K Nearest Neighbour (KNN)

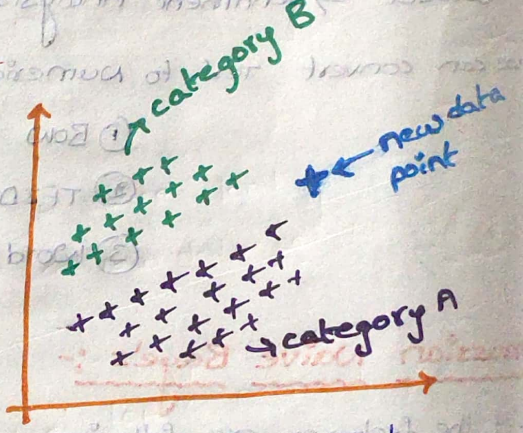
→ K-nearest Neighbour is one of the simplest Machine Learning Algorithms

→ K-NN algorithm assumes the similarity between the new and available cases and put the new case into the category that is most similar to available categories.

→ K-NN algorithm can be used for **Regression** as well as **classification**

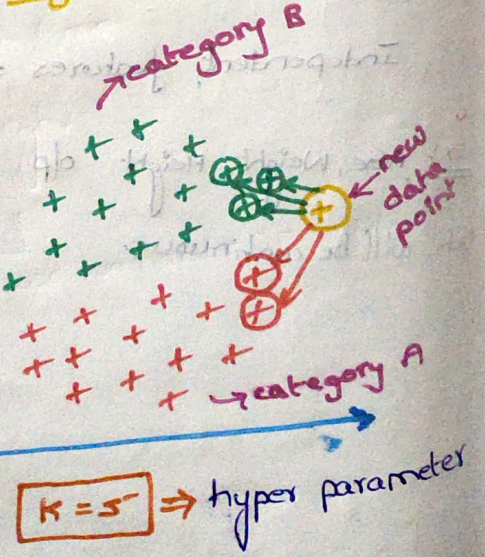
→ It is also called as Lazy Learner algorithm, because KNN algorithm at the training phase just stores the dataset and when it gets new data, then it classifies that data into a category that is much similar to the new data.

→ Suppose we have new data point and we need to put it in the required category.



→ We have to choose the number of neighbours in this case I am considering K=5.

→ Now calculate the distance between new data point and Nearest Neighbours.



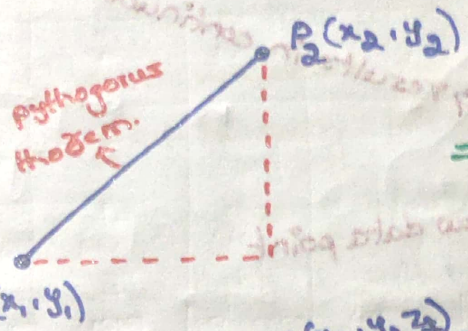
→ In order to calculate the distance we have two methods for distance Metrics.

① Eucledian distance

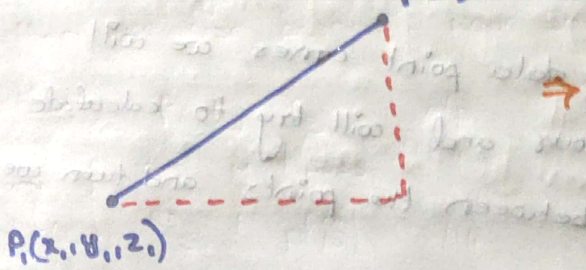
② Manhattan distance

K=5 ⇒ hyper parameter

① Euclidian Distance :-

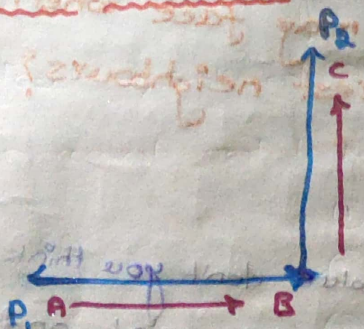


$$\Rightarrow \sqrt{(x_2-x_1)^2 + (y_2-y_1)^2}$$



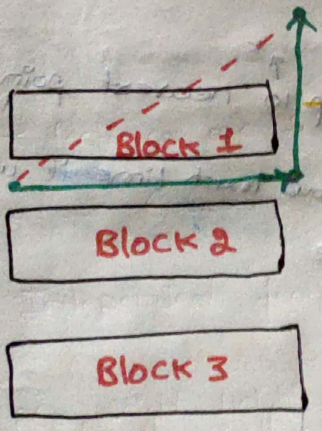
$$\Rightarrow \sqrt{(x_2-x_1)^2 + (y_2-y_1)^2 + (z_2-z_1)^2}$$

② Manhattan Distance :-



$$\Rightarrow |(x_2-x_1) + (y_2-y_1)|$$

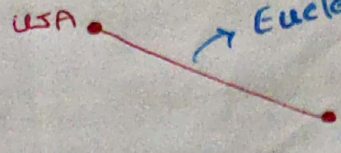
Ex 1 :-



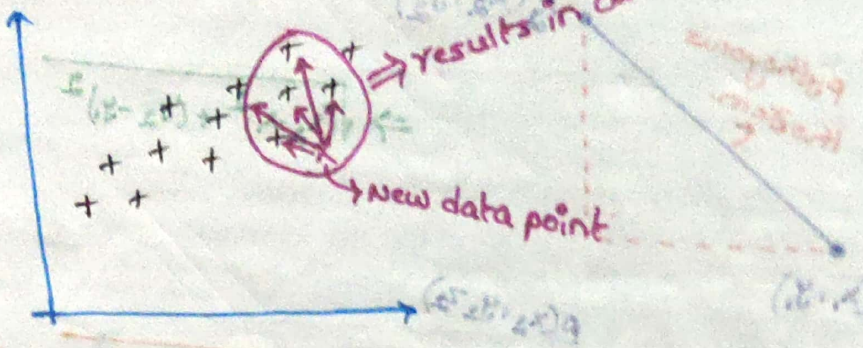
→ manhattan distance

Ex : 2

How do we compute the distance between USA and India?



Regression:-



→ Let's say when a new data point comes we will find 5 nearest neighbour and will try to calculate the average distance between the points and then we will get the continuous values.

→ Let's say in my dataset I have 1 million datapoints what is the problem that I may face when I am trying to findout the nearest neighbours?

→ Let's say K value = 10

→ Now inorder to find out that K-value don't you think we need find the distance between all new data point and all the points.

→ Then sort those values

→ And then try to findout the top 10 nearest points.

→ Incase of millions of datapoints how much time it will take obviously it will take much time.

→ Here Time complexity is very High.

→ How do we fix time complexity?

→ We will fix this by using two variants in KNN.

variants of KNN

1. K-Dimension tree
2. Ball Tree

→ In both these techniques first we ~~create~~ construct a decision Tree.

→ what will happen if we construct a decision tree?

→ In this case we don't need to calculate the distance from one point to every point if we have decision tree.

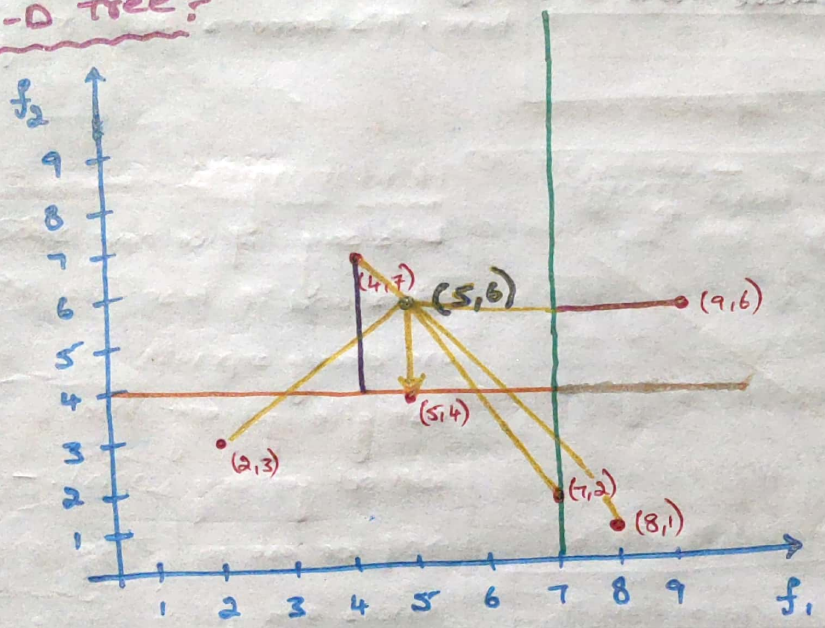
→ what is the importance of binary tree or Decision tree?

→ It takes $\log n$ time it will take

→ It will not take n time or n^2 time.

→ It reduces the time to reduce an Element.

① K-D Tree:



→ Let's consider few data points

(2,3) (4,7) (5,4) (7,2) (8,1) (9,6)

Aim: whenever we get a new data point I should not calculate the distance w.r.t every point.

→ Now will try to find out the median of all the x-axis values

2, 4, 5, 7, 9 ⇒ picking the value 7

$$\text{Median} = \frac{5+7}{2} = 6$$

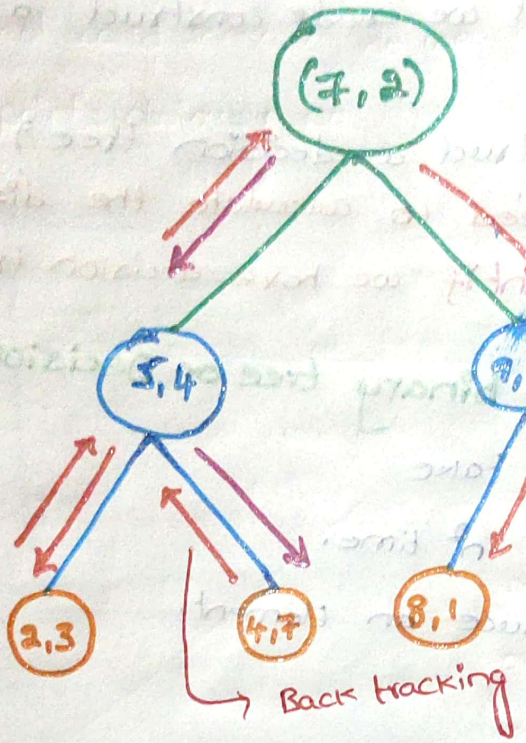
→ Now try to find out the median of all the y-axis val

3, 7, 4, 2, 1, 6

① sort the values

1, 2, 3, 4, 6, 7

$$\text{Median} = \frac{3+4}{2} = 3.5$$



K↓ No: of searches reducing.

→ From 7 ~~calculate~~ draw a vertical line. Root node is (7,2)