**K-nearest neighbor classifier**

Data input: data with label and data without label. Data with label is divided to train and test model. Data without label is what user wants.

Giving an option of choosing “Normalization” or “Not Normalization”. Default to be “Normalized”. To normalize or not is a judgement of the user himself, and need to know the background of the problem to make a decision. One example could be: if the range of values of one feature is obviously larger than other features, and this property of feature is very important and meaningful in determining the classes, then do not normalize. Otherwise, normalized. A safe bet is to normalized.

Being a non-parametric method, it is often successful in classification situations where the decision boundary is very irregular.

A case is classified by a majority vote of its neighbors, with the case being assigned to the class most common amongst its K nearest neighbors measured by a distance function.

Choose an odd K for 2 classes; k must not be a multiple of the number of classes

Drawback: the complexity in searching the nearest neighbors for each sample

Our focus will be primarily on how does the algorithm work and how does the input parameter effect the output/prediction.

KNN algorithm fairs across all parameters of considerations. It is commonly used for its easy of interpretation and low calculation time.

Distance functions:

Euclidean

Manhattan

Minkowski

It should also be noted that all three distance measures are only valid for continuous variables. In the instance of categorical variables the Hamming distance must be used. It also brings up the issue of standardization of the numerical variables between 0 and 1 when there is a mixture of numerical and categorical variables in the dataset.

**Hamming Distance**

Choosing the optimal value for K is best done by first inspecting the data. In general, a large K value is more precise as it reduces the overall noise but there is no guarantee. Cross-validation is another way to retrospectively determine a good K value by using an independent dataset to validate the K value. Historically, the optimal K for most datasets has been between 3-10. That produces much better results than 1NN.

Normalization of features:

Find the minimum and maximum, offset by the minimum and normalize.

Choose of K:

K: 1 to 10, cross validation to get the best K.