KNN Algorithm

**Introduction**

K-NN popularly known as a lazy learner, is a supervised nonparametric method used for both classification and regression (also used for imputing missing values and resampling datasets). It uses proximity (because it assumes that similar things exist in proximity) to make prediction about the grouping of an individual data point. It is mostly used for classification problem.

\* **K** represents the number of nearest neighbours the classifier will use to make its prediction

\* **Lazy leaner** because it memorizes the training instances.

\* **Nonparametric method** i.e., it doesn’t make strong assumption about the relationship between the dependent and independent variable or the distribution

\* **For classification** **problems**, a class label is assigned based on a majority vote—i.e., the label that is most frequently represented around a given data point is used, **for regression problems** the average of the k nearest neighbours is taken to make a prediction about a classification.

\* **K-NN** 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. For example, if we have a dataset of tomatoes and bananas. KNN will store similar measures like shape and colour. When a new object comes it will check its similarity with the colour (red or yellow) and shape.

**How are the distance between the data points are calculated?**

In order to determine which data points are closest to a given query point, the distance between the query point and the other data points will need to be calculated. These distance metrics help to form decision boundaries, which partitions query points into different regions. Some of the common distance includes:

1. **Euclidean distance (p=2):** This is the most used distance measure, and it is limited to real-valued vectors. Using the below formula, it measures a straight line between the query point and the other point being measured.
2. **Manhattan distance (p=1)**: This is also another popular distance metric, which measures the absolute value between two points. It is also referred to as taxicab distance or city block distance as it is commonly visualized with a grid, illustrating how one might navigate from one address to another via city streets.
3. **Minkowski distance**: This distance measure is the generalized form of Euclidean and Manhattan distance metrics. The parameter, p, in the formula below, allows for the creation of other distance metrics. Euclidean distance is represented by this formula when p is equal to two, and Manhattan distance is denoted with p equal to one.

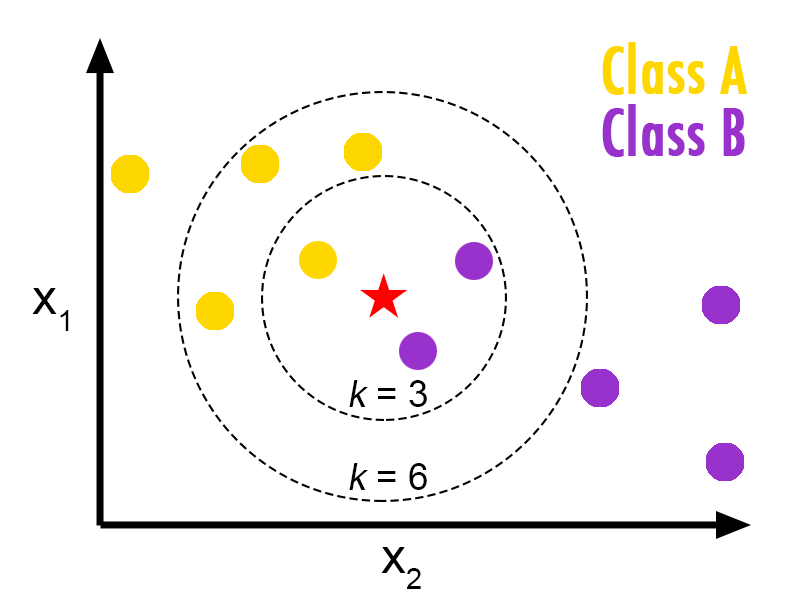
**Minkowski distance =**

1. **Hamming distance:** This technique is used typically used with Boolean or string vectors, identifying the points where the vectors do not match. As a result, it has also been referred to as the overlap metric. This can be represented with the following formula:

**How do we choose the best k?**

* it is recommended to have an odd number for k to avoid ties in classification
* Make a plot of different accuracy based on the value of k
* Using error curves

**The algorithm**

****

* Start with a dataset with known categories
* Select the number of K neighbours
* Calculate the distance of K numbers of neighbours
* Take the K nearest neighbours with reference to the distance
* Count the number of data points in each K nearest neighbours and get the labels
* Take the label with the maximum vote if it’s classification
* Or the mean of the vote if it’s a regression problem

**Required Data Preparation**:

1. Data Scaling: To locate the data point in multidimensional feature space, it would be helpful if all features are on the same scale. Hence normalization or standardization of data will help.
2. Dimensionality Reduction: KNN may not work well if there are too many features. Hence dimensionality reduction techniques like feature selection, principal component analysis can be implemented
3. Missing value treatment: If out of M features one feature data is missing for a particular example in the training set, then we cannot locate or calculate distance from that point. Therefore, deleting that row or imputation is required

**Advantages**

* It is intuitive and simple to implement
* It has assumption i.e., non-parametric
* No training steps as it simply stores the training data
* Variety of distance criteria to choose from
* Easy to implement for multi-class problem

**Disadvantage**

* Always needs to determine the value of K which may be complex some time.
* Slow algorithm
* The computation cost is high because of calculating the distance between the data points for all the training samples
* Curse of dimensionality: cannot work with high dimensional data, thus, the number of attributes has to be small
* Imbalanced data causes problems: E.g., if we consider two classes A and B and the majority of the training data is labelled as A, then the model will surely give a lot of preference to A. This might result in getting the less common class B wrongly classified.