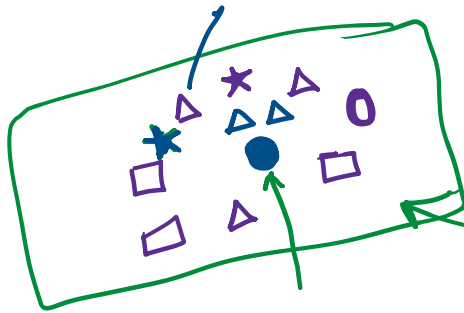


# K-Nearest Neighbour (K-NN)

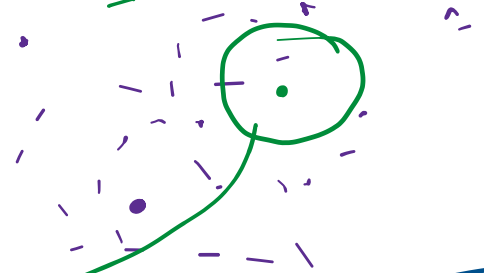
- Supervised Algo
- Classification

Aim: Identifies data points that are separated into different categories & it tries to classify the new sample from the set (based on nearest Neigh)

# Working:



(n-1)



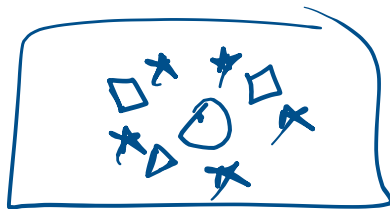
↑ 9 nearest Neighbours

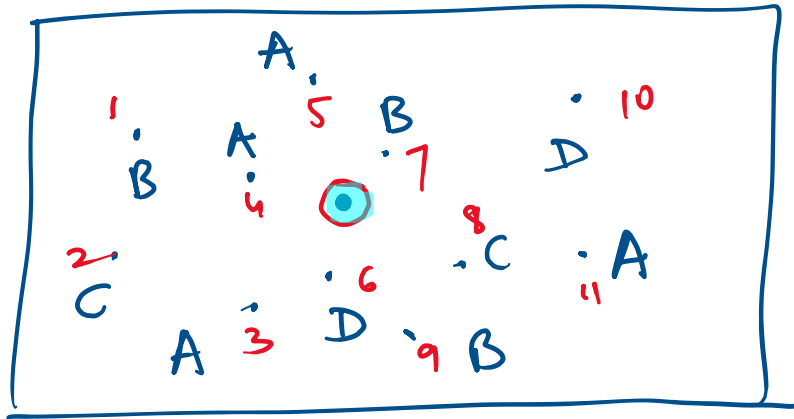
X 10

(11) - N~

$\Delta = A$

$\Delta = A \Rightarrow 3$  5  
★ = B  $\Rightarrow 2$   
□ = C  $\Rightarrow 3$   
○ = D  $\Rightarrow 1$





↑ 5 - Nearest Neighbor



4, 5, 7, 6, 8



A A B D C



Class of my Data Point = A

— 12  
11

$d_1$   
 $d_2$

$d_{11}$

Neighh

1	$d_1$
2	$d_2$
3	$d_3$
4	,
5	,
6	,
7	,
8	,
9	,
10	,
11	$d_{11}$

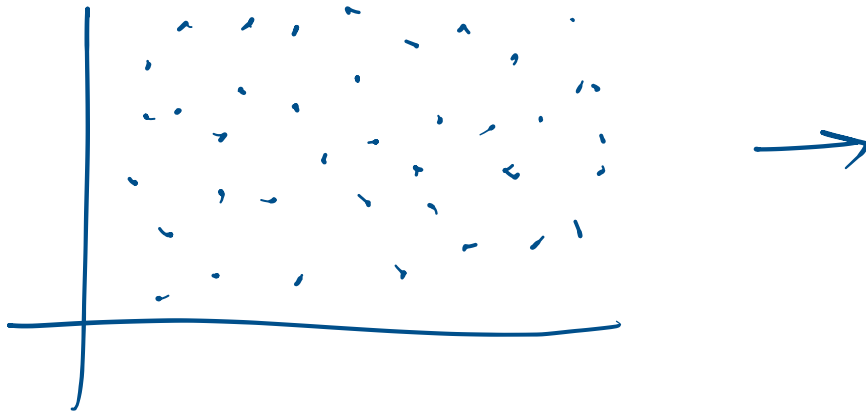
→ Sort.  
(Smallest  
to  
highest)

→ Take first  
k-distance

4	$d_{min}$	← A
5	$d_{min+1}$	←
7	$d_{min+3}$	←
6	$d_{min+4}$	←
8	$d_{min+5}$	←

# Lazy learner.





⇒ At the time of training it memorises the data points & finding the nearest neighbours are done at Prediction time.

Hence ⇒ Lazy learner.

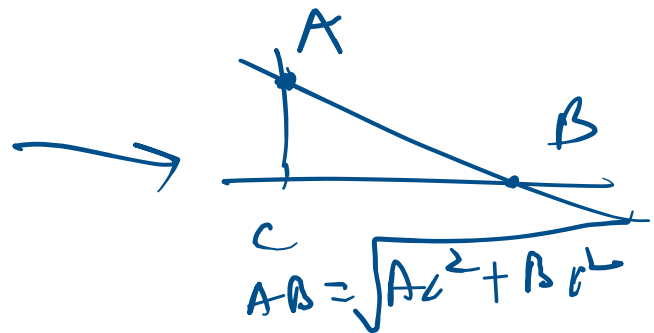
[Minimal Training but expensive learning]

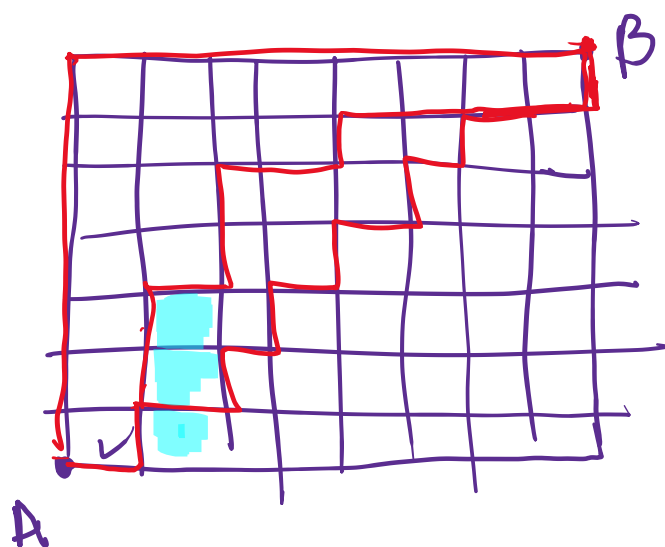
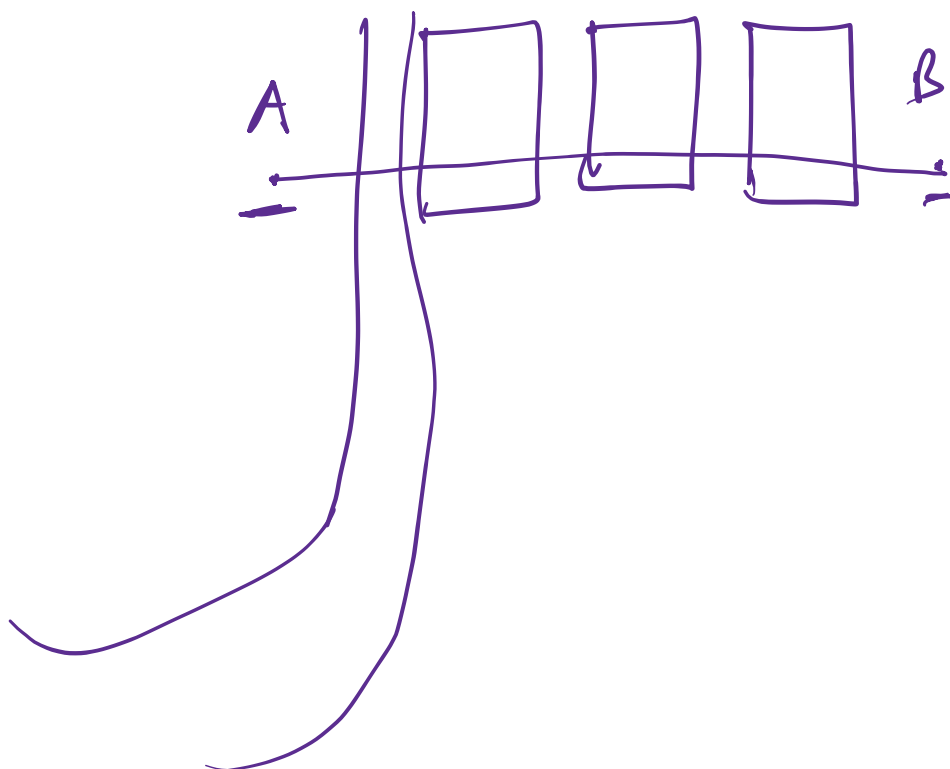
# Distance matrix.

① Euclidean distance

② Manhattan distance

↓  
City Block





$$\approx |y_2 - y_1| + |x_2 - x_1| + \dots$$

$$\sum_i |a_i - b_i|$$

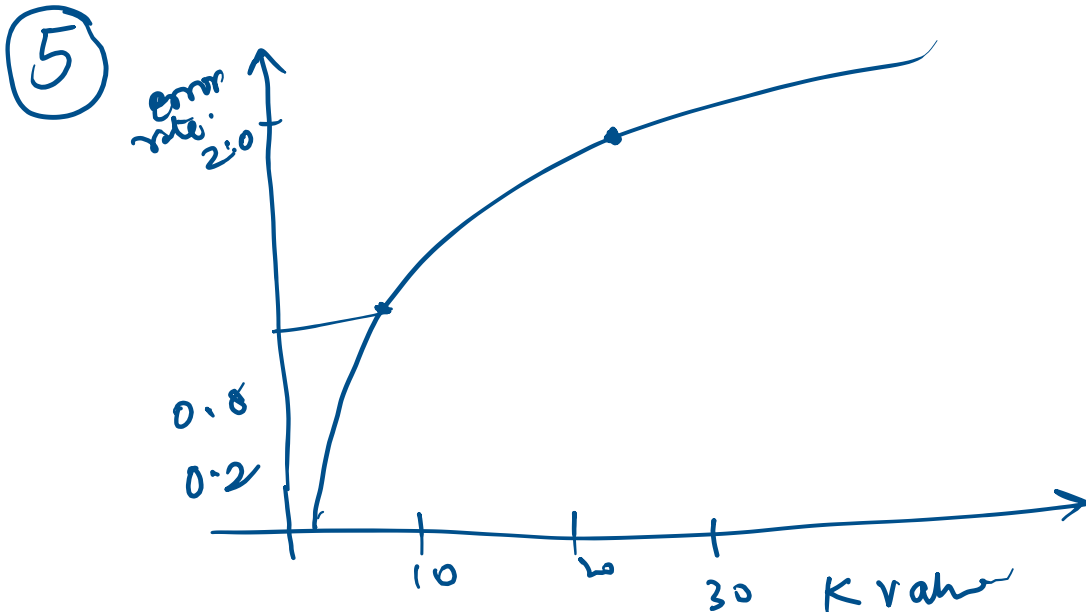
## # Finding the value of k.

① k should be odd - to avoid a tie in class voting.

② take help from accuracy/p/recall/f1-score  
↓  
range of k & then decide based on evaluation matrix.

③ Thumb Rule: optimal value of k =  $\sqrt{N}$

④ avoid very small value of k.



[illegible]