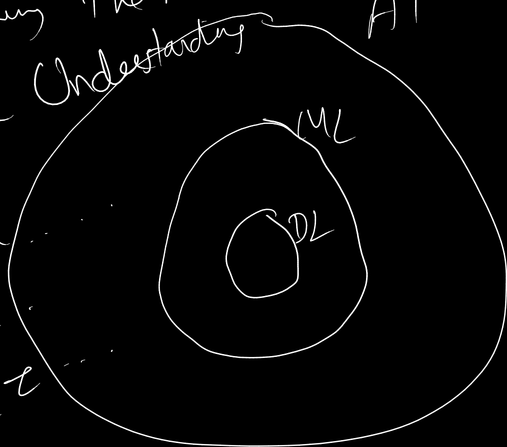
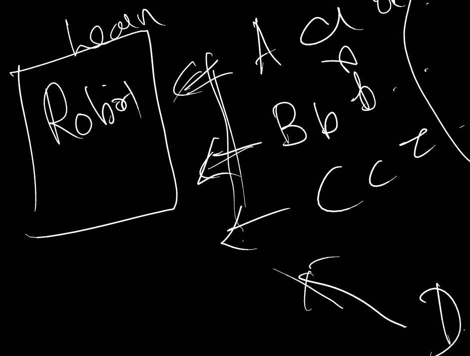
ML => You making the machine to Learn

MU => Machine Understanding




?




AI => Concepts

↳ Hill Climbing Algorithm...

Solving

Stol

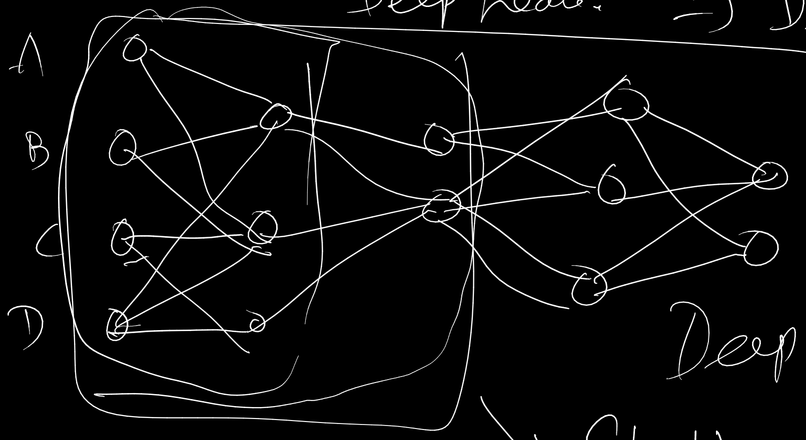
Tic Tac Toe

Generic

Deep Learning — Deept Network

DeepNet —

Deep learn. ⇒ Deep Network




Deep Networks.

Shallow Network

Data Science → ML, Deep

Rebranded

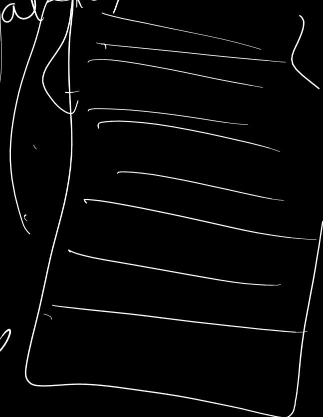
→ Data Mining

→ Data Warehouse → Indexing → Hashing → Categorization

→ Data Engineering

Tagging

Viz:



~~CBIR~~ CBR ⇒ Content Based Retrieval

Gom

CBIR → Contend Based Image Retrieval

Machine Learning.

→ Supervised ⇒ Classification
Labels

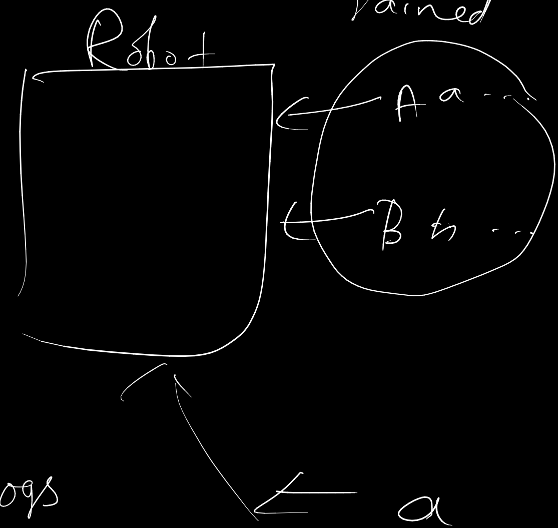
Unsuper →

Reinforce Le.

Semi Supervised Learn

Vijay
Ai
R.

Eg: Dogs
&
Cats.




Pandam

Unsupervised Learning -> Without Labels

Eg:

Amazon

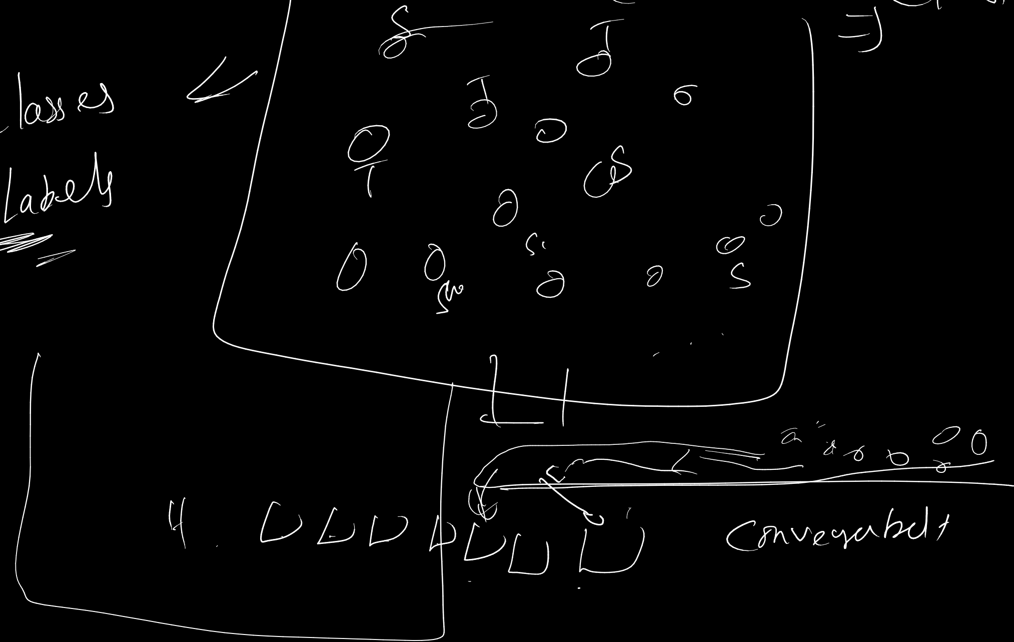
Billions => Clustering

-> Algorithm

# of classes

# of Labels

Bigbasket



features - Entity → Amarnath → height
→ weight
→ Color
→ Genry.
} feature

Samples

Samples.

Apple1 → Color → Red, Green.
Size → Round - 14 to 7 cm.
weight 150 gm
feat

Apple 2 → Green.
Size - 6
weight 60 gms

Apple 3 → Red
Size

Apple ← Class 1

Samples

Apple 1
2
3
4
5

Orang 1
2
3
4
.

$f_1$ $f_2$ $f_2$ $f_4$ . . . . . . $f_{1000}$

PCA

Two Labels

Classes

Target Class.

Unsupervised learning ⇒ Clustering — K means.

→ (Meanshift)

heirarchical — Divisive

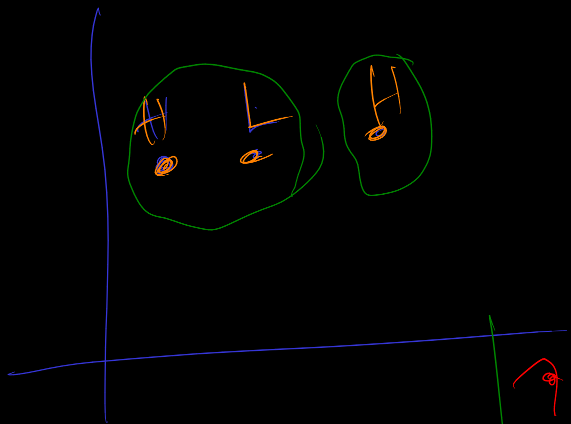
→ Agglomerative

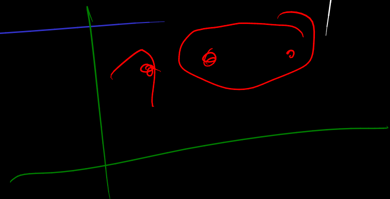
PCA

LDA

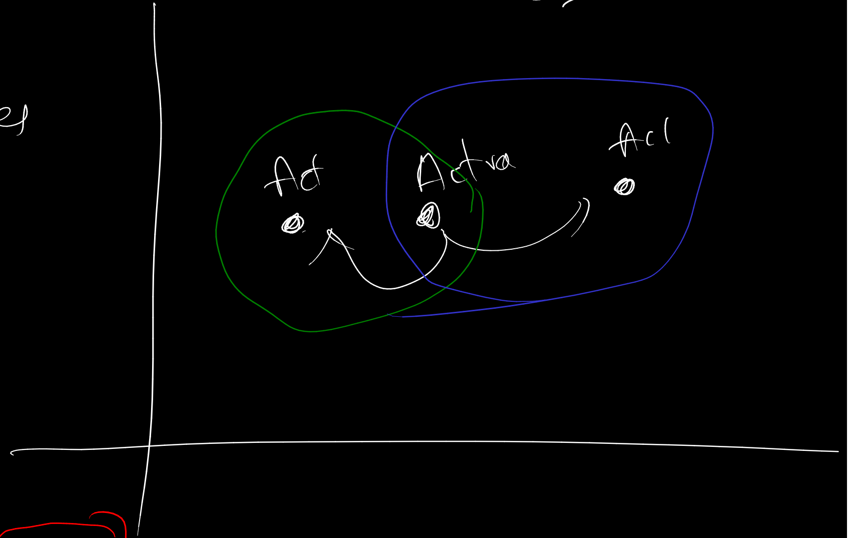
K means Clustering →

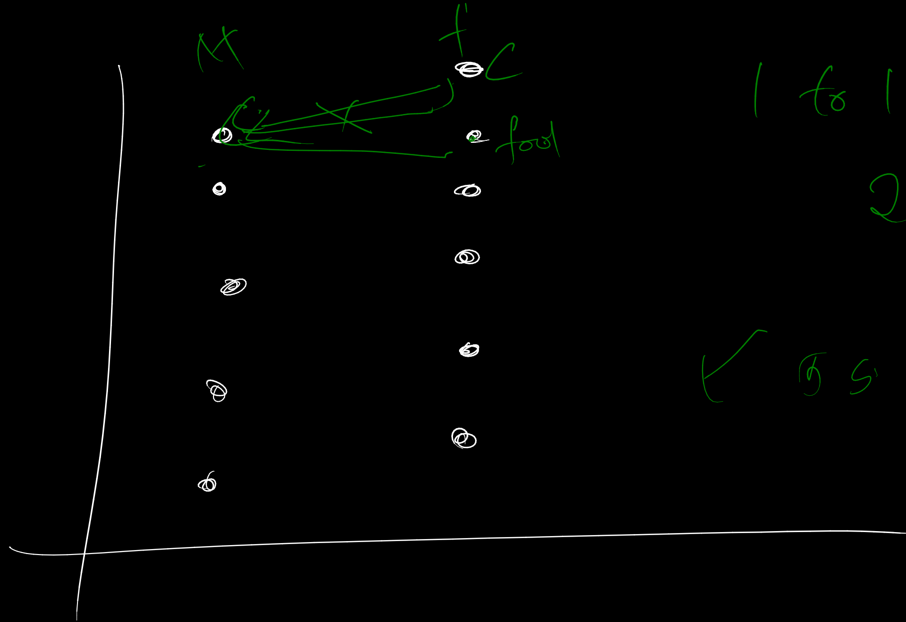
↓ # of labels/classes



Eg: 2 = K






1 to 1 mapping

20 matrimonial sites

Open Challenges

Problems.

Open Challenges in Project

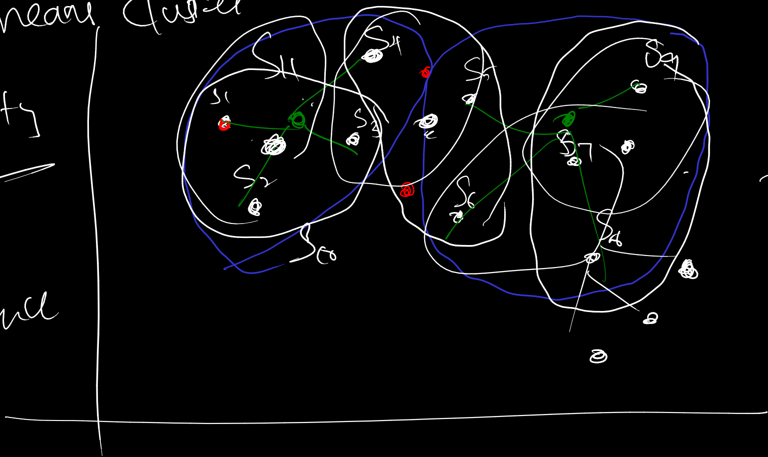
Next → 2 to 3 ...

K means Cluster:

Proximity

↓

Distance




Duduku

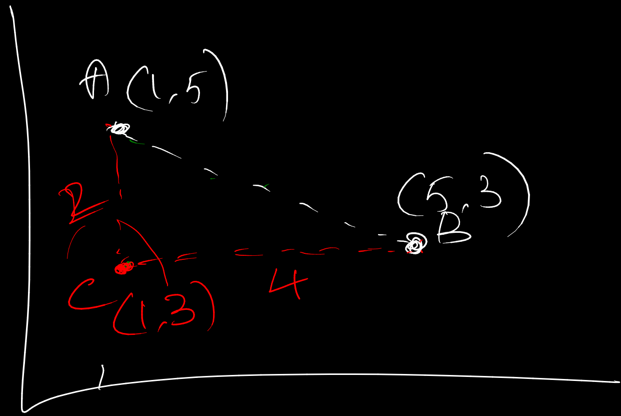
K=3 k=5 .6.

Install Proximity → Distance.

Water Tank

| | $f_1$ | $f_2$ |
|---|---|---|
| $H_1$ - | 1 | 1 |
| $H_2$ | 1 | 2 |
| $H_3$ | 2 | 3 |

Depth




$$AB^2 = AC^2 + CB^2$$

$$AB^2 = 2^2 + 4^2$$

$$= 4 + 16$$

$$AB^2 = 20$$

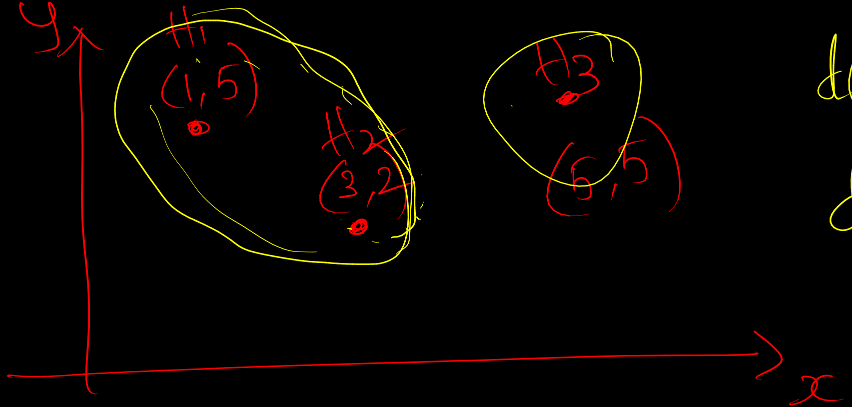
$$AB = \sqrt{20}$$

Distance Measurement

Euclidean Dist

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

$K = 2$




$$d(H_1, H_2) = \sqrt{(1-3)^2 + (5-2)^2} = \sqrt{4+9} = \sqrt{13}$$

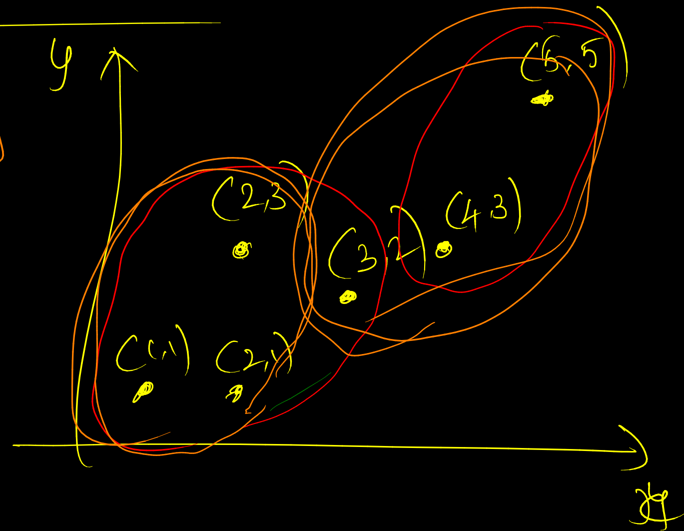
$$d(H_2, H_3) = \sqrt{(3-5)^2 + (2-5)^2} = \sqrt{4+9}$$

# K means clustering

Input : K=2 first step

| | $f_1$ | $f_2$ |
|---|---|---|
| $H_1$ | 1 | 1 |
| $H_2$ | 2 | 1 |
| $H_3$ | 2 | 3 |
| $H_4$ | 3 | 2 |
| $H_5$ | 4 | 3 |
| $H_6$ | 5 | 5 |

Proximity



Step 2: Randomly choose 2 points / Seed points

Step 3

Euclidean Distance

| | $f_1$ | $f_2$ | $H_1$ (1, 1) | $H_2$ (2, 1) | Class |
|---|---|---|---|---|---|
| $H_1$ | 1 | 1 | 0 | 1 | $H_1$ |
| $H_2$ | 2 | 1 | 1 | 0 | $H_2$ |
| $H_3$ | 2 | 3 | $\sqrt{5}$ | 2 | $H_2$ |
| $H_4$ | 3 | 2 | $\sqrt{5}$ | $\sqrt{2}$ | $H_2$ |
| $H_5$ | 4 | 3 | $\sqrt{13}$ | $\sqrt{8}$ | $H_2$ |
| $H_6$ | 5 | 5 | $\sqrt{32}$ | 5 | $H_2$ |

Step 4

$C_1 = \{H_1\}$

$C_2 = \{H_2, H_3, H_4, H_5, H_6\}$

| | $f_1$ | $f_2$ | $C_1$ (1,1) | $C_2$ (3.2, 2.8) | Centroid points classes |
|---|---|---|---|---|---|
| $H_1$ | 1, | 1 | | | $H_1$ |
| $H_2$ | 2, | 1 | | | $C_1$ |
| $H_3$ | 2, | 3 | | | $C_1$ |
| $H_4$ | 3, | 2 | | | $C_2$ |
| $H_5$ | 4, | 3 | | | $C_2$ |
| $H_6$ | 5, | 5 | | | $C_3$ |

Euclidean Distance

$$C_1 = H_1 \quad (1,1)$$

$$C_2 = \frac{1}{5}\{(2,1), (2,3), (3,2), (4,3), (5,5)\}$$

$$(16/5, 14/5)$$

Iteration 3

$C_1 = \{H_1 \ H_2 \ H_3\}$

$C_2 = \{H_4 \ H_5 \ H_6\}$

$\left.\begin{matrix}(n-1) \\ n\end{matrix}\right\}$ same Stop

Iteration 4:

$C_1 = \{H_1 \ H_2 \ H_3\}$

$C_2 = \{H_4 \ H_5 \ H_6\}$

Stop

Real time Example:

Randomly Generate - 20 points (python)

$k = 2$

$f_1$ $f_2$

$S_1$

$\vdots$

$S_{20}$

Cluster.

$C_1 = \{ S_1, S_3, S_4 \ldots S_6, S_n \}$

$C_2 = \{ S_2, S_5, S_7, S_8 \ldots \}$

Euclidean Distance

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

Single link // Nearest

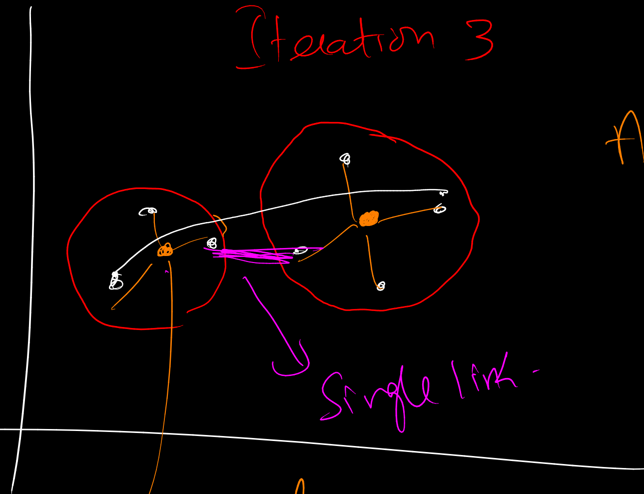
Complete link // farthest

Average link

Clusters will change

5

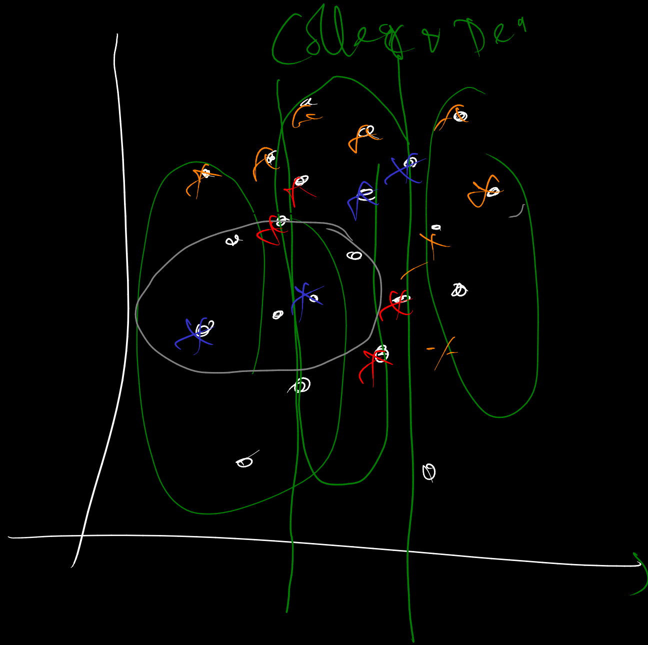
Iteration 3



Average link

Single link

Avg // Centroid



To find K automatically

K as an input

Elbow method

Shilhouette method.

↳ College $f_1$

dept $f_2$

Year $f_3$

Feature Selection/

DOB $f_4$

Feature Reduction

height $f_5$

Multi dimensional feature.

Weight $f_6$

Complexity $f_7$

$S_1$

$S_2$

$\vdots$

$D = \sqrt{\phantom{xx}}$

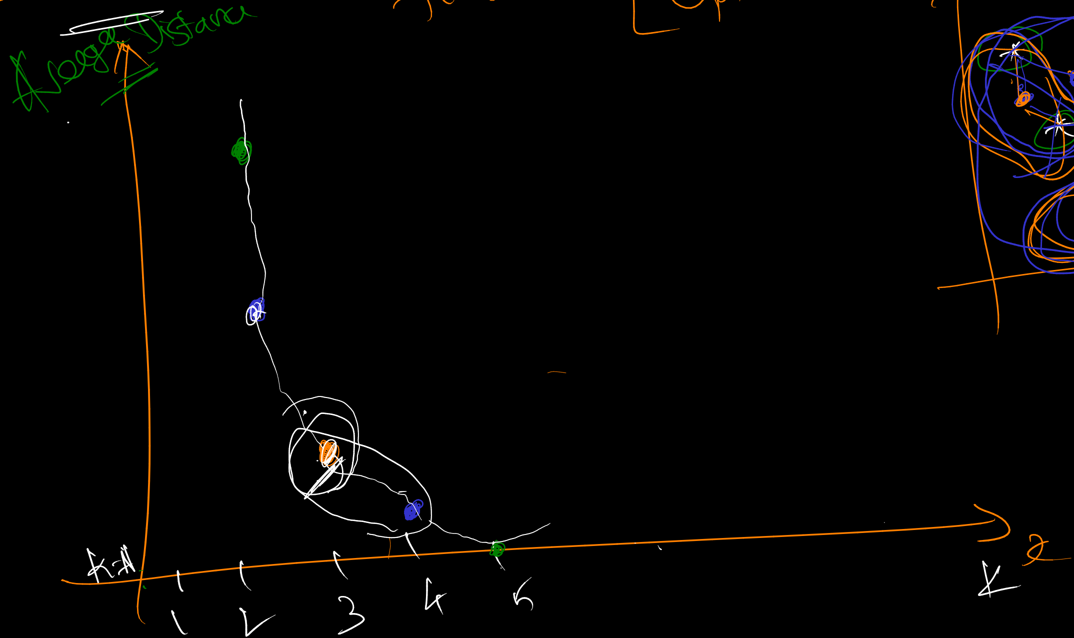
$K = 3$

$f_1 \checkmark - f_2 - f_3 \checkmark - f_4 - f_5 \checkmark\ f_6\ f_7 \cdots f_{10}$

$K = 3$

Clustering

Elbow method find K [Optimal





Area=

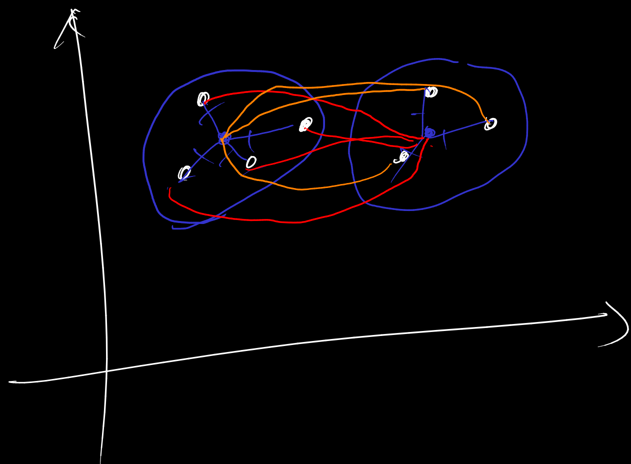
Shillouette Method.

↳ Cohesion ⇒ within the cluster distance of every point to its centroid

↳ Seperation. = b(i)

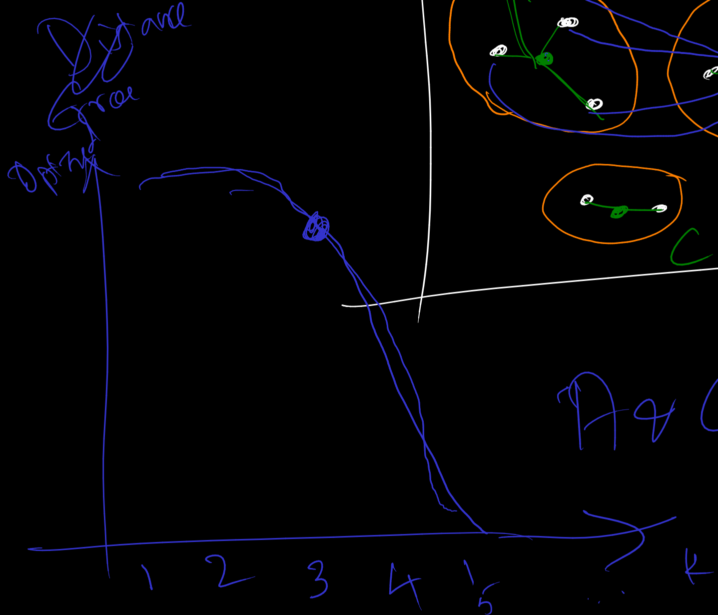
K = 2

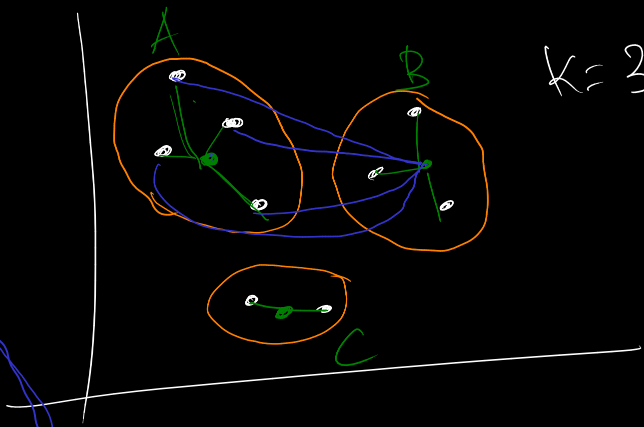
a(i) = average distance



Seperation.

$$b(i) = \frac{a(i) - b(i)}{\max(a(i), b(i))}$$




$K=3$

$A \longrightarrow B$

$a(i)$ A & B

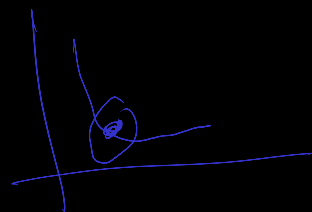
A & C

$a(i)$

$b(i)$ B & C

A & C , B & C

$$\frac{a(i) - b(i)}{\max(a(i), b(i))}$$



Elbow
Shilleuette. } To find $\underline{\underline{K}}$

K-means
Clusterins.

Dendogram

Agglomerative Clustering

Divisive Clustering

K=1 Dendogram

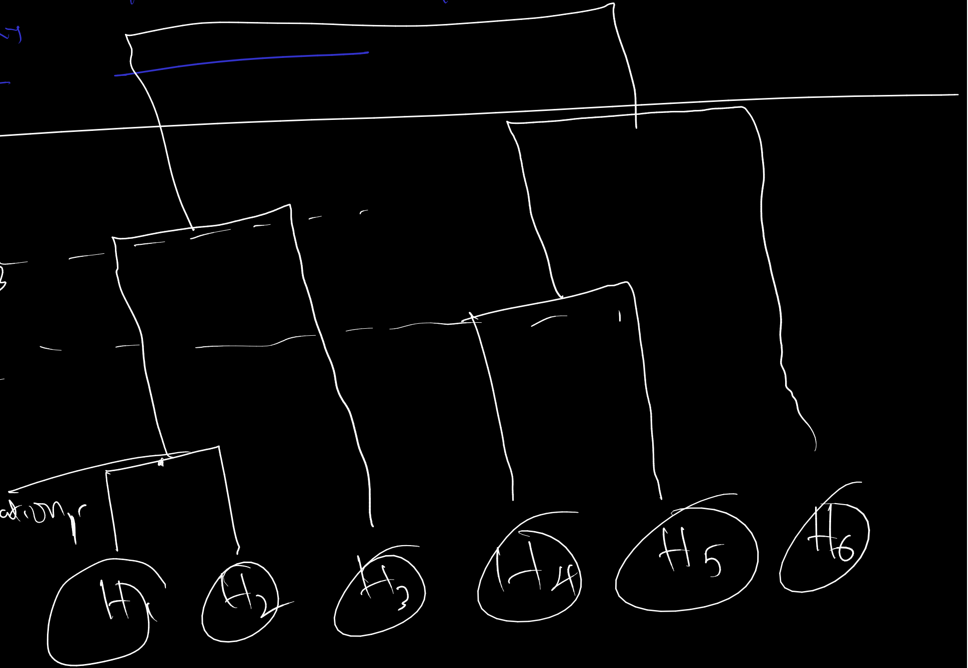
K=2 Iteration 3

K=3 Iteration 2

K=4

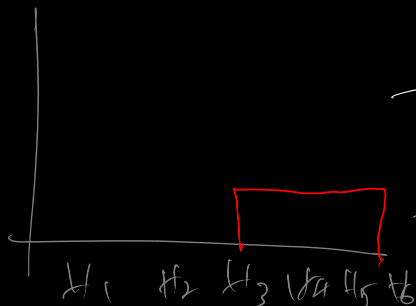
K=5 Iteration 1

K=6



| | $f_1$ | $f_2$ |
|---|---|---|
| $H_1$ | .40 | .53 |
| $H_2$ | .22 | .38 |
| $H_3$ | .35 | .32 |
| $H_4$ | .26 | .19 |
| $H_5$ | .08 | .41 |
| $H_6$ | .45 | .30 |

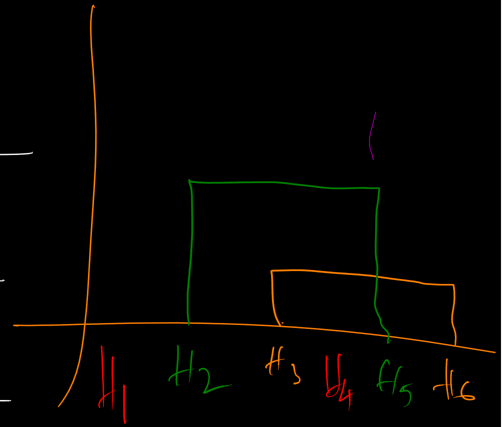
Euclidean




# Distance Matrix

| | $H_1$ | $H_2$ | $H_3$ | $H_4$ | $H_5$ | $H_6$ |
|---|---|---|---|---|---|---|
| $H_1$ | 0 | | | | | |
| $H_2$ | .24 | 0 | | | | |
| $H_3$ | .22 | .15 | 0 | | | |
| $H_4$ | .37 | .20 | .15 | 0 | | |
| $H_5$ | .34 | .14 | .28 | .29 | 0 | |
| $H_6$ | .23 | .25 | .11 | .22 | .39 | |

| | $H_1$ | $H_2$ | $(H_3, H_6)$ | $H_4$ | $H_5$ |
|---|---|---|---|---|---|
| $H_1$ | 0 | | | | |
| $H_2$ | .23 | 0 | | | |
| $H_3, H_6$ | .22 | .15 | 0 | | |
| $H_4$ | .37 | .20 | .15 | 0 | |
| $H_5$ | .34 | .14 | .28 | .29 | 0 |



| | $H_1$ | $(H_2, H_5)$ | $(H_3, H_6)$ | $H_4$ |
|---|---|---|---|---|
| $H_1$ | 0 | | | |
| $H_2, H_5$ | | 0 | | |
| $H_3, H_6$ | | | 0 | |
| $H_4$ | | | | 0 |

Distance Measures

Similarity Distance

Dissimilarity Distance

How close / nearby features are.

How farthest the feature are from two data points

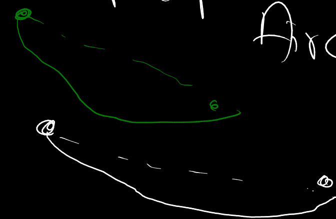
samples.

## → Distance Measures

- ↳ Euclidean Distance
  - Average
  - Single link
  - Complete
- ↳ Chessboard Distance.
- ↳ Cityblock.
- ↳ Jacob Distance
- ↳ Cosine Distance

Map

Aircraft



Road ways

Bus Stand

Sadar

Mosque.

# K-means Clustering image segmentation

Colour — RGB → Red Green Blue Channels

→ HSV

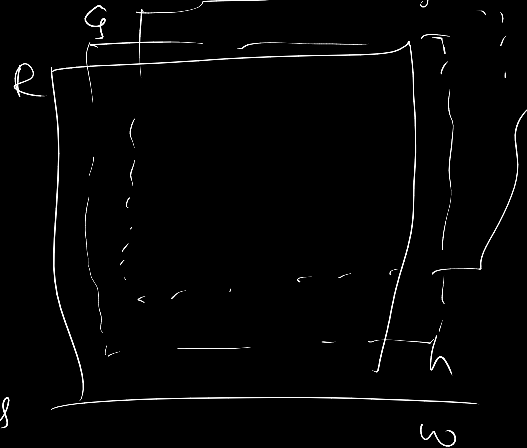
→ HIV R

h = 100

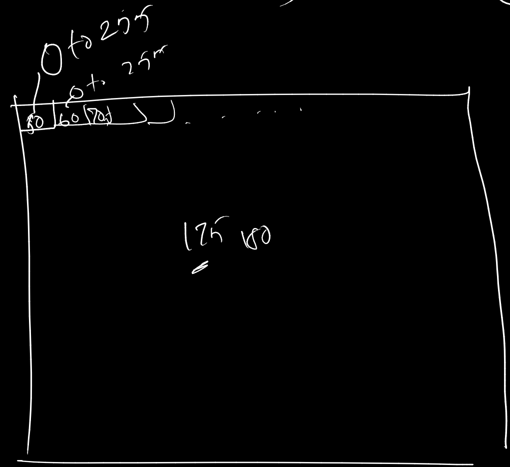
→

w = 100

3 × 100 × 100 pixels



3 Colour Channel → Gray Scale




K-means

K = 3

K = 4

Seed points

$S_2 = 200$

$S_1 = 50$

| | | |
|---|---|---|
| 50 | 50 | 70 |
| 20 | 200 | 215 |
| 10 | 100 | 175 |

$S_1 = 50$ $S_2 = 200$

Iter 1

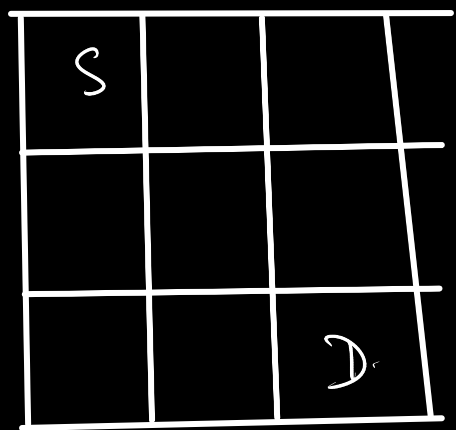
| | | |
|---|---|---|
| $S_1$ | $S_1$ | $S_1$ |
| $S_1$ | $S_2$ | $S_2$ |
| $S_1$ | $S_1$ | $S_2$ |

Iter 2

| | | |
|---|---|---|
| $S_1$ | $S_1$ | $S_1$ |
| $S_1$ | $S_2$ | $S_2$ |
| $S_1$ | $S_2$ | $S_2$ |

# Recursion

↳ A function that calls duplicate of the function



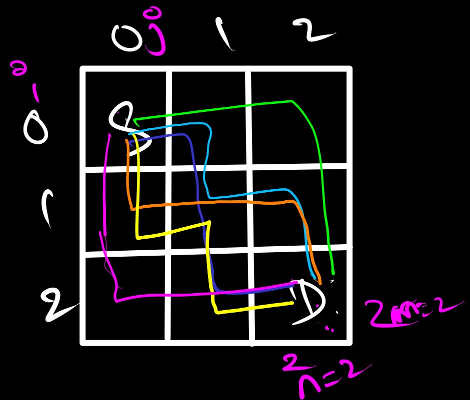

↳ Right

↳ Down

Exit Conditions

↳ Reaches Destination

↳ Go out of Boundary




Problem: # paths

Input: $\in [3,3]$

S : (0,0)

D : (2,2)

Output: # paths (2,2)

→ Right

↳ Down.

Counter = 0

1
2
3
4
5
6

b = (0,0)

if ($\cancel{i, j}$)
i = 2 & j = 2
return -

i > 2
return

j > 2 return.

6

```
def number_of_paths(n, i, j, c):
    if i==n & j==n:
        return c+1
    if i>n or j>n
        return c
    return number_of_paths(n, i+1, j, c):
    return number_of_paths(n, i, j+1, c)
number_of_paths(2, 0, 0, 0)
```

| | | | | | | |
|---|---|---|---|---|---|---|
| 1 | 1 | 2 | 3 | 5 | 8 | 13 |
| ↑ | ↑ | ↑ | ↑ | ↑ | ↑ | ↑ |
| n=1 | 2 | 3 | 4 | 5 | 6 | 7 |

Exit Condition:

if $n < 1$

Invalid

if $n = 1$

return 1

if $n = 2$

return 1

return $(n-1) + (n-2)$

```
def fibonacci(n):
    if n<0
        return -1

    if n==1:
        return 1
    if n==2:
        return 1
    return fibonacci(n-1) + fibonacci(n-2)
```

n=0 Invalid

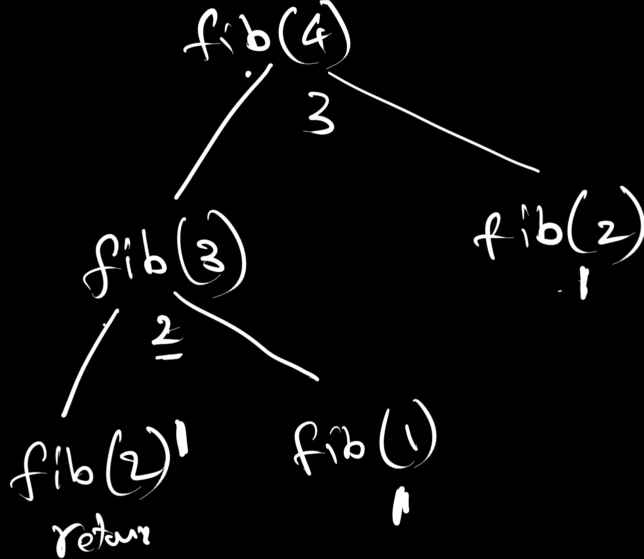
n=1 1

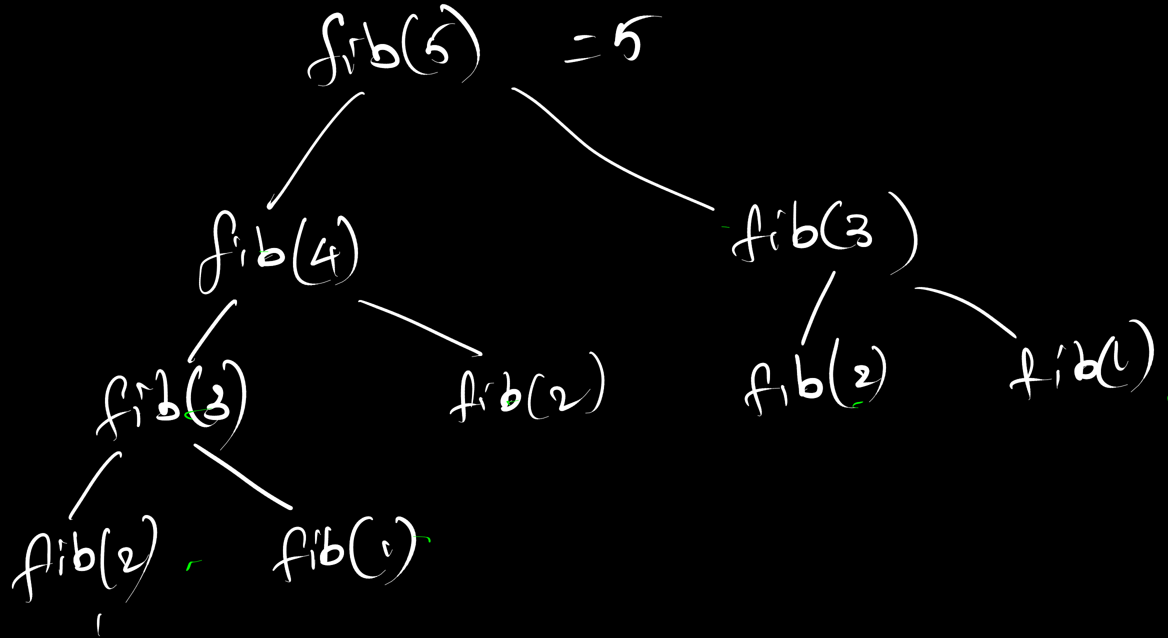
n=2 1

n=3 2

n=4 3

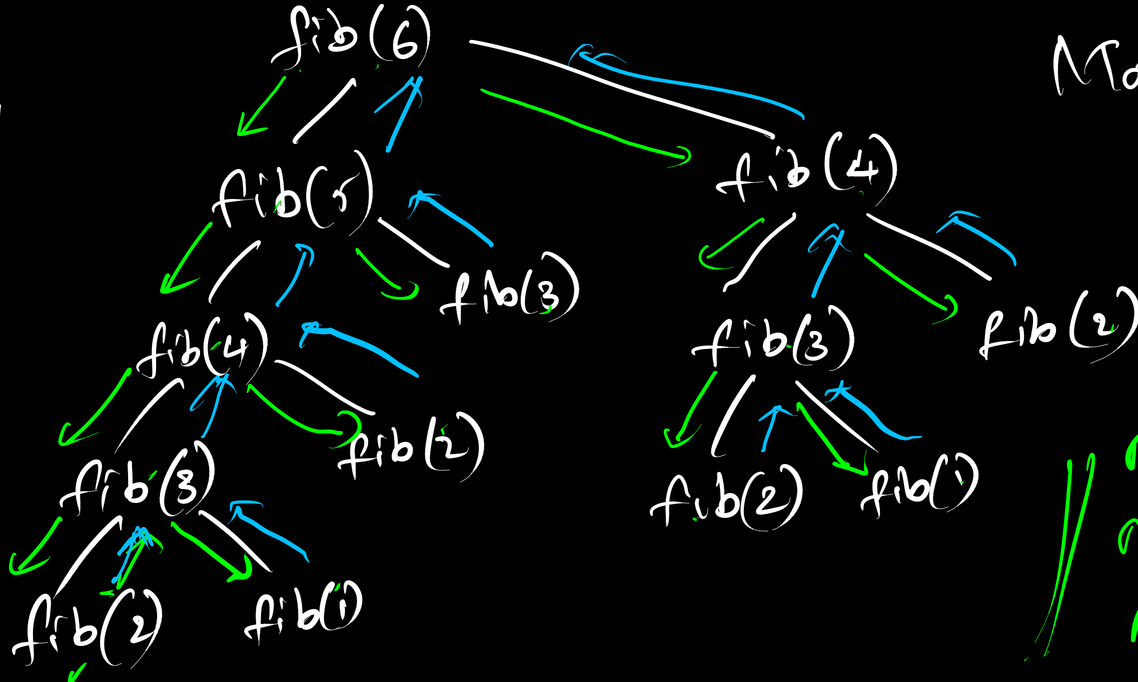
n=5





fib(5) = 5
fib(4)
fib(3)
fib(3)
fib(2)
fib(2)
fib(1)
fib(2)
fib(1)

| n | |
|---|---|
| 1 | 1 |
| 2 | 1 |
| 3 | 2 |
| 4 | 3 |
| 5 | 5 |
| 6 | 8 |
| 21 | ? |

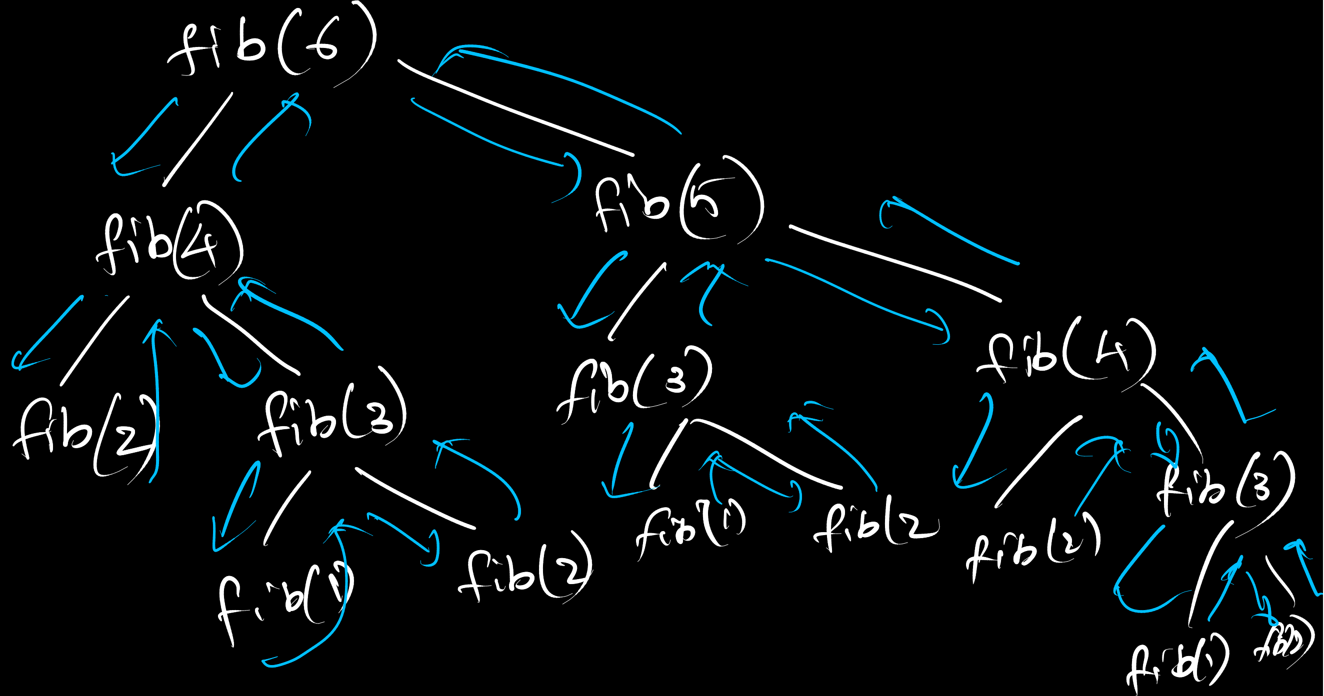



Mathem...
Model

Complexity
Time

| | |
|---|---|
| n=1 | 1 |
| n=2 | 1 |
| n=3 | 2 |
| n=4 | 5 |
| n=5 | 9 |

n=6 = 13


fib(6)
fib(4)
fib(5)
fib(2)
fib(3)
fib(3)
fib(4)
fib(1)
fib(2)
fib(1)
fib(2
fib(2)
fib(3)
fib(1)
fib(2)

# Heuristic Approach

Greedy Heuristic Approach

↳ Agent Travers the path

DFS - Depth first Search

| | 0 | 1 | 2 |
|---|---|---|---|
| 0 | S 0 | .5 | .3 |
| 1 | .2 | .4 | .6 |
| 2 | .7 | .4 | .1 D |

0 → 1 ·→

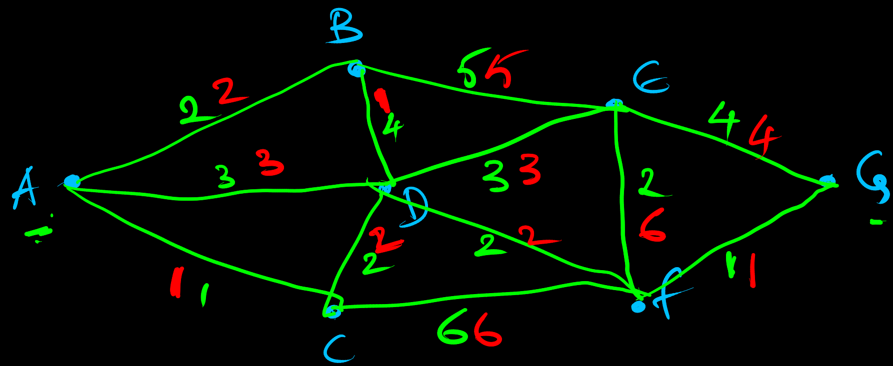
Greedy Approach

| | 0 | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|---|
| 0 | 0 | .2 | .1 | .6 | .5 | .2 |
| 1 | | 0 | .3 | .4 | .7 | .3 |
| 2 | | | 0 | .4 | .6 | .7 |
| 3 | | | | 0 | .2 | .3 |
| 4 | | | | | 0 | .4 |
| 5 | | | | | | 0 |

Graph

visited = [A✓, D✓ ]

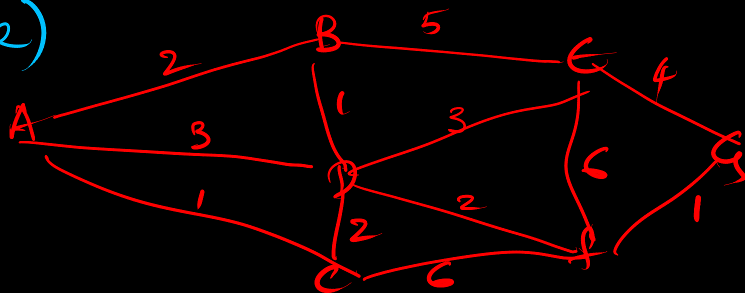
Greedy



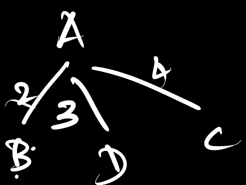

A =
|
C
|
D
|
f
|
G

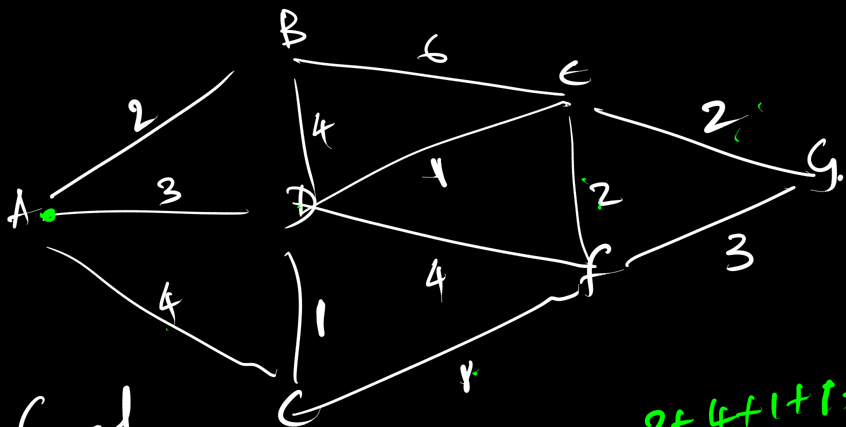
A
|
C
|
D
|
B
|
E
|
G

G(v, e)



A*




$B \xrightarrow{?} G$

$D \xrightarrow{?} G$

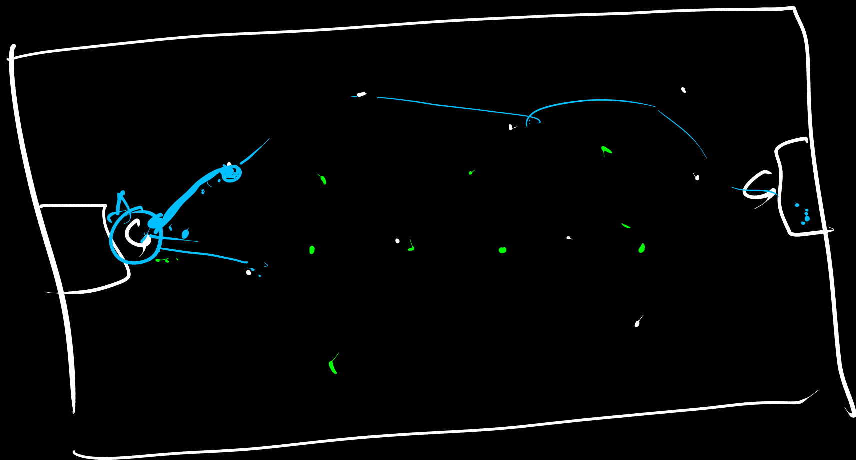
$C \xrightarrow{?} G$

Goal Orient

A*

2+4+1+1+2+2

12



$A^*$ → Goal

Compute the distance position to the destination

# Representation of a Graph

Vertices/nodes ↳ Matrix format → Adjacency Matrix

Edges/Arc Mwt ↳ Incidence Matrix

$G(V, e)$

## Adjacency Matrix:

Directed Graph

| | 0 | 1 | 2 |
|---|---|---|---|
| 0 | 0 | .4 | .2 |
| 1 | .6 | 0 | .1 |
| 2 | .6 | .3 | 0 |

$A = G(V, e) = V \times V$

$0 \rightleftarrows 1$

Un Directed Graph

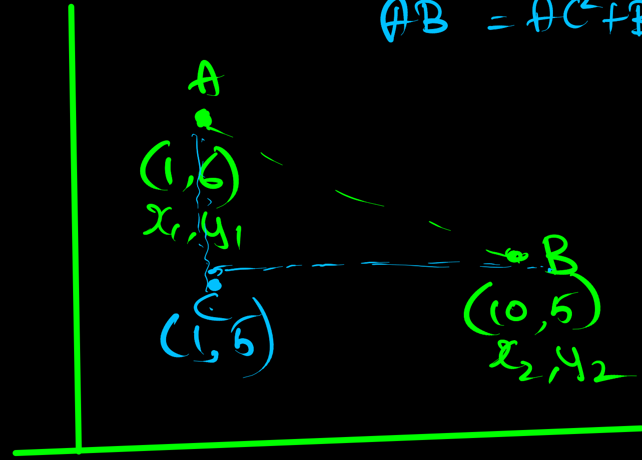
| | 0 | 1 | 2 |
|---|---|---|---|
| 0 | 0 | .2 | .3 |
| 1 | .2 | 0 | .4 |
| 2 | .3 | .4 | 0 |

# Distance Measures

- Chess Board
- City Block
- Jacard
- Sinusoidal
- Euclidean Distance

Pythogerous Theoram

$AB^2 = AC^2 + BC^2$




$$D(A,B) = \sqrt{82}$$

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

# Hill Climbing

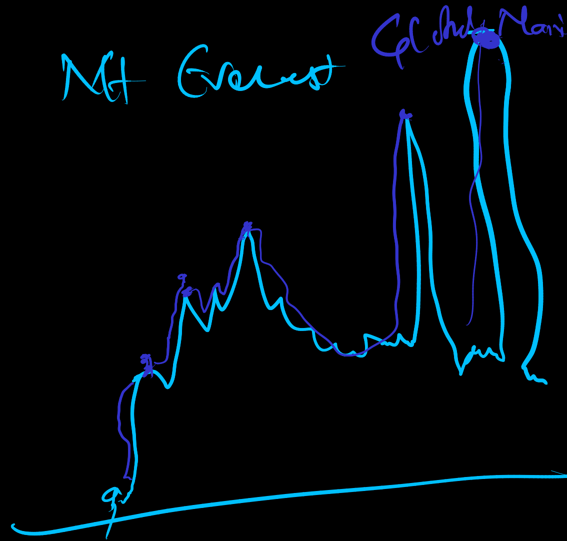
Gradient Descent

↳ Global Maxima

↳ Global Minima

↳ Local Maxima

↳ Local Minima

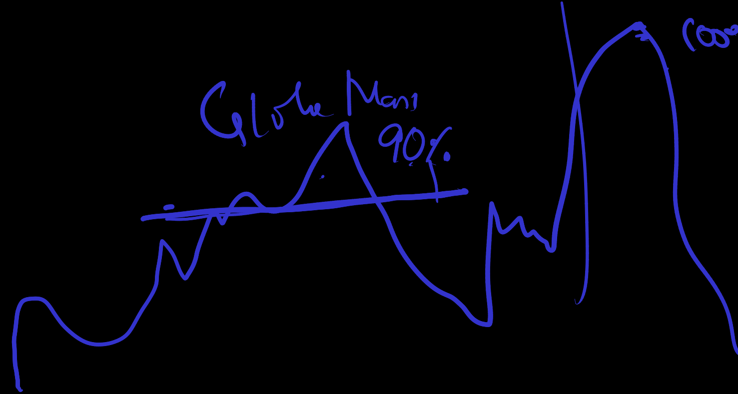


- Local Maxima

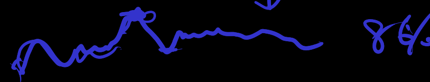
Local Minima

Local Minimum

Local Maxima

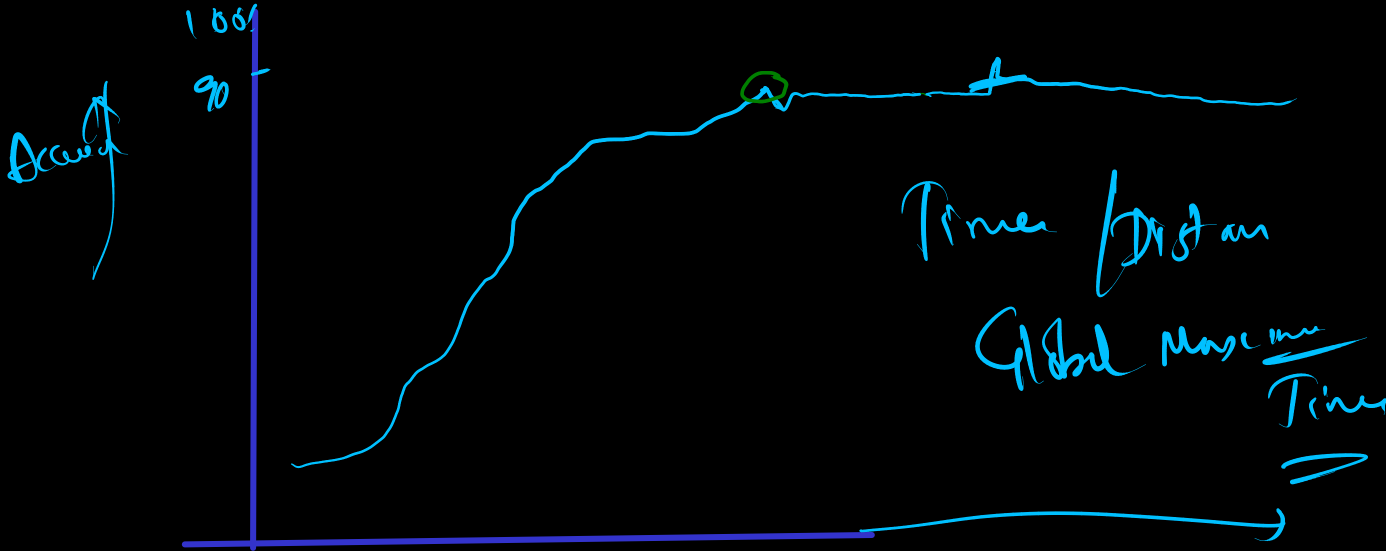



90% Dodabetta




Local { Area

Deep Learning Model



# Heuristic Approach

↳ Greedy

↳ A*

↳ Graph -

↳ ~~Distance Map~~ Repres..

↳ Adjacency

↳ Incr.: Matrix

Hill Climbing

Global Maxi

Global Min

Local Max

Local Min

Exhaustive Search

find the space

$$\begin{array}{c|cc} & 0 & 1 \\ \hline 0 & 2 & \\ 1 & 3 & 1 \end{array}$$

(0,0)

find 1 in the matrix

(1,1)

2 –
3 1

2 –
3 1

Move Right

↳ Left

↳ Up

↳ Down.

Recursion

2 1
3 –

↳ Move Right

↳ Move Down

2 1
– 3

– 1
2 3

Rigth
Up
Left
Down

Right
Down
Left
Up

Right
Down
R

— 3
2 1

3 —
2 1

3 1
2 —

3 1
— 2

0

3 2

0 1
3 2

1 0
3 2

1 2
3 0

Left
Up
Right
Down

Left
Down
Righ
Up

1 0
3 2

1 0
3 2

3 0
2 1

1 2
3 0

3 1
2 0

1 0
3 2

3 1
0 2

1 2
3 0

0 1
3 2

Left
Up
Right
Down

Initial State = [[1 0]
[3 2]]

(0,0) (0,1)
(1,0) (1,1)

Compare
Exit Condition

is == GS

Can't B
Borders

i, j

locat = find 0 location

Goal State = [[1 2]
[3 0]]

move(i-1, j ...)
mov(i, j+1 ...)

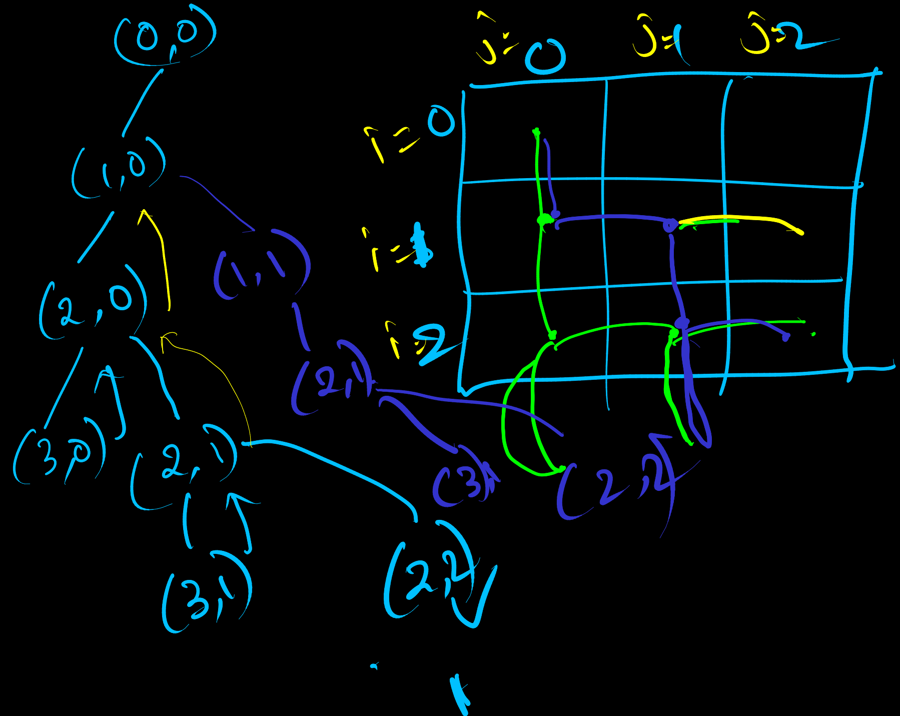
Code Chef . com

hackerrank . com

hackerworld . com

Leetcode . com

50 problems / revision



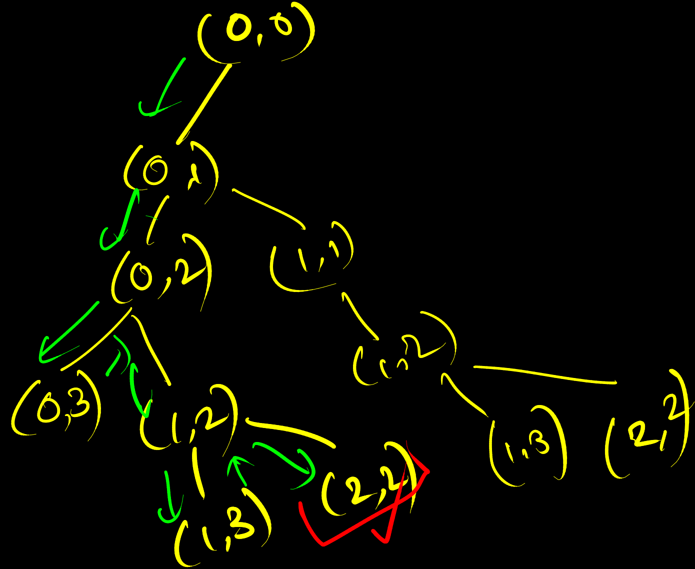

# No of paths

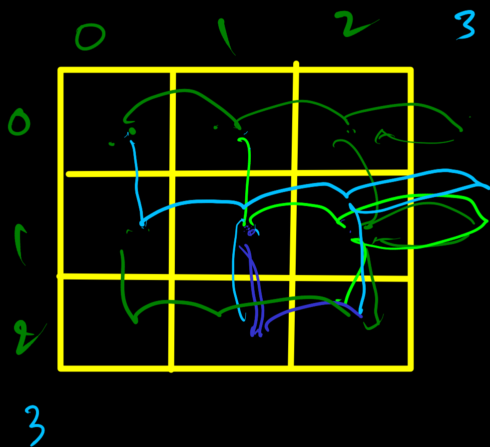
Down

1

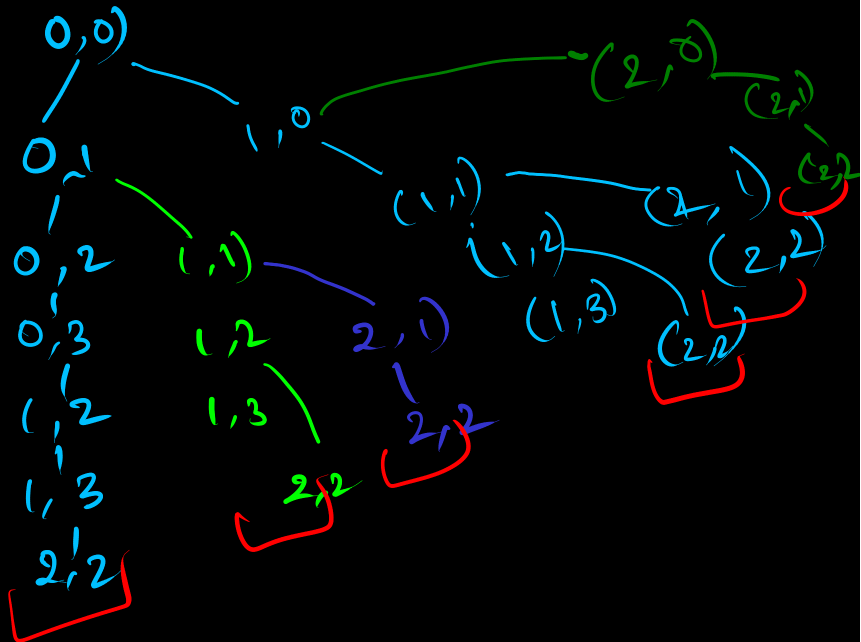
Right

if


(0,0)
(0,1)
(0,2)
(1,1)
(1,2)
(0,3)
(1,2)
(1,3)
(2,2)
(1,3)
(2,2)

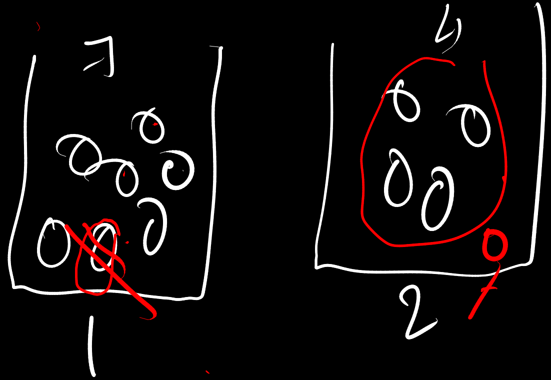


0
1
2
3
0
1
2
3

0,0)
0,1
1,0
(2,0)
(2,1)
0,2
(1,1)
(1,1)
(2,1)
0,3
1,2
2,1
(1,2)
(2,2)
(1,3)
(2,2)
1,2
1,3
2,2
1,3
2,2
2,2

# Piles of Stones : Game




2 players

1. Takes the last stone he wins

Player 1 starts

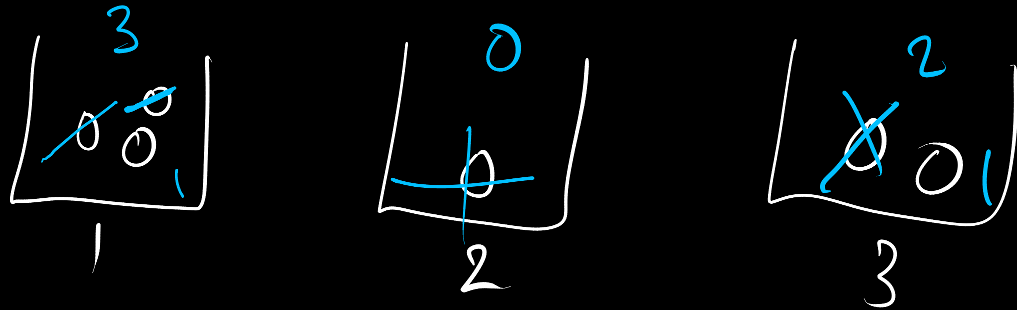
1 pile at a time

Any number of Stones

Choose Pile 1 - 6 Stones

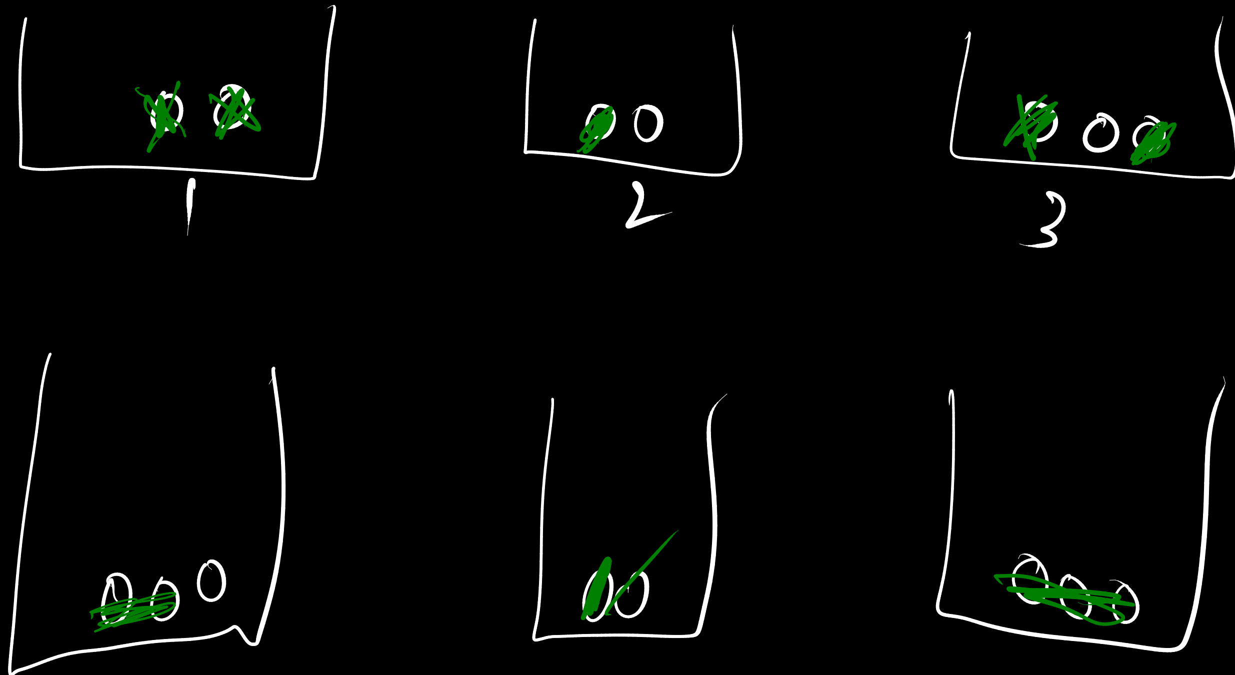
Play 1:

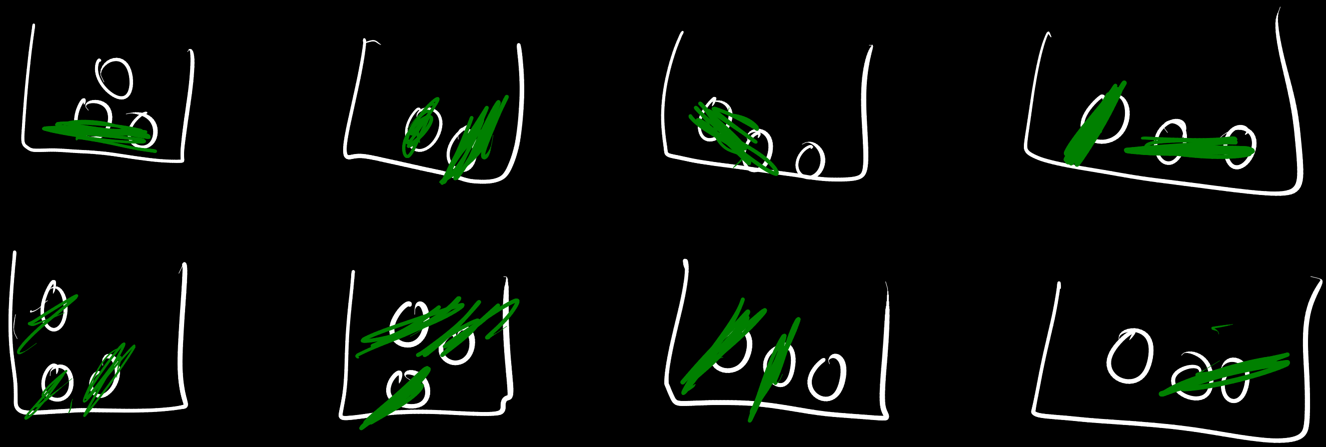
Player 2 : 3 Stones
Pile 2




Player 1 , Player 2

ple 2


1
2
3



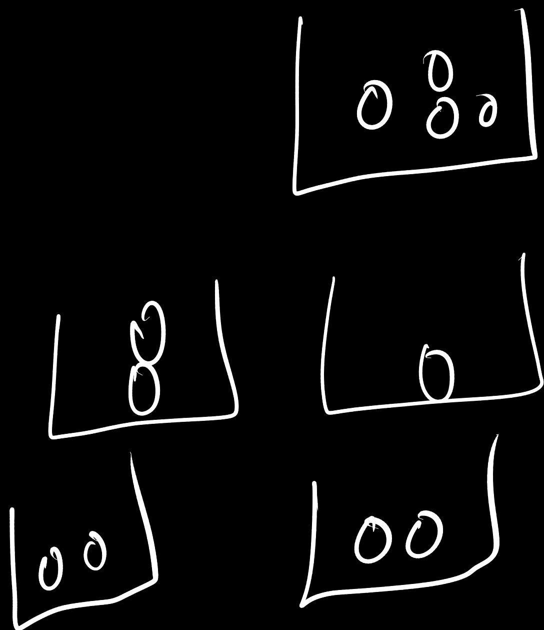
# Piles of Stones

2 Player

1.

2.

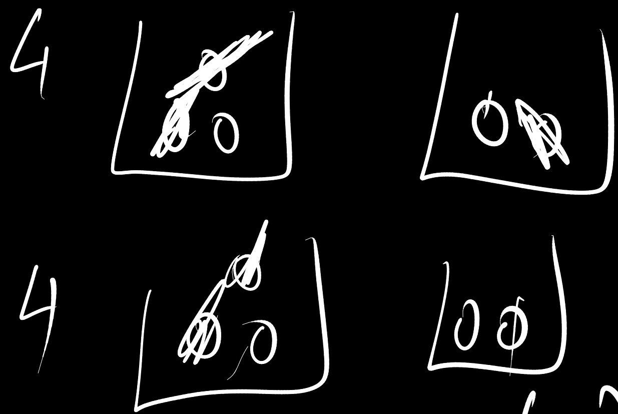
3



Player 1

player 1

player2




Player 1

3 players

$$P(P_1) = \frac{P(P_2 \cap P_1)}{P(P_2)} \text{ (Ans?)}$$

$P(P_2) = 0.6$

$P(P_1) = 0.5$

$P(P_3) =$

$$P(P_3) = \frac{P(P_2 \cap P_3)}{P(P_2)} \text{ (Approx)}$$

Steps!

[2, 2] = Input

Output P1 ?
P2 ?

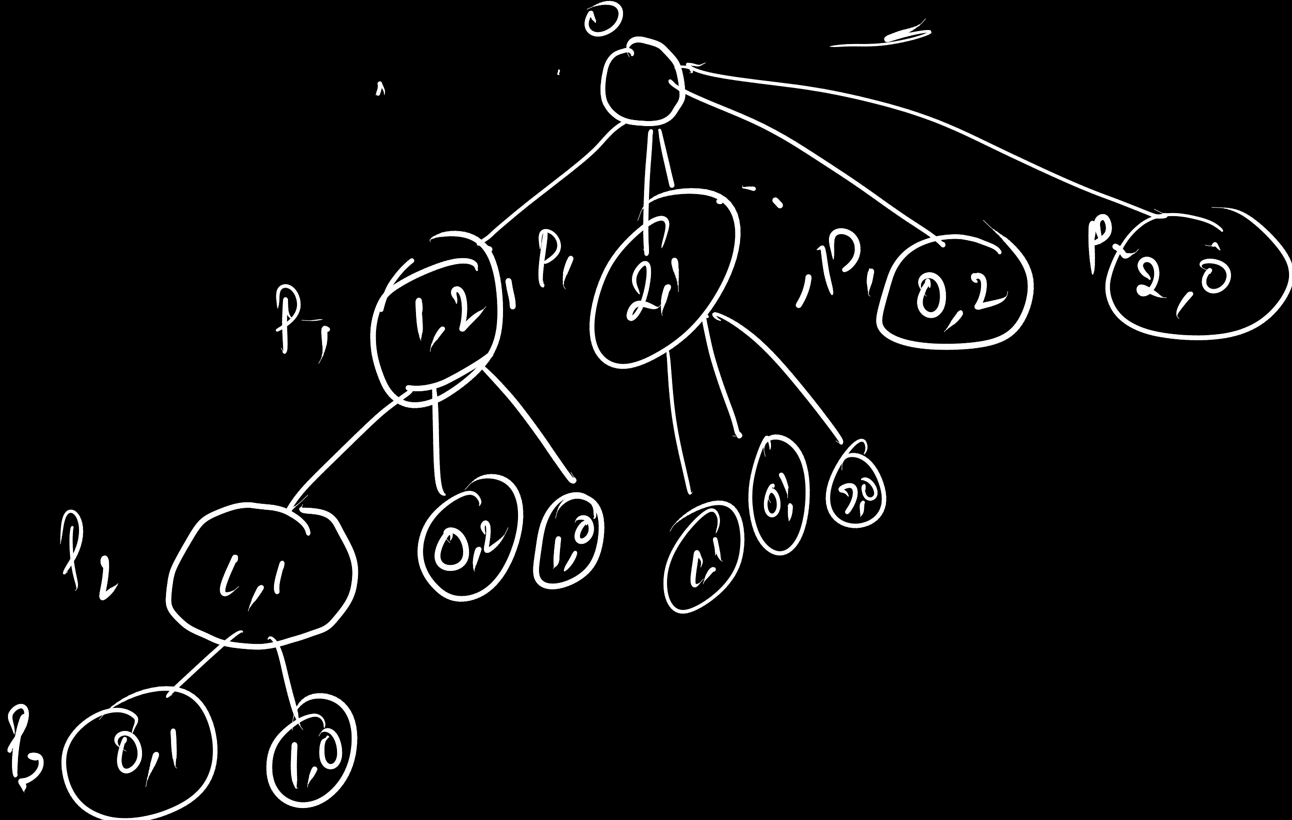
?

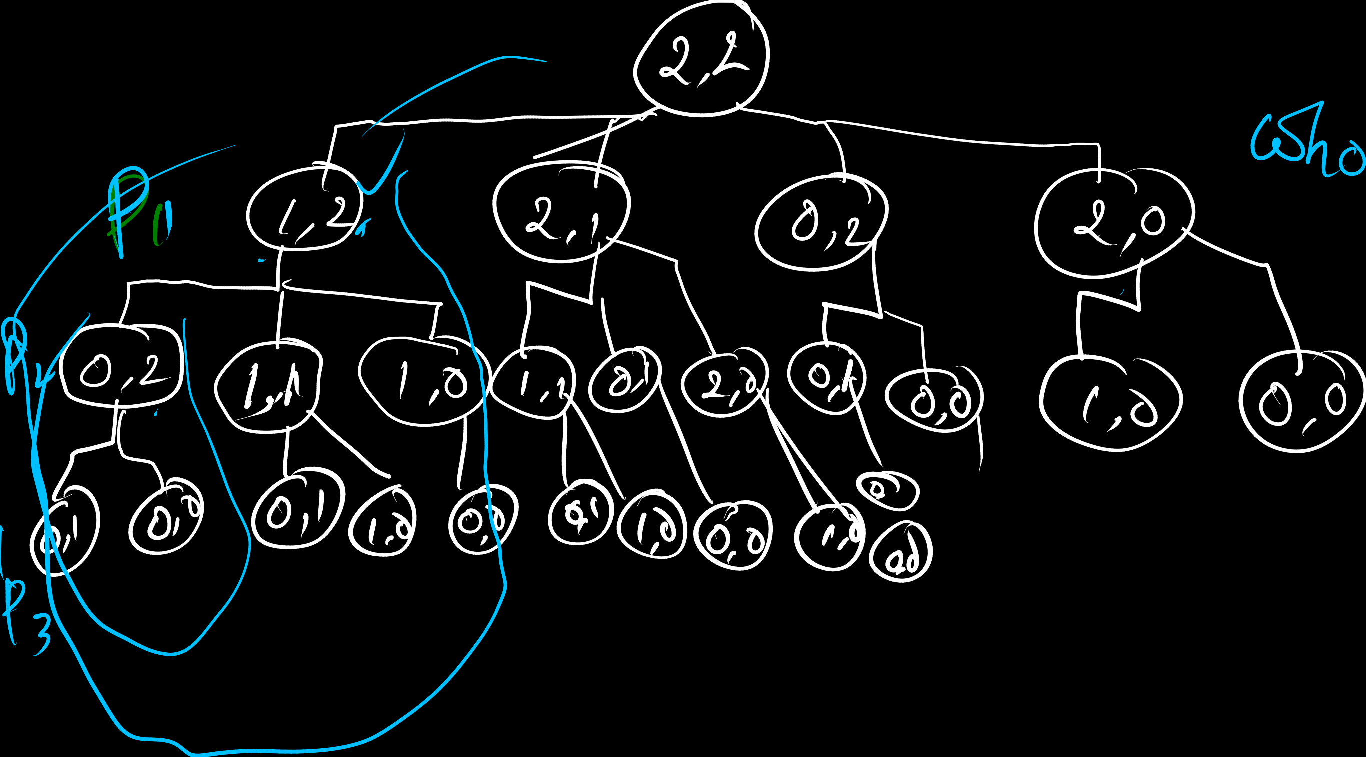
Data Structure

Algorithm:

# Tree Data structure

[2, 2]: empty [the




2,2
1,2
2,1
0,2
2,0
P1
0,2
1,1
1,0
1,1
0,1
2,0
0,1
0,0
1,0
0,0
0,1
0,0
0,1
1,0
0,0
0,1
1,0
0,0
1,0
0,0
P3

Who wins

2 players

↳ Tossing a Coin

10 times each

whoever gets maximum head wins

10 trials $P(P_1) = 0.5$

$P(P_2) = 0.5$

7 : 3

3 : 7

$P_1$ . 10 times

$P_1$ (Head) = 50 %

100 times

~~6.5~~

7 : 3

3 : 7

6 : 4

4 : 6

5 : 5

1000

6 : 4

4 : 6

100000

5.5 : 4.5

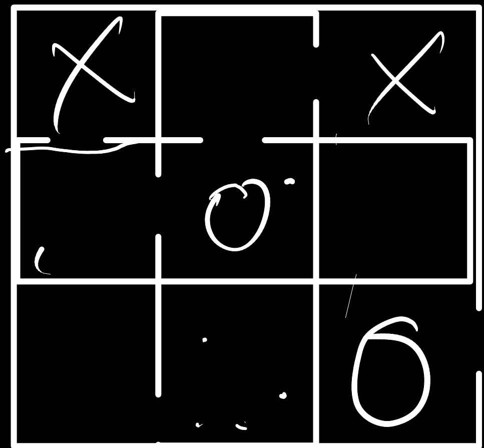
5 : 5

6.5 : 3.5

3.5 : 6.5


Accuracy
90%
99%
1% Error Rate
10
50% (Approx)
1cr. Trials

Player 0
Player X
Player



Tic Tac Toe Two players

Straight line

Data Structure

| | | |
|---|---|---|
| 0,0 | 0,1 | 0,2 |
| 1,0 | 1,1 | 1,2 |
| 2,0 | 2,1 | 2,2 |

winning positi

[0,0 0,1 0,2]

[1,0, 1,1 1,2]

[2,0 2,1 2,2]

[0,0 1,0 2,0]

[0,1 1,1 2,1]

[0,2 1,2 2,2]

[0,0 1,1 2,2]

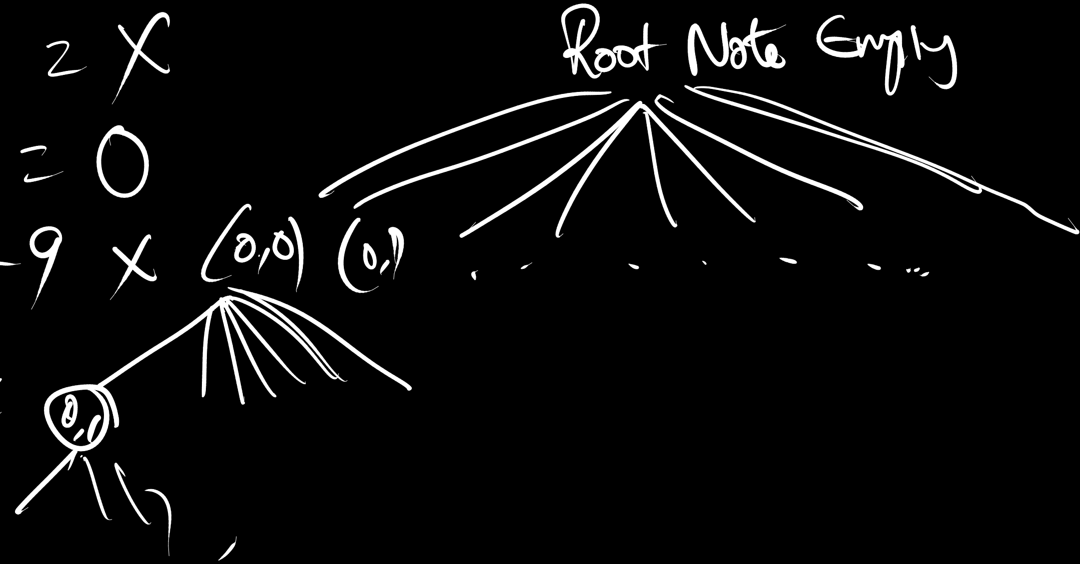
[0,2 1,1 2,0]

8x2

16 poss

Player 1 = X

player 2 = O



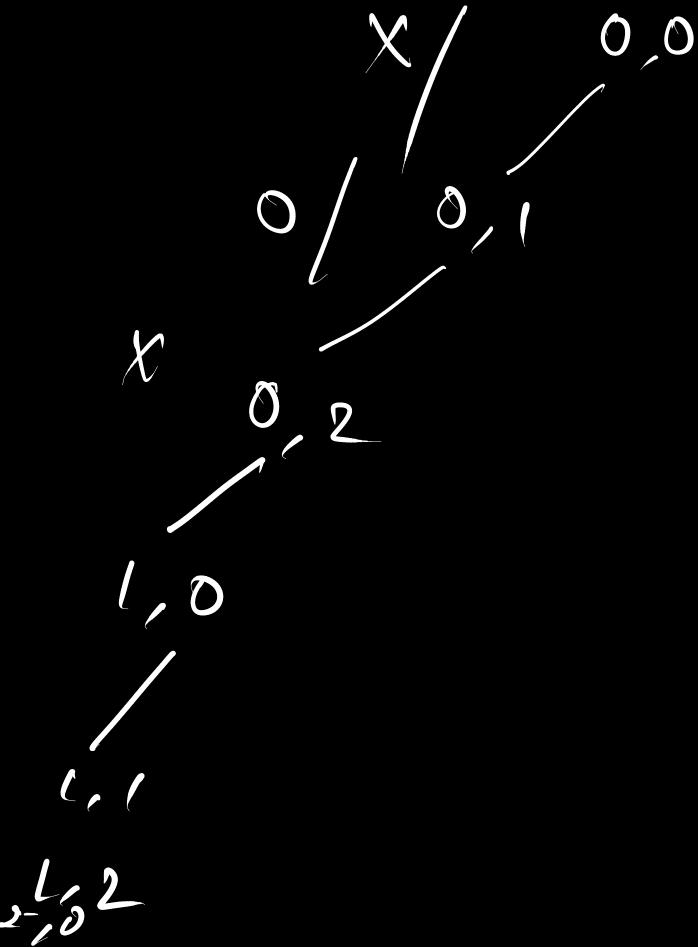

1-9 X

2nd = 8

3rd = 7

4th = 6

⋮




| | | |
|---|---|---|
| X | O | X |
| X | O | X |
| O | X | O |

$72 + 504 + \cdots$

1st 9 posn

2nd $9 \times 8 = 72$ nodes

3rd $72 \times 7 = 504$ nodes

4th $504 \times 6 =$

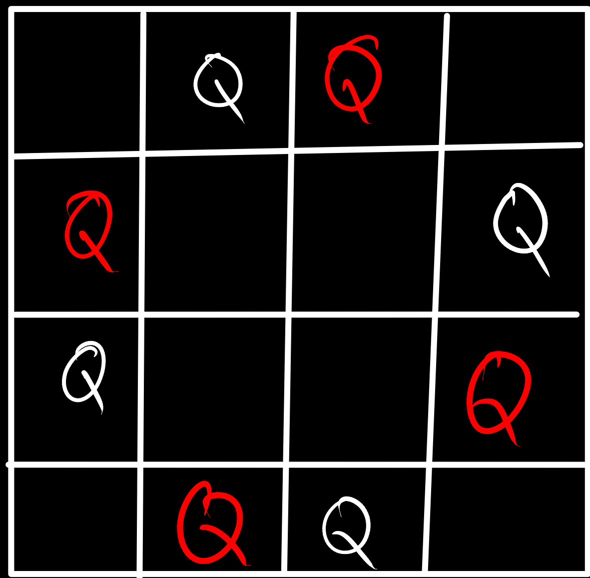
5th $504 \times 6 \times 5$

6th $504 \times 6 \times 5 \times 4$

$\vdots$

# 4 Queen Problem

4 x 4




Place All 4 queens

(0,0) (0,1) (0,2) (0,3) (1, ...

16 nodes

16 × 15 nodes

(0,1)

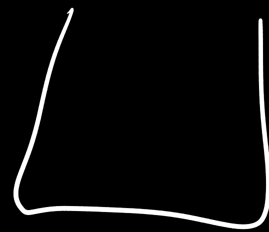
(0,2)

(0,3)

(1,0)

# Knapsack Problem

| Items | Weights | Value |
| --- | --- | --- |
| 1 [Bucket] | 1 | 1 |
| 2 [Water] | 3 | 4 |
| 3 [dress] | 4 | 5 |
| 4 [Weapon] | 5 | 7 |

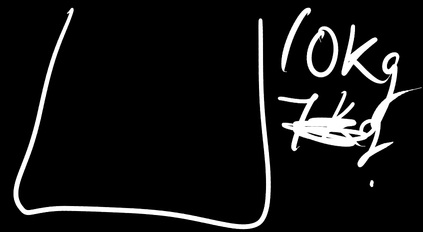


10Kg

~~7Kg~~?

# Knapsack Problem

| Items | Weights | Value |
| --- | --- | --- |
| 1 [Bu[illegible]] | 1 | 1 |
| 2 [Water] | 3 | 4 |
| 3 [dress] | 4 | 5 |
| 4 [Weapon] | 5 | 7 |



Sorting Algorithm

[2, 1, 4, 3, 6, 5]

[5 4 3 2 1] Alread Sorted

[1 2 3 4 5]

$a = [\, 1 <= 2 <= 3 <= 4 <= 5 \,]$

$\quad\;\; 0 \quad 1 \quad 2 \quad 3 \quad 4$

$O(5) = O(n)$

$O(n^2)$. Reduce

$n = 5$

for ( i = 0; i < n; i++) {

if a[i] > a[i++]

not sorted in ascending order

break;

[5 4 1 3 2]

Selection Sort → largest element will be placed in the last

Iter 1

4 5 1 3 2

4 1 5 3 2

4 1 3 5 2

4 1 3 2 5

4 1 3 2 5

1 4 3 2 $\boxed{5}$

1 3 4 2 $\boxed{5}$

1 3 2 $\boxed{4}$ $\boxed{5}$

---

1 3 2 $\boxed{4}$ 5

---

1 2 3 4 5

$n$

$n-1$

$n-2$

$n-3$

$n-4$

$O(n^2)$

$O(n \times (n-i))$

$i = 1$ to $n-1$

# Bubble Sort

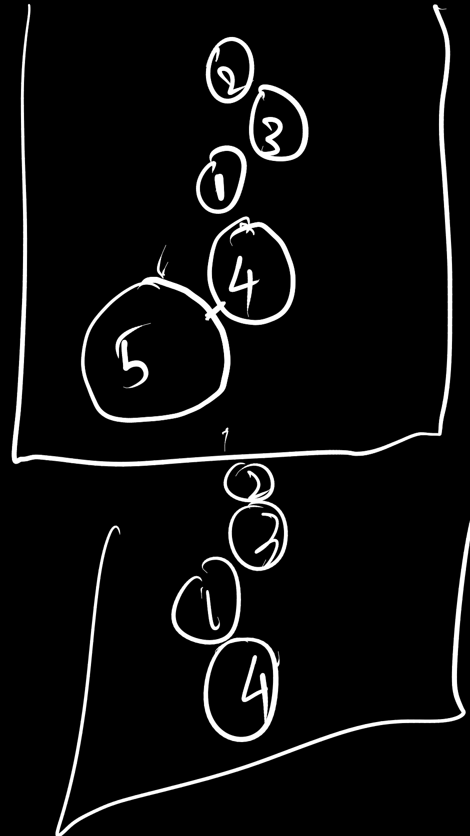
[5 4 1 3 2]

[4 5 1 3 2]

[4 1 5 3 2

4 1 3 5 2

4 1 3 2 5




$O(n^2)$

# Merge Sort

[5, 4, 1, 3, 2]

[5 4] [1 3 2]

[5] [4] [1] [3 2]

[5] [4] [1] [3] [2]

[4 5] [1 3] [2]

~~Con~~ Divide/

Merge

Conquer

[1 3 4 5] [2]

[1 2 3 4 5]

$O(n) + O(n) + O(n)$

$O(n \log n)$

→ Insertion    (Quick Sort)

$O(n\log n)$    $O(n\log n)$

[5 4 1 3 2]

1

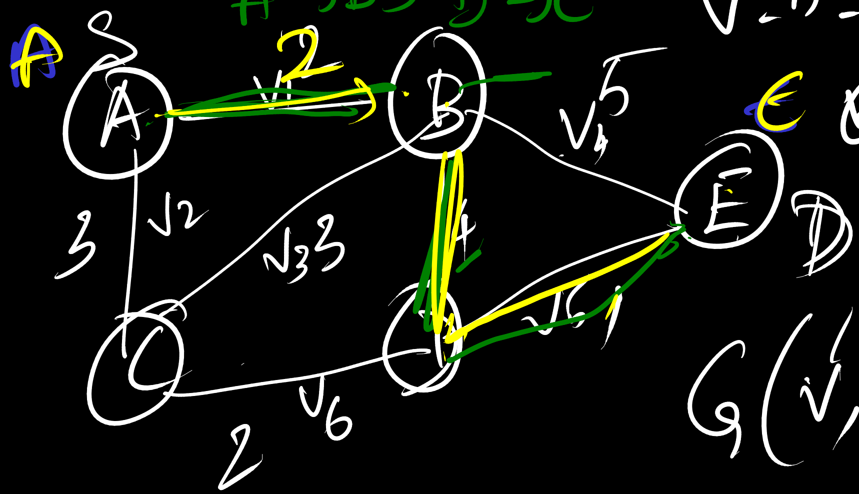
5

4 5

# Shortest Path Algorithm (AI - DS)

**Graph**

A → B → D → E




$V = n = \{A \ B \ C \ D \ E\}$

$E = \{V_1 \ V_2 \ V_3 \ V_4 \ V_5 \ V_6\}$

6

$G(V, e)$

Bruteforce

$n/2$

Graph → Traversal Problem [Mini Data]

↓

Tree

A(S)

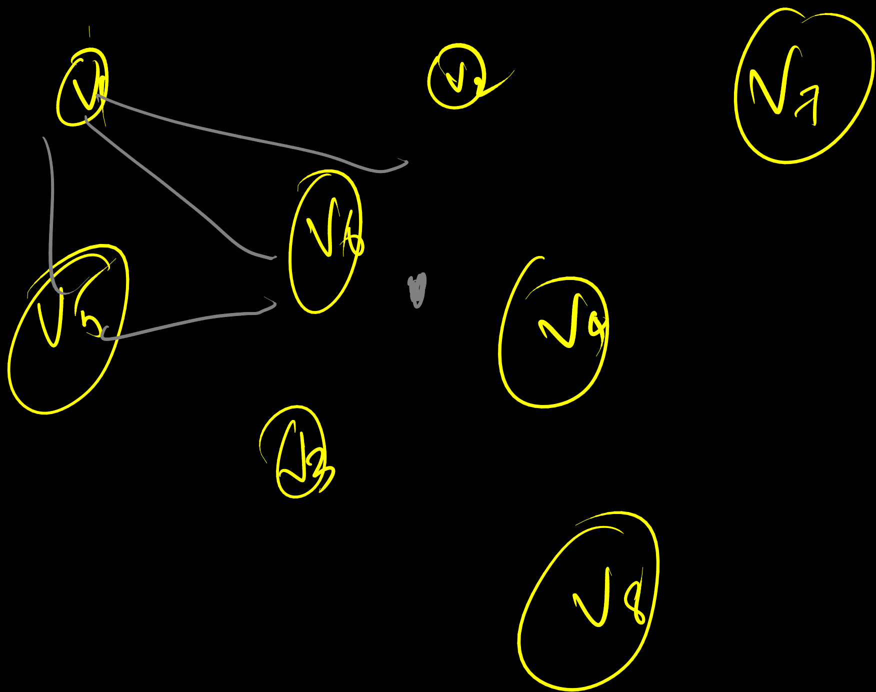
E(D)

Graph [loops]

Tree [no loops]

1 ways Connection

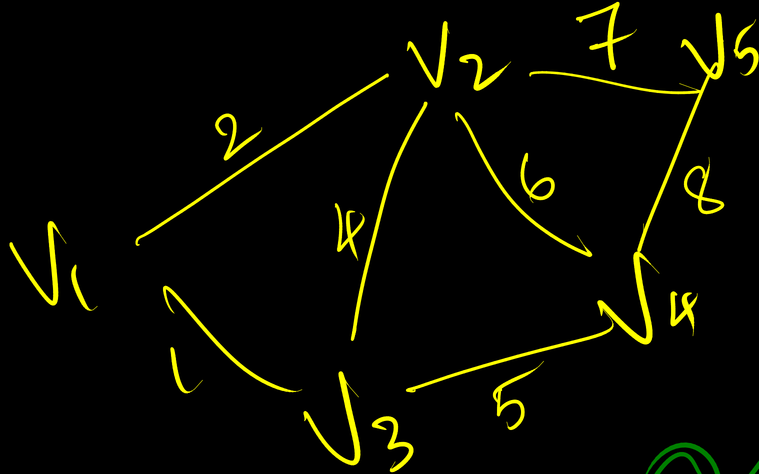
2 way Connection
(Bidirection)



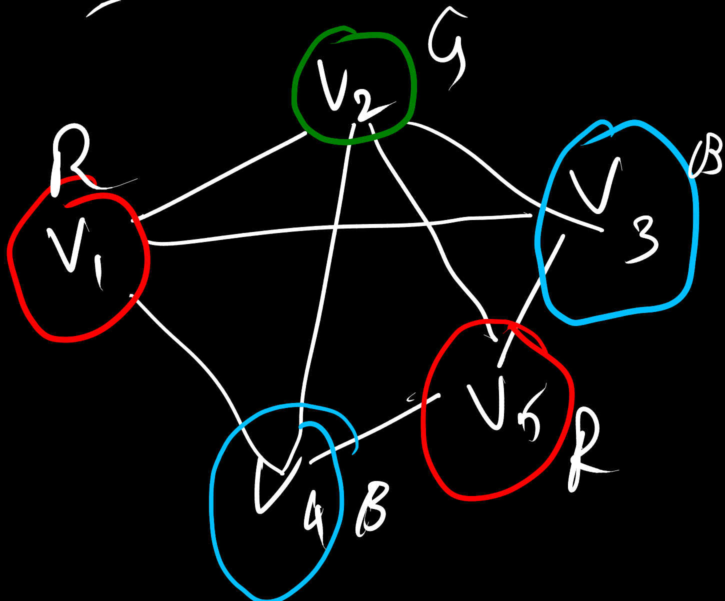
V1
V2
V7
V6
V5
V3
V8

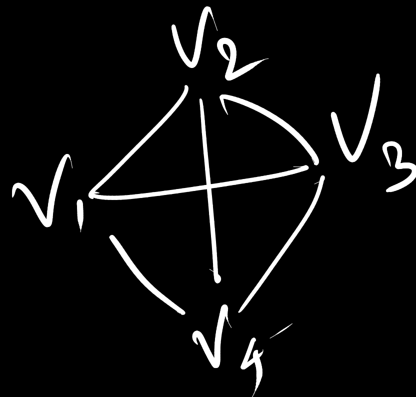
Graph Tree

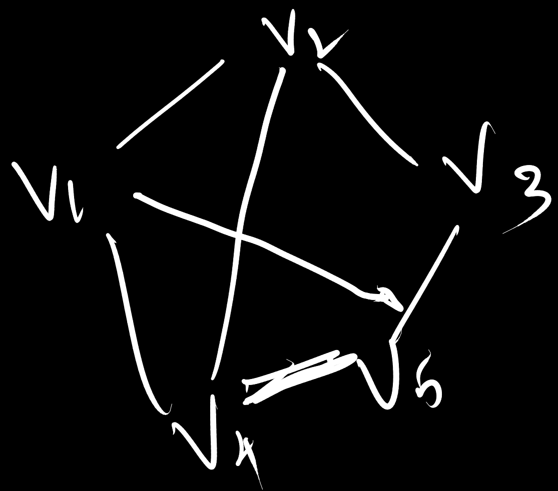
Minimum Spanning Tree




V = 5   Vertex Coloring Algorithm





V1
V2
V3
V4
V5

# Search Algorithm

[5 4 3 2 7 8 9]

Key 2 = Binary

[2 3 4 5 7 8 9]

Sort It

4) Data is Sorted

$O(n)$ = Linear Search

$O(\log n)$ = Binary Search

$O(2)$ = Linear Regression Search

AI Strategy Miss (th) Game → Mine Game

4 x 4

Player 1 - X

Player 2 - Z

Minesweeper

| | | | |
|---|---|---|---|
| ✓ | | Z | |
| Z ✓ | X | | |
| | Z | | X |
| X | | | |

Player 1 Strategy

Player 2

Hit / Miss

Sheets

Ship

Airwarrior

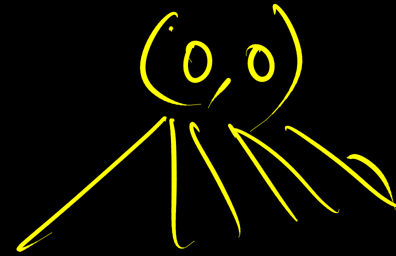
2 player Minesweeper

etc

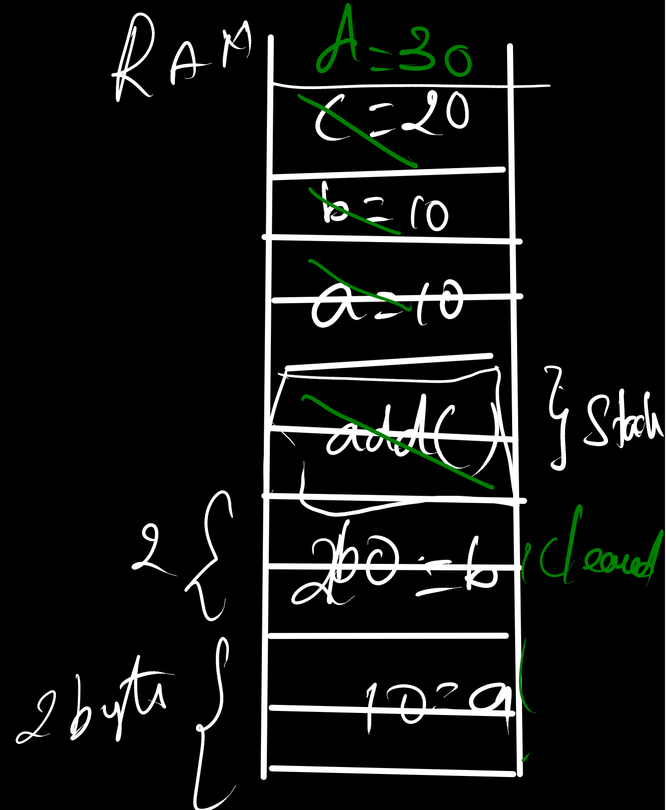
1. Game Theory Basics
2. Optimal Decisions
3. Alpha Beta Search.
4. Monte Carlo Search
5. Stochastic Games

# Sudoku Game

| | 0 | 1 | 2 | 3 |
|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 |
| 1 | 2 | 1 | 4 | 3 |
| 2 | 4 | 3 | 1 | 2 |
| 3 | 3 | 4 | 2 | 1 |

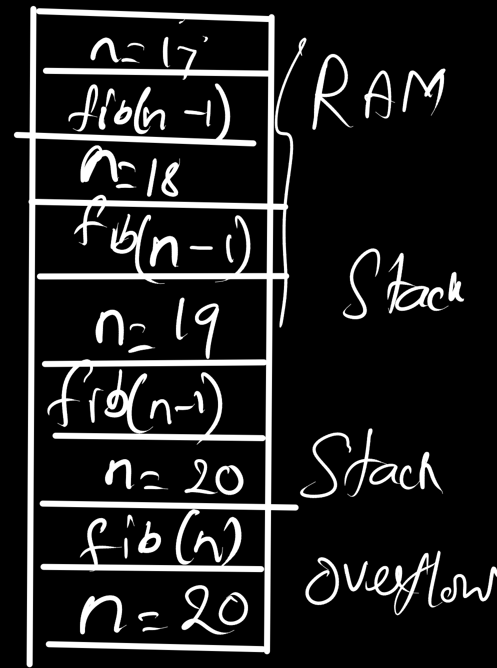


```
a = 10 [Integer]
b = 20
d=add(a,b)
def add(a,b):
    c= a+b
    return c
```



```
def fib(n):
    if n==1 or n==2
        return 1
    return fib(n-1)+fib(n-2)

n = 20
f = fib(n)
print(f)
```

Exit function




System will hang

Mobile RAM

} RAM

facebook

&

Mess

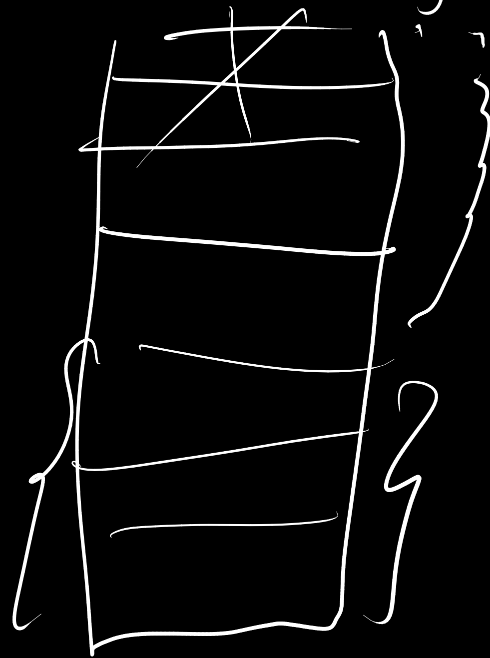
Wats

Youtube

:

filesystem

photos



Slow down

=

# Data Structures & Algorithms

↳ How optimal your code

[Memory] – RAM → Space Compl
How many variable declar

[Execution Time]
→ Time Complexity

$O(n^3)$ $O(n)$
$O(n^2)$ ...

Search Algorithm

Ascending order]

Input [ 4, 10, 20, 100, 111, 112 ... ]

Searching will be easier.

10 hours

Binary Search

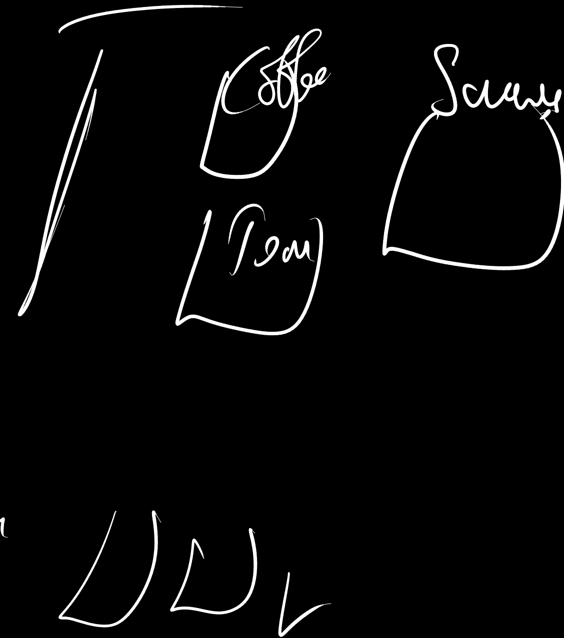
15 mins.

$O(\log n)$

Dress Cabn

Arrange.

Kitchen



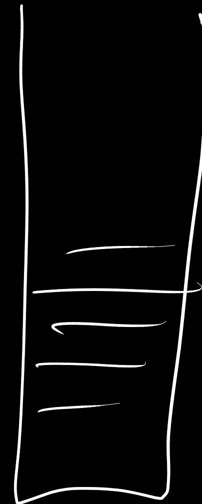

Salt

Array [ ]

a[10000] Continuous

Disadvant

linked list Continuous allocation



# Algorithms.

Searching → Linear
- Binary
- Regression

Sort → Selection
- Bubble Sort
- Insert
- Quic

Heap Sor.

Merge St

Strategies

- Brute force
- Divide & Conquer.
- Divide Merge Conquer.
- Breadth first Search
- Depth first Search
- Dynamic Programming

Tree Conns

↳ Binary Tree

↳ AVL Tree

↳ B Tree

↳ $B^+$ Tree

↳ . . .

Complete Tree

Partial Tree

full Tree

⋮

# Graphs

Adjancy Matrix

Incidence Matrix

# AI BootCamp

Non Polynomial Equation - DL

↳ DataSet -

↳ Computational Power / Resources.

DataSet - Data [plural]

Datum

Structural

Unstructured

↳ Text

Cater ↳ Image

↳ Video

↳ Numerical

↳ Signal

Hybrid Data

↳ Types

Interval Type of Data

Categorical Data

Multivalued Data

Mut Valued Interval Type of Data

AI → Concepts

↳ Strategies. → Gaming

↳ Hill Climbing. → Global Maxima

↳ → Global Minima

Gradient Descent

↳ Local Maxima

↳ Local Minima

Filters / Edge Detection Algorithm

↳ Canny
↳ Sobel
↳ Prewitt

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

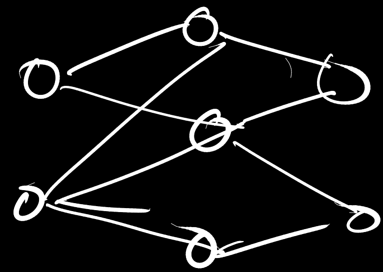
Local feature
feature

ECE →

$$\begin{bmatrix} 0 & 1 & 1 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

Kernels

filters 32 filters
64



Stored
Model

Hill Climbing

Search Strategy

Uninformed

└ Breadth first Search

└ Depth first Search

Algorithm Strategy

└ Dynamic Program

└ Brute force Search

└ NP Program

Local Maxima.

Hill Climbing

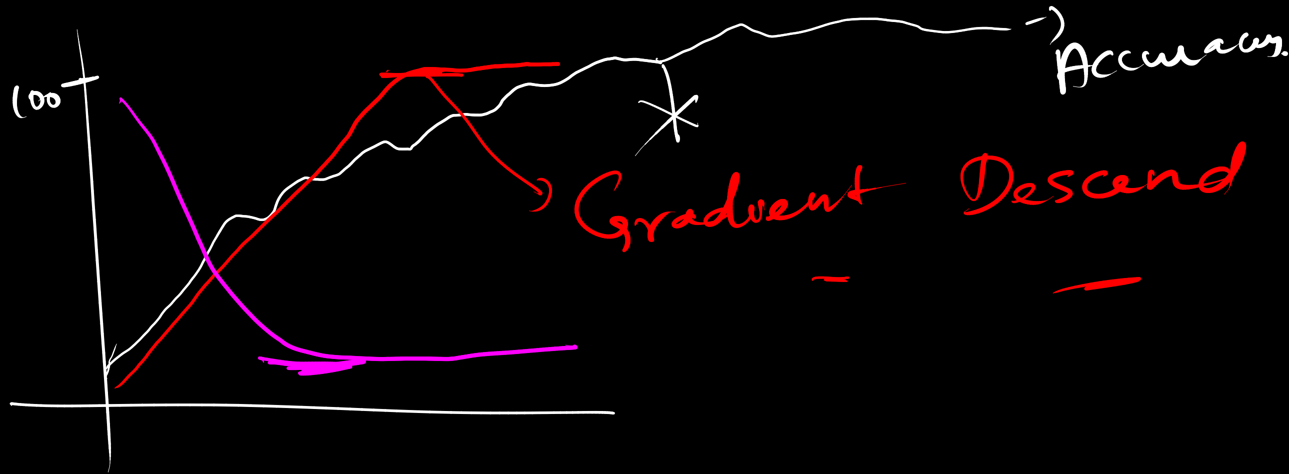
Local Max

Global Max

Local Maxima.

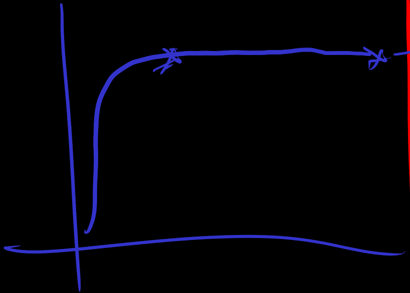
$T_1$ $T_2$ . . $T_3$ $T_5$

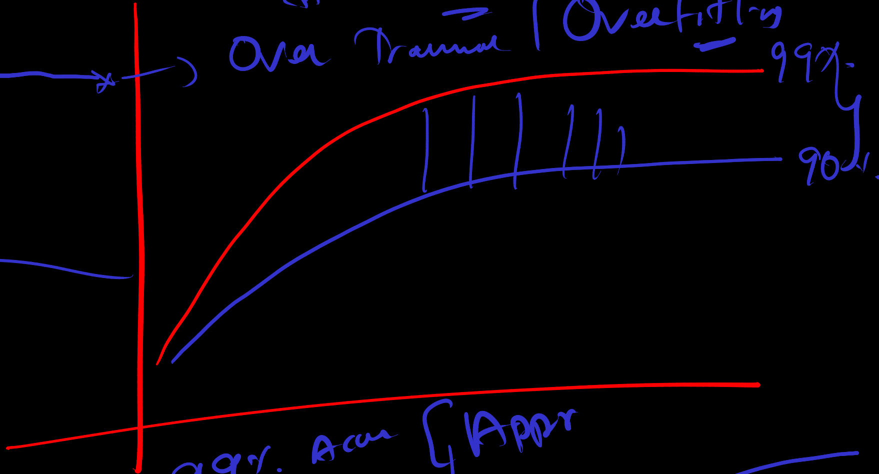
# Monotonically Increasing Graph



Efficiency in Over Training

{ Overfitting





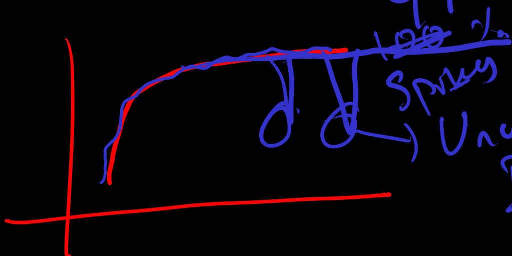

Underfitting Less Data

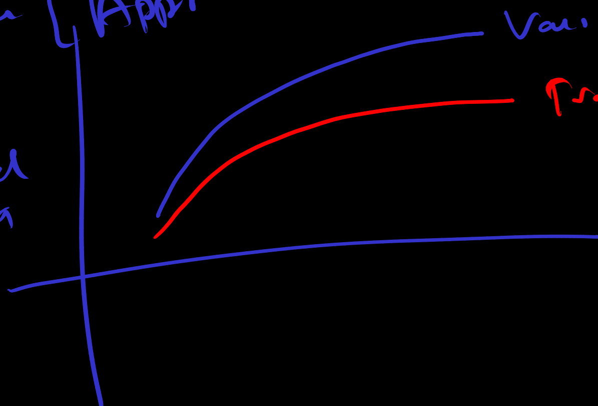
Proper Train Less Train

Converged

Sales

99% Accu { Appr




} failure...

Parameters | Hyper parameters

Parameters | Hyper Param. | Semi Supervised

Epochs → 200 / 1000

Learning Rate → Jumps

5 lakh

SKU

70% Accuracy

LR → .01

LR → 20 meters
100 m

Search, Strategy 3D

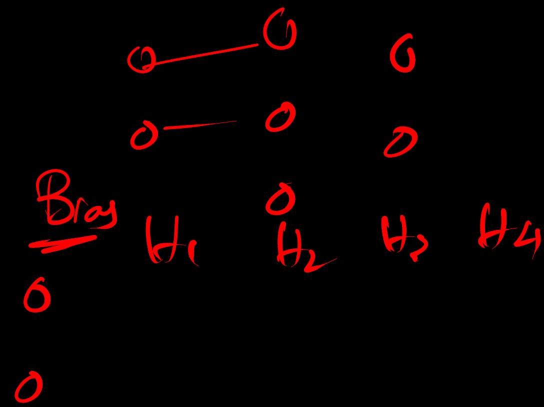
Networks →

Full Connected Network

Partial Connected Networks

Shallow Network

Deep Network



Bias - favourite

↳ Bias → Make the features Distribute.

→ Minimal Loss.

1 .9

0.8

0.3

0.4

0 0.1

0.345

$B_1$ 0

$B_2$ 0

$B_3$ 0

Dynamic & Strategy

↳ This Bias will be Recomputed for Every Iteration

Based on the loss

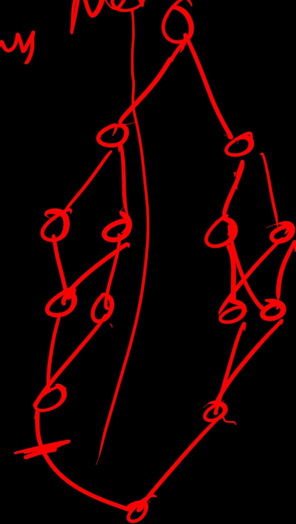
# Deep Network

↳ Connectivity

↳ Simple Network

↳ Parallel Network

Bypass Net...



# Strategies

Tossing a Unbiased Coin

10 Tries

$P(\text{Getting head}) = 0.5$

$P(\text{Getting Tail}) = 0.5$

0.3

$P(\text{H})$ 0.7 0.7

$P(\text{T}) = 0.3$

100 Trials

# Trials

1 lakh.

0.49.... P(H) = 0.6 / 0.4

P(H) = 0.5

0.59.... P(T) = 0.4 / 0.6

P(T) = 0.5

1000 Trial.

= 0.55

0.45

Approximation

Relationship

Term

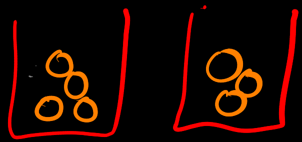
# Piles of Stones

↳ Who picks up final stone.

# Bags

2 players

Minimum of 1 Stone



Player 1

Alternate

Strategy Constraint Resource Problem

DFS

Search Structure Strategy

Can you Build an AI agent to be a player

Dfs

Algo

↳ Recursion

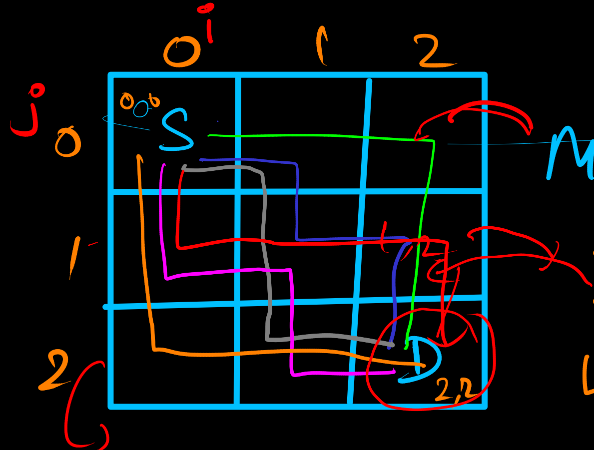
—

↳ Move Right

↳ Move Down

i+1, j

i, j+1




Maze Game

3 x 3

↳ Right

↳ Down

↳ Searching
↳ Sorting } Recursion

↳ Down

↳ Right

Robo

↳ Travel

Horizontally

Vertically

Not Diagonal.

Recursion

↳ Base Condition / Exit Condition

if (i == 2, i == 2)

Count ++

↳ Stack Overflow

Algo

Maze (c, i, j)

Templat

if i == 2 j == 2

c++

if i > 2

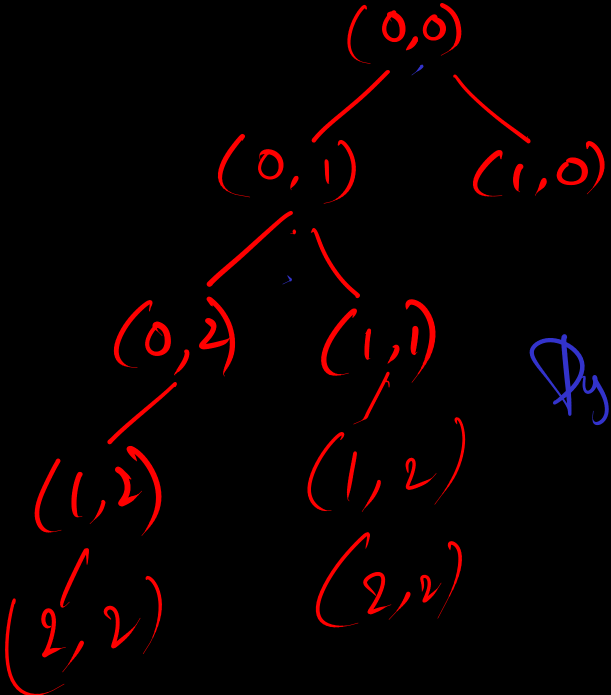
return

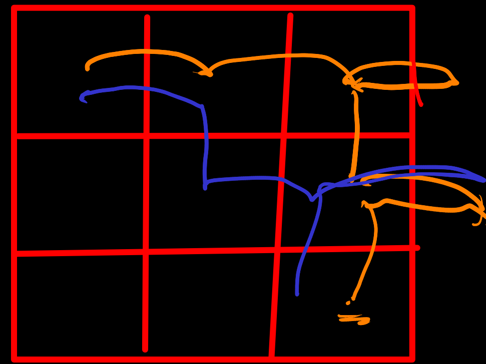
if j > 2

return

Maze(c, i+1, j)

Maze(c, i, j+1)



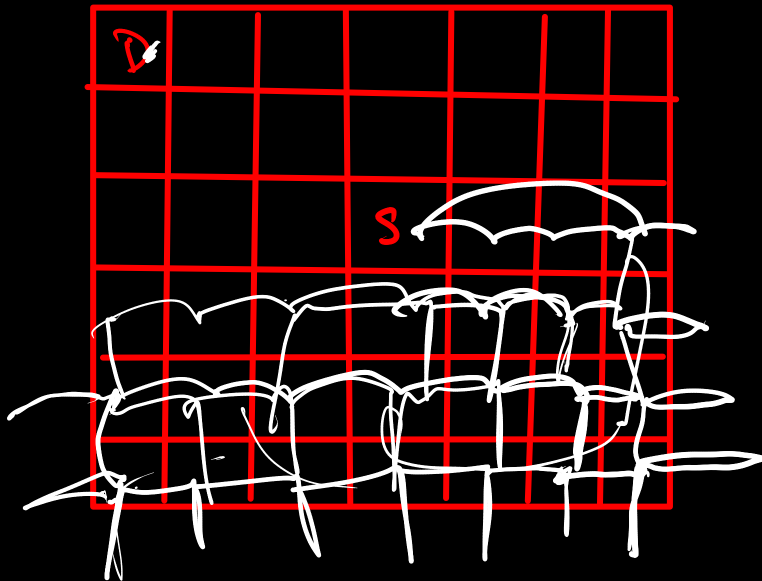



Dynamic

Memoization

Memoization

L
U
R
D




Right R

Left D

Bottom L

Up U

LR = 0.1

0.3

# Fibonacci Series :

# Goals Problem

1 1 2 3 5 8 13 . . .

↑ ↑ ↑ ↑ ↑ ↑ ↑

n = 1 2 3 4 5 6 7

if n = fib(n) = ?

Exit / Base Condition

if n == 1
return 1

if n == 2
return 1
(n-1) fib(n-2)

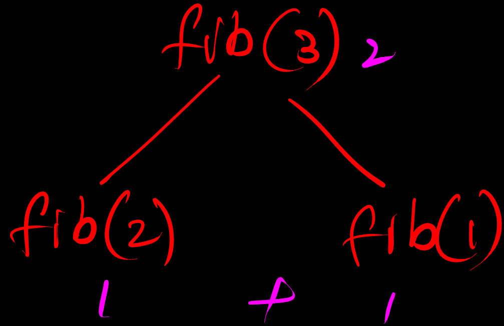
```
def fib(n):
    if n == 1
        return 1
    if n == 2
        return 1
    return fib(n-1) + fib(n-2)
```

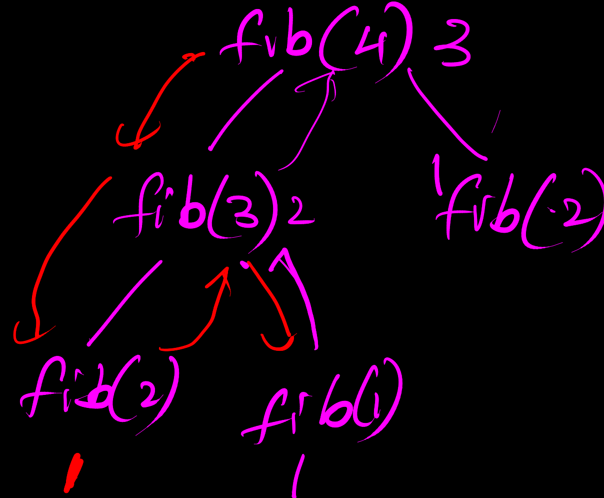
$n = 3$; $n - 1 = 2$; $n - 2 = 1$

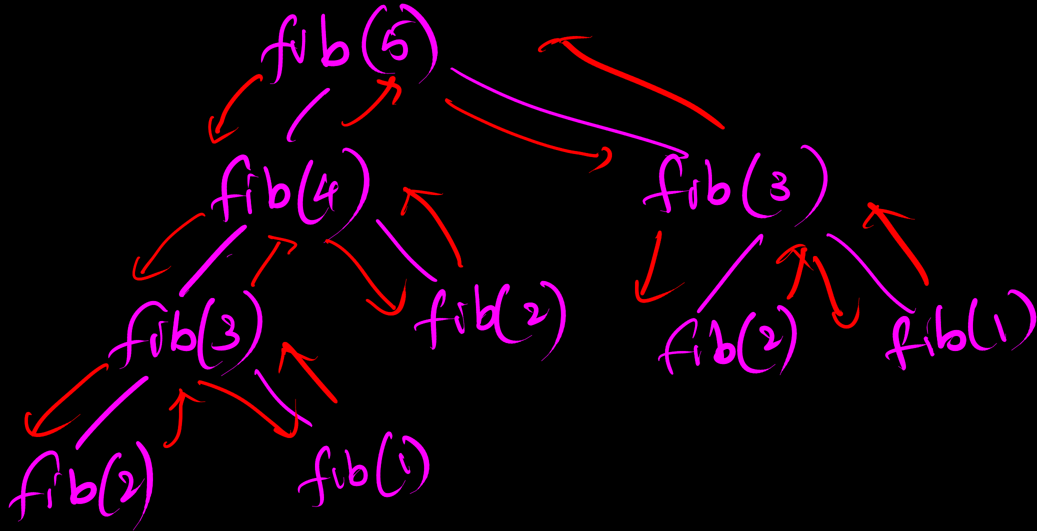
$= 1 + 1$

$fib(3) = \underline{2}$

```
def fib(2)
    if n = 1
        return 1
    if n == 2
        return 1
    return fib(n-1)+fib(n-2)
```

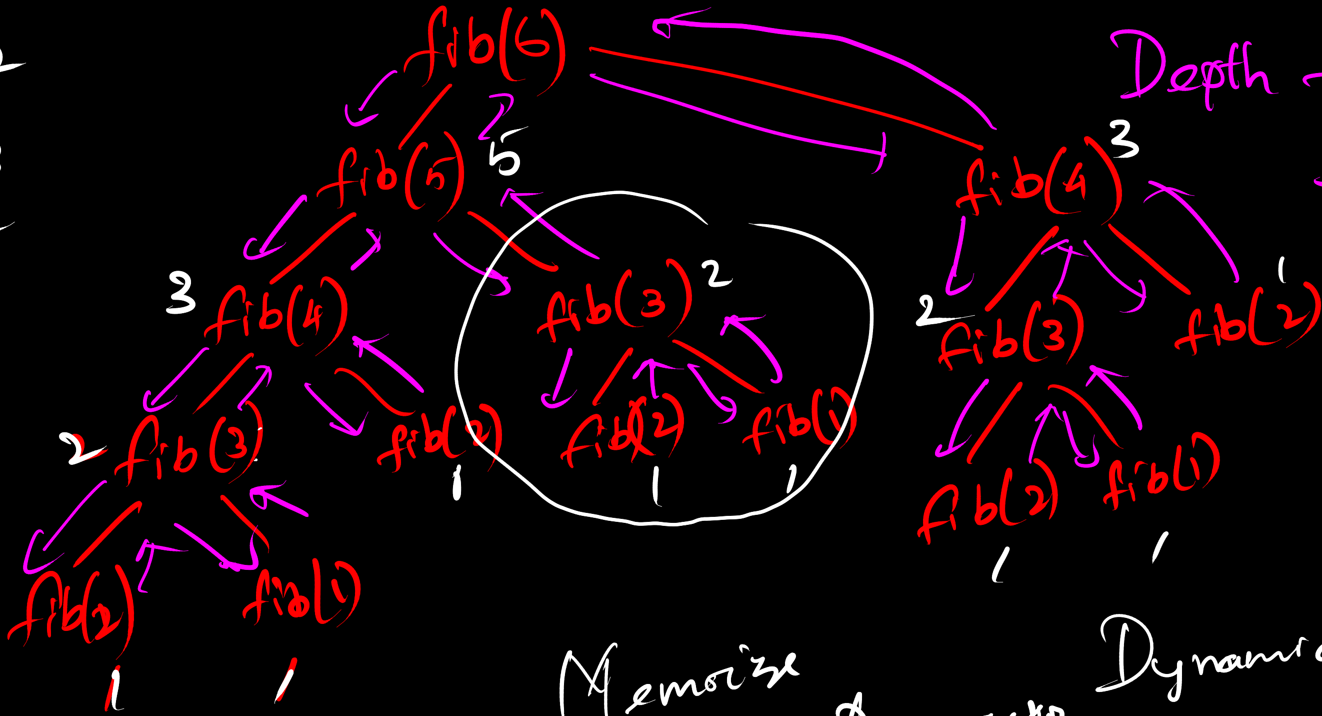

fib(3) 2
fib(2)
fib(1)
1
+
1


fib(4) 3
fib(3) 2
fib(2)
fib(2)
fib(1)
1
1


fib(5)
fib(4)
fib(3)
fib(3)
fib(2)
fib(2)
fib(1)
fib(2)
fib(1)

| | |
|---|---|
| fib(3) | 2 |
| fib(4) | 3 |
| fib(5) | 5 |

fib(6) = 8

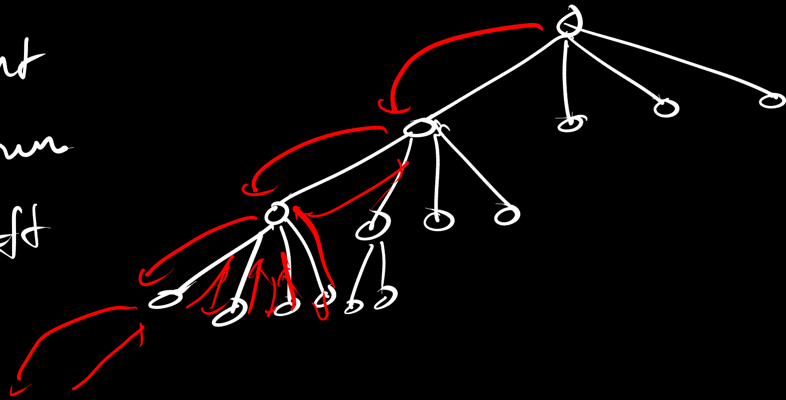



Depth first Search

Memoize

— Optimization: Dynamic Prog

mov Right
Down
Left



# Distance Measures

MSD

Euclidean Distance

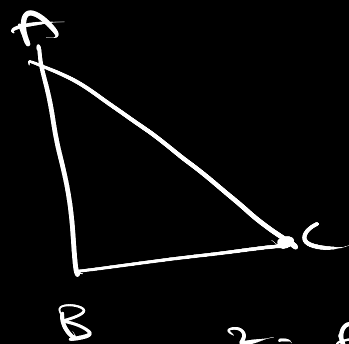
City Block

Chess Board

Jaccard

Cosine Distance Meass

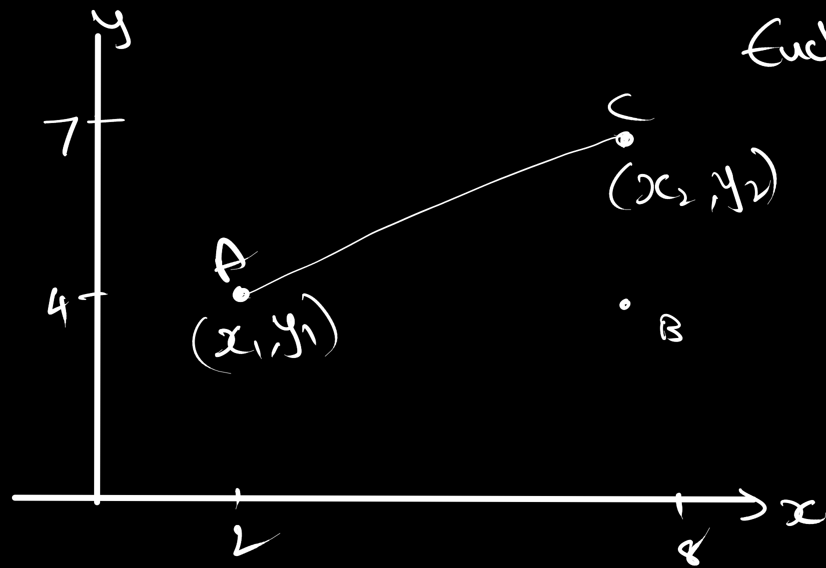
Pythagorean Theorem




$$AC^2 = AB^2 + BC^2$$

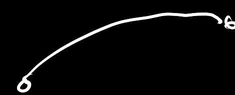
$$AC = \sqrt{AB^2 + BC^2}$$

Euclidean Dist




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

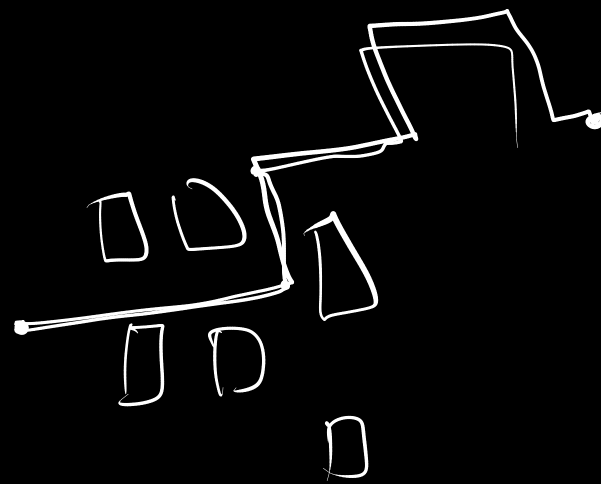
ABS



City Bloch

Chess Board

Highways



Jaccard Distance

Cose Distance measure

↳ Recommendation System

↳ Movie Suggest

Algorithm → Strategies

Maths Basics

Prob / Stats

Graphs

Narsingh Deo

$G(n, e)$ → edges → arcs

↓

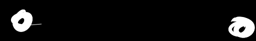
# vertices

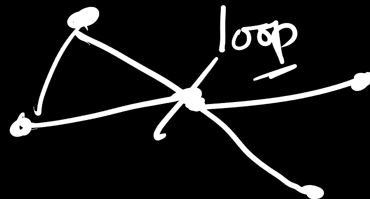
nodes

Cut Vertex

Cut Edges

Disconnected Graph





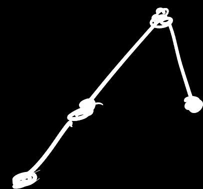
loop

Graph

→ Single node

→ Connect

Tree is also a Graph



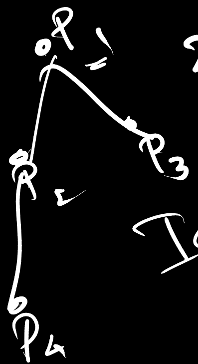
DSA → ✓ Graph

Friend Sugs.

Domain Transformation.

FaceBook

LinkedIn




2rd

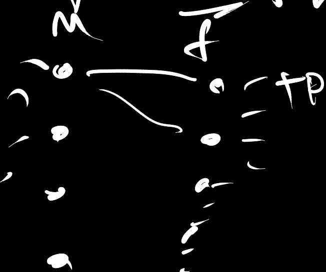
Algorithm. → Graph

Isomorphic Graph

Matrimonial Sites

Graphs

Bipartite Graph




Hamiltonian Graph

Walk

Cycle

Path

Shortest Path | Distance

Salesman Problem

Prims Algorithm

Kruskal's Algorithm

Tree

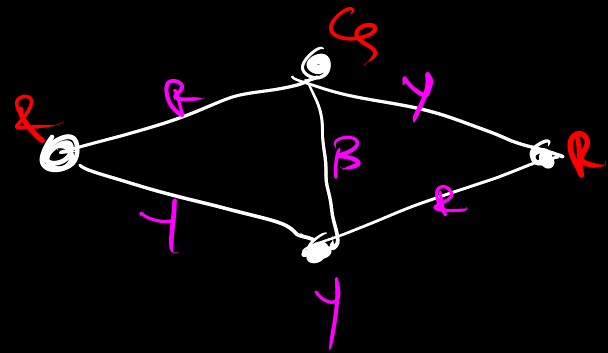
Graph Coloring

Algorithm

Vertex Color

Edge Coloring

Algorithm

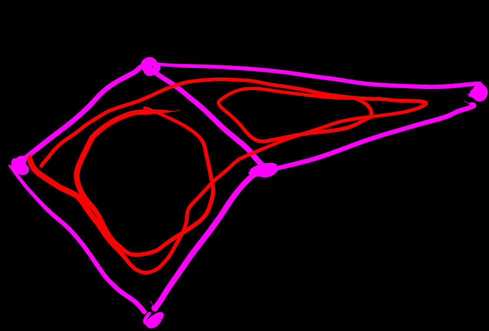



Vertex Color Alg

Minimum # Color

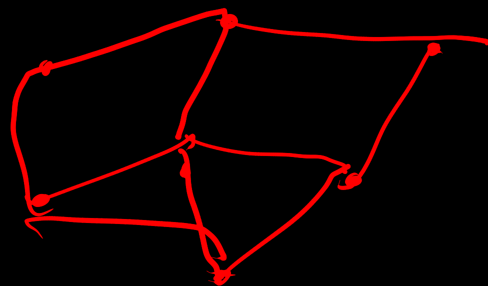
All Vertices

No adjacency Shall have same Color



Open Problem

# of Cycles in a General Graph.

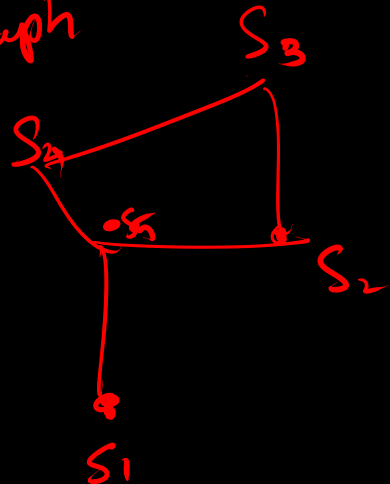


Sub Graphs

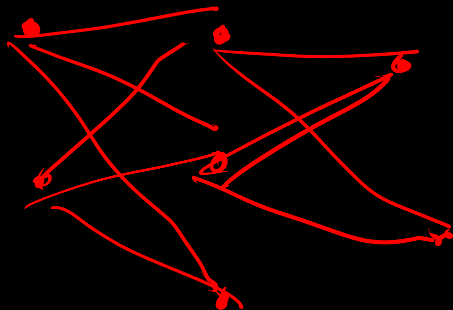
# Indian Railway Network

Graph

Optimize it



→ forward



Graph.

→ Directed Graph

Undirected Graph


→ One way

Parallel Edges

→ Chart

Adjacency Matrix

Incidence Matrix

| | $e_1$ | $e_2$ | $e_3$ | $e_4$ |
|---|---|---|---|---|
| $V_1$ | 1 | 0 | 1 | 0 |
| $V_2$ | 1 | 1 | 0 | 1 |
| $V_3$ | 0 | 1 | 1 | 0 |
| $V_4$ | 0 | 0 | 0 | 1 |

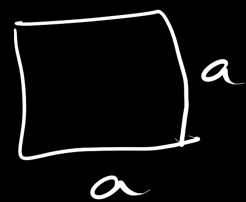
Undiret

| | $V_1$ | $V_2$ | $V_3$ |
|---|---|---|---|
| $V_1$ | $d_1$ | $d_2$ | $d_3$ |
| $V_2$ | | | |
| $V_3$ | | | |

Undiretd Gr

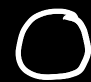
Dire

# Area of Square

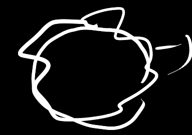


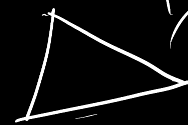

$a^2$

$l \times b$

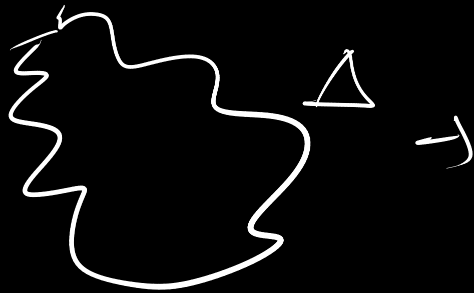


$\pi r^2$





$\frac{1}{2} bh$



↳ Prob / Statis ⟶ Hypothesis Testing

{ Condition — ⟶ Normal Dis

↳ T test

z test

f test

$H_0$ - Null Hypo

$H_1$ - Alternat Hypo

Algebra ⟶

↳ ML ⟶ Linear Regress

↳ Logistics Regression

↳ Trees Decisions

↳ Random forest

↳ Boosting & Bagging Concepts

↳ feature Selection ⟶ PCA, LDA, SVM!

Datasets.

Bagging

↳ DIA

Pattern Recogn

Distance Norms

Deep Learning

↳ Neural Networks

↳ Shallow Net

↳ Deep Networks

↳ CNN -

↳ RNN

↳ Conversation AI

↳ Generative AI

↳ Image to Text

Reviewer

Recent Research Architecture

Bias

Activation functions

AI Concepts

Meta

Data Science

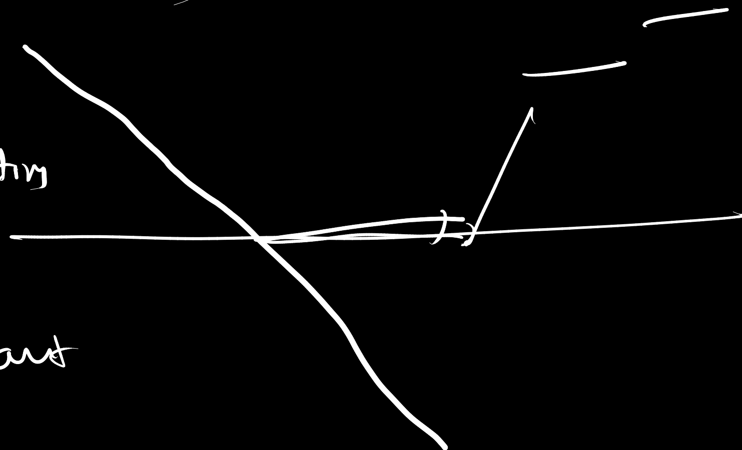
Excel

Pandas

# Activation functions

↳ Control the Signals

↳



Softmax Act function

Sigmoid Act funct

Relu Act Funt

TanH Act funt.

$\boxed{e^{-x}}$ $\quad e^{-x} \quad 0 - 1$

$e^{-1} = .36$

$e^{-2} = .13$

$e^{-3}$

$e^{-4}$

0 to 1

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

Soft max

$f(x) = \begin{cases} x, & \text{if } x \\ 0 \end{cases}$

$e^{-x} \quad e^{-1} \quad e^{-2} \quad e^{-3}$

NN

Decimal Points

Activat func

# Neural Network

1. Already have features
   ↳ Classify / predict / Range

Ne

$f_1$ $f_2$ $f_3$ . . . . . $f_n$

Level / Text

Img1 $S_1$

Img2 $S_2$

$S_3$

⋮

2. Extract features

↳ Deep Net

feature Extraction → | ⟶ Dense Layer / Classific

→ Pre

Dog Cat Images

→ CNN

Convolutional Neural Network

Text Data

→ RNN

→ Recurrent Neural Network.

CNN

Kernel / Mask

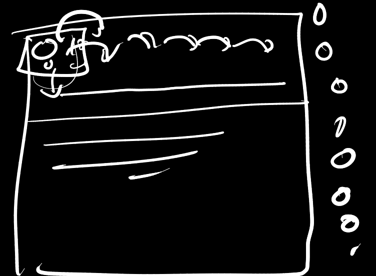
3x3 Kernel

filter

32

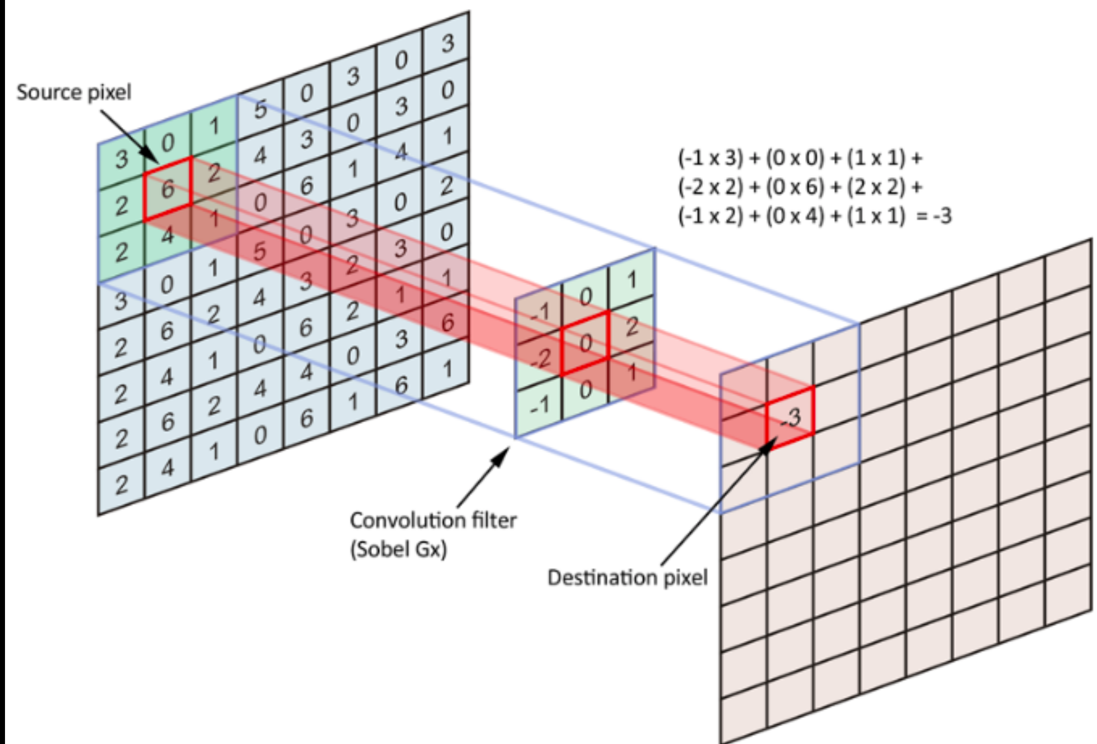
3x3 5x5

2x2 7x7

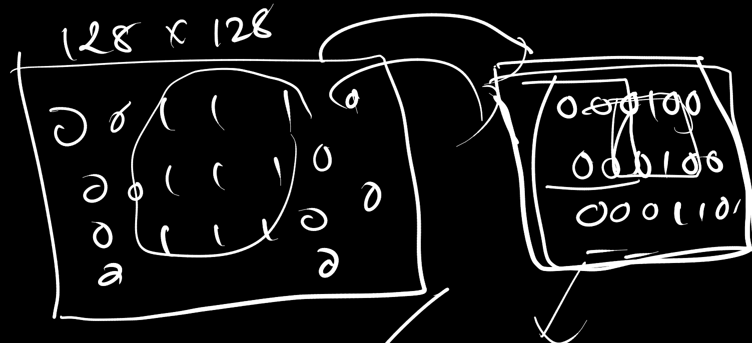


padding {}

Stride


Source pixel
(-1 x 3) + (0 x 0) + (1 x 1) +
(-2 x 2) + (0 x 6) + (2 x 2) +
(-1 x 2) + (0 x 4) + (1 x 1) = -3
Convolution filter
(Sobel Gx)
Destination pixel

Convolutional layer &
Pooling Layer




feature

Pooling Layer → Max
Min
Avg

| | | |
|---|---|---|
| 2 | 3 | 4 |
| 6 | 1 | 5 |
| 2 | 3 | 6 |

Averaging it out

6

1 x 1

1

C
↓
P
↓
C
↓
C
↓
P

Activation function

↳ Sigmoid

↳ TanH

↳ Soft Max

10K × 10K

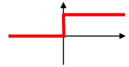
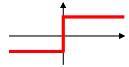
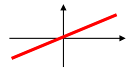
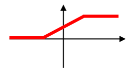
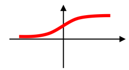
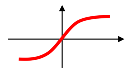
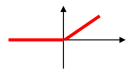
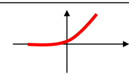
12 × 12

flatten → 1 dimensional

features (feature Map.

Relu

Leaky Relu

| Activation function | Equation | Example | 1D Graph |
|---|---|---|---|
| Unit step (Heaviside) | $\phi(z) = \begin{cases} 0, & z < 0, \\ 0.5, & z = 0, \\ 1, & z > 0, \end{cases}$ | Perceptron variant |  |
| Sign (Signum) | $\phi(z) = \begin{cases} -1, & z < 0, \\ 0, & z = 0, \\ 1, & z > 0, \end{cases}$ | Perceptron variant |  |
| Linear | $\phi(z) = z$ | Adaline, linear regression |  |
| Piece-wise linear | $\phi(z) = \begin{cases} 1, & z \geq \frac{1}{2}, \\ z + \frac{1}{2}, & -\frac{1}{2} < z < \frac{1}{2}, \\ 0, & z \leq -\frac{1}{2}, \end{cases}$ | Support vector machine |  |
| Logistic (sigmoid) | $\phi(z) = \frac{1}{1 + e^{-z}}$ | Logistic regression, Multi-layer NN |  |
| Hyperbolic tangent | $\phi(z) = \frac{e^{z} - e^{-z}}{e^{z} + e^{-z}}$ | Multi-layer Neural Networks |  |
| Rectifier, ReLU (Rectified Linear Unit) | $\phi(z) = max(0, z)$ | Multi-layer Neural Networks |  |
| Rectifier, softplus | $\phi(z) = \ln(1 + e^{z})$ | Multi-layer Neural Networks |  |

Sigmoid

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

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

Relu

$$f(x) \quad \max(0, x)$$

TanH

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

SoftMax

$$f(x_i) = \frac{e^{x_i}}{\sum_j e^{x_j}}$$

Average.

feature

G-M-D

flatten

6

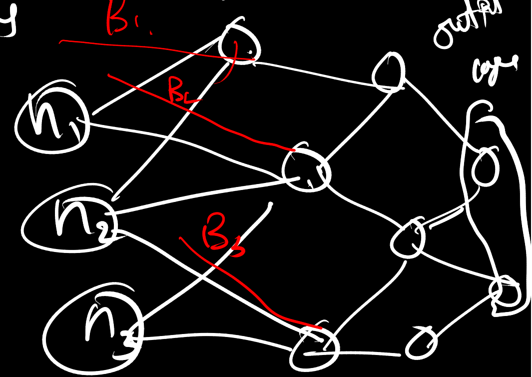
$f_1$ 0
$f_2$ 0
$f_3$ 0
$f_4$ 0
$f_5$ 0
$f_6$ 0

Prediction / Classification

Dense Network

Input Layer

Hidden layer



RNN — Recurrent Neural N/w

Text Data ↳ Movie Reviews — Positive
→ Negative

Number Data

"This movie is not interesting" —

↳ Bag of Words 2
↳ n grams
↳ Encoding Techniques

0 1 2 10 9 . . . . . .

Dictionary

5 7 8 9 2 1

flattend Encoding layer as a i/p to the

Dense Network

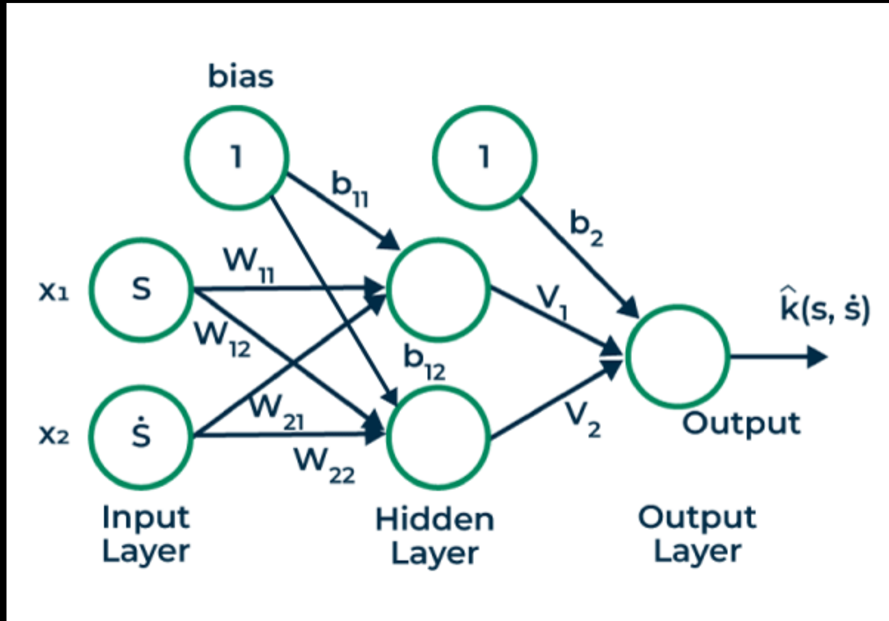
C
=
R

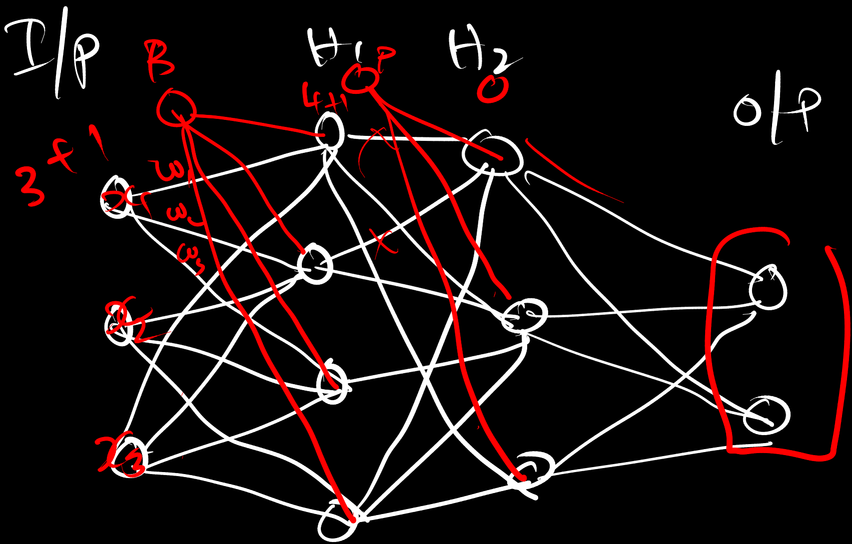
Represent

Feature Extraction

→ Dense Network

I/P Hidden Layer O/P


bias
1
1
b11
b2
x1
S
W11
W12
V1
k̂(s, ṡ)
b12
x2
Ṡ
W21
W22
V2
Output
Input Layer
Hidden Layer
Output Layer




Inputs × Weigh

$$x_1 \cdot w_1 + x_2 \cdot w_2 + x_3 \cdot w_3 + \dots + B \cdot w_i$$

Bias

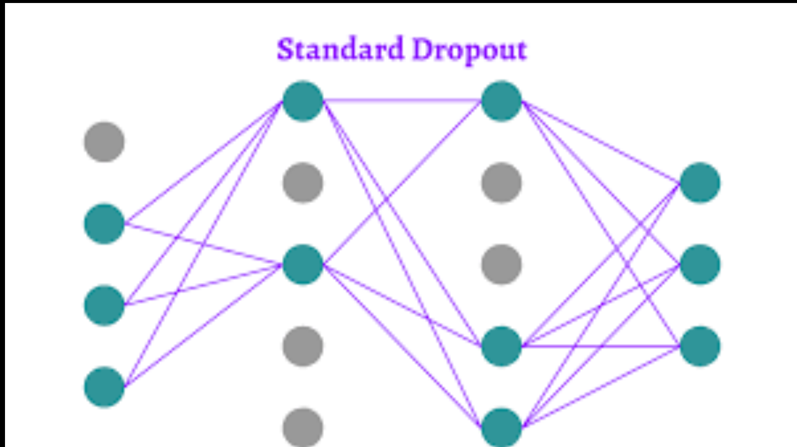
Classification

0 to 1

0 to 1

$$\sum x_i \cdot w_i + \text{Bias}.$$

→ Residue log y.


Standard Dropout

Dropout Layer

Regularization

1)
2) Bias | weights

Alter

Avoid overfitting Problems

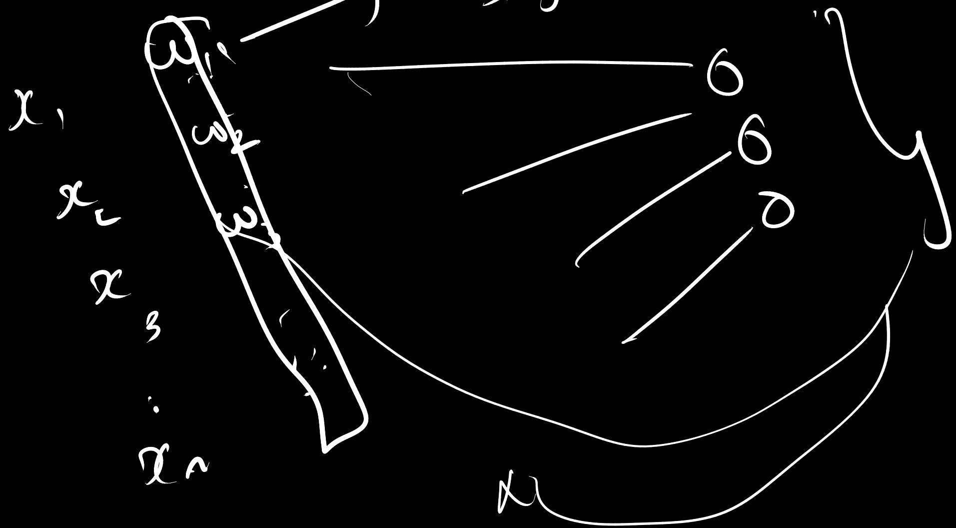
Learning Rate

Epochs

Epochs → # of iterations

C D

Dynamically Initialize




loss ↑

loss ↓

Reduced.

MSE

Model weights & Accuracy Layers

.h5

.h5

↳ Data
↳ Quantity
↳ Quality

Digit Classification

10 digits — 80k images.
1 8k images.

Data Augmentation ↳ Balanced Data

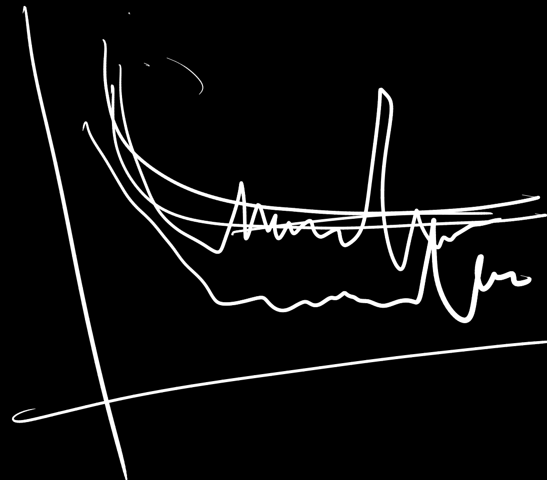
0 1 2 3 4 5 6 ... 9

6k-8k 5k 8k

↳ Rotation
↳ Zoom in/out
- Change the color
Splice

Filtering out the Data

↳ Manual Process



Classification → Dog / Cat

↳ Segmentation

Convoly Detection

→ Segmention

↓

Contour

Border

Classifi

Detection

Segmention

Image.

Text Analysis

Classification Movie Review

Segmentation → Key word Search

Generate the keys

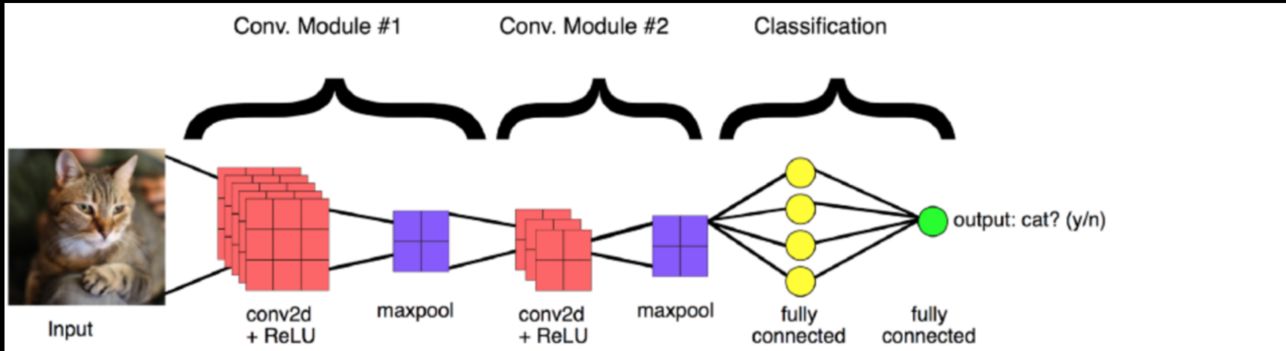
CBIR

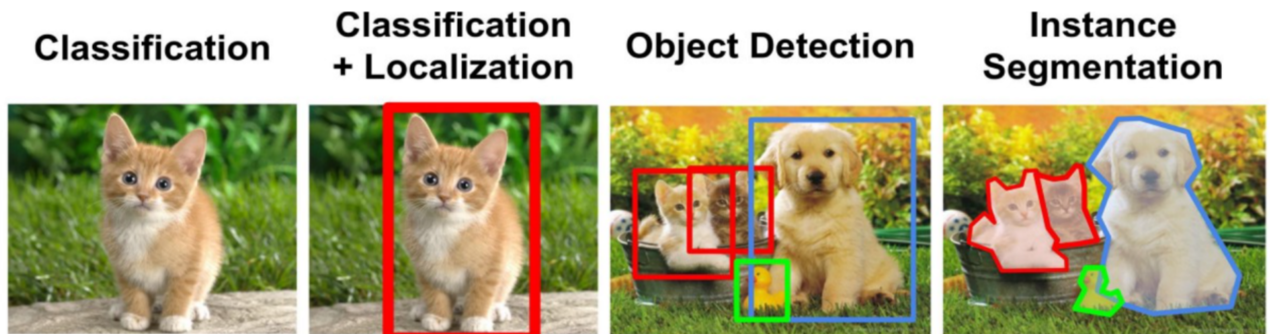
Content Based Search

Data Mining CB R

Data Engg C B

Data Science ← Data


Conv. Module #1
Conv. Module #2
Classification
output: cat? (y/n)
Input
conv2d
+ ReLU
maxpool
conv2d
+ ReLU
maxpool
fully
connected
fully
connected


Classification
Classification
+ Localization
Object Detection
Instance
Segmentation

Generative AI

↳

2016

Huge Datase

Huge Computation Power

LLama

Halli Labs → Conversation AI

Rule Based

+

Neural Network

Google Acquired

2018

Conversation AI

RAS framework

Sequencing

b/w words

b/w sentences

b/w paragraphs

b/w letters

Bag of words

Relationship

Sequence

Indent

Intend

RNN

Each Word
Numbers

Hi how are you?
12 13 45 2 10 15

Encoding Trimming

+ Sequencing

Flattening → Dense Layer

Multi Dataset

Airtel

Spam

Traffic Data

Number

Text

Image Data

→ Spam / Not

N/w Transformer

Predefined models

Created Network

↳ VGG 1

2

⋮

↳ Yolo 1

2

⋮

DeepNet 1

2

3

⋮

Training your own data

Class

Segmentation

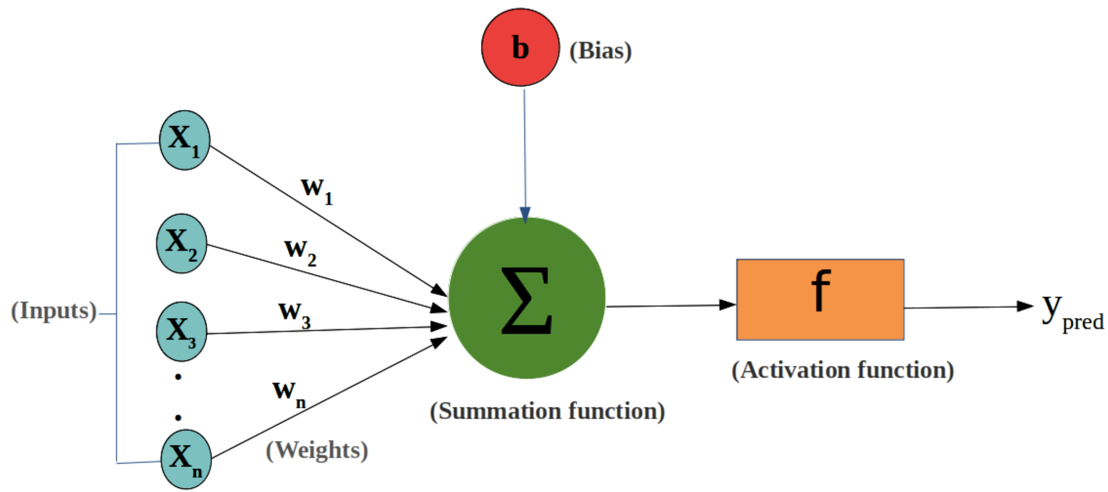
Deep Learning Models for Offline

Online

Inference Time

Tensorflow

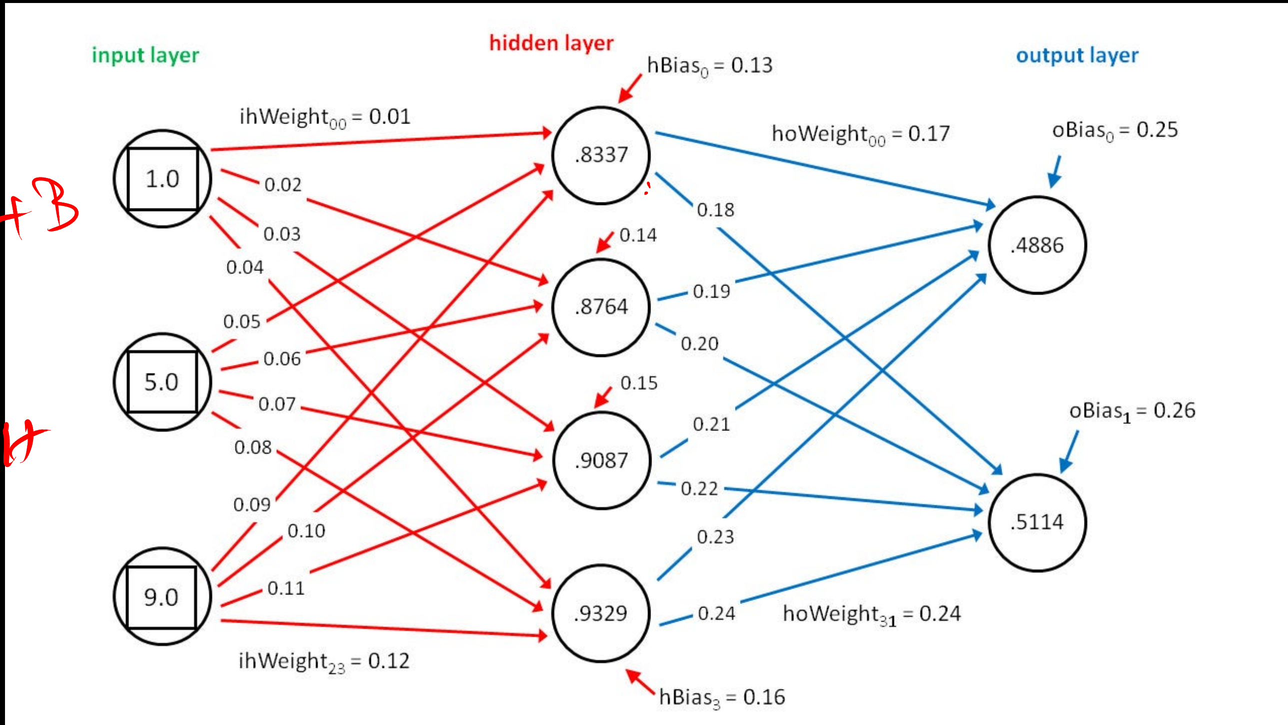
Tensor js - Blended in Browser


b
(Bias)
X1
X2
X3
Xn
w1
w2
w3
wn
(Inputs)
(Weights)
Σ
(Summation function)
f
(Activation function)
ypred

$Z = \sum x_i w_i + B$

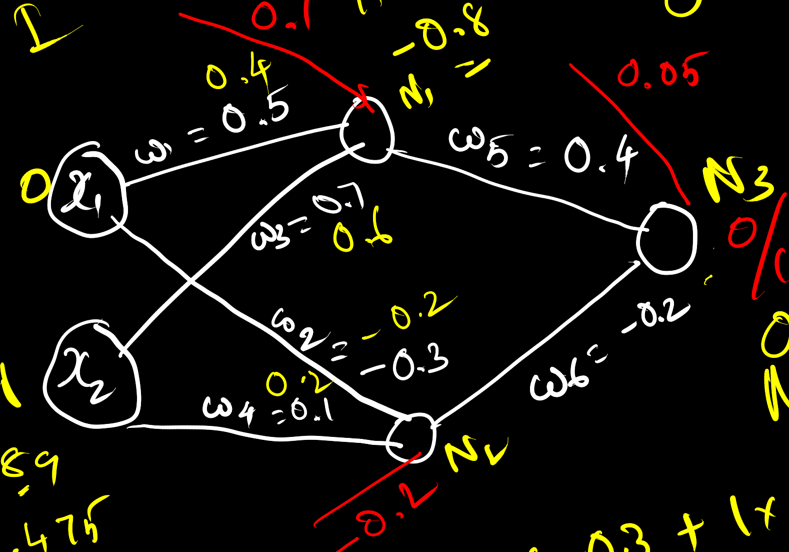
$\sigma(Z)$

TANH



I/P

$\sigma($




$N_3 = w_5 \times N_1 + w_6 \times N_2 + B_3$

$\sigma(N_3) =$

$N = \sum x_i w_i + B$

$N_1 = 0 \times 0.5 + 1 \times 0.7 + 0.1$

$\sigma(N_1) = \frac{1}{1+e^{-x}}$

$x = -0.8$

$O = 0.557$

$N_3 =$

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

$\sigma(-0.8) = \frac{1}{1+e^{0.8}}$

$N_1 = 0.689$

$N_2 = 0.475$

$N_3 = 0.557$

$E = O - N_3$

$= -0.557$

$N_2 = 0 \times -0.3 + 1 \times 0.1 + (-0.2)$

$x =$ $\sigma(x) =$

$E = O - 0.557$

$= 0 \times 0.5 + 1 \times 0.7 + \underline{0.1}$

$= 0.8$

$\sigma(0.8) = \dfrac{1}{1+e^{-0.8}} =$

$N_1 = 0.689$

$0.689 \times 0.4$
$+ 0.475 \times -0.2$
$+ 0.05$
$= 0.2306$
$\sigma$

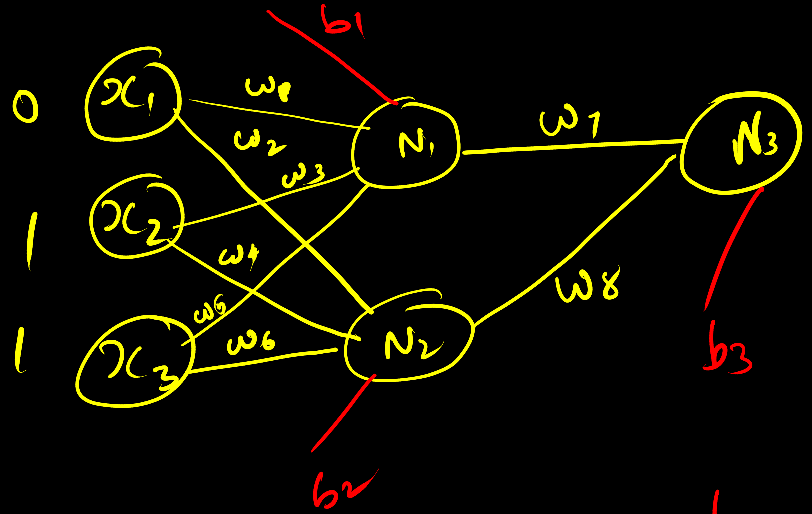
$= 0 \times -0.3 + 1 \times 0.1 - 0.2 = -0.1$

$E = 0 - 0.2306$
$= -0.2306$

$\sigma(0.1) = \dfrac{1}{1+e^{+0.1}} = N_2 = 0.475$

Op

$E = 0 - 0/p$




$x_1 = 0$

$x_2 = 1$

$x_3 = 1$

$w_1 = 0.1$

$w_2 = 0.5$

$w_3 = 0.7$

$w_4 = 0.5$

$w_5 = 0.2$

$w_6 = 0.3$

$w_7 = 0.6$

$w_8 = 0.2$

$b_1 = 0.1$

$b_2 = 0.6$

$b_3 = 0.4$

Epochs = 5

After Wi & Bi

9894104930

amarnathresearch@gmail.com

codeidea.in

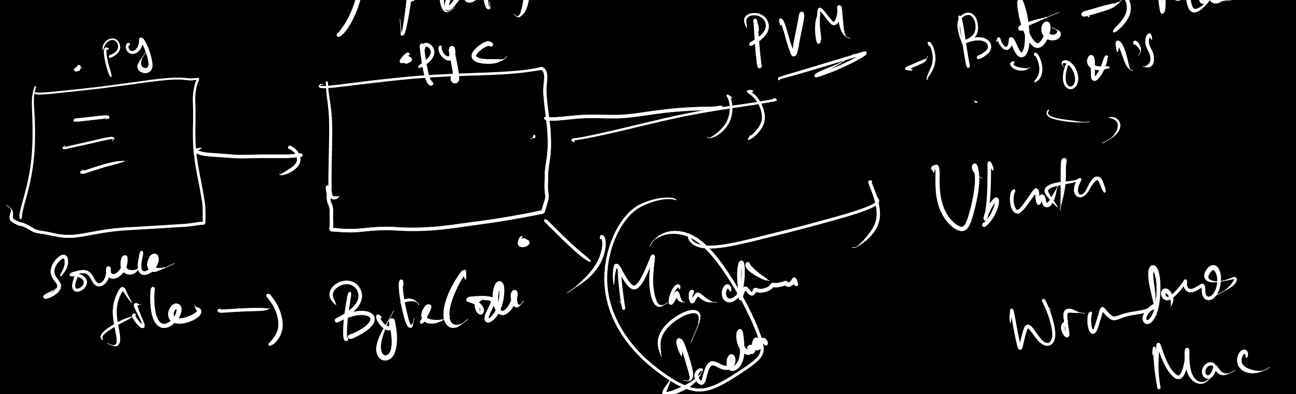
Talent Development

CTO

# Python Language

↳ Python Virtual Machine.

↳ Platform Independent



# DVM

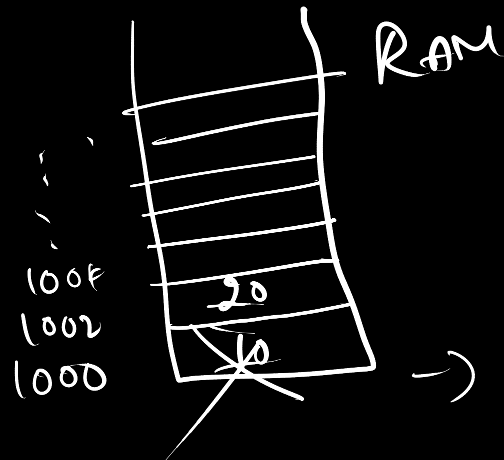
- ↳ Converting
- ↳ Execution
- ↳ Garbage Collection.

a = 10

del a

a = 10

b = 20




→ Stack Memory

Heap Memory

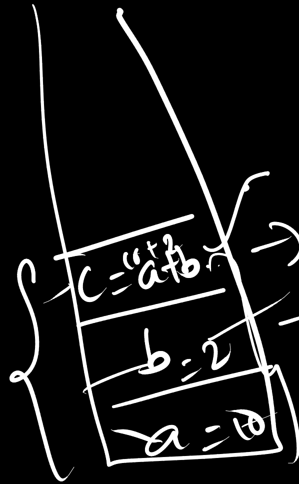
→ Clean it

↳ Memory Management

a = 10
b = 20
c = a+b

TTL

(Time to Live)




→ Execution.
→ 30 se
60 se
1 hr

Python Garbage collector

a = 10 // int

b = "sigmoid"

c = 'sigmoid'

d = 0.01

e = true

f = false

g = None

Standardization

Naming the Variable

arg_a =

argb =

sum = arg a + argb

Camel case

water_bottle

waterBottle.

WaterBottle

# Installation of Python

↳ Windows → Download python.org

↳ Ubuntu → sudo apt install python

2.x

3.6

3.8 3.11

Shell → Interactive

file → Execute

↳ Check python is installed

↳ Which Version.

python --version

Python -V

Indent

Space

2 Spaces

4 spacs

functions

main ()

↳ add ()

↳ mull ()

↳ Div ()

// Modularity

↳ Code Understanding

↳ Cohesion | high

↳ Coupling | Low.

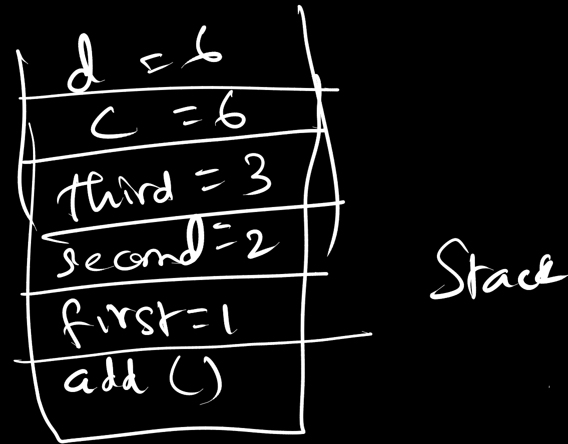
Utility

main()
↳ encrypt(a) ⟶ encrypt(a)
& Hashfun (Ukl
level.

Python:

```
def add(first, second, third):
    sum = first + second + third
    return sum.
c = add(1, 2, 3)
print(c)
```

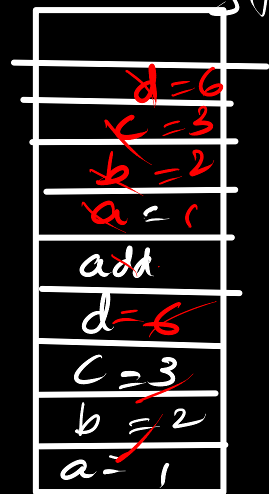
```
def add(first, second, third):
  c = first+second+third
  return c

d = add(1, 2, 3)
print(d)
```



```
def add(a, b, c):
  d = a+b+c
  return d

a = 1 // 2
b = 2 // 2
c = 3 // 2
d = add(a, b, c) // 8
print(d)
```

16 bytes

Stack mem




() Popped out

Stack DataStru

Conditions

if

if else

if a > b and b > c:

else:

if a > b

a < b

a >= b

a <= b

a == b

and &&

or ||

not !

Combinations

and or not

<u>**Loops**</u>

for

while

for i in range(1,10)

for i in range(1,2,10)

$$\sqrt{x} = y$$

$$x = y * y$$

$y = 16$

Given $y = 4$

Guess $x$

$y = 4$

Compute

$x = ?$

$x = ?\ 16$

Given $x = ?$

$y = ?$

$\sqrt{x} =$

$x = 16$

$l = 1 \quad h = 16$

$m = (1 + 16)/2$

$m * m == 16$

$l = 1 \quad h = m$

$\sqrt{x} =$

$x = 16$

$l = 1 \quad h = 16$

$m = (1 + 16)/2$

$m * m == 16$

$l = 1 \quad h = m$

Singleton Class

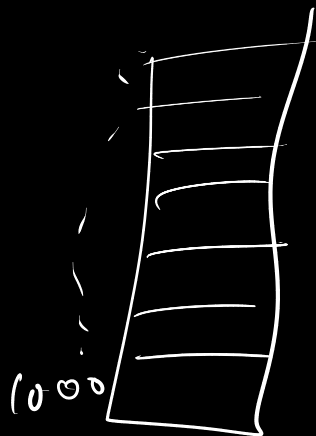
// One Time Instance

↳ X Global variables

List

a = [10, 20, 30 ... ~~50~~, 1000]

a[2]



```
a = [10, 20, 30]
print(a[2])
```

```
a = [[[10, 20], [30]], [4], []]
```

```
def example():
    a = [10, 20]
    return a
print(example())
```

```
def example():
    return [10, 20]
```

```
a = set()

a= [10, 20, 30]
print(a)
```

```
a=[10, 20]
a[0] = 30
print(a)
```

```
a=set()
a={20,30}
print(type(a))
```

```
a = set()
a.add(10)
print(a)
```

```
a=set()
a.add(10)
a.add(20)
a[0]=30
print(a)
```

a = set(10, 20)

---

a = (20, 30)
b = (40, 50)
print(a+b)

```
a = {}

a = {10:20, 30:40}

a[10] = ? = 20

for k,v in a.items():
    if v==20:
        return k
```

a = {10 : "hello" , 20 : "welcome", 10 : "Blr"}

---

a = {10 : [20,30,40] , 20 : "welcome"}

---

a = {20 : {30 : 40}, 20 ....

# Strings

```
a = "hello__world"
print(a.split())
print(a.split("_"))
```

a = input()

a = input(" Enter a value")

string

a = int(input(" Enter a value"))

Oops

↳ Classes

& its attributes

User → Class / Entity / Object

↳ id

↳ name

↳ DOB

↳

UML

ER

Class Diagram

Sequence Diag

Flow Diag

Str

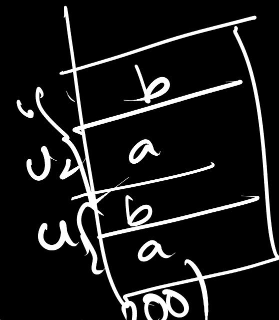
```python
class User:
    a = 10
    b = 20
    name = 'abc'

u = User()
print(u.a)
print(u.b)
print(u.name)
```

```python
class User
    a = b = name = 10
    def info():
        return a, b, name

u = User()
print(u.a)
print(u.b)
print (u.name)
```

```
class User:
    a=10

u= User
u.b =20
u.c=30
print(u.a)
print (u.b)
print(u.c)
```

```
class User:
    a = 20 , b = 30
    def __init__(self, a, b):
        self.a = a
        self.b = b

u1 = User(40, 50)

print(u1.a)
print(u1.b).
```

```
u2 = User(100, 200)
=
print(u2.a)
print(u2.b
```



# File Systems

# Recursion:

A func which calls ~~itself~~ replication of the function.

Fibonacci Series

Goals

1 1 2 3 5 8 . . . . .

1 2 3 4 5 6 . . .

$n = 5 \neq 5$

$n = 6 ?$ 8

Base Condition

Edge Condition

Goal Case

Stack Overflow

```
def fib(n):
    if n==1
        return 1
    elif n==2
        return 1
    return fib(n-1)+fib (n-2)
```

fib(3)

fib(3) = 2

3

```
def fib(n)
  if n == 1 or n == 2:
    return 1
  return fib(n-1) + fib(n-2)
```

2 1

1 1

2

```
def fib(n)
  if n == 1 or n == 2:
    return 1
  return fib(n-1) + fib(n-2)
```

1

```
def fib(n)
  if n == 1 or n == 2:
    return 1
  return fib(n-1) + fib(n-2)
```

fib(4) = 3

```
def fib(n)
  if n == 1 or n == 2:
    return 1
  return fib(n-1) + fib(n-2)
```

```
def fib(n)
  if n == 1 or n == 2:
    return 1
  return fib(n-1) + fib(n-2)
```

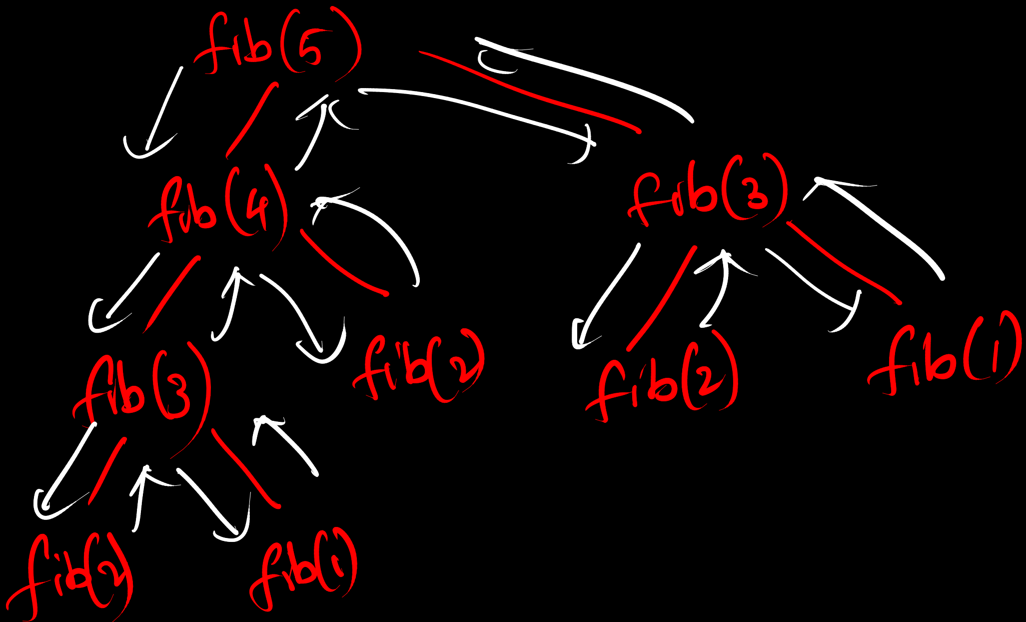
```
def fib(n)
  if n == 1 or n == 2:
    return 1
  return fib(n-1) + fib(n-2)
```

1 + 1

level = 2

```
def fib(n)
  if n == 1 or n == 2:
    return 1
  return fib(n-1) + fib(n-2)
```

```
def fib(n)
  if n == 1 or n == 2:
    return 1
  return fib(n-1) + fib(n-2)
```




DfS

= Depth first Search

fib(10)

user.py

User
id
name
ins_id

institution.py

Institution
id
name

1 aman 1
2 Ram 1

$u_1$ = User(id, na, i)
$u_2$
$u_3$ ...

$i_1$ = Instul(id, nam)
$i_2$ = Ins (id nam)

Install Mysql

Set Environment Variable
To access via Terminal

mysql -u root -p

Big O → Time Complexity & Space Complexity

Linear Search:

a = [10, 20, 30, 40, 50]

Key = ?

Best Case ? = O(1)

Average Case ? Ω(n/2)

Worst Case ? O(n)

Binary Search

Worst Case : ? $O(\log_2 n)$

Best Case = ? $O(1)$

= [50, 20, 10, 100, 200, 5]

$= O(n^2) + O(\log n)$

$n(n+1)$

O(log n) : Binary Search (sorted)

Monotonically Increasing

" " Decreasing

Selection Sort

↳ $O(n^2)$

Bubble Sort

↳ $O(n^2)$

Merge Sort

O

Division $\log n$

↓

Sort(conquer) $n$

↓

Merge $\log n + n$

$\log n + n$ $\log n + n$

$\log n * n * (\log n + n)$

$O(\log n * n * n)$

$O \log n *$ $n \log n$

Quick Sort

$O(n^2)$

$n \log n$

$O(n)$

Heap Sort

$\log_2 n = K$

$n = ?$

$n = 2^K$

Insertion Sort

$O(n^2)$

Radix Sort

$O($

Selection Sort

$n = 5$

for $i = 1$; $i < \cancel{n}\ n-1$; $i{+}{+}$)

for $j = i+1$; $j < n$; $j{+}{+}$)

swap

$i = 1$ ///

$i < n-1$ // $n$

$i{+}{+}$ $n-1$

$j = i+1$ //

Covid Dataset

Ex 1: Total Provinces

Total deaths,

Suspected

Confirmed

Recovered

Ex 2: Second CSV

find the same as Ex 1

# added provinces & its names

# What is the increase / decrease in the cases:

| Confirmed | Death | Recovered | Suspected |
|---|---|---|---|
| 200+ | 17 | 3 | -32 |

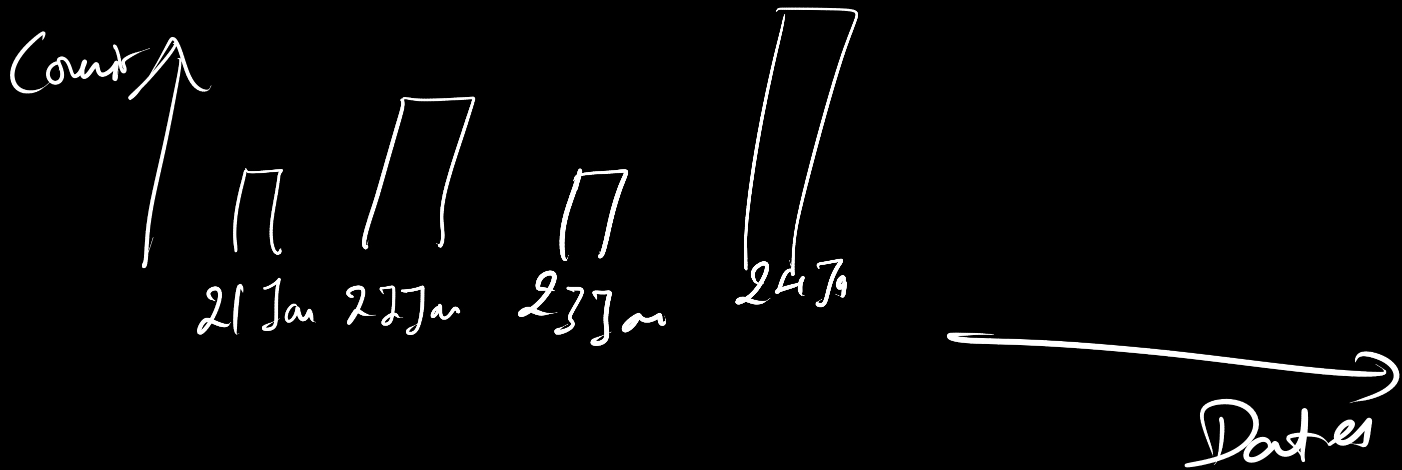
Day 1

| | Country | Confirmed | Deaths | Recovered | Suspecte |
|---|---|---|---|---|---|
| 21 Jan 2020 | China | – | – | – | – |
| 21 Jan 2020 | Thai | – | – | – | – |
| | ⋮ | | | | |
| 22 Jan 2020 | Chin | +10 | –20 | – | – |
| | ⋮ | | | | |

Matplotlib / Seaborn

Deaths

China



China.csv —

Aus.csv

US.csv.

Dates

Death Sum - - - .

# Data Analysis

↳ Visualization

1. Preparation of the Data

↳ Data Collection

— Sources

↳ Data Cleaning

Duplications =

NULL

↳ Data Preparation:

↳ Raw Data → Simplified Data

4) China.csv

thailand .csv

malasya. .csv

deaths, suspe, confo

Data1

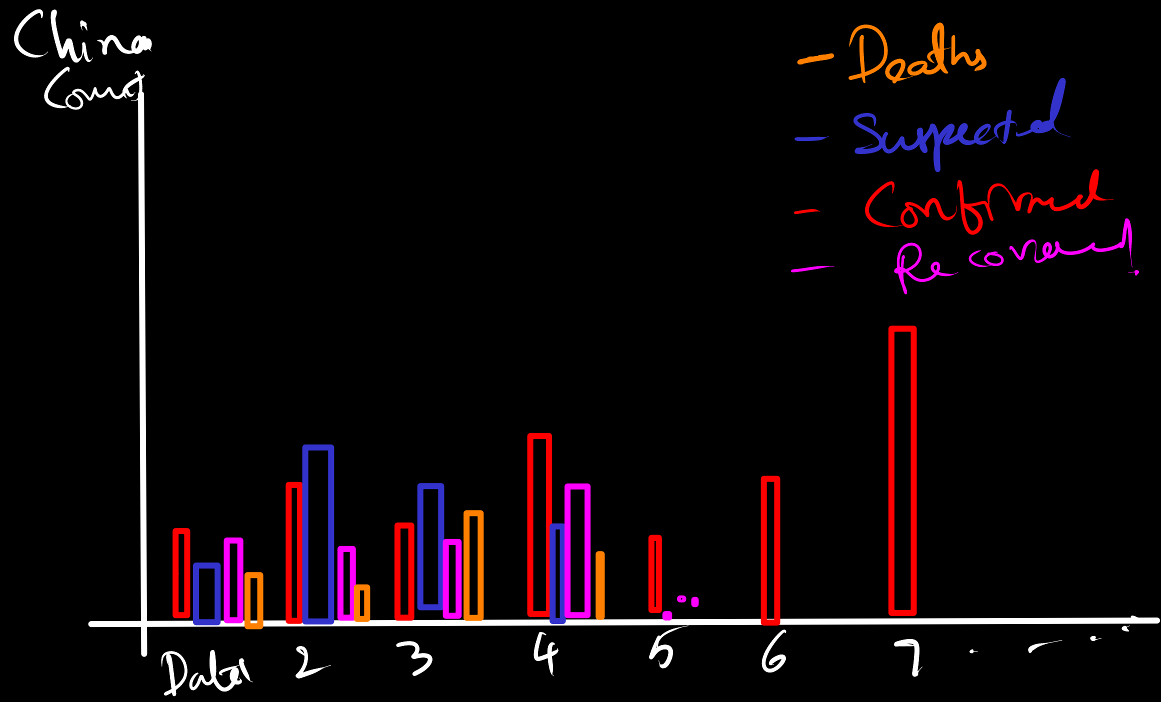
Data2

Data3

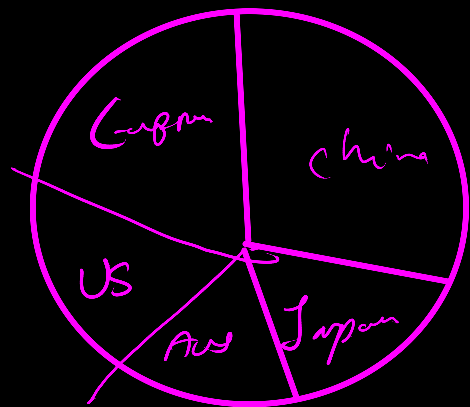
.

.

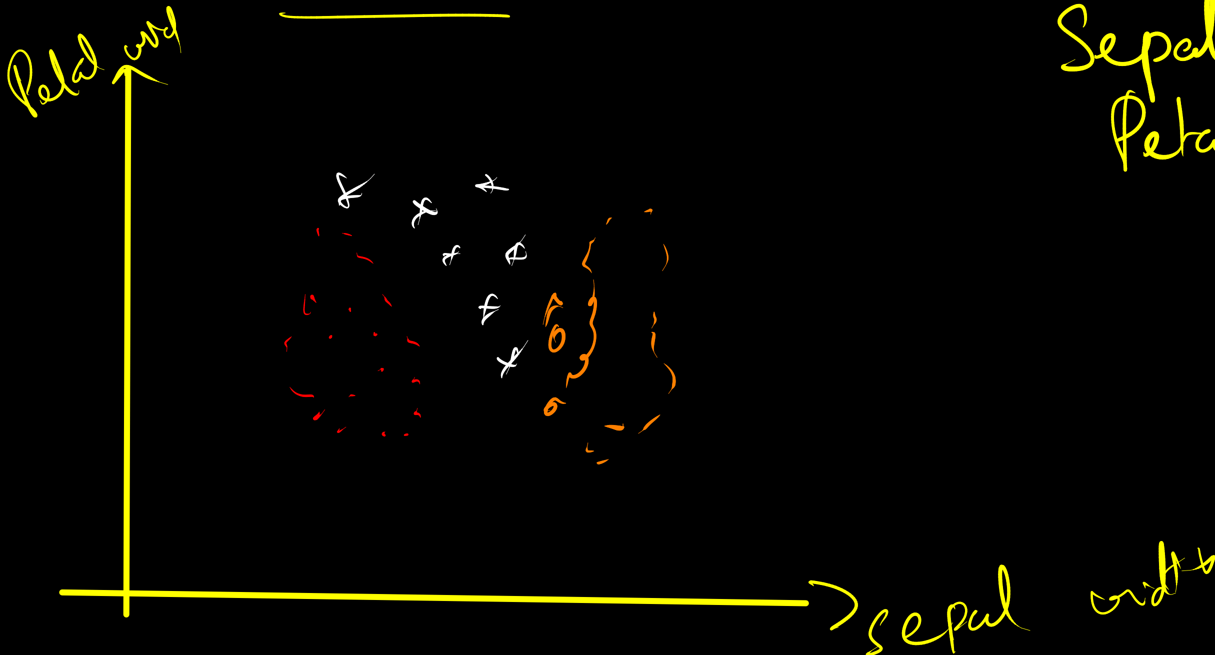
..


China
Count
Deaths
Suspected
Confirmed
Recovered
Data1
2
3
4
5
6
7

Deaths



# IRIS Dataset




Sepal width
Petal width

Geographical plot

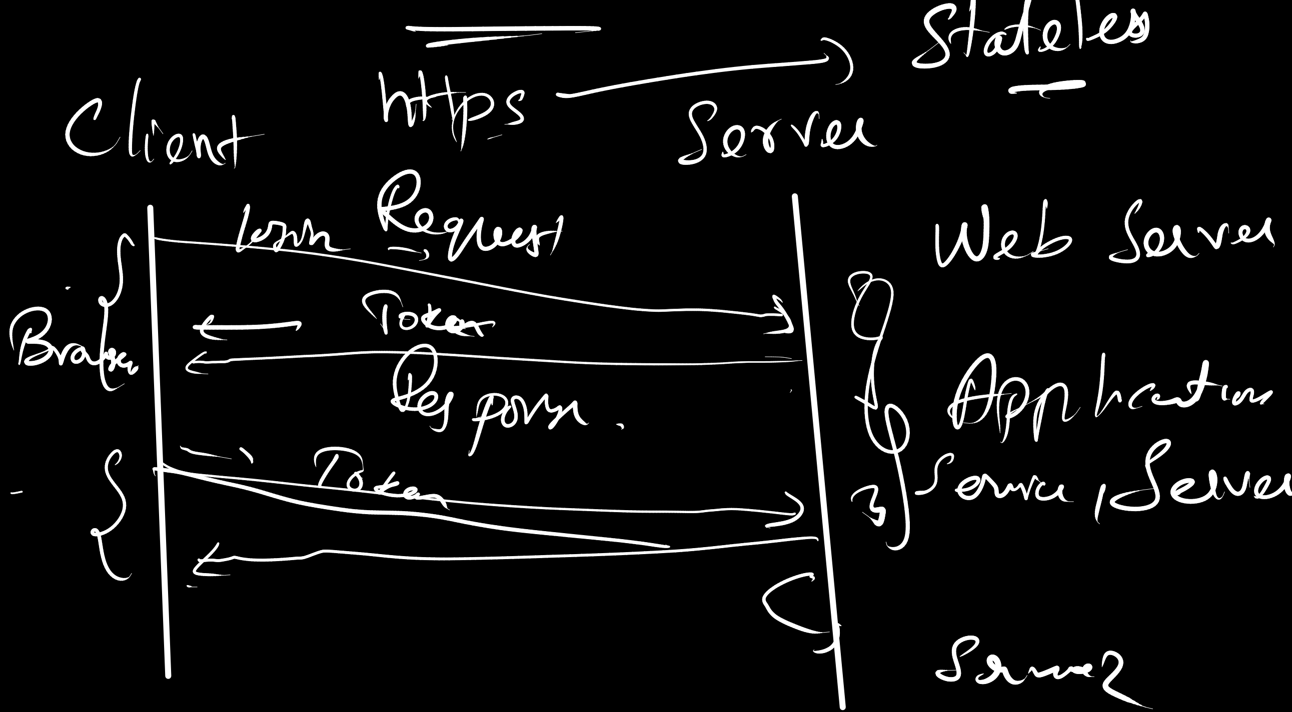
Mark different

Asia Korea

China Japan

ge

# API calls



Stateless
https
Client
Server
login Request
Token
Browser
Response
Token
Web Server
Application Server / Server
Server2

Web Server

↳ Static files

Content

Apache

[Container]

App | Server

↳ Dynamic Content

Tomcat

Weblogic

IBM Sphere

Web Serva → Apache ← Many request

↓ less power

Computation

Consump

↳ Computa

Dynam

Web ← Load Balancing

← Distribution of Request

Application ← Dynamic

← Micro Services

Monolothic Appl.

API

request get data

get data

Server

Application

Python

Get Request

POST Request

Option

Upload Request

Response Code

Status 200

Request

Browser

GUI

Mobile

Mobile App

Server

404

500

502

Python

# Project

Machine Learning

Prob & Stats

↳ Singular → one dimensional Data

4 0 & 1 $P(H) = 1/2 = P($

$P(T) = 1/2 = 1 - P(H)$

↳ Number Data

## Conditional Probability

$P(A) = 0.8$

$P(B) = 0.4$

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

What is the pro

rain & carry umbrella.

Stats

Inferflation

$$\text{Mean} = \frac{\sum x_i}{N}$$

$$SD = \sqrt{(\bar{x})^2 - (x^2)}$$

$$\sqrt{\frac{\sum(x_i - \bar{x})^2}{N}}$$

10
20
30
5
2
1

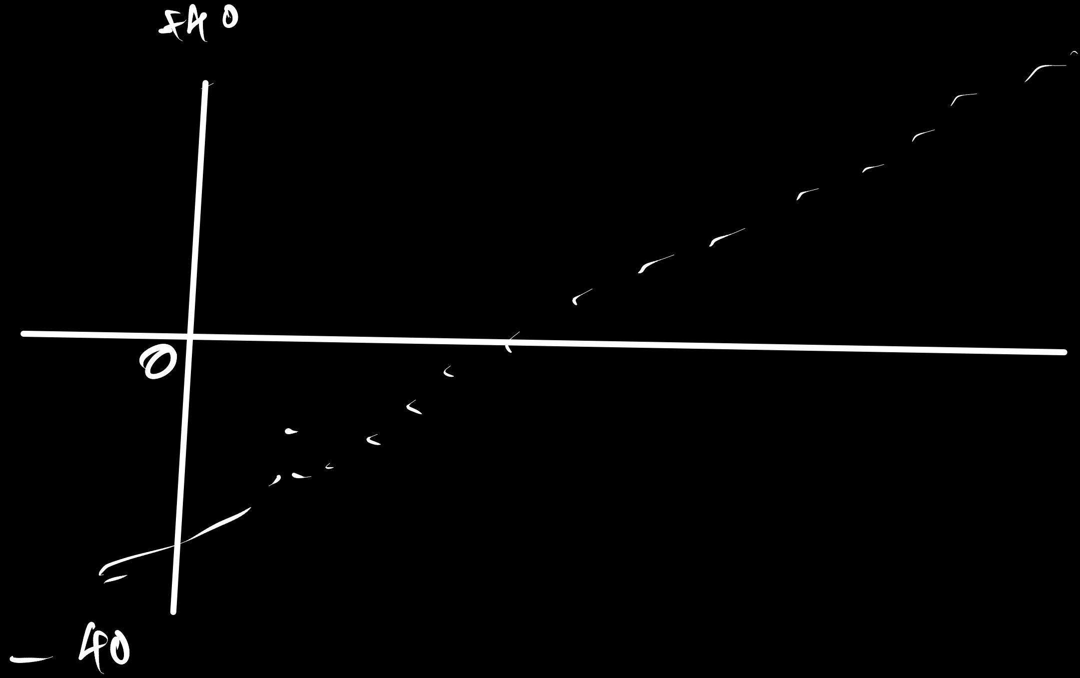
## Correlation

$X \qquad Y$

$$\operatorname{corr}(x,y) = \sqrt{\frac{(x_i y_i)^2 - x_i^2 \cdot y^2}{N+1}}$$

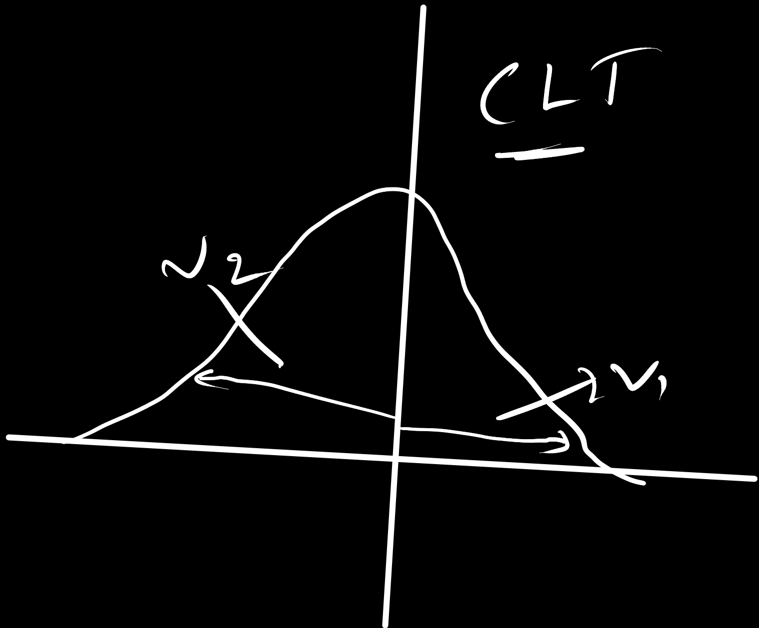
$N > 0$

# Covariance



CLT

Normal Distribution



Limitations of Stats & Prob

↳ ML

↳ Algebra

$x = 0$ // $x = y = 0$

$x^2 = y$

$x = \log_2 y$

or

$y = 2^x$

$x = \log_3 y$

$y = 3^x$

$x = \log_{10} y$

$y = 10^x$

$x = \log y$

and

$x = e^y$

$y = mx + b$

Given $x$ & $y$

$m ?= \frac{y_2 - y_1}{x_2 - x_1}$ / $x$ of

$m = \frac{x_2 - x_1}{y_2 - y_1}$, ~~x~~ as ______

$$3x^2+2x+10=0$$

solve $x$

$$32x^3+16x^2+8x=24$$

solve $x$

# Machine Learning

- Data
  - Text
  - Numerical
  - Image
  - Video
  - Audio
  - Hybrid data

↳ 1 $x$ [Single value]

↳ [1, 10, 15 ....] - Multivalued data.

↳ Interval Type of Data {(10 - 20)}

↳ Range.

↳ Multivalued Interval type of data

{(10 - 20) (15 - 100)...}

Categorical Data

{Apple, Orange, Emn ....}

Sourcing

↳ Camera / Video / Audio Device

↳ Text

↳ Type

↳ Radar.

↳ Websites

↳ pdf

html

⋮

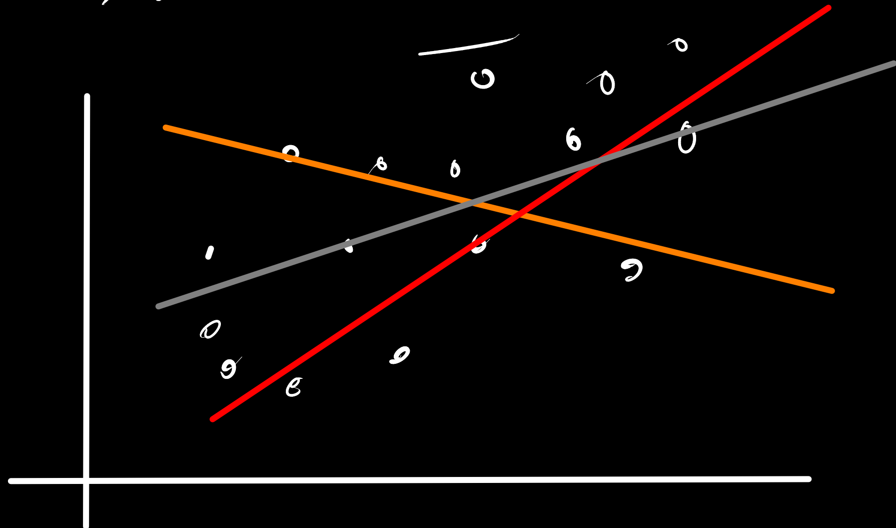
ML : What type of problem :

How will solve the problem

Solution / Algorithms.

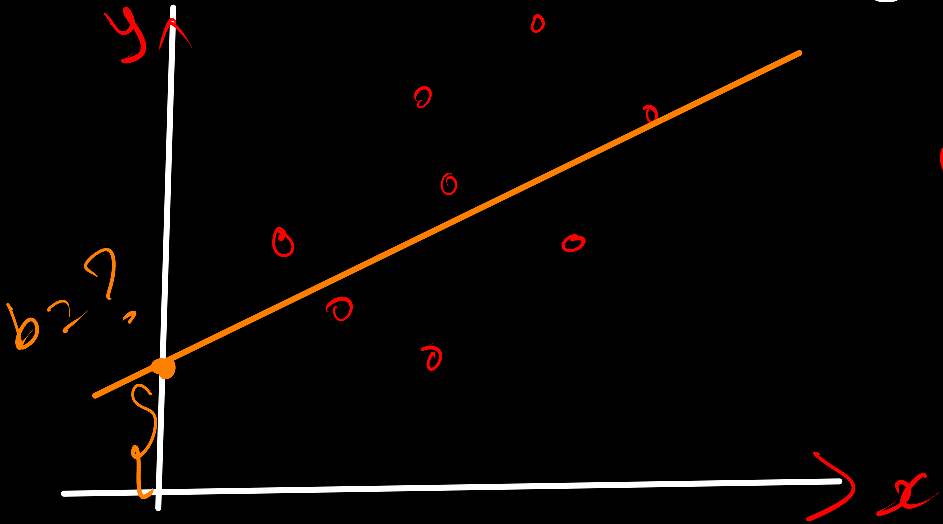
↳ Regression.

↳ Linear Regression



# Regression [Linear]

→ Straight line




$$y = mx + b$$

intercept
shift
offset

if $x = 10$

$y = ?$

$m = 1.31$

$b = 2.415$

$y = mx + b$

$y = 1.31x + 2.415$

| $x$ | $y$ |
|---|---|
| 1 | 2 |
| 2 | 6 |
| 3 | 8 |
| 4 | 7 |
| 5 | 10 |
| 6 | 9 |

$y = mx + b$

$m = ?$

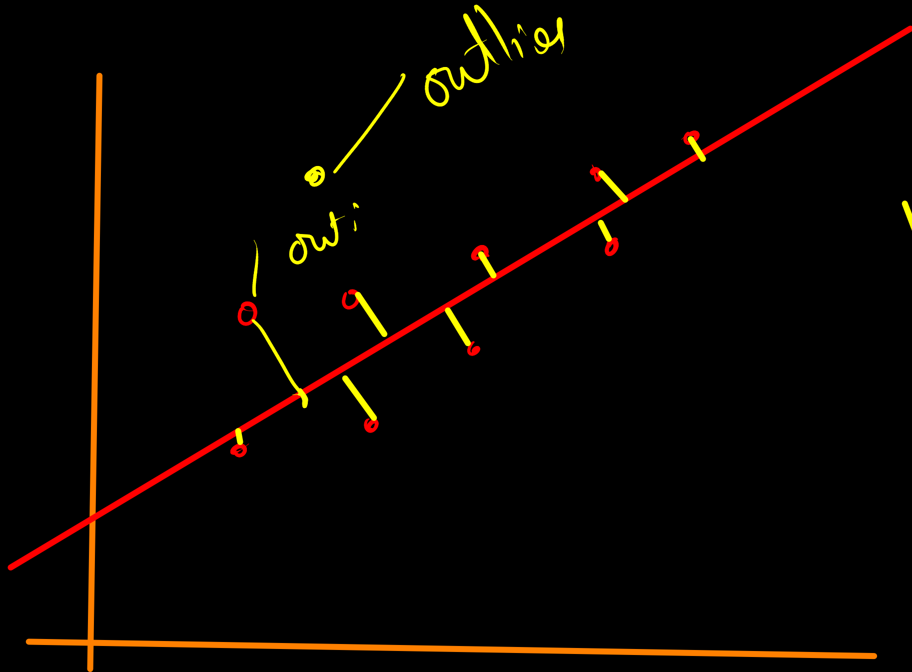
$$m = \frac{n\sum x_i y_i - \sum x_i \sum y_i}{n\sum x_i^2 - (\sum x_i)^2}$$

$y = 1.31(10) + 2.415$

if $y = 15$

$x = ?$

$$b = \frac{\sum y_i - m \sum x_i}{n}$$




$$Loss = \frac{1}{N}\sum (y_i - \hat{y})^2$$

$L$

$$\hat{y} = mx + b$$

| | $x$ | $y$ | $\hat{y}$ |
|---|---|---|---|
| 0 | 6.89 | 3.26 | – |
| 1 | 5.12 | 1.98 | – |
| 2 | 7.82 | 3.25 | – |
| 3 | 7.42 | 3.67 | – |
| 4 | 6.94 | 3.57 | – |
| 5 | 7.89 | 2.99 | – |
| 6 | 6.73 | 2.60 | |
| 7 | 6.75 | 2.48 | |
| 8 | 6.09 | 2.31 | |

$$\hat{y} = \underline{m}x + \underline{b}$$

$$\text{Loss} = \frac{1}{N} \sum (y_i - \hat{y}_i)^2$$

$\underline{\underline{\text{MSE}}}$ = Mean Square Error.

# Logistic Regression

↳ Sigmoid function

$$\sigma(z) = \frac{1}{1+e^{-z}}$$

$$e = \sim 2.718$$

$$z = b_0 + b_1 x_1 + b_2 x_2 + \dots + b_n x_n$$

$$\sigma(z) = \frac{1}{1+e^{-z}}$$

$$\sigma(-10) = ?$$

$$\sigma(5) = ?$$

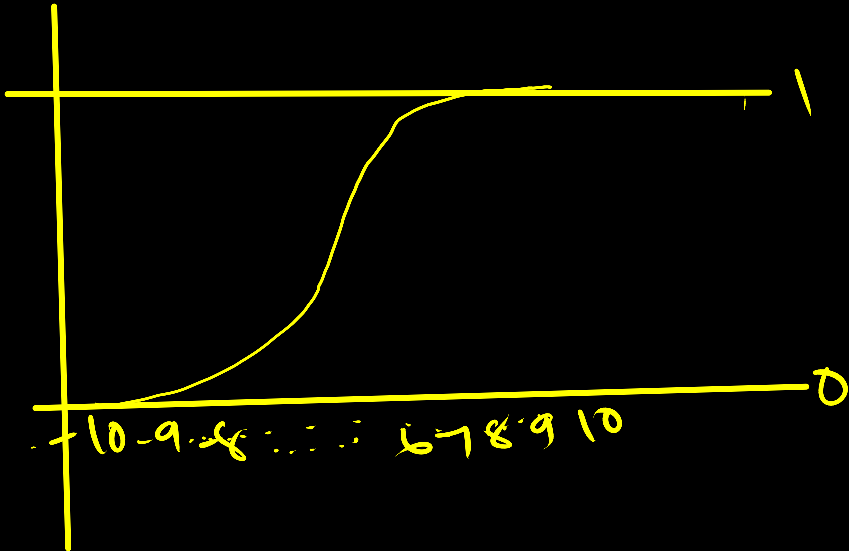
$$\sigma(1) = ?$$

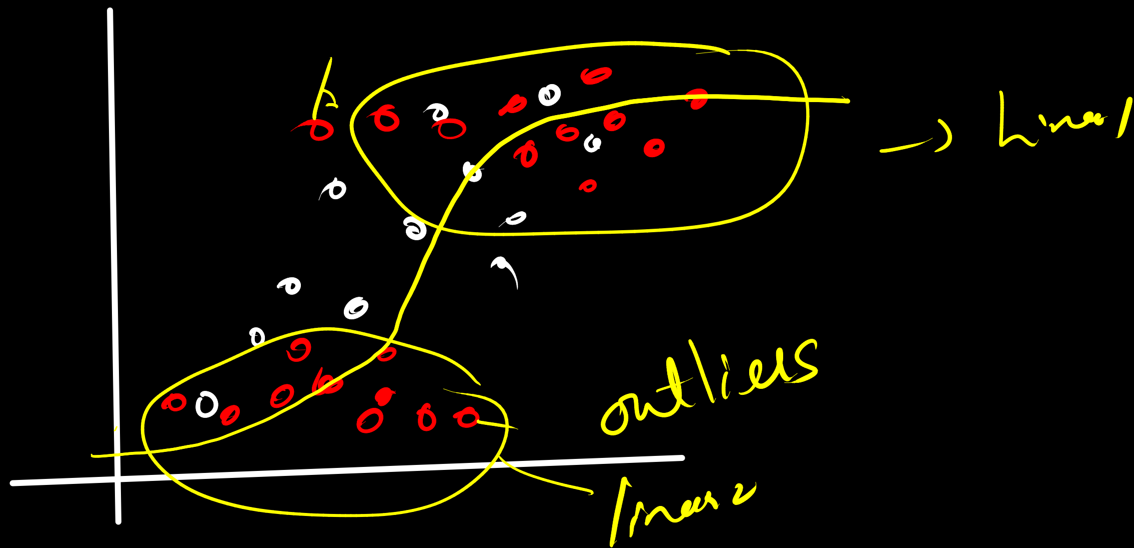
$$\sigma(10) = ?$$

$$\sigma(-1000) = ?$$

$$\sigma(+1000) =$$

Why Logistics





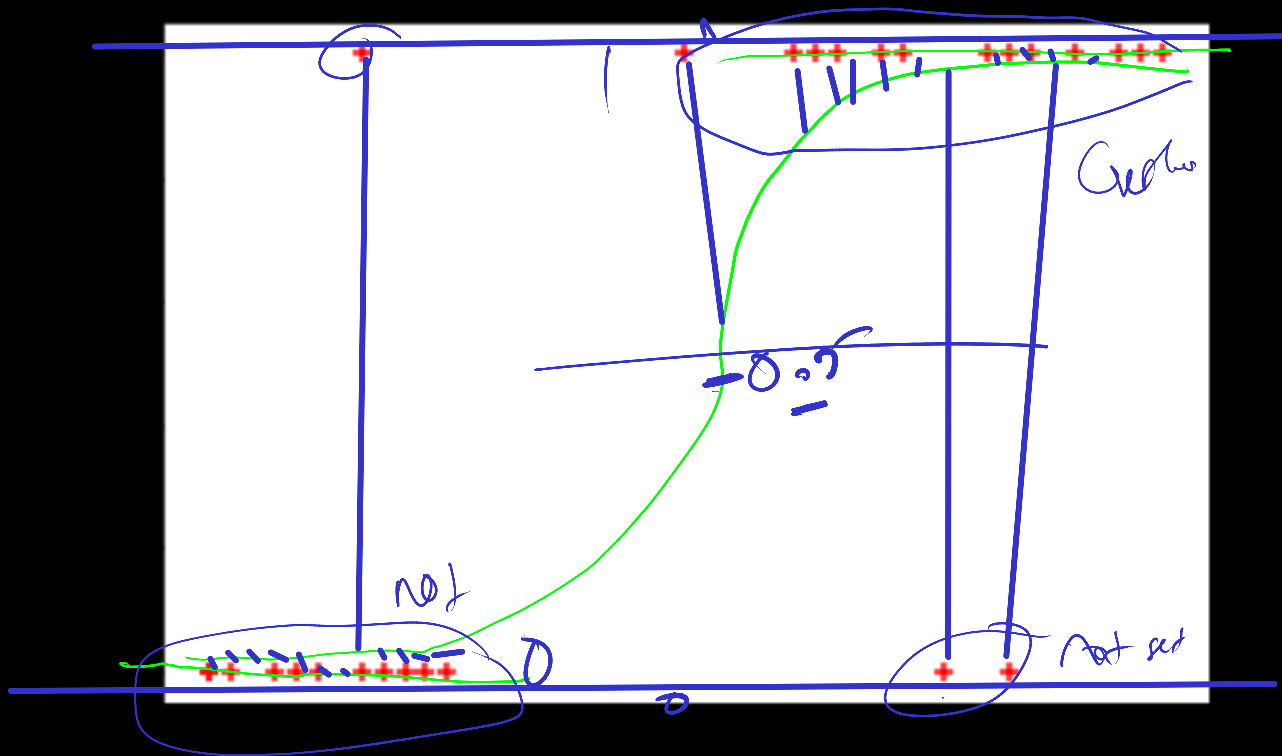
linear
outliers
linear 2

$$\sigma(x) = \frac{1}{1+e^{x}} \longrightarrow \text{Linear Regression}$$

Logistics $\rightarrow$ Binary Classification

0 or 1

Two Class problem


not
not set

# Decision Tree

Samples

| f1 | f2 | Target |
|---|---|---|
| Green | 3 | apple |
| Yellow | 3 | apple |
| Red | 1 | grape |
| Red | 1 | grape |
| Yellow | 3 | Lemon |

dimensions

$f_1$ = Color

$f_2$ = Diameter

Target = Label

= Classes

# Impurities

Gini $= 1 - \sum_{i=1}^{n} p_i^2$

Entropy

$G = 16/25$

T $P(A) = 2/5$

$P(G) = 2/5$

$P(L) = 1/5$

$1 - \sum$

$1 - \left[\left(\frac{2}{5}\right)^2 + \left(\frac{2}{5}\right)^2 + \left(\frac{1}{5}\right)^2\right]$

$= 16/25$

if color = Red

$$1-\left[\left(\tfrac{2}{3}\right)^2+\left(\tfrac{1}{3}\right)^2\right]$$

Apple
Apple
Lemon

0.44

Grape
Grape

0

=

if f2 = 1

0.44 Apple
Apple
Lemon

Grape
Grape
G = 0

Stump

$f_1 = \text{Green}$

0.6

$G_1 = 0$

Step 3

Green   Yellow   Red   3   1

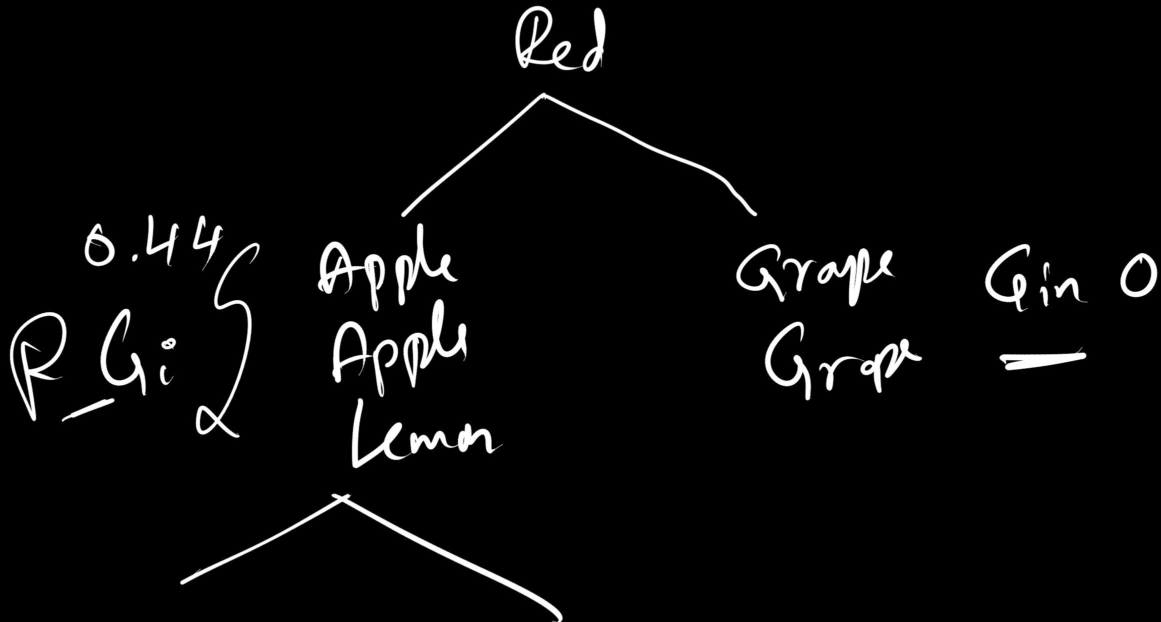
$S_1$   $S_2$   $S_3$   $S_4$   $S_5$

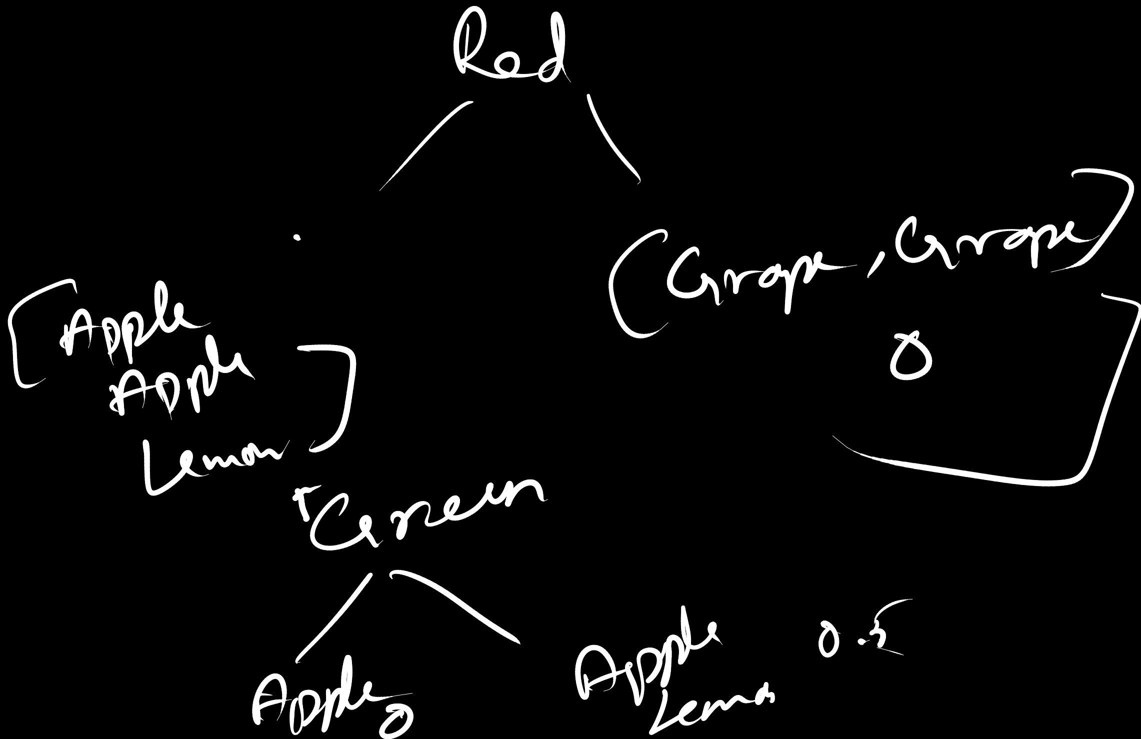
$G_1$   $G_2$   $G_3$   $G_4$   $G_5$   $G_6$   $G_7$   $G_8$   $G_9$   $G_{10}$

Weight Gini (WG)

$$= \left( \frac{\overset{n_1}{\# l}}{N=5} \times G_1 \right) + \left( \frac{\overset{n_2}{\# r}}{N=5} \times G_2 \right)$$

Information Gain (IG) = Parent Gin − WG


Red
0.44
R_Gi
Apple
Apple
Lemon
Grape
Grape
Gin 0


Red
Apple
Apple
Lemon
Grape, Grape
0
Green
Apple
0
Apple
Lemon
0.5

# RANDOM forest

Problem DT SD1

$f_1$ $f_2$ $f_3$ .... T

$S_1$
$S_2$
$S_3$
$S_4$
⋮

$f_1$ $f_3$

$S_1$
$S_2$
$S_{10}$
⋮

$SS_1$ $f_1$ $f_2$

$S_1$

$S_2$

$S_3$

$S_4$

$DT_1$

$SS_2$ $f_2$ $f_3$

$S_1$

$S_5$

$S_6$

$\vdots$

$DT_2$

$SS_3$

$DT_3$

Clustering

Distance Measures

Euclidean

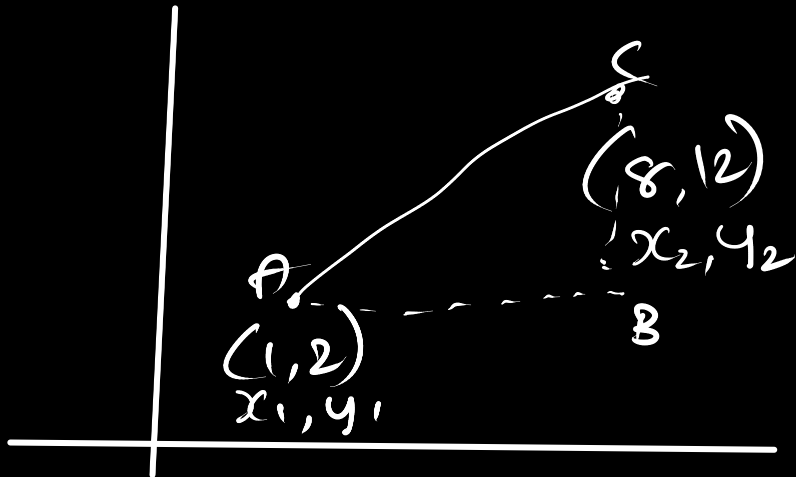
City Block

Chess Board

Jaccard

Cosine

# Eucludean Distan




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

$$AC^2 = AB^2 + BC^2$$

$$AC = \sqrt{AB^2 + BC^2}$$

Clustering vs Classification

No labels — Labels

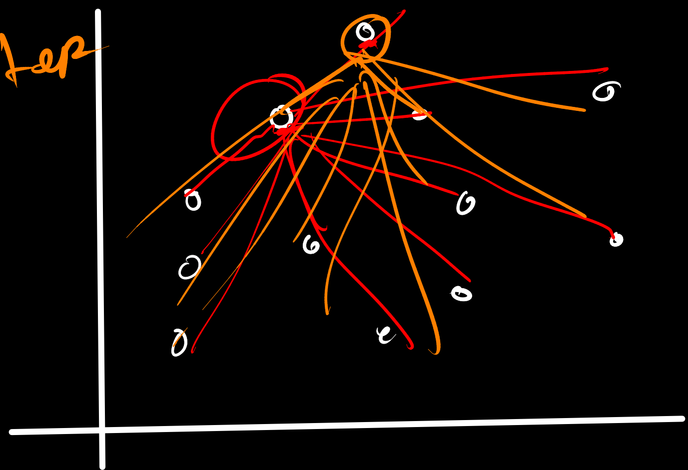
Clustering

KNN ≠

Euclidean Distance

$k$ = # Clusters

Step



$SN_1$

$SN_2$

Step 1 : Chose $SN_1$ $SN_2$

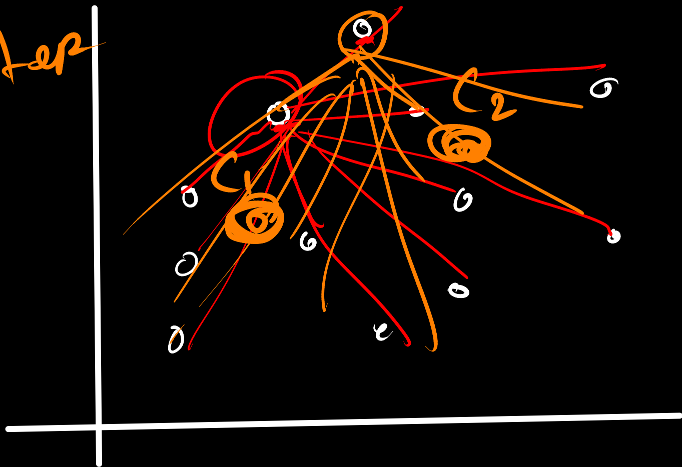
Step2 : find the distance All the nodes $SN_1$

from the distan all the node $SN_2$

Step 3 : $M(SN_1)$

$M(SN_2)$

Step




$SN_1$

$SN_2$

KNN $S_1$ & $S_2$

Clustering

$ED(P_1, P_2)$

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

| | x | y | D($S_1$) | D($S_2$) | Min |
|---|---|---|---|---|---|
| $S_1$ | 1 | 2 | 0 | | $S_1$ $C_1$ |
| $S_2$ | 1 | 4 | | 0 | $S_2$ |
| $S_3$ | 2 | 3 | | | $S_1$ |
| $S_4$ | 6 | 5 | | | $S_2$ $C_2$ |
| $S_5$ | 7 | 6 | | | $S_2$ |
| $S_6$ | 8 | 5 | | | $S_2$ |

$x \quad y \quad (c_1) \quad (c_2) \quad D_1((x_1, y), c_1) \quad D_1((x, y), c_2)$

$$SE \quad \sqrt{\sum (x_i - \hat{x}_i)^2}$$

$$MSE = \frac{SE}{N}$$

$\hat{x}_i$ = clustering label / prediction

$x_i$ = original label

# Random forest

↳ Bagging

↳ Boosting

↳ AdaBoost

| Sample | Feature 1 (X1) | Feature 2 (X2) | Target (y) |
|---|---|---|---|
| 1 | 2 | 3 | 0 |
| 2 | 1 | 2 | 0 |
| 3 | 3 | 5 | 1 |
| 4 | 4 | 6 | 1 |

Bagging

$SS_1$

s $f_1$ $f_2$ T

1

2

3

2

$SS_2$

s $f_1$ $f_2$ T

1

2

3

3

$SS_3$

s $f_1$ $f_2$ T

2

3

3

↳ All the features should be present in every sub samples

↳ Samples in SS can be repeated (with replacement)

↳ In SS, few samples can be absent

↳ All the samples may be present in one or other SS

$SS_1$ $SS_2$ $SS_3$

$DT_1$ $DT_2$ $DT_3$ Parallel

Some Sampling are also

Vote

Whichever is the best gets elected/chosen

# Data Engineering

- PCA - Principal Component Analysis
- LDA -
- SVM - Classification

PCA

↳ Orthogonal

↳ Perpendicular

Highest

Eigenvalues

PCA1

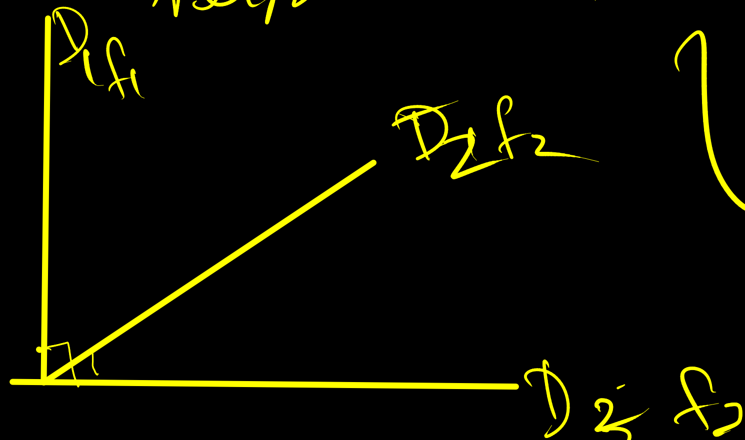
PCA2

PCA3

⋮

Lowest

Descending order




Initial features

↳ Eigen values

↳ Eigenvectors

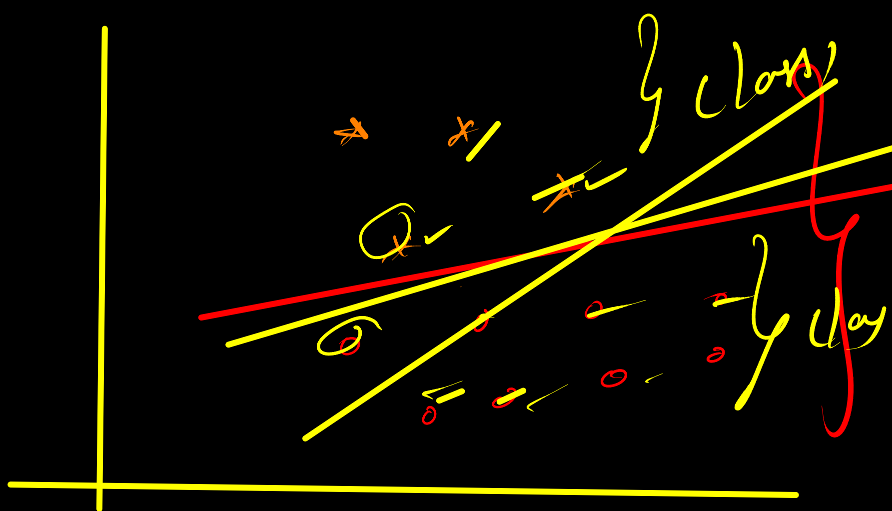
Eigen Space

Ef1 Ef2 ....

perpendicular

# Support Vector Machine [SVM]

Support vectors




Classification
Plane Purpose
↳ Binary classification

Linear

Non Linear

→ Plane

→ Classifier

y Choosing a boundary

Support vectors

↳ 1 or more from each class

→ Difference

– Residuals

→ Loss function

Linear, ← ↳ Regression

Logistic (Sigmoid)

↳ Classification

Decision Trees

Random forest

↳ Bagging

↳ Boosting

↳ Supervised Learning

↳ Labels

↳ Un supervised Learning

↳ X Labels

↳ Clustering.

↳ Semi Supervised Learn.

((weightage)

NLP [Natural Language Processing]

" Text Data."

"The cats are running quickly towards the garden"

↳ Categorical

↳ Bag of words

↳

Preprocessing

↳ Tokenization
Split the words

Lematization [The, cats, are, runn~~ing~~, quick~~ly~~, towards the garden]

Stemming [The cat are run quick toward the garden

Stop word Removal

| | cat | run | quick | toward | garbn | o |
|---|---|---|---|---|---|---|
| f1 | 3 | 1 | 10 | 20 | 40 | 2 |

Encoding

$f_1 \quad f_2 \; - \; - \; - \; - \; - \; f_{10} \; - \; - \; .$

$S_1 \quad - \quad - \quad - \quad 1 \quad 0 \; 0$

$S_2 \quad - \quad - \quad - \quad f_{20}$

Threshold value 20

Padding

Trimming

NLTK → token → Lemmatization Encoding

or

space

NLTK Onehot encoder

Use Spacy,

Sklearn - Tfid

Load the entire csv.

text | Labels

Preprocess it

token, lemmatizat

Stopwords

Simple Classifier

Tfid

f1 f2 . . . . . fn      Labels

↳ Random forest / SVM

Classification

Soft Max

$$\sigma(z) = \frac{e^{z_i}}{\sum e^{z_i}}$$

Sigmoid

Relu

TANH

LekyRelu

Feature Extraction

- ↳ Classification
- ↳ Segmentation
- ↳ Prediction
- ↳ Detection

Multi class

Multi label

→ Neural N/w

Inputs

Hidden

Output

losses
Derivey

Input
→ weights

Black Box

# Image Classification

I/p images → feature Extraction

↳ Dense N/w

Shallow N/w

Deep N/w

# Image Classification

i/p → Color – $n \times m \times c$ — $100 \times 100 \times 3$

Red
Green
Blue / Ye

3 band

Grey – $n \times m$

$100 \times 100$

Features in the maps

↳ Edges / Corners

↳

1000 x 1000

→ Clearity

↳ 100 x 100

↳ Reducing the resolution

↳ Extraction importan features

↳ Global fea

↳ Local featu

Big Data

1000 x 1000

1 million

= 1000 x 1000 x 1 million

b) 50 x 50 x 1 million

c) 24 x 24 x 1 million

↳ CNN

↳ filter

3 x 3

5 x 5

7 x 7

↳ Pooling

Min

max

~~Avg~~

↳ flatten → 1 dimension vector

$$f_1 \quad \begin{array}{|c|c|c|} \hline 0 & 1 & 0 \\ \hline 1 & 1 & 1 \\ \hline 0 & 1 & 0 \\ \hline \end{array}$$

H T T

11

$$\begin{matrix} 0 & 0 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{matrix}$$

$$f_2 \quad \begin{matrix} 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \end{matrix}$$

$$\begin{matrix} 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{matrix}$$

$3 \times 3 \rightarrow$ Edges / Corner

$5 \times 5 \rightarrow$ Edges / Corners

MNIST