**Part 2**

**(b)**

**Problem statement –**

There are a range of different techniques you can investigate that could potentially improve the performance of either the distance-weighted k-NN and the basic k-NN from part 1. For example, you could look at the performance of these algorithms for different hyper-parameter settings (such as the value of k).

You should describe and investigate a comprehensive range of techniques that could potentially improve the accuracy of your basic k-NN and distance-weighted k-NN. Your report should document fully the different techniques, provide a justification for selecting and investigating these techniques and present the resulting accuracy. Please note that when you incorporate additional techniques you should implement these techniques in your own code using core Python or NumPy (rather than relying on imported functionality or high level functions).

1. Min Max Scaling –

Normalize all dimensions independently

Using Range Normalization minimum and maximum value of a feature can be identified

New value = (original value – minValue) / (maxValue - minValue)

1. Ignore all the irrelevant feature vectors from the dataset during compute
2. Selection of an appropriate value of k is very important

* Too small value of k can make the algorithm susceptible to noise and can overfit
* Larger value can reduce the noise but make classes less distinct

Assess the value of k based on few selected values.