* K- Nearest Neighbour (KNN):

- Supervised, non-parametric algorithm, which means it doesn't make any assumption on underlying data.
- Distance based approach.
- Used for classification and Regression both.
- It stores all available data points and classifies a new data point based on the similarity, i.e distance from 'K' nearest points.
- The data point which we try to classify or predict target for is called query data point.
- It is lazy learner algorithm as it doesn't learn from training set immediately, instead it stores the dataset and at time of classification or regression of query data point, it performs action on the dataset.

A. Working

1. Initialize value of K
2. For each query data point in dataset:
   - 2.1 Calculate distance between query data point and all of the data points from training dataset.
   - 2.2 Sort the distances in ascending order.
   - 2.3 Get top K rows from sorted distance array.
   - 2.4 Get labels of selected entries

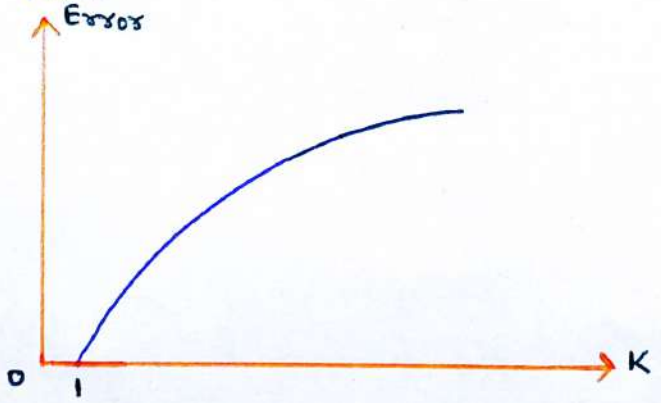
KNNR → If regression, return mean of K Labels

KNNC → If classification, return mode or most frequent of K Labels.

- KNNR: KNN regressor.
- KNNC: KNN classifier.

B. Selection of K

- K is hyperparameter, so we need to tune it.
- In case of KNNC as we are choosing the label with max vote, so we should choose K value as odd to do the tie breaking.
- Never choose K=1, as it simply means we will decide the group which query data point belongs based on single data point.



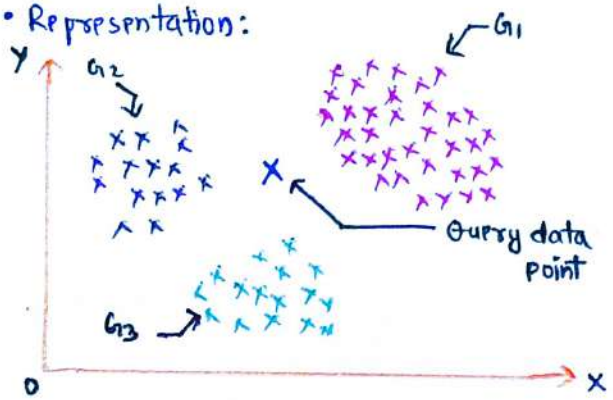
→ At K=1, error is 0 as only one class.

- Various ways to calculate distance, but most popular one is euclidian distance (E).

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

where,
two points $P_1(x_1,y_1)$ and $P_2(x_2,y_2)$ has E distance.

- Representation:



C. Advantages

1. Robust to noisy data.
2. No assumptions
3. Versatile
4. No model training, easy tuning.

D. Disadvantages.

1. Works better on bigger dataset.
2. High computation cost