

## Introduction:

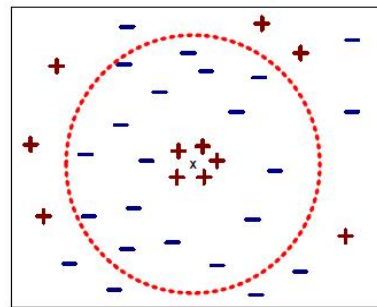
1. This series is aimed to step by step approach at Image Recognition problem.
2. The first in the series approach is to implement the image recognition problem through KNN

## Why KNN?

K Nearest Neighbor (KNN from now on) is one of those algorithms that are very simple to understand but works incredibly well in practice. Also it is surprisingly versatile and its applications range from vision to proteins to computational geometry to graphs and so on.

## Algorithm :

1. Consider the following pic, suppose we want to find the class of 'x'
  - a. Calculate the euclidean distance of  $|x-y_i|$ ,  $y_i$  being element.
  - b. Get the closest neighbors of 'x' in terms of distances.
  - c. Get the class of the top voted class of the neighbors.



## Implementation(knn\_iris):

1. LoadDataset: loads the iris dataset, and divides the dataset in training and test sets.
2. euclideanDistance: Computes the Euclidean distance between any two elements, upto a specified length:
  - a. `euclideanDistance([1,2,3,'a'],[3.5,4.2,6,'c'],3) = 4.4821869662029945`
3. getNeighbors: The function calculates the closest neighbors with their class for a given 'testInstance'. 'k' implies how many neighbors we want.
4. getResponse: The function gives the most occurred class
5. getAccuracy: The function compares with the predicted and actual to give an accuracy.

I will upload the documentation for KNN implementation in Image recognition.

