KNN (K- nearest neighbour)

It is a supervised ML algo used for both classification & regression tasks.

Basic Idea

Find K closest points

Use those to predict output

- Classification → Majority voting

  Regression → Avg. of values.

Steps

① Choose K (hyperparameter), common values → 3, 5, 7

somewhere use $\sqrt{N}$ to find K, K odd- help in classification to avoid ties.

② Calculate distance b/w test & all training points.

most used euclidian $D = \sqrt{\sum_{i=1}^{N} (x_i - y_i)^2}$

③ Sort distances

④ Pick first K points after sorting

⑤ Make prediction

- Classification → Count class label of K- neighbours

  class with high freq is predicted

- Regression → Take mean of K neighbour values

$$\left\{ \hat{y} = \frac{1}{K} \sum_{i=1}^{K} y_i \right\}$$

ex → • Classification

K = 3, nearest neighbor labels = {Red, Blue, Red}

Prediction → Red (majority vote)

• Regression

K = 4, neighbor values → 10, 12, 14, 16

$$\hat{y} = \frac{10+12+14+16}{4} = 13$$

• Point about K

small K → sensitive to noise, low bias, high variance

Large K → Smoth decision boundary, high bias, low variance

• Adv

→ Simple & easy

→ No training phase

↳ As it does not build model during training instead it stores the training data & makes prediction only at test time.

→ Works well with small datasets.

Dis adv

→ Slow for large dataset

→ High memory usage

→ Sensitive to noise & outliers.