K-Nearest Neighbour (KNN)

① KNN Classifier
② KNN regressor

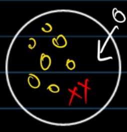
* idea

You will be like people surrounding you.

Scenario 1

Since majority of data is white colour in the surrounding of new data, then it might be also of white colour.

Scenario 2.



o → This ball might be of yellow.

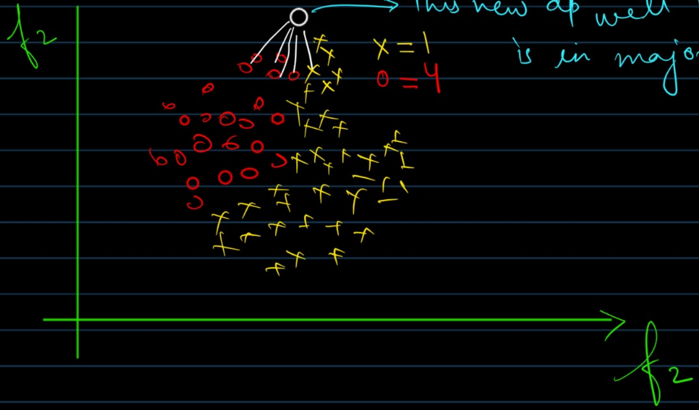
In both of the cases you have considered the majority colour ball in the bag as deciding factor.

① KNN Classifier

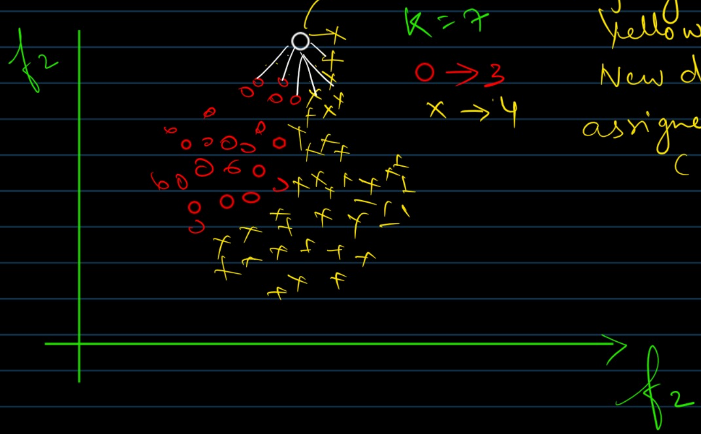
f1	f2	y
—	—	0
—	—	1
—	—	0
—	—	0
—	—	1

K=5 (How many nearest dp you want to see)

→ This new dp will be assigned to 0th class as it is in majority.



→ Majority is Yellow class, New dp will be assigned to yellow class.



* As k changes the class of new dp might also change

⇓

K - hyperparameter.

KNN Classifier Algorithm

① Plot the datapoint in n-d space.

② Initialise the k-value (No of Neighbours you went to consider)

for each k, k ∈ 1 → ∞ (generally is taken greater than 3 and it should always be greater than 0)

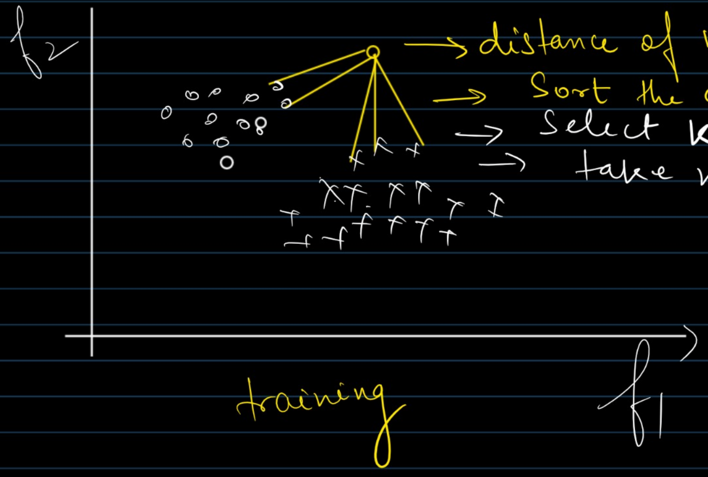
→ Calculate the distance of new dp wrt to all datapoints.

→ Sort the distance

→ based on k find the class of that k nearest datapoint.

③ Find mode of the class

④ Assign the class.



→ distance of new dp from all the dp's

→ Sort the distance.

→ Select k nearest distance dp.

→ take mode and assign the class.

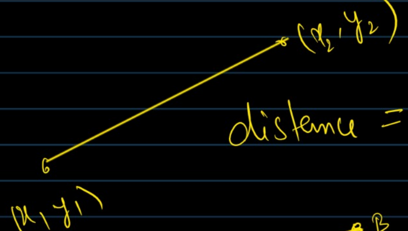
new dp = (100, 50)

f_1	f_2	y	distance of new dp with rest dp	sort	k = 3
—	—	1	40	38	(0) ✓
—	—	1	58	40	(1) ✓
—	—	1	60	58	(1) ✓
—	—	0	38	60	(1)

Majority is 1 for k=3, ∴ Prediction is class 1.

To calculate distance

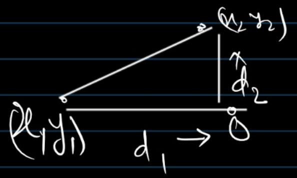
① Euclidean distance



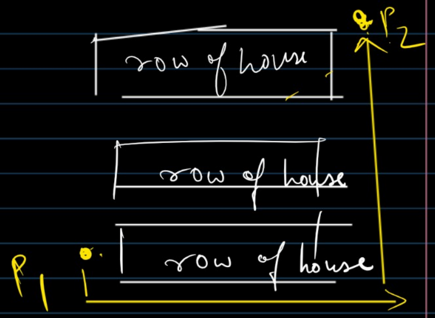
$$\text{distance} = \sqrt{(x_2-x_1)^2+(y_2-y_1)^2}^{\,2}$$

by Aeroplane → Euclidean distance.

② Manhatta distance.

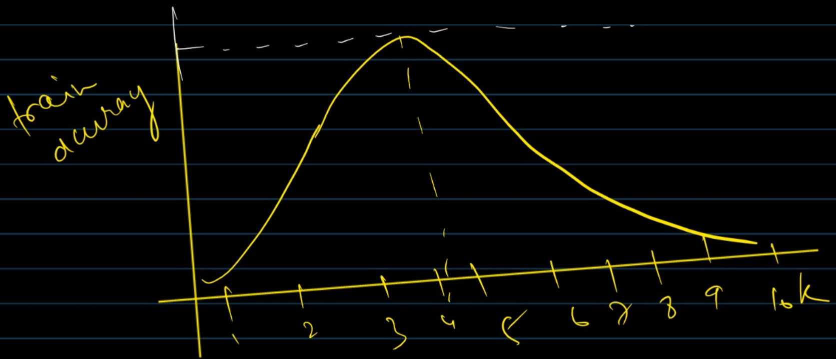


distance = $d_1 + d_2$



K - hyperparameter

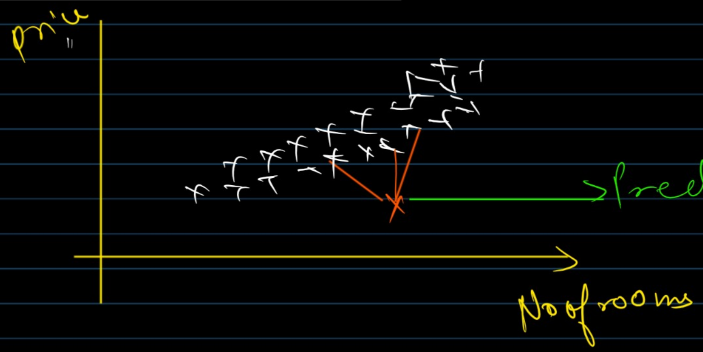
→ for different K, you keep the track of train / test accuracy.



→ for the K which gives highest train accuracy, that will be optimal K.

Summary of Classification
→ calc dist
→ sort distance
→ select nearest K distance.
→ Assign the mode to class

② KNN Regression



→ Prediction of new dp will be average (mean) (y) price of house.

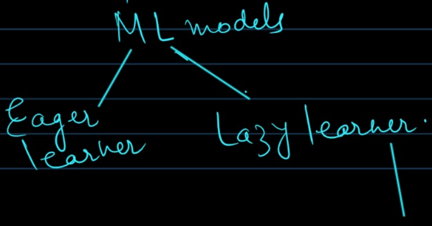
→ In case of outlier instead of Avg, you take median of K nearest dp.

* Advantage

→ Easy to Understand / Very intuitive
→ Performance of model in terms of evaluation metric is good. (Rsq, acc)

* disadvantage

→ Lazy learner.



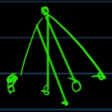
↓
At real time the distance of test data from each of dp is Calulated ⇒ Brute force.

→ All the model parameter are calculated while training and used for prediction

↓
→ you are Calculating parameters of model in real time.

⇓

training | computation time is more.



→ n dp → n-1 distance you will be Calculating.
→ Computationally expensive $(O(N))$

* Variants of KNN

$\left.\begin{array}{l}\text{① KD Tree}\\ \text{② Ball tree}\end{array}\right\} \rightarrow$ Optimizers