**3.1 Introduction Proposed Models:**

Explaining the model, we implemented here in five steps,

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

What is K in KNN algorithm?

K in KNN is the number of nearest neighbors considered for assigning a label to the current point. K is an extremely important parameter and choosing the value of K is the most critical problem when working with the KNN algorithm. The process of choosing the right value of K is referred to as parameter tuning and is of great significance in achieving better accuracy. If the value of K is too small then there is a probability of overfitting the model and if it is too large then the algorithm becomes computationally expensive. Most data scientists usually choose an odd number value for K when the number of classes is 2. Another formula that works well for choosing K is, k- sqrt(n) where n is the total number of data points.

Selecting the value of K depends on individual cases and sometimes the best method of choosing K is to run through different values of K and verify the outcomes. Using cross-validation, the KNN algorithm can be tested for different values of K and the value of K that results in good accuracy can be considered as an optimal value for K.

**Conclusion 3.6 Proposed Models**

The algorithm can be summarized as:

* A positive integer k is specified, along with a new sample
* We select the k entries in our database which are closest to the new sample
* We find the most common classification of these entries
* This is the classification we give to the new sample

A few other features of KNN:

* KNN stores the entire training dataset which it uses as its representation.
* KNN does not learn any model.
* KNN makes predictions just-in-time by calculating the similarity between an input sample and each training instance.