# KNN Classification Algorithm

o 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.

o K-NN algorithm stores all the available data and classifies a new data point based on the similarity. This means when new data appears then it can be easily classified into a well suite category by using K- NN algorithm.

o K-NN algorithm can be used for Regression as well as for Classification but mostly it is used for the Classification problems.

o 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.

o KNN algorithm at the training phase just stores the dataset and when it gets new data, then it classifies that data into a category that is much similar to the new data.
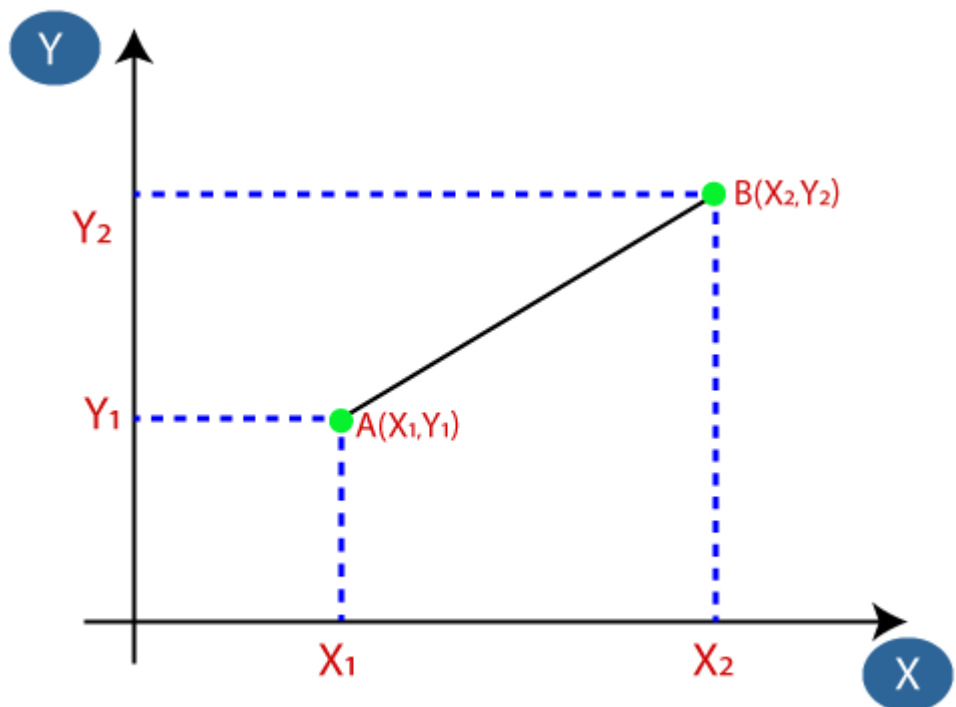
Below are some points to remember while selecting the value of K in the K-NN algorithm:

o There is no particular way to determine the best value for "K", so we need to try some values to find the best out of them. The most preferred value for K is 5.

o A very low value for K such as K=1 or K=2, can be noisy and lead to the effects of outliers in the model.

o Large values for K are good, but it may find some difficulties.

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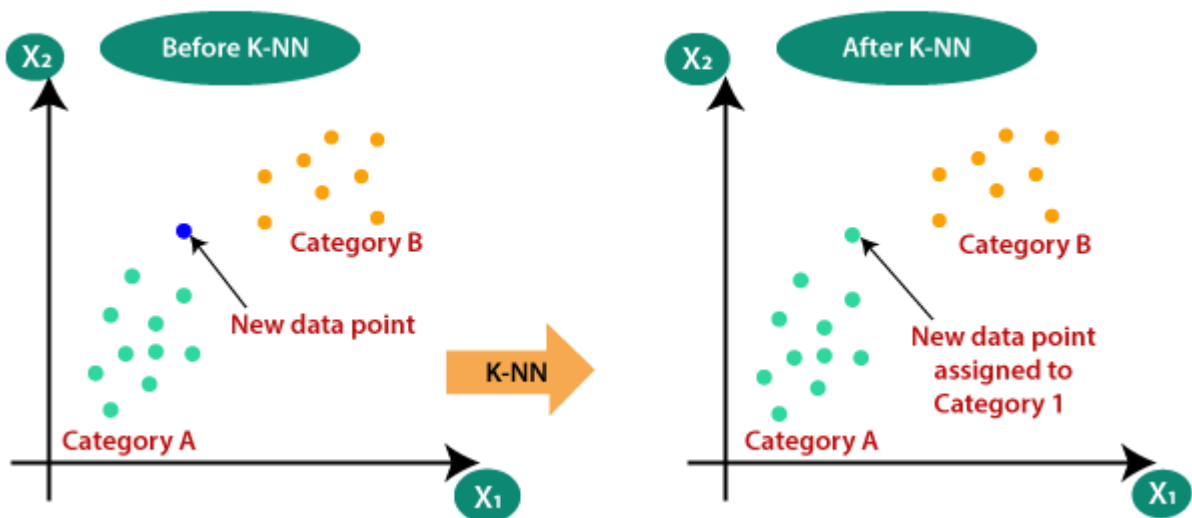
The K-NN working can be explained on the basis of the below algorithm:

- o **Step-1:** Select the number K of the neighbors
- o **Step-2:** Calculate the Euclidean distance of **K number of neighbors**
- o **Step-3:** Take the K nearest neighbors as per the calculated Euclidean distance.
- o **Step-4:** Among these k neighbors, count the number of the data points in each category.
- o **Step-5:** Assign the new data points to that category for which the number of the neighbor is maximum.
- o **Step-6:** Our model is ready.

Euclidean Distance between A_1 and $B_2 = \sqrt{(X_2-X_1)^2+(Y_2-Y_1)^2}$

Category A:3 neighbors
Category B:2 neighbors

Category B

New Data point

Category A

Polynomial Regression

- Polynomial Regression is a regression algorithm that models the relationship between a dependent(y) and independent variable(x) as nth degree polynomial. The Polynomial Regression equation is given below:

$$y = b_0 + b_1x_1 + b_2x_1^2 + b_2x_1^3 + \ldots \ldots b_nx_1^n$$

- It is also called the special case of Multiple Linear Regression in ML. Because we add some polynomial terms to the Multiple Linear regression equation to convert it into Polynomial Regression.

- It is a linear model with some modification in order to increase the accuracy.

- The dataset used in Polynomial regression for training is of non-linear nature.

- It makes use of a linear regression model to fit the complicated and non-linear functions and datasets.

The need of Polynomial Regression in ML can be understood in the below points:

- If we apply a linear model on a **linear dataset**, then it provides us a good result as we have seen in Simple Linear Regression, but if we apply the same model without any modification on a **non-linear dataset**, then it will produce a drastic output. Due to which loss function will increase, the error rate will be high, and accuracy will be decreased.

- So for such cases, **where data points are arranged in a non-linear fashion, we need the Polynomial Regression model**. We can understand it in a better way using the below comparison diagram of the linear dataset and non-linear dataset.

Simple linear model

$y=b_0+b_1x$

Polynomial model

$y=b_0+b_1x_1+b_2x_1^2$
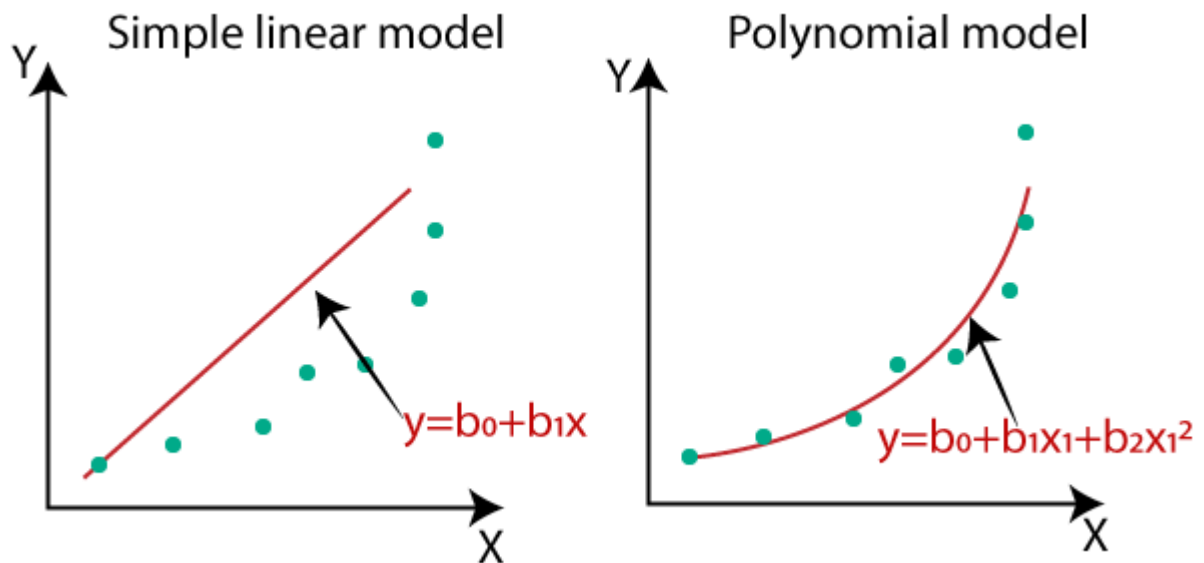
- In the above image, we have taken a dataset which is arranged non-linearly. So if we try to cover it with a linear model, then we can clearly see that it hardly covers any data point. On the other hand, a curve is suitable to cover most of the data points, which is of the Polynomial model.
- Hence, *if the datasets are arranged in a non-linear fashion, then we should use the Polynomial Regression model instead of Simple Linear Regressio*

Equation of the Polynomial Regression Model:

Simple Linear Regression equation: $y = b_0+b_1x$ ....(a)

Multiple Linear Regression equation: $y= b_0+b_1x+ b_2x_2+ b_3x_3+....+ b_nx_n$ ....(b)

Polynomial Regression equation: $y= b_0+b_1x + b_2x^2+ b_3x^3+....+ b_nx^n$ ....(c)

-