social media interactions, proactive satisfaction indicators (e.g., exit clicks).

Sentiment:

NLP analysis on customer support or feedback, social proof/referral influence, churn based on sentiment scores.

Monetary and Transactional:

Recency, frequency, monetary scoring (RFM), CLV, cross-sell/upsell history.

External Data:

Competitor pricing, economic downturns, market conditions, and community involvement metrics.

Psychographic Data:

Personal values, lifestyle choices, brand sentiment analysis based on third-party reviews.

Milestone Progression:

Key points in the customer journey (onboarding, first purchase, retention markers).

8121350256

Resource Optimization: Focus retention efforts on high-value customers, maximizing ROI on marketing and customer service efforts.
Data-Driven Decision Making: Use insights from data science to refine strategies, measure impact, and continuously improve retention tactics.
Leveraging data science in customer retention helps businesses stay competitive by turning data into actionable insights, driving more targeted and effective retention strategies.

Why User Retention Matters
Higher Profitability: Retained customers tend to spend more over time and are more likely to refer others, lowering acquisition costs.
Brand Loyalty: Consistent retention efforts build stronger relationships, leading to long-term brand loyalty and advocacy.
Data Insights: Engaged users provide valuable data that can help improve products and services, further driving innovation.
Competitive Advantage: A high retention rate signals to the market that your business provides lasting value, setting you apart from competitors.

Feature Generation from Raw data.

1. Recency, Frequency features
   ↳ Time since last transaction or interaction: Customer with long gap since their last transaction are more likely to churn (leave).
   → Frequency: No of transaction within a specific time frame. (eg. Monthly, quately).

2. Engagement features:-
   → Days Active.
   → Average Transaction Interval: Measure average time between transaction. An increased time interval may indicate to churn.
   → Subscription plan:- changes from yearly to month & from ~~Automatic to Manual~~ frequent downgrades or cancellations.

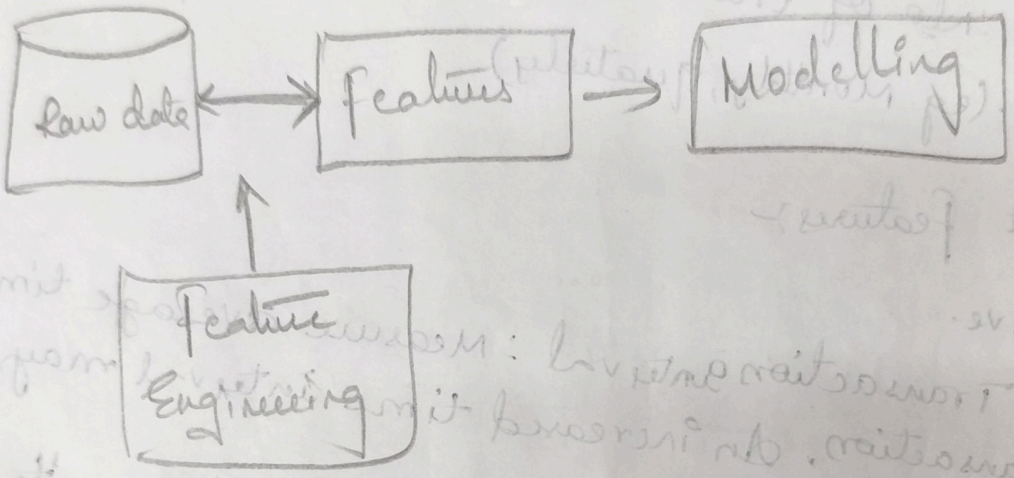
3. Behavioural features:-
   → Preffered Transaction time:- time & day of transaction occur. Change in behaviour might suggest churn.

4. Payment Behaviour:-
   → Failed payment Attempts:- High frequency of failed payments can indicate financial difficulties.
   → paymethod changes:- change in payment method (from automatic to manual payment) early sign of churn.

5. Discount & promotion Response:-
   → customer responses to discount or promotional offers.

6. Customer Support Interaction :-

→ Complaints & Transaction : Ratio of complaints or refund requests to total transaction
→ Combine several behavior patterns into a score that estimate the likehood



Feature selection algorithms-

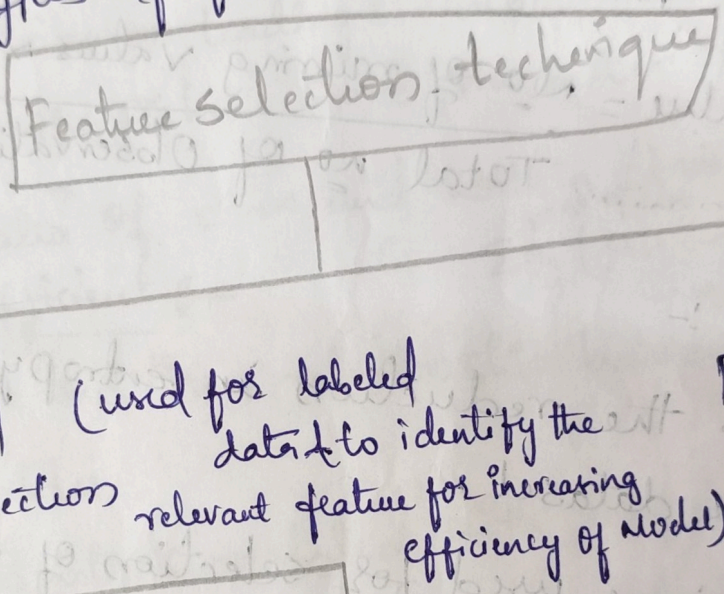
→Feature selection :-
→ A feature is an attribute that has an impact on a problem or is useful for the problem, & choosing the important features for the model is known as feature selection.
→Feature selection is a way of reducing the input variable for the model by using only relevant data in order to reduce overfitting in the model.
→ "It is a process of automatically or manually selecting the subset of most appropiate and

Relevant features to be used in model building

Benefits Of using feature selection in ~~Data~~ Data science—

1. It helps in avoiding the curse of dimensionality.
2. It reduces the training time.
3. It reduces overfitting hence enhances the generalization.

→ There are 2 types of feature selection techniques.



6. Missing Value Ratio :-

→ The value of missing value ratio can be used for evaluating the feature set against the threshold value.

→ The formula for obtaining the missing value is ratio of number of missing value in each col^n divided by total number of observation.

→ The variable which is having more than the threshold value can be dropped.

$$\text{Missing value} = \frac{\text{No of missing values} * 100}{\text{Total no of Observations.}}$$

Information Gain :-

→ It determines the reduction in entropy while transforming the dataset.

→ It is a technique used for selection of feature by calculating the information gain of each variable with respect to target variable.

entropy is a measure of uncertainty of a random variable.

$$\text{Gain}(S, A) = \text{Entropy}(S) - \sum \frac{S_v}{S} \text{Entropy}(S_v)$$

S → dataset, A → a particular attribute

$$\text{Info}_A(D) = \sum_{j=1}^{v} \frac{|D_j|}{D} * \text{Info}(D_j)$$

Chi-square test :- (X^2)

↳ It is generally used to test the relationship between Categorical variable.

→ It Compares the observed value from different attributes of dataset to its expected value.

$$X^2 = \sum \left(\frac{\text{observed value} - \text{Expected value}}{\text{Expected value}} \right)^2$$

Fisher's Score ←

→ It Select each feature independently according to their score under fisher criterion leading to suboptimal set of features.

→ The larger the fisher's score, the better is the selected feature.

→ It is ratio of class and within class variance.

$S_i = \sum n_j (\mu_{ij} - \mu_i)^2 \; \sum n_j * p^2_{ij}$

(n_j → no of inst, μ_i → mean, μ_{ij} → mean of i,j, p^2_{ij} → variance of i,j)

Correlation Coefficient ←

→ It Measures the linear relationship between the features and target variables.

→ Features highly correlated with target are considered important.

Pearson correlation (for nominal value).

$$P_{x,y} = \frac{Cov(x,y)}{\sigma_x \sigma_y}$$

$Cov(x,y)$ = covariance

σ_x – Standard deviation of X

σ_y – " " of Y.

Dispersion Ratio:-

→ It is defined as the ratio of Arithmetic Mean (AM) to that of Geometric Mean (GM) for a given feature.

→ Its value ranges from +1 to ∞.

→ Higher the dispersion ratio implies a more relevant feature.

Relief:-

→ This method measures the quality of attributes by randomly sampling an instance from dataset & updating each feature & distinguishing between instances.

Mutual Dependence:-

→ This method measures if two variables are mutually dependent, & thus provides the amount of information obtained for one variable on observing the other variable.

→ Depending on presence/absence of a feature, it measure the amount of information that feature contribute to making the target prediction.

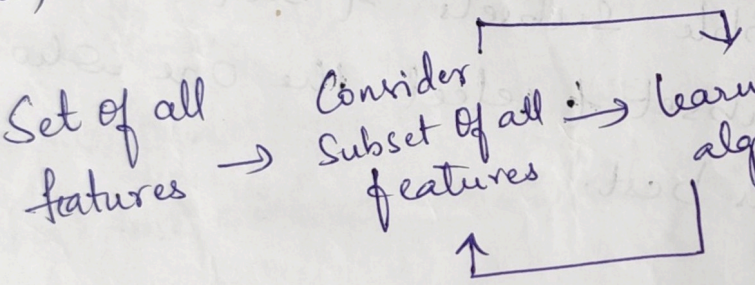
Set of all features → Selecting the best subset → Learning algorithm → performance.

Wrapper Methods:-

→ Wrapper Methods, also referred to as greedy algorithms train the algorithm by using subset of features in an iterative manner.

→ Stop selecting the subset when the performance of model decreases or a specified number of features has been achieved.

→ The advantage of wrapper method over filter methods is that they provide an optimal set of features for training model, thus resulting in better accuracy.



Techniques-

1. forward Selection:-

→ This method is an iterative approach where we initially start with an empty set of feature & keep adding a feature which best improves model after each iteration.

→ The stopping is done until the addition of new variable does not improve performance of model.

Backward Elimination :-

→ This is an iterative approach, where we initially start with all features and after each iteration, we remove the least significant feature.

→ The stopping is till no improvement in the performance of model is observed after the feature is removed.

Bi-Directional Elimination :-

→ This method uses both forward selection and backward elimination technique simultaneously to reach one unique solution.

Exhaustive Selection :-

→ This technique is considered as the brute force approach for the evaluation of feature subset.

→ It creates all possible subsets & build a learning algorithm for each subset & select the one whose model's performance is best.

Recursive Elimination :-

→ This greedy optimization method selects features by recursively considering the smaller & smaller set of features.

→ The estimator is trained on an initial set of feature & importance is obtained using feature-importance-attribute.

→ The least important features are then removed from current set of features till we are left with required number of features.

Embedded Methods:-

→ Embedded methods perform feature selection during model training process.

→ These methods are combination of filter & wrapper approach. & are less computationally expensive than wrapper methods.

Techniques /methods:-

1. Lasso Regression (L_1 Regularization) :-
   → Adds a penalty to cost function that encourages the model to shrink less important feature coefficients to zero.
2. Ridge Regression (L_2 Regularization) :-
   → Adds a penalty to cost function that shrinks feature coefficient, but without eliminating them.
3. Elastic Net :- A combination of L_1 & L_2 Regularization.
4. Decision Trees & Randomforest :-
   Can naturally rank features, based on their importance.

Steps in feature selection :-

1. understant the Data :- perform EDA to understand rel^{n}ship b/w variable & target variable
2. Remove Redundant feature :- Drop highly correlated feature
3. Apply Feature selection Techinque :- Use any Filter, wrapper Method to select feature.

④.

4. Evaluate model performance:-

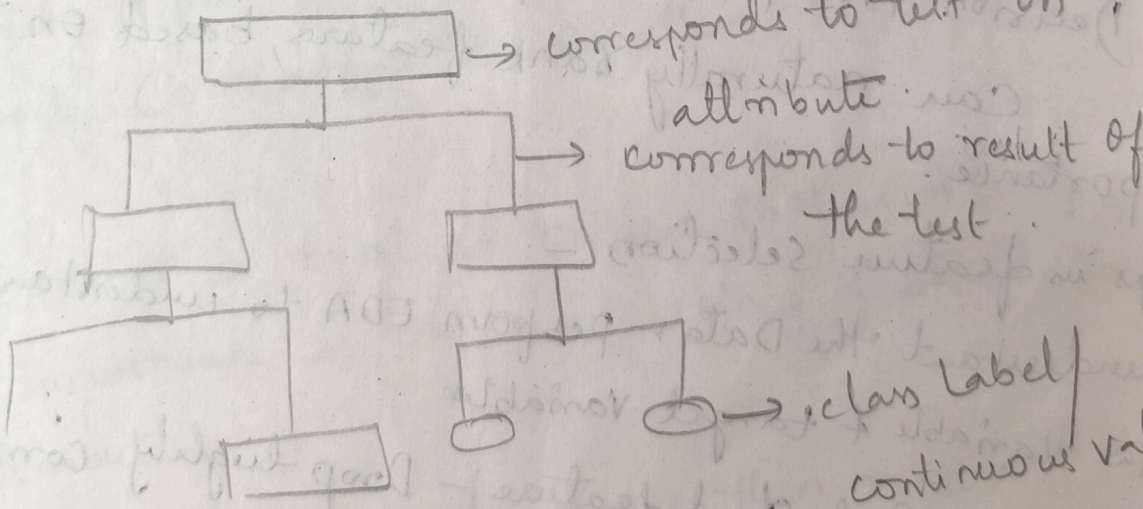
After selecting features, evaluate model performance using cross-validation or other techniques.

Decision Trees:-

→ Decision Tree is powerful machine learning algorithm that can be used for feature selection.

→ They provide a clear & intuitive way to make decision based on data by modeling the relationships b/w different variable.

→ A decision tree is a flow-chart like structure used to make decision or predictions. It contains nodes representing decision or test on attributes, branches representing the outcome of ~~predictio~~ decision & leaf representing final outcome or prediction.



in R

```
install.package("titanic").  // package installation
library(titanic)
data("titanic_train").  // load dataset
str(titanic_train)  // structure of dataset
titanic_clean <- na.omit(titanic_train)  // remove missing value
~~tree_model <- rpart(~~
titallic_clean$survived <- as.factor(titanic_clean$survived)
                    // factor for classification
tree_model <- rpart(survived ~ pclass + Age + Gender + fare,
                    data = titanic_clean, method = "class")
Summary(tree_model)  // view tree model
pred <- predict(tree_model, titanic_clean, type = "class")
confusion_matrix <- table(titanic_clean$survived,
                          pred)
print(confusion_matrix).
```

attribute

Advantages :-

1. Simplicity & Interpretability :-
   → They are easy to understand & interpret. The visual representation closely useful for human decision-making process.

2. No need of Feature Scaling:-
It does not require Normalization or Scaling of data

3. Handle Non-linear data:-
Capable of handling Non-linear relationship b/w feature & target variable.

Disadvantages:-

1. Overfitting:- Decision Trees can overfit the training data, especially when they are deep with many nodes

2. Bias towards features with more levels:-
Features with more level can dominate the tree structure.

3. Instability:- Small change in data can result in completely different tree being generated.

Random Forest :-

→ Random forest is a powerful tree learning technique in Machine Learning.

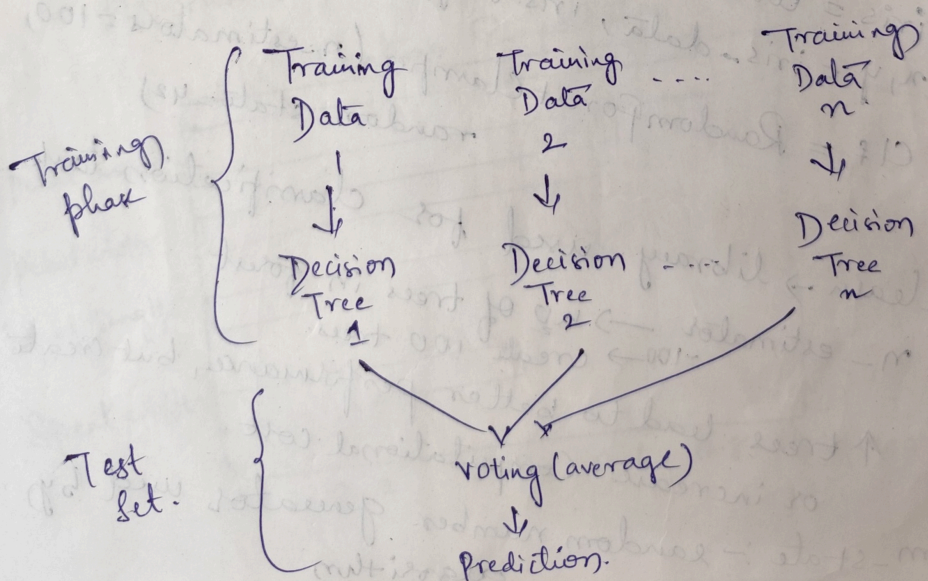
→ It works by creating a number of Decision Trees during the training phase.

→ It is a Supervised learning technique can be used for both classification & Regression problem in ML.

→ It is based on Concept of ensemble learning, which is a process of combining multiple classifier to solve a complex problem & to improve performance of model.

→ Random Forest is a classifier that contains a number of decision Trees on various subsets of given dataset and takes the average to improve the predictive accuracy of dataset.

→ The greater the number of trees in the forest leads to higher accuracy & prevents the problem of Overfitting.



Why to use Random forest:-

→ It take less training time as compared to other algorithm.
→ It predict o/p with high accuracy, even for large dataset & it run efficiently

How feature selection is Done:-

1. Load the dataset
2. Split it into training & testing set.
3. Fitted a random forest classifier.
4. Extract and visualize feature importance.
5. Select features based on importance.
6. Evaluate the model using only selected feature.

ex:-
```
from sklearn.datasets import load_iris
from sklearn.ensemble import RandomForestClassifier
iris = load_iris()
x, y = iris.data, iris.target
clf = RandomForestClassifier(n_estimators=100,
                             random_state=42)
```

Sklearn → library used for classification task.
n_estimator → no of trees in forest.
=100 → create 100 trees.
↑ trees lead to better performance, but create or increase Computational cost.
random_state :- random number generator used by algorithm

clf.fit(x,y)

→ fit method is called to train the model on provided data. It adjust the model parameters on training data.

x - input features.

Y - target variable.

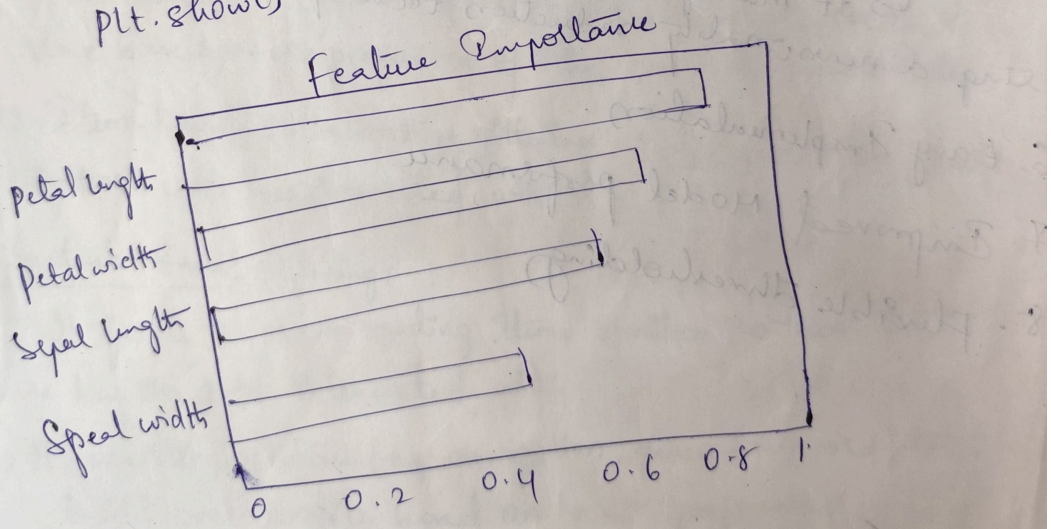
→ It create Multiple Decision trees based on training data

importances = clf.feature_importance.

Plot the feature importance

```
import matplotlib.pyplot as plt.
plt.bar(range(X.shape[1], importance)
plt.xticks(range(X.shape[1], iris.feature_names,
                    rotation = 90)
plt.title(" ").
plt.show()
```

↓ names of dataset



shape → return a tuple representing array or Dataframe

First column → no of rows

Second column → no of col^n x.shape[1] = no of col^n

Need of Random Forest in feature selection :-

1. Robustness to Overfitting
2. Automatic Feature importance measurement.
   ↳ MDI (Mean Decrease Impurity)
3. Handling Non-linear Relationships.
   ↳ It Capture non-linear Rel^ship b/w features & target variable.
4. Handling High-Dimensionality
5. Ability to Handle correlated Features.
   ↳ It manage correlated feature without requiring dimensionality reduction technique.
6. Easy Implementation.
7. Improved Model performance.
8. Flexible Thresholding

UNIT-IV

Recommendation System

→ Recommendation System provides personalized suggestions to users by leveraging data about their preferences, behavior and interaction.

→ In simple words, it is an algorithm that suggests relevant items to users.

Types of Recommendation Systems :-

1. Collaborative filtering :-

→ It relies on the assumption that users with similar preferences in the past will continue to have similar preferences in future.

→ It can be of two types.

limit :-
→ It cannot handle new items b/c → not trained, this is called cold-start problem

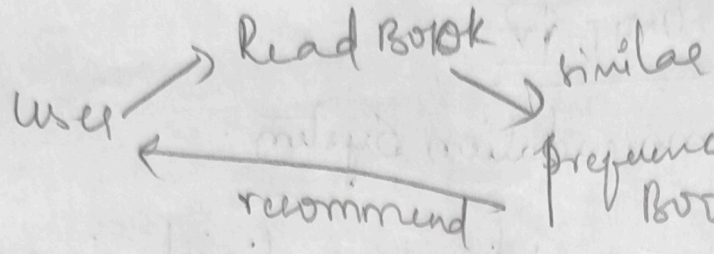
1. User Based collaborative filtering :- Recommends items to user based on the preference of other user with similar taste.

2. Item-Based collaborative filtering :- Recommends item similar to those user has interacted with or rated highly.

2. Content-Based Filtering :-

→ It focus on recommending items similar to those the user has liked or interacted with.

→ It uses item feature (eg. description, genre, & keyword) to build user profile based on past preferences.



Limitation:-
→ Make recommendation based on existing interest of user.

3. Hybrid Method:-
→ It Combine both Collaborative & Content-based filtering to overcome the limitation of each approach.
eg:- It can recommend new items (avoiding cold-start problem) & take advantage of user ~~pref~~ preference to improve accuracy

4. Matrix factorization:-
→ Techniques such as SVD & Alternating least Squares (ALS) decompose the user-item interaction matrix into lower-dimensional matrices to identify features, which capture rel^{n}ship between user & item.

Challenges in Recommendation System:-

1. Cold-start problem:-
→ It occurs when there is insufficient data for new users or items, making it difficult to generate recommendation.

2. Data Sparsity:-
→ In large recommendation system, the user-item interaction matrix is often sparse (most user do not interact with most items). Matrix Factorization ~~Here~~ helps in filling the gap by latent factors.

3. Scalability:-

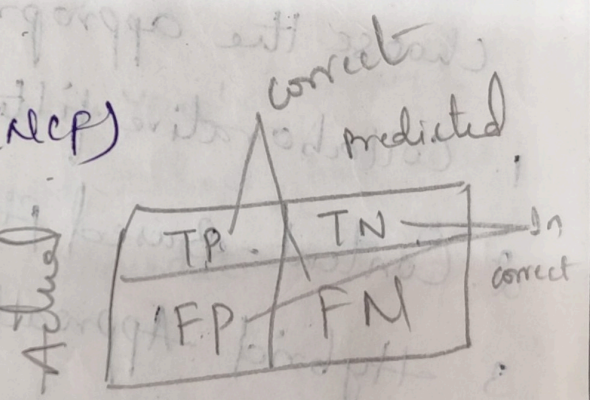
→ Large Scale Systems need to process millions of users and items in real-time, requiring efficient algorithm & distributed systems for computation.

4. Diversity Vs Accuracy:-

→ A balance must be made between recommending relevant items (accuracy) & offering a diverse range of suggestion to prevent monotonous recommendation.

Popular Algorithms in Recommedation System:-

1. Singular Value Decomposition (SVD)
2. K-NN
3. Alternating Least Square (ALS)
4. Neural collaborating Filtering (NCF)
5. Latent Dirichlet Allocation (LDA)



Evaluation Metrics:-

1. Precision = $\frac{TP}{TP+FP}$
2. Recall = $\frac{TP}{TP+FN}$
3. F1 Score = 2 * recall * precision / recall + precision
4. Root Mean Square Error
5. Mean Average precision

Application:-

1. E-commerce
2. Streaming servic
3. Social Media
4. Content platform - youtube.

Building a Recommender System :-

1. Data Collection :-
Gather data on user interactions, preferences and item attribute. This data can be collected from user rating, clicks, views and purchase histories.

2. Data Preprocessing :-
clean & preprocess the data to handle missing value, normalize & transform into a suitable form.

3. Method Selection :-
choose the appropriate recommendation techniques.
1. Collaborative filtering for user behaviour pattern
2. Content-Based filtering for utilizing item attribute
3. Hybrid Approach for combining multiple method.

4. Model Training :-
→ Train the choose model using historical data

5. Evaluation :- Evaluate the model using different Metrics for prediction accuracy, precision and recall for classification models.

6. Deployment:-

→ Deploy the model to production environment, where it can generate real-time recommendation for users.

7. Monitoring & optimization:—

→ Continuously monitor the performance of the recommender system & fine-tune the model based on user feedback & changing behavioural pattern.

Algorithmic review:—

→ In the nearest neighbour algorithm, if you want to predict whether user A likes something, you look at a user B closet to user A.

→ Once we identified a similar user, you'd then find something that user A had not rated, but user B has rated and liked and use that as recommendation for user A.

Some problems with Nearest Neighbour:—

1. Curse of dimensionality.

→ There are too many dimensions, so the closest neighbors are too far away from each other to realistically be considered.

2. Overfitting

→ Overfitting is one of the problem

3. Correlated features

→ There are tons of features, moreover, that are highly

Correlated with each other.

4. Sparseness
→ If you table is too Sparse, have lot of missing data, then most things are unknown & distance means nothing.

5. Measurement errors:-
→ There may be a measurement error (called reporting errors)

6. Computational Complexity.
→ There is a Calculation cost - Computational Complexity.

7. Preferences change Overtime
→ User preference may also change overtime, which falls outside the model.

8. Cost to update
→ It is also expensive to update the model as we add more data
→ The biggest issues are the first two on the list.

Dimensionality Reduction

→ In dimensionality reduction, data encoding or transformations are applied to obtain a reduced or "compressed" representation of original data.

→ If the original data can be constructed from compressed data without any loss of information, the data reduction is called lossless.

→ If we construct only an approximation of the original data, then the data reduction is lossy.

→ It categories the techniques based on Feature selection & Feature extraction.

→ Feature extraction involves creating a new features that are combination of original ones, often in a lower dimensional space.

1. principal Component Analysis:- projects data onto a lower dimensional space by finding the directions of maximum variance.

→ PCA transforms the data into a new co-ordinate system, where the greatest variance of data lies on the first co-ordinate (called the first principal component), the second greatest variance on second coordinate.

Steps in PCA:-

1. Standardize the data
→ Since PCA is sensitive to scale of data, it's essential to standardize features to have a mean and standard deviation.

2. Compute the Covariance Matrix.
→ Calculate the Covariance Matrix, which shows the relⁿship between the variables in the data.

3. Calculate Eigenvalues and Eigenvectors.
→ Eigenvector (directions) and Eigenvalues (Magnitude) of the covariance matrix are calculated to understand the Principal Components direction & importance.

4. Sort Eigenvalues and Select principal Components.
→ Sort eigenvalues in descending order and chose the top k eigenvector based on amount of variance.
→ the top eigenvectors form the new feature space

5. Transform the original dataset.
→ Finally, transform the data into new features space by multiplying original data with Selected eigenvectors.
→ the result is a lower-dimensional representation of original dataset.

Example of PCA :-

→ Given data in Table, reduce the dimension from 2 to 1 using the principal component Analysis (PCA) algorithm.

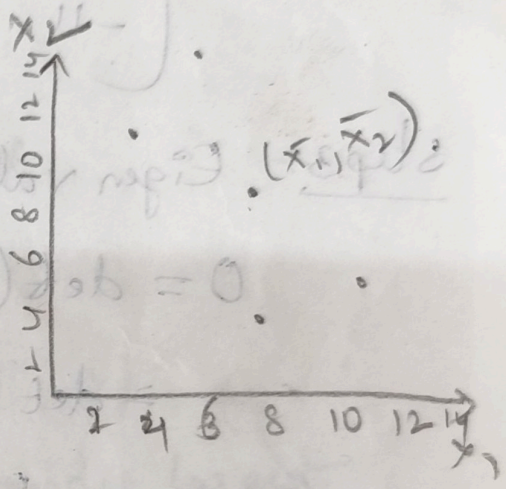
Features	EX1	EX2	EX3	EX4
x_1	4	8	13	7
x_2	11	4	5	14

Step 1:- Calculate Mean

$$\Rightarrow \bar{x}_1 = \frac{1}{4}(4+8+13+7)$$

$$= \frac{1}{4}(32) = 8$$

$$\Rightarrow \bar{x}_2 = \frac{1}{4}(11+4+5+14) = 8.5$$



Step 2:- Calculate the Covariance matrix.

$$S = \begin{bmatrix} \text{cov}(x_1, x_1) & \text{cov}(x_1, x_2) \\ \text{cov}(x_2, x_1) & \text{cov}(x_2, x_2) \end{bmatrix}$$

$$\text{cov}(x_1, x_1) = \frac{1}{N-1} \sum_{k=1}^{N} (x_{1k} - \bar{x}_1)(x_{1k} - \bar{x}_1)$$

$$= \frac{1}{3}\left((4-8)^2 + (8-8)^2 + (13-8)^2 + (7-8)^2\right)$$

$$= 14.$$

$$\text{Cov}(x_1, x_2) = \frac{1}{3}(x_{1k} - \bar{x}_1)(x_{2k} - \bar{x}_2)$$

$$= \frac{1}{3}\left((4-8)(11-8.5) + (8-8)(4-8.5) + (13-8)(5-8.5) + (7-8)(14-8.5)\right)$$

$$= -11$$

$$Cov(x_2, x_1) = Cov(x_1, x_2)$$
$$= -11$$

$$Cov(x_2, x_2) = \frac{1}{N-1} \sum_{k=1}^{N} (x_{2k} - \bar{x}_2)(x_{2k} - \bar{x}_2)$$
$$= \frac{1}{3}\left((11-8.5)^2 + (4-8.5)^2 + (5-8.5)^2 + (14-8.5)^2\right)$$
$$= 23$$

$$S = \begin{bmatrix} 14 & -11 \\ -11 & 23 \end{bmatrix}$$

Step 3:- Eigen value of covariance matrix.

$$0 = \det(S - \lambda I)$$
$$= \det\left(S - \lambda \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\right) = \det\left(S - \begin{pmatrix} \lambda & 0 \\ 0 & \lambda \end{pmatrix}\right)$$

$$\det\begin{pmatrix} 14-\lambda & -11 \\ -11 & 23-\lambda \end{pmatrix}$$

$$= \begin{vmatrix} 14-\lambda & -11 \\ -11 & 23-\lambda \end{vmatrix}$$

det =

$$(14-\lambda)(23-\lambda) - (-11)(-11)$$
$$= \lambda^2 - 37\lambda + 201$$
$$\lambda = \frac{1}{2}(37 \pm \sqrt{565})$$
$$= 30.3849,\ 6.615$$
$$\lambda_1 \quad \lambda_2$$

Step 4:- calculate eigenvectors

$$U = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \qquad (S - \lambda I)U = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$\begin{vmatrix} 14-\lambda & -11 \\ -11 & 23-\lambda \end{vmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$$

$$= \begin{bmatrix} (14-\lambda)u_1 - 11u_2 \\ -11u_1 + (23-\lambda)u_2 \end{bmatrix}$$

$$= (14-\lambda)u_1 - 11u_2 = 0 \quad , \quad -11u_1 + (23-\lambda)u_2 = 0$$

$$= 14-\lambda u_1 = 11u_2$$

$$\frac{u_1}{11} = \frac{u_2}{14-\lambda} = 0 \Rightarrow \frac{u_1}{11} = \frac{u_2}{14-\lambda} = t$$

$$u_1 = 11t, \; u_2 = (14-\lambda)t$$

$$\frac{u_1}{u_2} = \frac{11}{14-\lambda}.$$

$$U = \begin{bmatrix} 11 \\ 14-\lambda \end{bmatrix}$$

→ for calculate principal components consider largest eigen values

$$\lambda = 30.3849.$$

→ To find the unit eigen vector,

$$\|U_1\| = \sqrt{(11)^2 + (14-\lambda_1)^2}$$

$$= \sqrt{(11)^2 + (14-30.3849)^2}$$

$$= 19.7348$$

$$e_1 = \begin{bmatrix} 11/u_1 \\ 14-\lambda/u_1 \end{bmatrix} = \begin{bmatrix} 11/19.734 \\ \frac{14-30.38}{19.734} \end{bmatrix} = \begin{bmatrix} +0.5574 \\ -0.8303 \end{bmatrix}$$

Step 5 :- Compute first principal components

$$e_1^T \begin{bmatrix} x_{1k} - \bar{x}_1 \\ x_{2k} - \bar{x}_2 \end{bmatrix}$$

$$\begin{bmatrix} 0.5574 \\ -0.8303 \end{bmatrix}^T \begin{bmatrix} x_{11} - \bar{x}_1 \\ x_{21} - \bar{x}_2 \end{bmatrix}$$

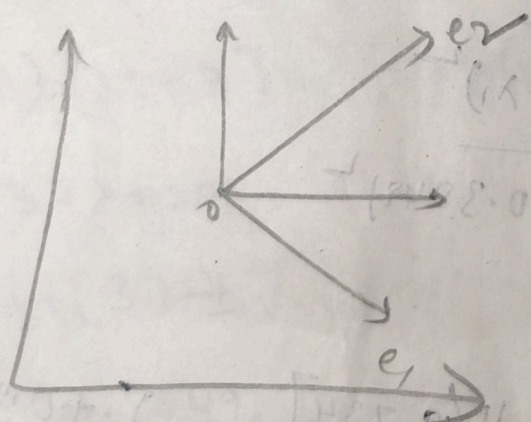
$$\begin{bmatrix} 0.5574 & -0.8303 \end{bmatrix} \begin{bmatrix} 4-8 \\ 11-8.5 \end{bmatrix}$$

$$= 0.5574(-4) - 0.8303(2.5)$$

$$= -4.30535$$

Calculate for all.

features	EX1	EX2	EX3	EX4
x_1	4	8	13	7
x_2	11	4	5	14
first principal components	-4.305	3.7361	5.6928	-5.1238



Ex-2

f1	2.5	0.5	2.2	1.9	3.1
f2	2.4	0.7	2.9	2.2	3.0
f3	3.0	0.5	2.5	2.0	3.5

Impact on Computational Efficiency:-

1. Faster model Training and prediction:

→ Reducing dimensionality with PCA lowers the number of features, which reduces data preprocessing requirements.

→ This leads to faster training times for ML models, including collaborative filtering or matrix factorization models often used in recommendation.

2. Reduced Storage & Memory Requirements:

→ Smaller dataset require less memory for storage & are easier to manipulate.

→ PCA's dimensionality reduction allow for compact representation that save computational resources.

Impact on Recommendation Accuracy:-

1. Better Generalization:-

→ A better reduced, noise-free features set often leads to model that generalize better to new data.

2. Reduced Overfitting:-

→ Removing redundant and irrelevant features with PCA reduces the risk of overfitting, particularly in high-dimensional data.

→ Overfitting can lead to inaccurate recommendations, as the model might become too sensitive to noise & specific pattern in training data.

C → Creating recommendation engine with matrix factorization involves multiple steps from data preprocessing to model training & tuning.

4. → Matrix factorization technique like singular value Decomposition (SVD) or Alternating least Square (ALS) are popular in recommendation system as they effectively handle Sparse datasets.

5. 1. Data collection & preparation
   → Gather data
   → Handle sparsity
   → Train/Test

6. 2. Matrix factorization Application
   → Use either ALS or SVD

3. Model Training
   → Training the model use an optimization Algorithm (ALS) to learn the optimal value for user & item matrices.
   → Evaluate on validation set using metrics such as Root Mean Squared Error (RSME) for rating based system.

4. Improve Recommendation Accuracy:

5. Model Evaluation & Deployment.
   → Evaluate on training set
   → Deploy the model
   → Monitor & update

Application
1. Data visualization
2. Noise reduction
3. Feature engineering
4. Compression:

Singular Value Decomposition (SVD) :-

→ Singular Value Decomposition is a powerful matrix factorization technique used in many areas, including recommendation systems, data compression & signal processing.

→ It decomposes a matrix into 3 distinct matrices, which can simplify the understanding & manipulation of high-dimensional data.

Given a Matrix A with dimensions $m \times n$, the SVD of A is defined as

$$A = U \Sigma V^T.$$

U is an $m \times m$ orthogonal matrix, whose columns are the left singular vector of A.

Σ is an $m \times n$ diagonal matrix, containing the singular values of A

V^T is transpose of an $n \times n$ orthogonal matrix, whose columns are right Singular vector of A.

Application of SVD.

1. Dimensionality Reduction.

→ SVD helps to reduce dimensions by keeping only largest singular value, representing most critical information.

2. Recommender System

→ SVD can decompose a user-item rating matrix, identifying latent factor representing user preferences & item characteristic.

Example:-

$$A = \begin{bmatrix} 1 & 1 \\ 7 & 7 \end{bmatrix}_{m \times n}$$

SVD of $A = U_{m \times m} \Sigma_{m \times n} V^T_{n \times n}$

$$A = \begin{bmatrix} 1 & 1 \\ 7 & 7 \end{bmatrix}, A^T = \begin{bmatrix} 1 & 7 \\ 1 & 7 \end{bmatrix}$$

Calculate $A \times A^T = \begin{bmatrix} 1 & 1 \\ 7 & 7 \end{bmatrix} \begin{bmatrix} 1 & 7 \\ 1 & 7 \end{bmatrix}$

$$U = AA^T = \begin{bmatrix} 2 & 14 \\ 14 & 98 \end{bmatrix}$$

Calculate $A^TA = \begin{bmatrix} 1 & 7 \\ 1 & 7 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 7 & 7 \end{bmatrix}$

$$V = A^TA = \begin{bmatrix} 50 & 50 \\ 50 & 50 \end{bmatrix}$$

Left Singular vector U,

find eigen values & eigen vectors of $AA^T = U$.

$$AA^T - \lambda I = 0$$

$$\begin{bmatrix} 2-\lambda & 14 \\ 14 & 98-\lambda \end{bmatrix} = 0$$

$$\Rightarrow (2-\lambda)(98-\lambda) \Rightarrow 14 \times 14 = 0$$

$$= \lambda^2 - 100\lambda + 196 - 196 = 0 \Rightarrow \lambda^2 - 100\lambda = 0$$

$$\lambda(\lambda - 100) = 0$$

$\lambda = 0, \lambda = 100$

eigen vector 1st $\lambda = 0$ call $V = \begin{bmatrix} x \\ y \end{bmatrix}$

$$\begin{bmatrix} 2 & 14 \\ 14 & 98 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = 0$$

$$\begin{bmatrix} 2x + 14y \\ 14x + 98y \end{bmatrix} = 0.$$

$2x + 14y = 0$

$2(x + 7y) = 0$

$x = -7y$ if $y = 1$.

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} -7 \\ 1 \end{bmatrix}$$

Length of eigen vector

$$= \sqrt{(-7)^2 + (1)^2} = \sqrt{49 + 1}$$

$$= \sqrt{50} = 7.07$$

1st $$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \frac{-7}{7.07} \\ \frac{1}{7.07} \end{bmatrix} = \begin{bmatrix} -0.99 \\ 0.14 \end{bmatrix}$$

2nd eigen value.

$$\begin{bmatrix} 2 & 14 \\ 14 & 98 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = 100 \begin{bmatrix} x \\ y \end{bmatrix}$$

$$\begin{bmatrix} -98x + 14y \\ 14x - 2y \end{bmatrix} = 0.$$

$14x - 2y = 0$

$2(7x - y) = 0$

$7x - y = 0$

$7x = y$

if $x=1$

$7=y$

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 1 \\ 7 \end{bmatrix}$$

length

$$= \sqrt{(1)^2+(7)^2} = \sqrt{50}$$

$$= 7.07$$

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 0.14 \\ 0.99 \end{bmatrix}$$

Sort the eigen vector according to eigen value

$\lambda=100$ $\lambda=0$

$$V = \begin{bmatrix} 0.14 & -0.99 \\ 0.99 & 0.14 \end{bmatrix}$$

Right Singular vector

$A^T \times A \Rightarrow A^TA - \lambda I = 0$

$$\begin{bmatrix} 50-\lambda & 50 \\ 50 & 50-\lambda \end{bmatrix} = 0$$

$$(50-\lambda)(50-\lambda) - 50 \cdot 50 = 0$$

$$\lambda^2 - 100\lambda + 2500 - 2500 = 0$$

$$\lambda^2 - 100\lambda = 0$$

$$\lambda = 0 \text{ or } \lambda = 100$$

$$\begin{bmatrix} 50 & 50 \\ 50 & 50 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = 0 \text{ or } \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 50x+50y \\ 50x+50y \end{bmatrix} = 0$$

$x = -y$ $\quad 50x = -50y$

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} -1 \\ 1 \end{bmatrix}$$

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

$= 1.41$

$$x = \begin{bmatrix} \frac{-1}{1.41} \\ \frac{1}{1.41} \end{bmatrix} = \begin{bmatrix} -0.70 \\ 0.70 \end{bmatrix}$$

when $x = 0$

$$\begin{bmatrix} 50 & 50 \\ 50 & 50 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = 100 \times \begin{bmatrix} x \\ y \end{bmatrix}$$

$$\begin{bmatrix} -50x + 50y \\ 50x - 50y \end{bmatrix} = 0$$

$-50x + 50y = 0$

$-50x = -50y$

$x = y$

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

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

$= \sqrt{2}$

$= 1.41$

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 1/1.41 \\ 1/1.41 \end{bmatrix} = \begin{bmatrix} 0.70 \\ 0.70 \end{bmatrix}$$

$\Rightarrow$ Sort the eigen vector according to eigen value

$$V^T = \begin{bmatrix} 0.70 & -0.70 \\ 0.70 & 0.70 \end{bmatrix}$$

Diagon matrix.

It is a matrix of $m \times n$ where all elements are 0 except the diagonal elements ~~matrix~~.

$$\Sigma = \begin{bmatrix} \sqrt{100} & 0 \\ 0 & \sqrt{0} \end{bmatrix}$$

$$\Sigma = \begin{bmatrix} 10 & 0 \\ 0 & 0 \end{bmatrix}$$

Hence SVD of $\begin{bmatrix} 1 & 1 \\ 7 & 7 \end{bmatrix}$ can be written as following

$$\begin{bmatrix} 1 & 1 \\ 7 & 7 \end{bmatrix} = U \Sigma V^T$$

$$= \begin{bmatrix} 0.14 & -0.99 \\ 0.99 & 0.14 \end{bmatrix} \begin{bmatrix} 10 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} 0.70 & -0.70 \\ 0.70 & 0.70 \end{bmatrix}^T$$

$$= \begin{bmatrix} 0.98 & +0.98 \\ 6.93 & +6.93 \end{bmatrix}$$

→ We can approximate A using just that component, which gives a compressed but lower-quality representation of A.

→ SVD simplifies complex data structure, making it highly effective for applications like recommendations systems & image compression.

⇒ Apply SVD to real-world recommendation engine, such as e-commerce or streaming service.

1. Data preparation.
   → Load Data, Handle Sparsity, Fill missing value.
2. Apply SVD to Decompose Matrix
3. Evaluate the model using rsme (RSME)
4. Improve Recommendation Accuracy

UNIT-V

MINING SOCIAL NETWORK GRAPHS:-

→ Mining social n/w graph is a central aspect of data Science, especially for analyzing relationship, behaviors, and influence patterns within a network.

→ Social n/w analysis (SNA) is often applied to understand how entities (like people, organization) are interconnected.

Key Concepts of Mining social network graphs:-

1. Graph Structure & Representation.

i) Nodes:- Represent entities such as users, companies or pages.

ii) Edges (Links):- Represent the relationship or interaction between entities, which can be directed (one way) or bi-directed (two-way)

iii) Adjacency Matrix:- A mathematical representation where rows and columns represents nodes, and cell value indicate the presence or weight of connection.

iv) Edge List:- pair of connected nodes, these lists can be simpler for storage but might lack detail without additional attribute.

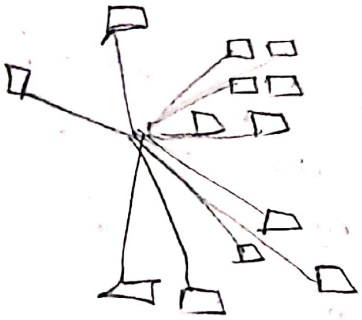
Scanned with OKEN Scanner

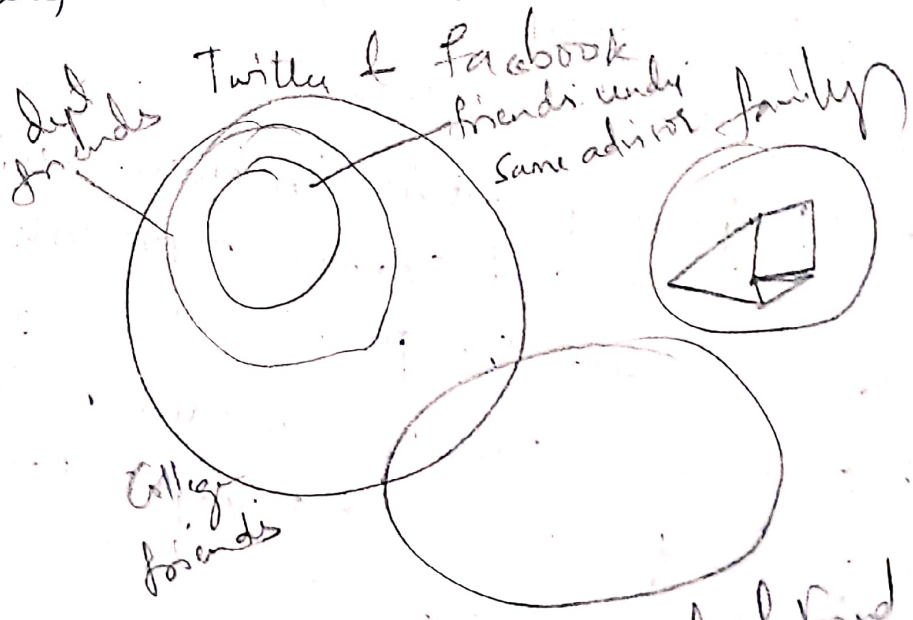
2. Centrality Measures:-

- Degree Centrality :- Counts the number of direct connections (edge) a node has, indicating its influence or popularity within the network.
- Betweenness Centrality :- Measures the extent to which a node lies on the shortest paths between other nodes, showing its importance as a connector.
- closeness centrality :- Calculate the average distance from a node to all other nodes, identifying those that are "close" to many others in network.

Networks & Communities-

→ We often think of networks being organized into modules, clusters, communities.





Social n/w:-

→ A collection entities (nodes)

→ At least one relationship b/w entities of n/w (edges)

Ex:- facebook (friends → boolean - either friends / not)

Scanned with OKEN Scanner

Types of social n/w:-

→ Collaboration Graph.

→ Locality (non-randomness) property

→ The relationship tends to cluster.

→ If A is related to B and C, then the probability that B and C are related is higher than random.

→ Having mutual friends increases the chance of two people knowing each other.

Let B and C are two random entities.

P(B & C) are very small.

But if there is a mutual friend/connection A then probability is significantly high.

Types of social n/w:-

→ The popular one: Facebook, Twitter, Instagram etc.

→ But also several other types.

→ Telephone n/w: nodes are phone numbers

→ AB is an edge if A and B talked over phone within one week, or month, or ever.

→ Edge could be weighted by no of times phone calls were made or time of conversation.

- similarly, any messenger n/w (whatsapp).

Scanned with OKEN Scanner

Email network:- nodes are email address.
→ AB is an edge if A & B sent mails to each other within last one week, or month, or ever
→ One directional edges would allow spammers to have edges.
→ Edges could be weighted.
→ Other networks:- collaboration network.
→ author of papers, jointly written papers or not.

Author-paper graphs:-
→ Both author paper can be nodes.
→ Two authors are connected if they have written a paper together.
→ can put weights to such edges based on the count or relative count.
→ Two papers are connected if there are common authors.

query-page graph:-
→ If user clicked a page p after searching with query q, then put an edge q→p.
→ It can infer query-query relationship.

Networks with several edge types
1. Consider a popular social n/w as facebook.
→ If user A and B are friends; then AB is an unidirected (or bidirectional) edge

Scanned with OKEN Scanner

→ If A follows B, then A → B is a directed edge.

* Edges can be defined based on many other criteria.

→ Does A like/comment in B's posts? (unidirected)

→ Does A and B appear together in any photo? (bidirectional).

Clustering of social network Graphs:-

→ clusters are Communities.

↳ people in same institute / company.

↳ people in photography club

↳ set of people with "something in common" between them.

→ Need to define a distance b/w points (nodes)

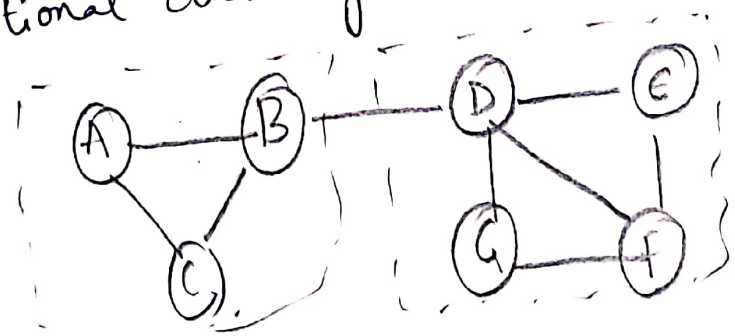
→ In graphs with weighted edges, different distances exist.

→ For graph with "friends" or "not friends" relationship

→ Distance is 0 (friend) or 1 (not friends).

or 1 (friend) or ∞ (not friends).

Traditional clustering:-



→ 2 Communities

→ Traditional clustering depends on distance.

Scanned with OKEN Scanner

→ Likely to put two nodes with small distance in same clustering

→ social n/w graph. would have cross-community edges.

→ may join B & D with to not so low probability

Betweenness of an Edges:—

→ It is defined by no of pair of nodes (x, y) such that AB lies on the shortest path between x & y

→ There can be more than one shortest path between x & y.

High score of betweenness means!

→ the edges runs "between" two communities.

Betweenness gives a better measures.

→ In this example, edges such as BD get a higher score than edges such as AB.

Scanned with OKEN Scanner

Girvan - Newman Algorithm :-

→ The Girvan - Newman algorithm is a popular method used in network analysis and Community detection to identify and separate communities or group in a n/w based on concept of edge betweenness.

Key Concepts :-

Edge like Betweenness :- The "betweenness" of an edge is a measure of how often that edge lies on shortest path between pair of nodes in a network.

→ Edges with high betweenness are likely to act as bridge between different Communities.

2. Communities :- The algorithm aims to detect groups of nodes that are more densely connected internally than with nodes in other group.

Steps :-

1. Calculate Edge Betweenness.

→ For each edge in the network, calculate the edge betweenness score, this is done by counting the number of shortest paths between all pair of nodes that pass through the edge.

Scanned with OKEN Scanner

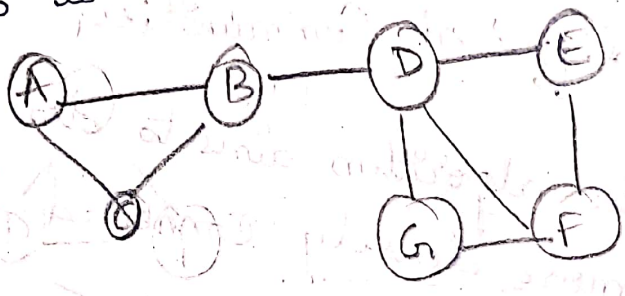
2. Remove Edge with Highest Betweenness:-

→ Remove the edge with the highest edge betweenness score from the n/w. This step weakens the Connectivity between Communities.

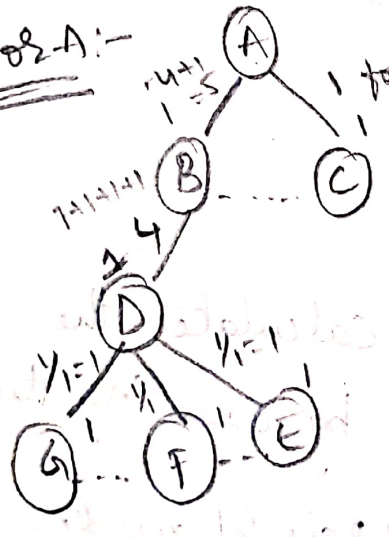
3. Recalculate the Edge Betweenness

4. Repeat until the network is divided into distinct Communities.

5. Stop when a predefined number of Communities are identified. or when the network is Completely divided into isolated nodes.



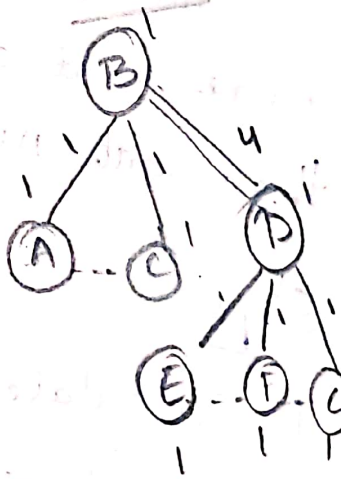
for A:-



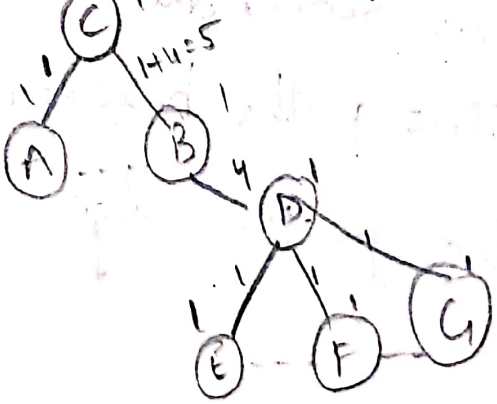
for leaf = child / parent = 1/1 = 1

for non-leaf add (1+1+1+1)

for B -

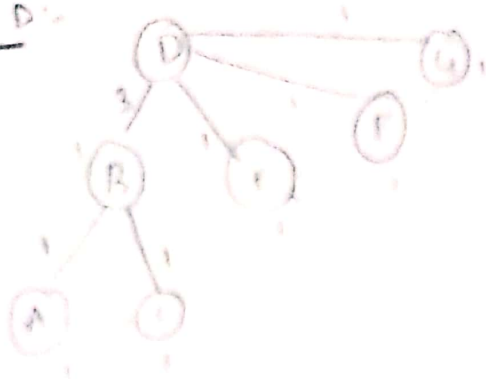


for C

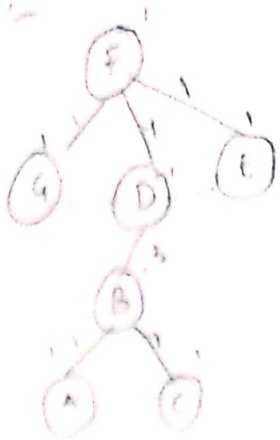


Scanned with OKEN Scanner

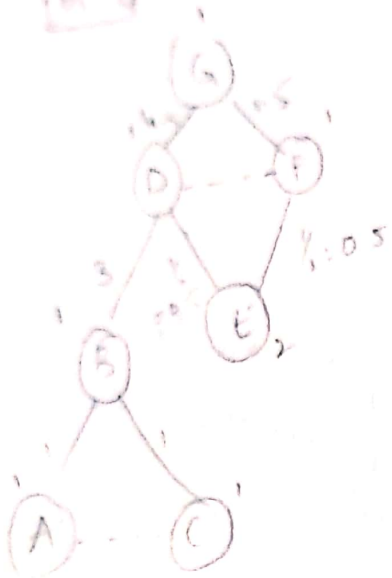
For D :-



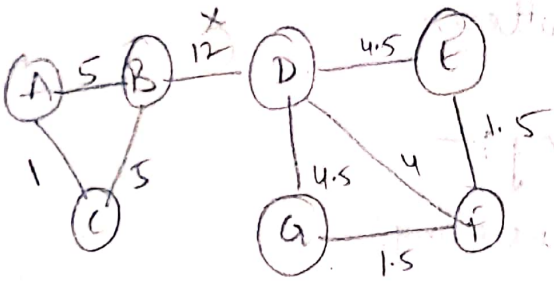
For F :-



For G :-

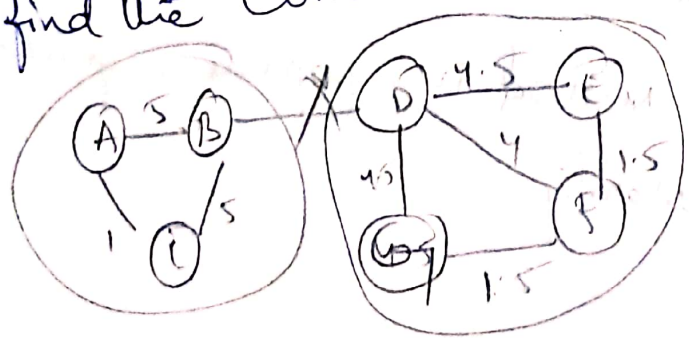


Edges	Betweenness Centralities							Total	Avg
	A	B	C	D	E	F	G		
AB	5	1	0	1	1	1	1	10	10/6 = 5
AC	1	0	1	0	0	0	0	2	1
BC	0	1	5	1	1	1	1	10	5
BD	4	4	4	3	3	3	3	24	12
DG	1	1	1	1	0.5	0	4.5	9	4.5
DF	1	1	1	1	0	4	0	8	4
DE	1	1	1	1	4.5	0	0.5	9	4.5
EF	0	0	0	0	1.5	1	0.5	3	1.5
FG	0	0	0	0	0.5	1	1.5	3	1.5



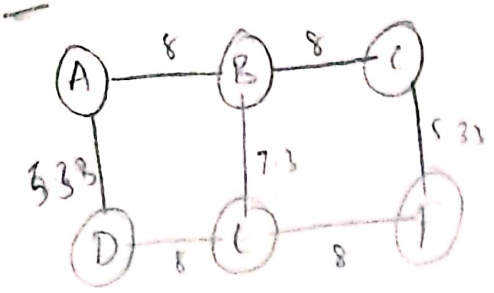
Betweenness Centrality of Graph

→ To find the Communities in a graph



Scanned with OKEN Scanner

Q1-2



Edges

AB => 3.165+1.5+1.33+0.835+0.5+0.66 = 8

AD => 1.835+0.5+0.33+1.83+0.5+0.33 = 5.32

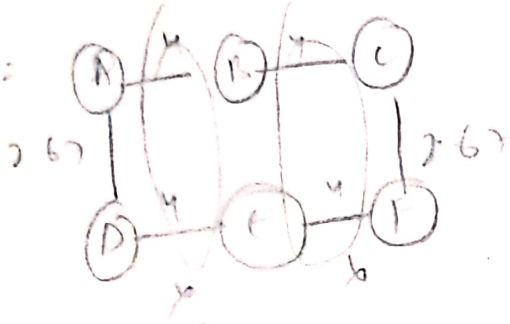
BC => 8

BE => 1.34

CF => 5.33

DE => 8

EF => 8



Clustering of Graphs:-

→ clustering in graphs is the process of grouping nodes into clusters where nodes within the same cluster are more connected to each other than to nodes outside the cluster

→ Techniques is widely used to analyse complex networks, including social network, biological network to reveal underlying structure or pattern.

Methods of Graph clustering :-

1. Modularity-Based clustering:-

Modularity measures the density of edges within clusters compared to the edges between clusters,

Example:- Louvain Method is a popular algorithm for Maximizing Modularity

Scanned with OKEN Scanner

Louvain Method is an efficient algorithm for Community detection in large network.

→ It identifies Communities by maximizing modularity

Modularity formula.

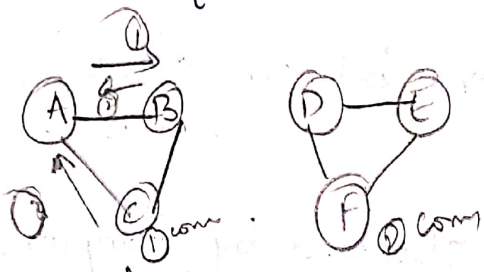
$$Q = \frac{1}{2m} \sum_{ij} \left(A_{ij} - \frac{k_i k_j}{2m}\right) \delta(c_i, c_j)$$

Where A_{ij} is 1 if is an edge between i & j and 0 otherwise.

k_i & k_j are degree of node i & j

m is total no of edges.

$\delta(c_i, c_j)$ is 1 if nodes i & j belong to same Community & 0 otherwise.



2. Spectral clustering :-

→ uses graph laplacian (a matrix derived from the adjacency matrix) to find cluster based on the eigenvalue & eigenvectors.

→ Effective for finding clusters with low connectivity between them.

Scanned with OKEN Scanner

3 Hierarchical Clustering:-

-> Builds a hierarchy using either a divisive or agglomerative approach

* Agglomerative Methods starts with each node as a separate cluster & merge them iteratively.

-> while divisive Method starts with a single cluster & divide it.

How Modularity Helps Identify Meaningful clusters:-

-> It allows us to

* Maximize internal Connection within clusters, identifying tightly knit-group.
* Minimize external Connections between clusters, capturing Community boundaries.
* Quantify how well nodes are clustered by Comparing the actual density of edge within Communities to a random distribution of edge.

Louvain Algorithm:-

Phases of Louvain Algorithm:-

phase 1:- Local modularity Optimization.

-> Each node is initially treated as its own Community. The algorithm Evaluates whether moving a node from its Community to a neighboring community

Scanned with OKEN Scanner

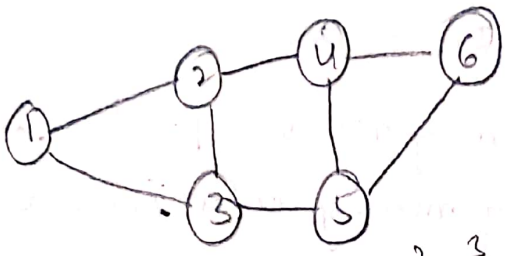
increase modularity.

Phase 2 :- Community Aggregation.

→ After, phase 1 communities are aggregated into "Super-nodes", & the graph is reduced.

→ A new, smaller graph is created where each super-node represents a community found in the previous step.

→ these two phases are repeated iteratively until no further modularity gain is achieved, or until the improvement falls below threshold.



Adjacency Matrix

	1	2	3	4	5	6
1	0	1	1	0	0	0
2	1	0	1	1	0	0
3	1	1	0	0	1	0
4	0	1	0	0	1	1
5	0	0	1	1	0	1
6	0	0	0	1	1	0

Step 1:- Initialize Communities:-

Each node starts as its own Community.

Initial Communities : 1, 2, 3, 4, 5, 6

2. Phase 1 : Modularity optimization.

→ for each node, we evaluate moving it to a neighboring node's community & calculate if this improves modularity

Scanned with OKEN Scanner

Iteration over Nodes.

Node 1 : It has connection to 2 & 3

→ Moving from Node 1 to community of Node 2 & 3 increases Modularity, so we select one {1,2}

Node 2 : Already in the community of Node 1, check other Neighboring Nodes (3 & 4).

→ Moving Node 2 & 3 as a pair to Node 3 community could increase modularity.

→ Resulting community could be {1,2,3}

* Repeat this process for other nodes.

→ After several iterations, we may reach a point where modularity can no longer be increased by moving nodes individually.

updated communities {1,2,3} & {4,5,6}.

Louvain algorithm:-

1. Optimizes modularity locally by moving nodes to neighboring communities.
2. Aggregates nodes into super-nodes to form a reduced graph.
3. Repeat these steps until modularity gains are negligible.

Scanned with OKEN Scanner

Direct Discovery of Communities in Graph:-

→ Communities in a Social network Graph are group of nodes with dense connection within the group and sparse connection outside the group.

1. Girvan-newman algorithm:-

→ uses edge-betweenness to iteratively remove the edges, breaking down the graph into Communities.

2. Clique-percolation Method:-

→ find overlapping Communities by detecting Cliques (Complete Subgraphs) and "rolling" them over to discover larger Overlapping Communities.

Key Concept of clique-percolation:-

k-clique:-

→ A k-clique is a Complete Subgraph of k-nodes, meaning that every pair of node in the k-clique is connected by an edge.

→ For example 3-clique is a triangle, while 4-clique is a fully connected quadrilateral.

Percolation of cliques:-

→ percolation refers to the process of connecting k-clique that share k-1 nodes to form larger, connected structure.

Scanned with OKEN Scanner

→ If two k-clique share k-1 nodes, they are considered adjacent.

Clique Communities:-

→ A clique Community is a set of k-clique that can be reached from each other through a sequence of adjacent k-clique.

Steps of the Clique-percolation method:-

1. Choose a clique size (k):-
Select a value for k-based on desired size of clique.
Larger value of k result in more tightly knit communities, while smaller value yield larger more loosely connected communities.

2. Identify k-cliques:
Find all k-cliques in the network.

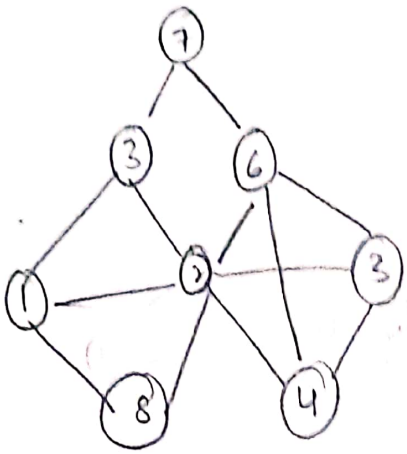
3. Connect Adjacent k-clique:
Identify pair of k-clique that share k-1 nodes.
Link these k-clique to form a larger structure, where each node has atleast k-1 connections to others within the structure.

4. Form clique communities:
Every set of Connected k-clique forms a clique Community.
A node can belong to multiple communities if it part of multiple overlapping k-clique.

* Community is a group of cliques such that all cliques must have k-1 nodes in common.

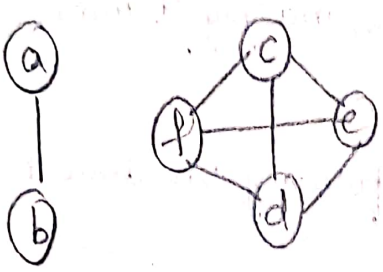
Scanned with OKEN Scanner

Example 1:-



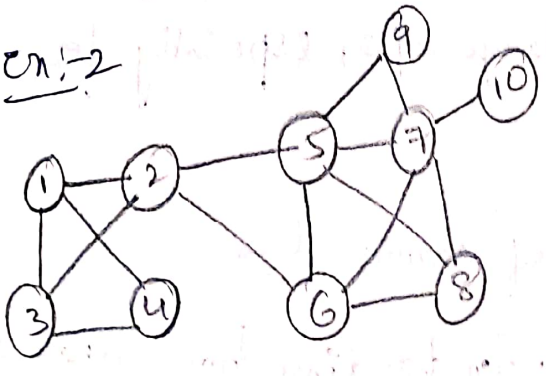
In given example there are six k=3 cliques

a: 1,2,3
b: 1,2,8
c: 2,4,5
d: 2,4,6
e: 2,5,6
f: 4,5,6



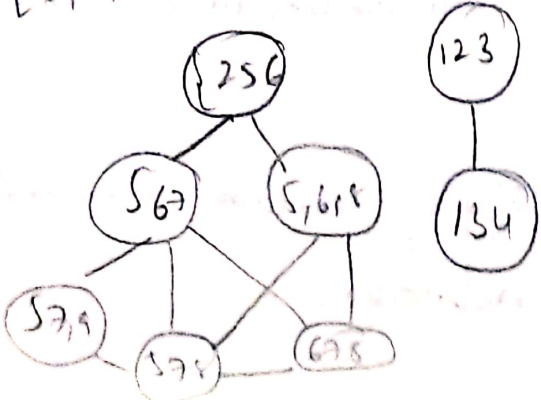
2 communities

Ex:-2



k=3

{1,2,3}, {1,3,4}, {2,5,6}, {5,6,7}, {5,6,8},
{6,7,8}, {5,7,8}, {5,7,9}.

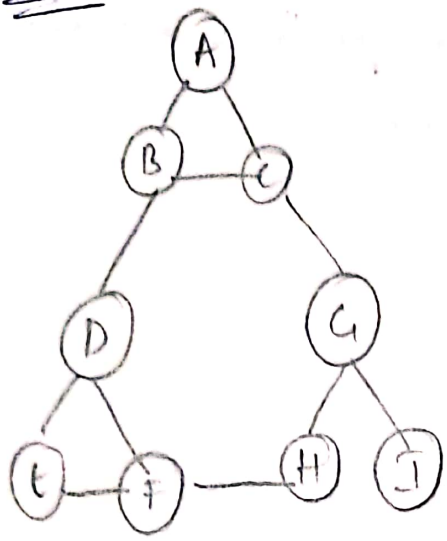


k=4

{1,2,3,4}
{2,5,6,7,8,9}

Scanned with OKEN Scanner

Ex3:-



C_1 : a,b,c

C_2 : d,e,f

C_3 : g,h,i

Community : 0.

Advantages:-

→ Allow for Overlapping communities, making it more relaistic for social networks.

→ Detects tightly-knit groups by focusing on dense k-cliques.

Limitation:-

→ Computationally intensive for large n/w, especially for higher value of k.

Application of Direction Discovery of communities.

1. Social influence Analysis:- Understanding how ideas or products spread within Social Communities.

2. Marketing & Targeting:-
   Segmenting Users into Communities to deliver targeted marketing messages.

3. Anomaly detection:- Identifying unusual Communities in financial or Security networks.

Scanned with OKEN Scanner

Partitioning of Graphs:-

→ partitioning is dividing a graph into disjoint sets of nodes (Subgraphs) such that certain properties are optimized, like minimizing the number of edges or balancing the number of nodes in each partition.

Partitioning Techniques:-

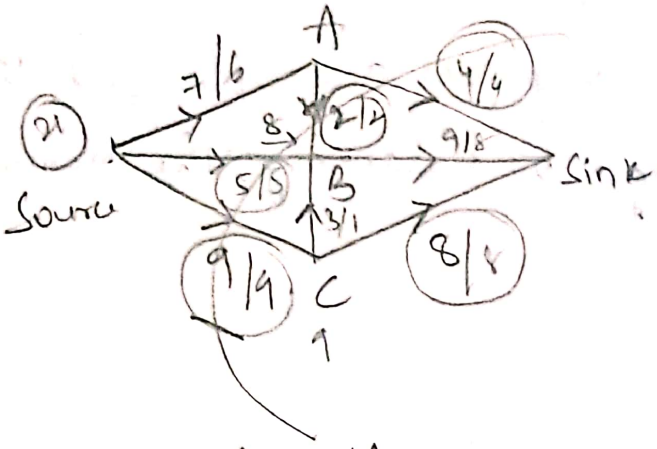
* Minimum cut / Max flow:-
  → finds ~~pattern~~ partitions by minimizing the number of edge that need to be cut to separate the graph.
  → used to balance the no of nodes in each partition.
  → the maximum value of flow from a source to a sink i a flow network is equal to minimum capacity of a cut tha separates the source & sink.

Terms:-

1. Flow Network:- A directed graph where each edge has a capacity, & flow cannot exceed this capacity.
2. Maximum flow:- The greatest amount of flow that can be sent from source to sink without violating capacity constraint
3. Minimum cut:- The smallest total capacity of edge that, if removed, would disconnect the source & sink.

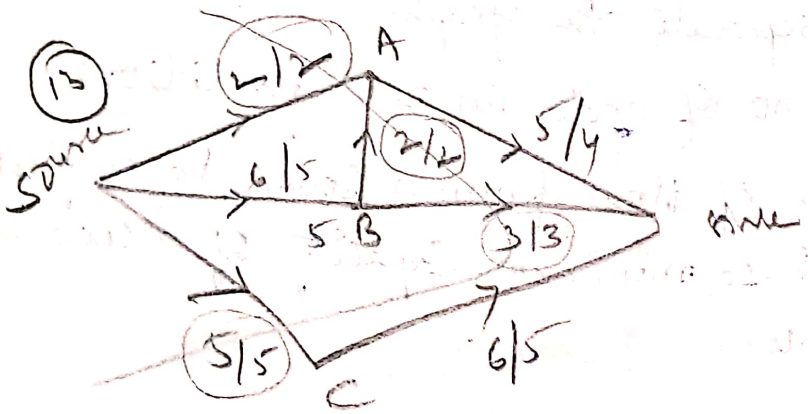
Scanned with OKEN Scanner

Example – 6



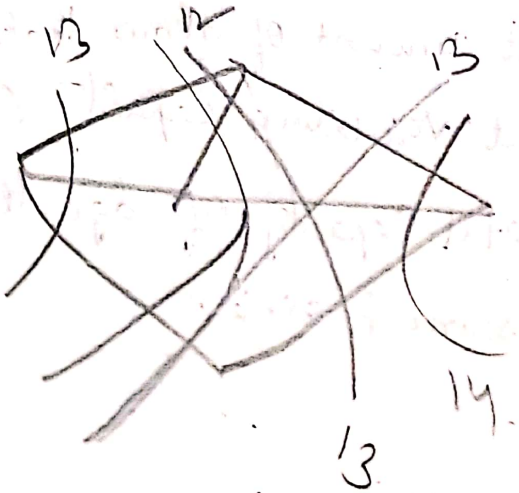
Maximum flow => 4 + 2 + 5 + 9

= 20



Manflow => 2 + 2 + 3 + 5

= 12



Spectral partioning:-

Spectral partioning is a graph partioning method that uses eigenvalues & eigen vectors of matrices associated with a graph, such as Laplacian Matrix, to divide the graph into clusters or Communities.

Key Terms:-

Graph Representation:-

A graph $G=(V, E)$ consist of vertices V & edges E which can be represented using adjacency matrices.

Laplacian Matrix:-

It is defined as

$$L = D - A$$

D → Degree Matrix, A → Adjacency Matrix

L has important properties -

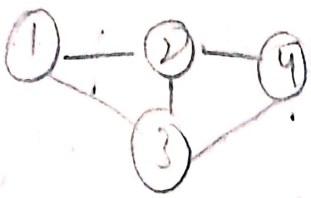
1. It is symmetric & positive semi-definite.
2. The smallest eigenvalue of L is always 0, with the corresponding eigenvector being constant vector.

Spectral Clustering:-

→ Spectral clustering relies on eigenvector of L to partition the graph.

→ The second smallest eigenvalue and its corresponding eigenvector are particularly useful for partitioning

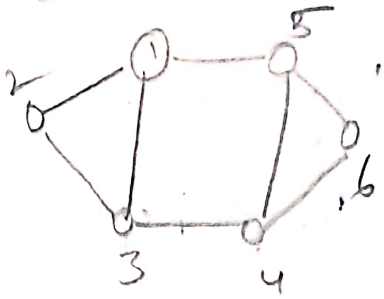
Ex:- Consider a simple graph



$$A = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}$$

$$D = \begin{bmatrix} 2 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 \\ 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 2 \end{bmatrix}$$

Ex 2



$$L = D - A$$

$$= \begin{bmatrix} 2 & -1 & -1 & 0 \\ -1 & 3 & -1 & -1 \\ -1 & -1 & 3 & -1 \\ 0 & -1 & -1 & 2 \end{bmatrix}$$

find eigen value & eigen vector

→ Sort the eigen values in ascending order.

3. Select the fiedler vector

Identify the eigenvector corresponding to the second smallest eigent value of L.

4. partition the Graph:-

Split vertices based on the sign (positive or -ve) or threshold value of the entries.

→ To partition the graph into more than two cluster, repeat the process recursively on subgraph.

Scanned with OKEN Scanner

Neighborhood properties in Graph :-

→ In Graph theory, neighborhood properties describe the relⁿship and connections between vertices based on their adjacency in a graph. These properties are fundamental in understanding the structure and characteristics of a graph.

Key Concepts :-

1. Neighborhood of a vertex :-

The neighborhood of a vertex v, denoted as N(v), is the set of all vertices adjacent to v.

$$N(v) = \{u \in V : (v,u) \in E\}$$

Ex:- In a graph with edges $E = \{(a,b), (a,c), (b,d)\}$, the neighbood of a is $N(a) = \{b,c\}$.

2. closed Neighborhood :-

The closed neighborhood of a vertex v, denoted $N[v]$, includes v itself and its neighbors.

$$N[v] = N(v) \cup \{v\}.$$

3. Degree of vertex :-

The degree of a vertex v, denoted deg(v), is the size of its Neighborhood.

$$deg(v) = |N(v)|$$

In Directed Graph :-

1. In-degree : No of edges pointing to v
   Out-degree : No of edges pointing away from v.

Scanned with OKEN Scanner

4. Neighborhood Overlap:-

The Neighborhood overlap of two vertices u and v measures proportion of common neighbors relative to their union.

$$\text{Overlap}(u,v) = \frac{|N(u) \cap N(v)|}{|N(u) \cup N(v)|}.$$

5. Common Neighbors:

Two vertices u and v share common neighbor if:

$$N(u) \cap N(v) \neq \emptyset.$$

This property is used in link prediction and community detection.

6. Neighborhood connectivity

The average neighborhood connectivity of a vertex v is the average degree of its neighbors:

$$C(v) = \frac{1}{\deg(v)} \sum_{u \in N(u)} \deg(u).$$

Extended Concepts:-

1. Clustering Coefficient:-

Measures how connected the neighbors of a vertex are:

$$C(v) = \frac{\text{Number of edges between neighbor of } v}{\left(\deg_2(v)\right)}$$

Scanned with OKEN Scanner

2. Neighborhood Subgraph :-

The Subgraph induced by neighborhood of v, denoted by $G[N(v)]$, is the graph formed by vertices in $N(v)$ and all edges between them.

3. Neighborhood Diversity :-

The number of distinct types of neighborhoods in graph, often used in optimization and algorithmic graph problems.

Applications:-

1. Social networks
2. Machine learning
3. Recommendation Systems
4. Graph Algorithms

Scanned with OKEN Scanner

Data visualization

→ Data visualization is the graphical representation of information and data using visual elements like charts, graphs, and infographics.

→ The goal is to make complex data more accessible, understandable and usable for analysis and decision-making.

→ Basic principles Of Data visualization.

a) Clarity :-

* Make the visualization easy to interpret with Concise labels, and titles.

* Avoid unnecessary elements like 3D effects or excessive gridlines.

b) Accuracy :-

* Ensure the visual representation accurately reflects the data.

* use Consistent Scales and intervals to avoid misleading viewers

c) Relevance :-

* Tailor the visualization to target audience and their goals.

* Highlight Key insights or tends relevant to decision-making process.

Scanned with OKEN Scanner

D. Simplicity:-

* Remove unnecessary Complexity by limiting the number of Colors, fonts or visual elements.
* Focus on Conveying One Key message per chart or visualization.

E. Aesthetics :-

* use visually pleasing and professional color schemes.
* Maintain balance, alignment & proportionality in design

~~2. Ideas for~~ 1. Define the purpose.

↳ Know your Audience.

↳ Set clear Objectives.

2. Choose the Right visualization:-

* Match chart-to Data; use appropriate charts to graph based on the data type & analysis.

Comparison - Barchart.

Trends - Line charts.

Distribution - Histogram.

Relationship - scatter plot

proportions - pie chart or TreeMap.

* Avoid Misleading visuals:- Ensure that Chosen representation align with data's actual story.

Scanned with OKEN Scanner

3 Simplify the Design :-

* Minimize clutter :- Remove unnecessary elements
* Highlight key points :

4. Maintain Accuracy :-

* Avoid Distortion : Represent data, with avoiding proportion or perspective.
* Label clearly : provide enough context without overwhelming the viewer.

5. Use Effective visual elements :-

* color
* size & proportion

2 Ideas & Tools for Data visualization :-

1. Idea for visualizing Different Type of data.

1. Trend over time : Line & area charts
2. Comparison : Bar chart, grouped bar charts
3. proportion : pie chart, donut chart, treemap.
4. Distribution : Histogram, box plot & violin plot
5. Relationship : Scatter plot, bubble charts
6. Geospatial Data : choropleth maps, dot density maps.

Scanned with OKEN Scanner

B. Popular Tools for Data visualization

1. Coding Tools:-
   Python : Matplotlib, Seaborn, plotly, Altair
   R : ggplot2, shiny, plotly.

2. BI Tools:
   Tableau, power BI, Google Data studio.

3. Online platforms:
   Flourish, canva, Infogram.

4. Spreadsheet Tools:
   * Microsoft Excel, Google sheets

5. Interactive visualization ~~Tools~~ libraries.
   D3.js, Charts.js.

Examples of Inspiring Industry projects:-

A. Healthcare:-
   * Covid-19 Data visualization
   → Johns Hopkins Dashboard: A real dashboard tracking global Covid-19 cases, deaths, and recoveries using interactiv map & trend graphs.
   → visualization principle:- Interactive maps help users explore data at different geographic levels.

Scanned with OKEN Scanner

B. Finance:-

Financial Market Dashboard.

→ Bloomberg's terminal visualize Stock Movements, volume & financial trends with time-series plots

→ visualization principle: Time Series data uses Consistent Scales & annotation for clarity.

C. Retail :- Customer Segmentation

→ Retail companies like Amazon or Target visualize Purchasing trends using bubble charts & customer personas

→ visualization principle:- Bubble size represent revenue which color-coding distinguishes Customer Segments.

D. Sports:-

Performance Analysis Dashboard.

* Websites like FiveThirtyEight display player stats using radar charts & bargraph.

Visualization principle:- clear labels and Consistent color emphasize player strengths and weakness.

Scanned with OKEN Scanner