K-Nearest Neighbors (KNN)

→ KNN is a simple ML Algorithm used for classification and Regression tasks

→ The main idea behind KNN is to clasify a data points based on the majority class of its neighboring data points.

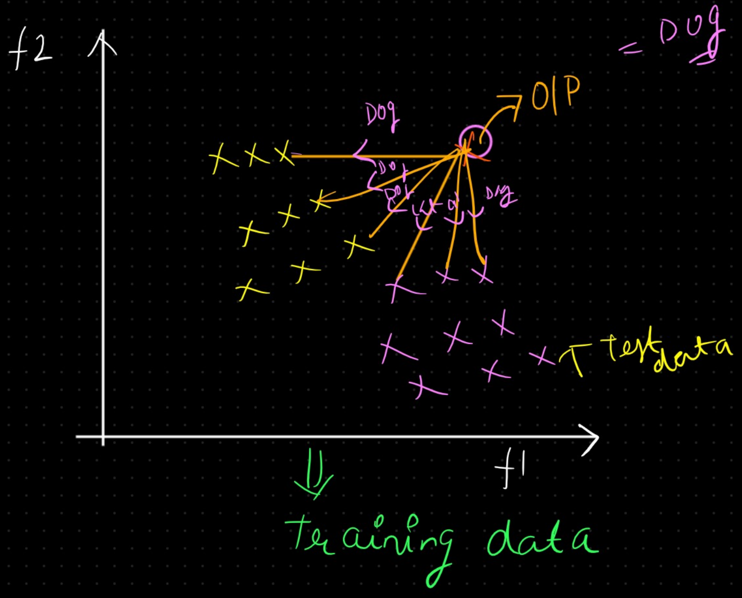
1. Classification
2. Regression

KNN → Classification

(f1) Height (f2) weight

O/P → categorical

Animal
↓
Dog / Cat



K = Parameter

K = 6

① Initalize → K > 0

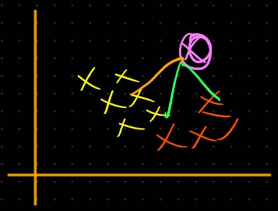
K = 1, 2, 3, 4, 5 ---

② Find the K-Nearest Neighbor from the Test Data

③ K = 5, majority of Class

How KNN works →

① Collect the Data



② K = 3

③ Distance Metrics

Euclidean distance

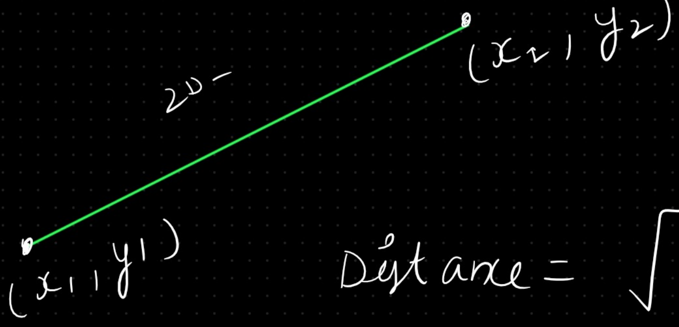
Manhattan distance

④ Find Nearest Neighbours

⑤ Counts Votes (majority)

⑥ Prediction

Eucledian distance



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

3D

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

$x_1, x_2, x_3 - - - x_n$

$y_1, y_2, y_3 - - y_n$

$$\text{Eucledian distance} = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$$

This distance metric is intuitive and measures the straight-line distance between two points in space.

Manhattan Distance

↓

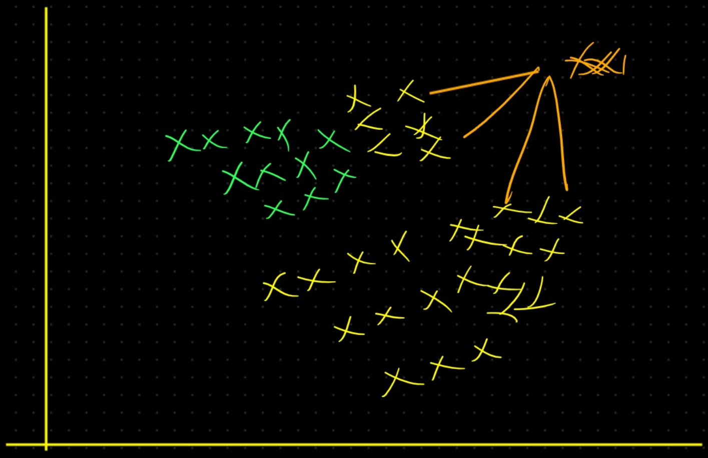
taxi cab / block distance

$$= |x_2 - x_1| + |y_2 - y_1|$$

→ Euclidean distance measures the shortest straight-line path between two points, while Manhattan distance measures the distance along the grid lines.

→ Euclidean distance is often used when the data points are continuous and can be represented in a Cartesian plane, whereas Manhattan distance is useful when dealing with data points in a grid-like structure, such as images or maps.

→ Euclidean distance is sensitive to outliers, while Manhattan distance is less sensitive since it measures the sum of absolute differences.



Large Dataset → Time complexity ↑↑

[Auto] KNN ↓

→ ① Ball Tree } Binary Tree

→ ② KD Tree

↓

→ Reducing the number of distance calculation

Regression