K-Nearest Neighbors (KNN)

(non-parametric, supervised learning classifier)

a) What it is:

- The k-nearest neighbors algorithm, sometimes referred to as KNN or k-NN, is a supervised learning classifier that employs similarity to produce classifications or predictions about the grouping of a single data point.
- K-Nearest Neighbors (KNN) algorithm is a non-parametric technique, because it makes no assumptions about the implicit or from the inherited data.
- The KNN method simply saves the information during the training phase, and when it receives new data, it categorizes it into a category that is quite similar to the new data.
- It is also known as a lazy learner algorithm since it saves the training dataset rather than learning from it straight away. Instead, it uses the dataset to perform an action when classifying data

Why it is:

Suppose we have a dataset of two categories named category ‘x’ and category ‘y’. we have a new sample or feature named ‘s’. Now we need to find which category of ‘x’ and ‘y’ the new sample ‘s’ belongs to. To solve this problem or to classify the new data sample K- Nearest Neighbors algorithm is required. K-Nearest Neighbor algorithm makes it simple to determine the category or class of a given data sample or feature, here in our case ‘s’.

b) How it does:

Step-1: Choose the variable K that is number of nearest neighbors.

We have to choose the parameter K optimally by using the thumb rule of square root of number of data samples in the training dataset we can consider the value k.

Step-2: Calculate the Euclidean distance of K nearest number of neighbors between the given sample from all the training samples.

After the Step-1, to calculate the Euclidean distance we can use the below formula which we have learnt in high school geometry. The Euclidean distance between the points M (x_1, y_1), N (x_2, y_2) is $\sqrt{(x_2-x_1)^2+(y_2-y_1)^2}$

Step-3: Sort the calculated Euclidean distances and determine the K nearest neighbors as per the calculated Euclidean distance.

After calculating the Euclidean distance sort all the values of distances and determine the nearest neighbors based on the k^{th} minimum distance.

Step-4: Count the number of data points in each category among these k neighbors.

After sorting collect and count all the data points of each category of the nearest neighbors.

Step-5: Assign the new or given sample data points to the category where the neighbor count is at its highest.

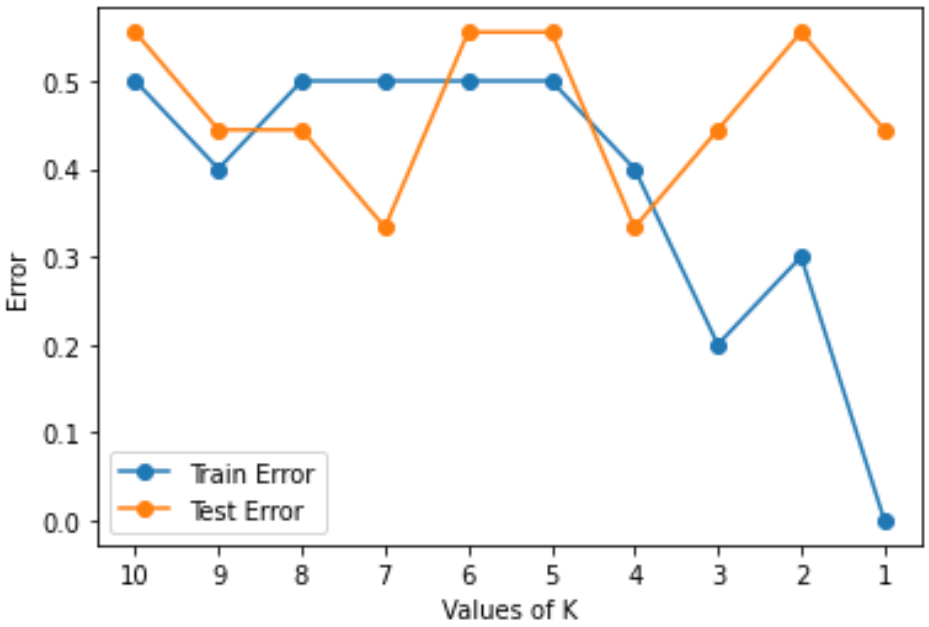
By using the majority of the nearest neighbor count assign the new or given sample data points to that category.

c) Application:

There are so many real world applications of KNN algorithm. Few of them are prediction of stock prices, calculating credit score, Facial recognition, Recommendation systems and also in some parts of agriculture and medical sectors.

RESULTS for KNN Model

a) Plot Results with Training errors and Test errors.



b) On x-axis indicating values of K. On y-axis indicating Error rate.

c) Observation:

The most common thing in the K- Nearest Neighbor in obtaining the optimal value of k is training error is always zero when the value of k is 1. The rule of thumb for KNN says that square root of the number of samples gives the value of k without plotting error rate. Test error is same at k=4 and k=7.

d) Conclusion:

The critical and important point is Testing error should be minimum to obtain the optimal value of K. From the plot the test error is same at k=4 and k=7. And from comparing the train error for k=4 and k=7 the train error is minimum at k=4. So the optimal value of K is 4.