# MLNC CW

## Kth Nearest Neighbour

In the case of a k-th nearest neighbour classifier, training consists in computing the optimum value of K i.e. the value of K which yields the least number of misclassifications. The distance between points in the dataset was computing using Euclidean Distance.

The data was normalised so that all dimensions are in the range 0 – 1 thus assuming the same weight for each dimension. This was done in order to reduce the differences between dimension ranges. Such large differences have a great impact when computing the distance between points resulting in a bias towards

### Equally Sized Training/Data Sets

Below is the output class distribution sing upper half of the dataset (284 points) for training and lower half (285 points) for testing.

|  |  |  |
| --- | --- | --- |
|  | Training Set | Testing Set |
| Class 0 | 139 | 218 |
| Class 1 | 145 | 67 |
| Total | 284 | 285 |

#### Testing

Below are is performance and the computed optimum value of K:

|  |  |
| --- | --- |
| Optimum K | Testing Error Rate |
| 3 | **92.25%** |

### “leave-one-data-point-out” Cross Validation

#### Testing

Below are is performance and the computed optimum value of K:

## Generative Classification Approach (one Gaussian per class)

In the generative classification approach, training consists in computing the optimum parameters to model the distribution of the data. Assuming one Gaussian per class this resolves into finding the optimal values for the mean of each class and the covariance matrix for the entire dataset.

The training function devised also returns the prior probability for each class. This is used in Bayes theorem to infer the probability of the testing point belonging to each class k.