

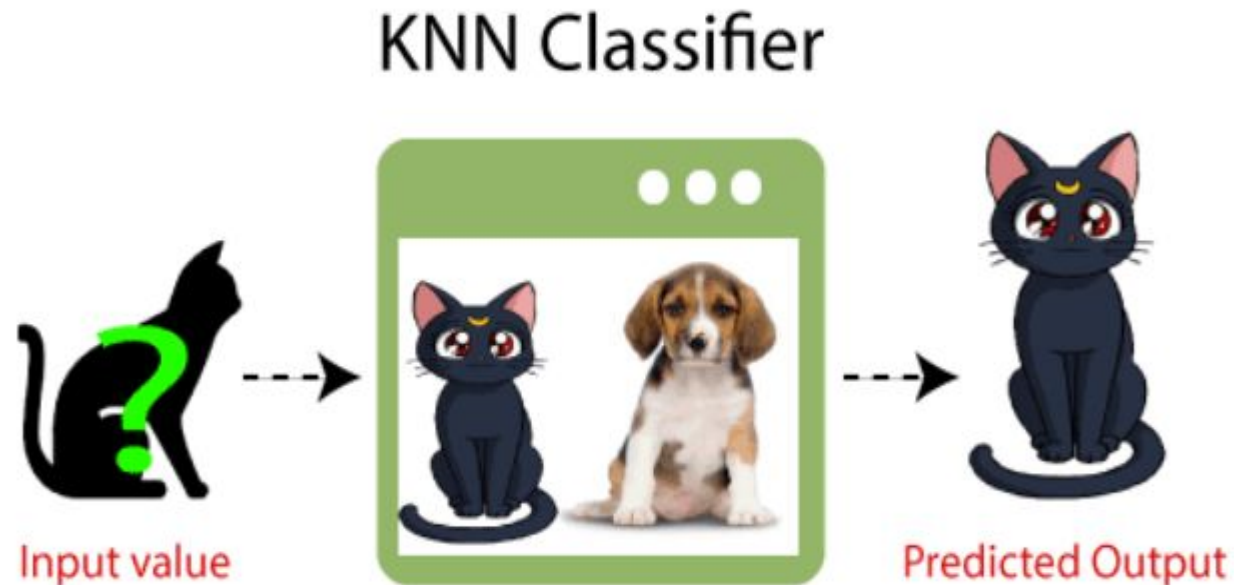
K Nearest Neighbor

- K-Nearest Neighbour is one of the simplest Machine Learning algorithms based on Supervised Learning technique.
- K-NN algorithm assumes the similarity between the new case/data and available cases and put the new case into the category that is most similar to the available categories.
- K-NN algorithm stores all the available data and classifies a new data point based on the similarities of neighbour. This means when new data appears then it can be easily classified into a well suite category by using K- NN algorithm.
- K-NN algorithm can be used for Regression as well as for Classification but mostly it is used for the Classification problems.
- K-NN is a **non-parametric algorithm**, which means it does not make any assumption on underlying data.

- It is also called a **lazy learner algorithm** because it does not learn from the training set immediately instead it stores the dataset and at the time of classification, it performs an action on the dataset.
- KNN algorithm at the training phase just stores the dataset and when it gets new data, then it classifies that data into a category based on the similarity of nearest neighbours.

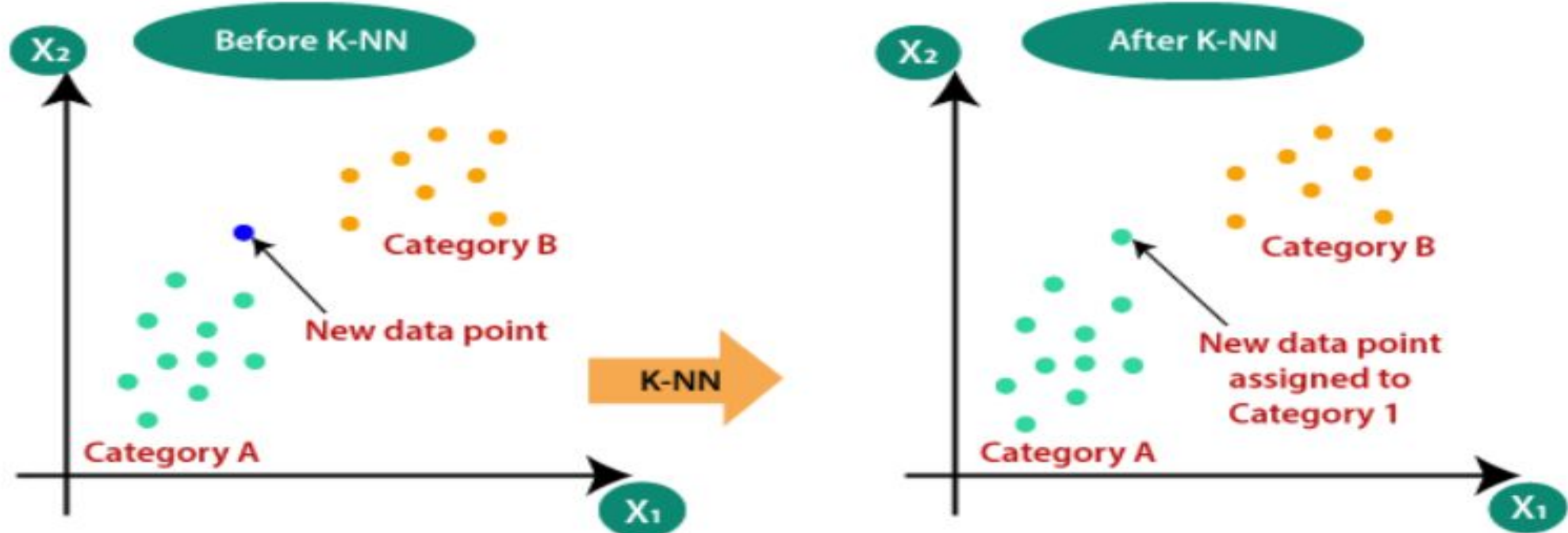
Example

- Suppose, we have an image of a creature that looks similar to cat and dog, but we want to know either it is a cat or dog.
- So for this identification, we can use the KNN algorithm, as it works on a similarity measure of nearest neighbours.
- Our KNN model will find the similar features of the new data set to the cats and dogs images and based on the most similar features of nearest neighbours it will put it in either cat or dog category



Why do we need a K-NN Algorithm?

- Suppose there are two categories, i.e., Category A and Category B, and we have a new data point x_1 , so this data point will lie in which of these categories.
- To solve this type of problem, we need a K-NN algorithm. With the help of K-NN, we can easily identify the category or class of a particular dataset.
- Consider the below diagram:



How does K-NN work?

The K-NN working can be explained on the basis of the below algorithm:

Step 1 – For implementing any algorithm, we need dataset. So during the first step of KNN, we must load the training as well as test data.

Step 2 – Next, we need to choose the value of K i.e. the nearest data points. K can be any natural number.

Step 3 – For each point in the test data do the following –

3.1 – Calculate the distance between test data and each row of training data with the help of any of the method namely: **Euclidean, Manhattan or Hamming distance**. The most commonly used method to calculate distance is **Euclidean**.

3.2 – Now, based on the distance value, sort them in ascending order.

3.3 – Next, it will choose the top K rows from the sorted array.

3.4 – Now, it will assign a class to the test point based on most frequent class of these rows.

Step 4 – End

- Euclidean distance formula is given by:

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

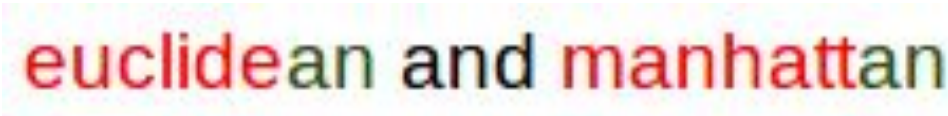
Where,

- “d” is the Euclidean distance.
- (x_1, y_1) is the coordinate of the training data point
- (x_2, y_2) is the coordinate of the test data point.

- Manhattan distance formula is given by:

$$d = |x_2 - x_1| + |y_2 - y_1|$$

Hamming Distance

- Hamming Distance measures the similarity between two strings of the same length.
- The Hamming Distance between two strings of the same length is the number of positions at which the corresponding characters are different.
- Let's understand the concept using an example. Let's say we have two strings: **“euclidean”** and **“manhattan”**
- Since the length of these strings is equal, we can calculate the Hamming Distance.
- We will go character by character and match the strings.
- The first character of both the strings (e and m respectively) is different. Similarly, the second character of both the strings (u and a) is different. and so on.
- Look carefully – seven characters are different whereas two characters (the last two characters) are similar:

- Hence, the Hamming Distance here will be 7. Note that larger the Hamming Distance between two strings, more dissimilar will be those strings (and vice versa).

Example: Calculate the output for 6th data item in the table using KNN algorithm. Assume that first 5 are training data item and 6th is testing data item. Assume K=3.

S. No.	Maths	English	Result
1.	4	3	Fail
2.	6	7	Pass
3.	7	8	Pass
4.	5	5	Fail
5.	8	8	Pass
6.	6	8	?

Ans: Euclidean distance from each training data point:

1. $d = \sqrt{(6-4)^2 + (8-3)^2} = 5.38$

2. $d = \sqrt{(6-6)^2 + (8-7)^2} = 1$

3. $d = \sqrt{(6-7)^2 + (8-8)^2} = 1$

4. $d = \sqrt{(6-5)^2 + (8-5)^2} = 3.16$

5. $d = \sqrt{(6-8)^2 + (8-8)^2} = 2$

So, from the above calculation, Euclidean distance for data point no. 2,3, and 5 are lowest. As the 3-Nearest Neighbor belongs to pass category; therefore, the 6th data point also belongs to pass category.

Advantages of KNN Algorithm:

- It is simple to implement.
- It is robust to the noisy training data
- It can be more effective if the training data is large.

Disadvantages of KNN Algorithm:

- Always needs to determine the value of K which may be complex some time.
- The computation cost is high because of calculating the distance between the data points for all the training samples.