**KNN IMPUTER**

Missing values exist in almost all datasets and it is essential to handle them properly in order to construct reliable machine learning models with optimal statistical power.

**KNNimputer** (K-Nearest Neighbors imputation method) is a scikit-learn class used to fill out or predict the missing values in a dataset. It is a more useful method which works on the basic approach of the KNN algorithm rather than the naive approach of filling all the values with mean or the median. In this approach, we specify a distance from the missing values which is also known as the K parameter. The missing value will be predicted in reference to the mean of the neighbours.

It is implemented by the **KNNimputer()**method which contains the following arguments:

**n\_neighbors:**number of data points to include closer to the missing value.  
**metric:**the distance metric to be used for searching.  
values – {nan\_euclidean. callable} by default – nan\_euclidean  
**weights:**to determine on what basis should the neighboring values be treated  
values -{uniform , distance, callable} by default- uniform.

To see this imputer in action, we will import it from Scikit-Learn’s impute package –

from sklearn.impute import KNNImputer

One thing to note here is that the KNN Imputer does not recognize text data values. It will generate errors if we do not change these values to numerical values.

**Usage of KNN**

The KNN algorithm can compete with the most accurate models because it makes highly accurate predictions. Therefore, you can use the KNN algorithm for applications that require high accuracy but that do not require a human-readable model.

The quality of the predictions depends on the distance measure. Therefore, the KNN algorithm is suitable for applications for which sufficient domain knowledge is available. This knowledge supports the selection of an appropriate measure.

The KNN algorithm is a type of lazy learning, where the computation for the generation of the predictions is deferred until classification. Although this method increases the costs of computation compared to other algorithms, KNN is still the better choice for applications where predictions are not requested frequently but where accuracy is important.