

## \* HPC - Assignment - 04 \*

\* Aim:- Parallel Implementation of  $k$  Nearest Neighbour classifier

\* Objective:- To understand  $k$ -nearest neighbor classifiers.

\* Theory:-

o  $k$ -nearest neighbors classifier:-

Given a dataset of items, each having numerically valued features (like height, weight, Age, etc.). If the count of features is  $n$ , we can represent the items as points in an  $n$ -dimensional grid.

- Given a new item, can calculate the distance from the item to every other item in the set.
- Pick the  $k$ -closest neighbors & see where most of these neighbors are classified in.
- How to calculate the distances between items - The solution to this depends on the data set. If the values are real we usually use Euclidean distance.
- If the values are categorical or binary, usually use the Hamming distance.



## \* Algorithm:-

Given a new item-

- ① Find distance b/w new item & all other items.
- ② Pick k shorter distances.
- ③ Pick the most common class in these k distances.
- ④ That class is where we will classify the new item.

## \* Finding Euclidean Distance:-

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

## \* Conclusion:-

Understood the k-nearest neighbor classifier.