

## kNN Introduction

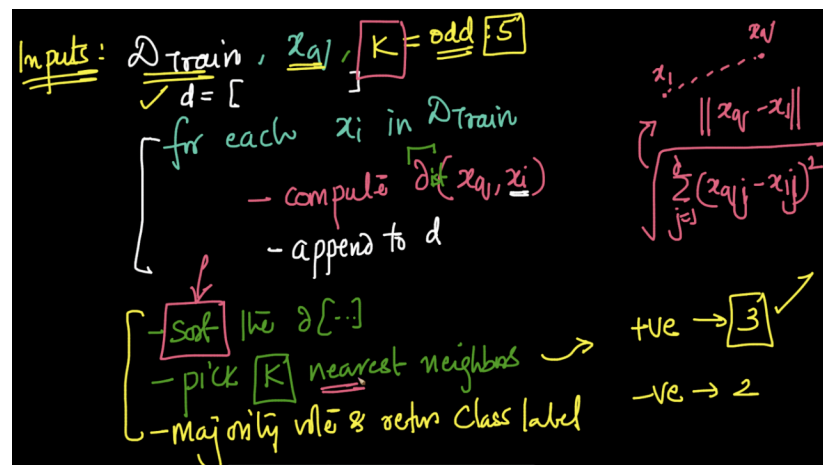
- The k-Nearest Neighbors (kNN) algorithm is a versatile and intuitive supervised approach used for classification and regression tasks in ML
- kNN is non-parametric and instance-based, making minimal assumptions about the underlying data distribution and offering simplicity in implementation.
- kNN can easily handle non-linear and multiclass problem

## Why kNN?

- Simple to use for multiclass classification
  - One v/s rest method would have to be employed for Logistic regression
- Non-linear data handling
  - Logistic regression needs to create polynomial features

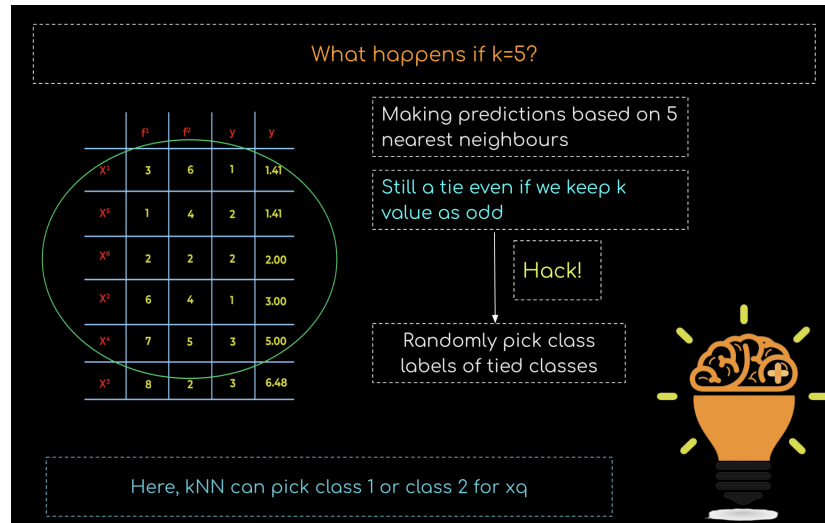
## kNN Algorithm

1. Training → No training required, just store the training data
2. Testing Prediction:
  - a. Find out the distance of all the points from each point using a distance metric
  - b. Filter out k-nearest neighbors of each point
  - c. Assign majority class label to the point



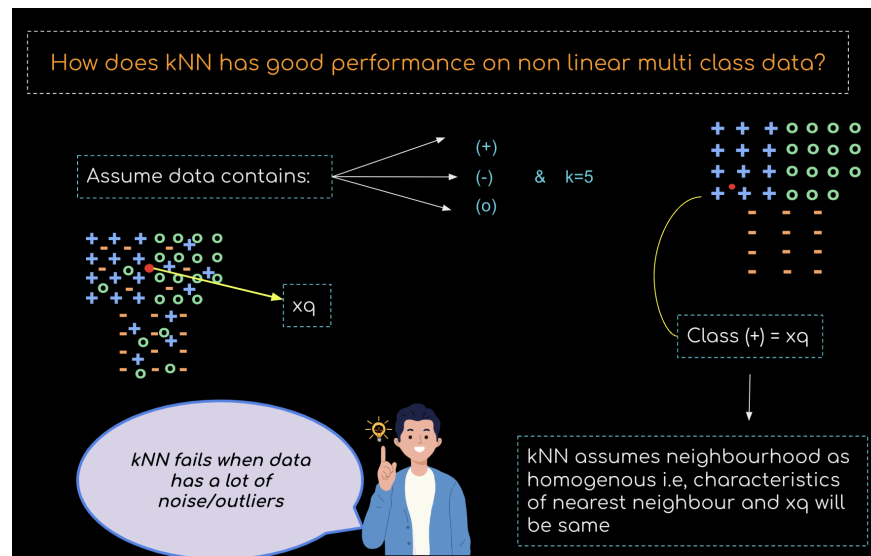
## k value must be odd

1. If k is even → can have a tie situation resulting in no accurate prediction
2. If k is odd → In a tie situation, randomly pick the class label for any of the tied Classes



## kNN Assumption

- Homogeneous Neighborhoods -> Similar things are close to each other
- kNN fails if there are a lot of outliers in the data



## Bias-Variance Tradeoff in kNNs

- High variance: Small  $K$  -> Starts fitting outliers/noise
- High bias: Large  $K$  -> Higher inaccurate predictions since they are affected by faraway points also

