KNN (K Nearest Neibour)

- K-nearest neighbors (KNN) algorithm is a type of **supervised ML algorithm** which can be used for both **classification as well as regression** predictive problems. However, it is mainly used for classification predictive problems in industry. The following two properties would define KNN well −
- **Lazy learning algorithm −** KNN is a lazy learning algorithm because it does not have a specialized training phase and uses all the data for training while classification.
- **Non-parametric learning algorithm −** KNN is also a non-parametric learning algorithm because it doesn’t assume anything about the underlying data

- **K-NN Classification :-** The output is in the form of class 0 or 1. An object is classified by a plurality(More than One) vote of its neighbors, with the object being assigned to the class most common among its k nearest neighbors (k is a positive integer, typically small). If k = 1, then the object is simply assigned to the class of that single nearest neighbor.
- **K-NN Regression:-** The output is the property value for the object. This value is the average of the values of k nearest neighbors

- K-NN is a type of **instance-based learning, or lazy learning,** where the function is only approximated locally and all computation is deferred until classification.
- Both for classification and regression, a useful technique can be to assign weights to the contributions of the neighbors, so that the nearer neighbors contribute more to the average than the more distant ones.
- The neighbors are taken from a set of objects for which the class (for k-NN classification) or the object property value (for k-NN regression) is known. This can be thought of as the training set for the algorithm, though no explicit training step is required.
- A peculiarity of the k-NN algorithm is that it is sensitive to the local structure of the data.

- The training examples are vectors in a multidimensional feature space, each with a class label. The training phase of the algorithm consists only of storing the feature vectors and class labels of the training samples.
- In the classification phase, k is a user-defined constant, and an unlabeled vector (a query or test point) is classified by assigning the label which is most frequent among the k training samples nearest to that query point.
- A commonly used distance metric for continuous variables is Euclidean distance. For discrete variables, such as for text classification, another metric can be used, such as the overlap metric (or Hamming distance).

Parameter selection.

- The best choice of k depends upon the data; generally, larger values of k reduces effect of the noise on the classification,but make boundaries between classes less distinct. A good k can be selected by various heuristic techniques (see hyperparameter optimization). The special case where the class is predicted to be the class of the closest training sample (i.e. when k = 1) is called the nearest neighbor algorithm.
- The accuracy of the k-NN algorithm can be severely degraded by the presence of noisy or irrelevant features, or if the feature scales are not consistent with their importance. Much research effort has been put into selecting or scaling features to improve classification. A particularly popular approach is the use of evolutionary algorithms to optimize feature scaling.Another popular approach is to scale features by the mutual information of the training data with the training classes.
- In binary (two class) classification problems, it is helpful to choose k to be an odd number as this avoids tied votes. One popular way of choosing the empirically optimal k in this setting is via bootstrap method.