Practical Machine Learning Assignment 1

Student number: R00183157
Student Name: Karthik Murugadoss

Part1- Development of basic k-NN algorithm

File Name: *Part1 ML assignment.ipynb*

The above ipython notebook contains the python code for basic k-NN algorithm.

Pre-processing of Data:

- The data was checked for missing values and it had no missing values.
- The data did not require normalisation because the range of data was almost in the same range

Training Data instances : 4000
Test Data instances; 1000
Features:10
There are two functions used in the code:

1) getdata(filename)

   reads the data from the file and returns the features and class separately as two NumPy arrays

2) calculate_distance(2Darray, 1Darray) –

   Takes two arguments which takes the entire training data array and one single test instance.
   Returns a vector of distances between the single test instance and all the training instances.

The accuracy for the basic k-NN algorithm for k=1 is 89.5

Part2- Investigating k-NN variants and hyper-parameters

a) A distance weighted k-NN was developed and implemented. The implementation is available in the file- *Part2 ML assignment.ipynb*

The performance achieved for the distance-weighted variant of k-NN for K=10 is **92.7** the distance metric used was **Euclidean distance** and the n value used was **2** in calculating w_i

$$f(\mathbf{x}_q) := \frac{\sum_{i=1}^{k} w_i f(\mathbf{x}_i)}{\sum_{i=1}^{k} w_i}$$

Where

$$w_i = \frac{1}{d(\mathbf{x}_q, \mathbf{x}_i)^2}$$

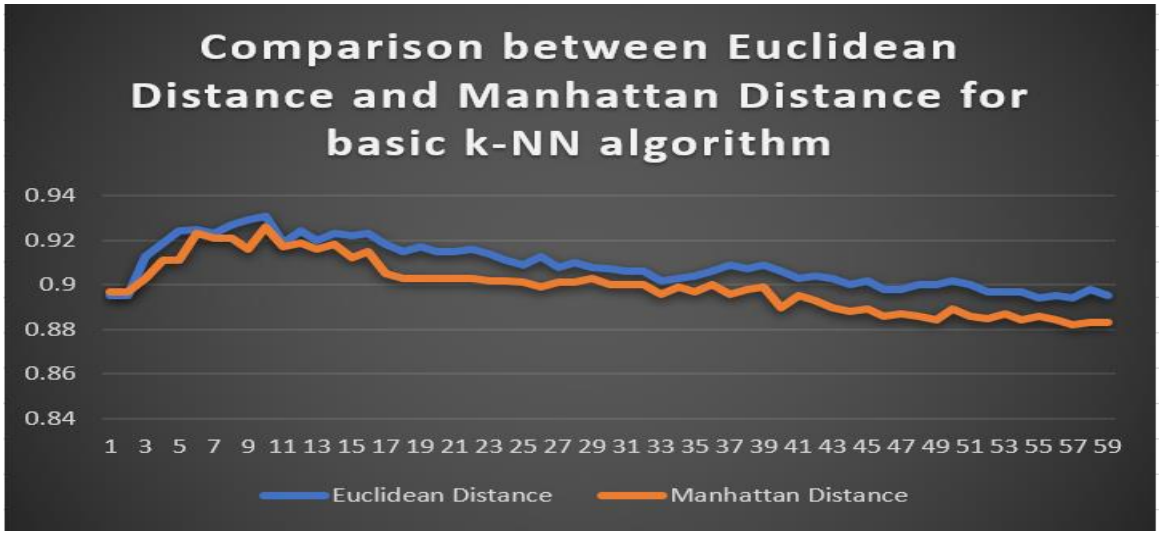
The highlighted part denotes **n**

b) Investigating on improving the performance of basic k-NN and distance-weighted k-NN.

Basic k-NN investigation:
Changing Hyper parameter k and distance metric for Basic k-NN and seeing how the performance varies to pick the right k value and distance metric for this particular data set.

The below graph explains how the accuracy varied with different values of **K** for both Euclidean distance and Manhattan distance

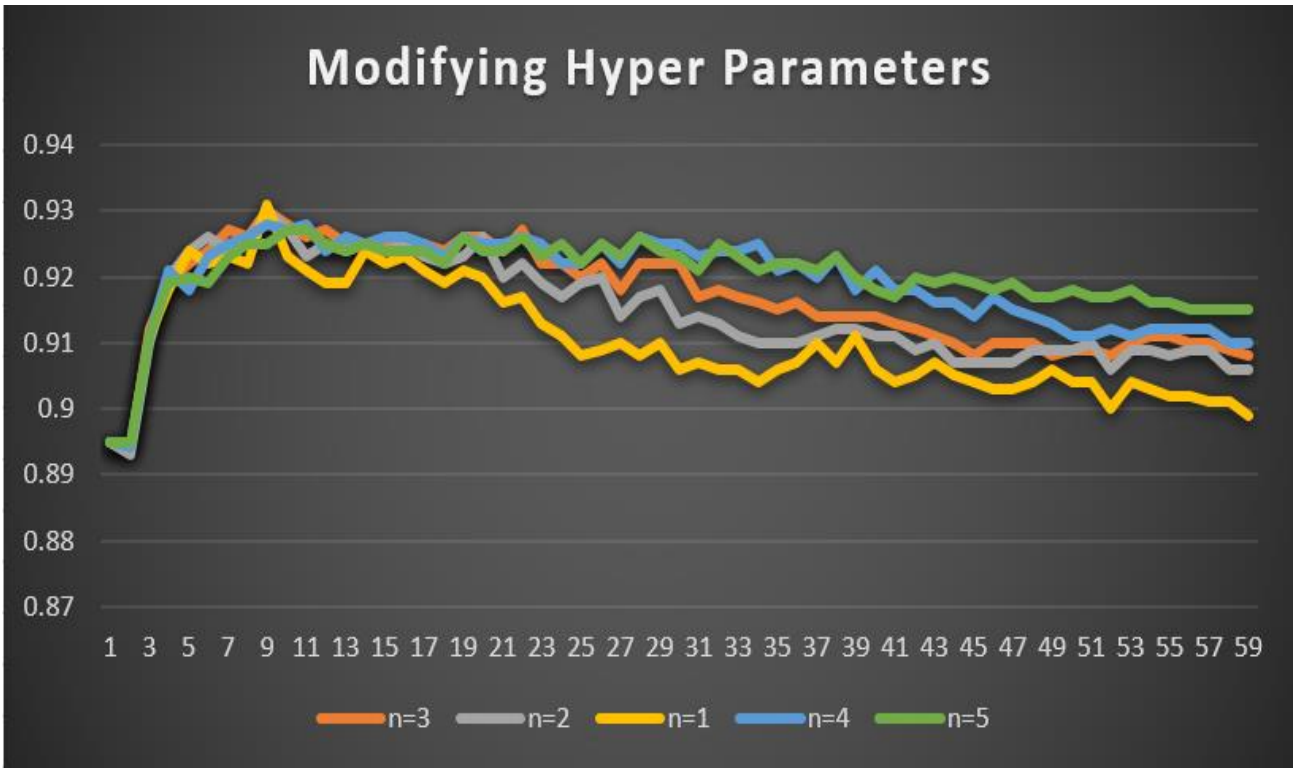
From the below line chart it can be inferred that the Euclidean distance works well for this data set than Manhattan and the approximate value for K would be 10 where the accuracy is at it peak after which it declines.



Distance weighted k-NN investigation:

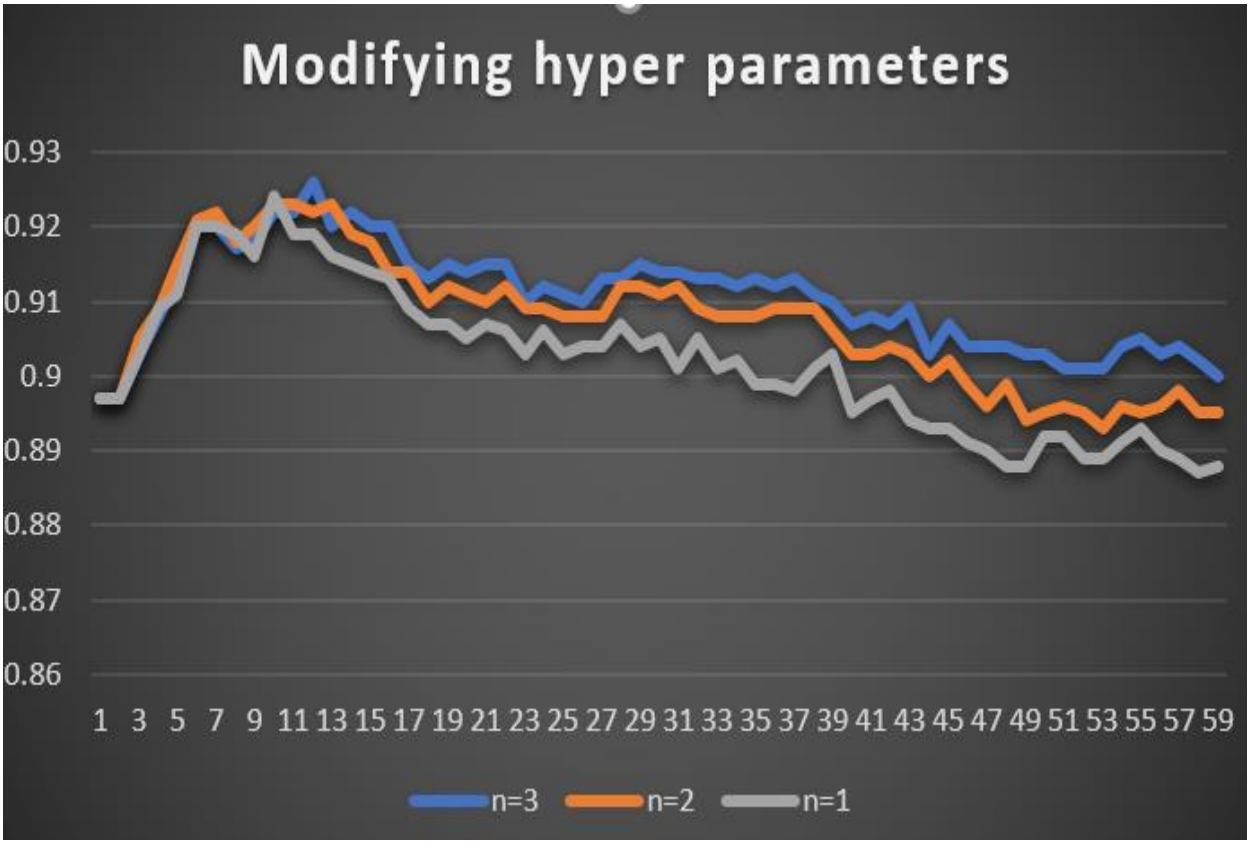
For the distance weighted k-NN there are two hyper parameters k and n

For Euclidean distance metric the below graph shows how the hyper parameters change the accuracy



It could be seen that as n value increases the accuracy for a wide range of K values also increases.

Similarly, For Manhattan distance metric the below graph shows how the hyper parameters change the accuracy

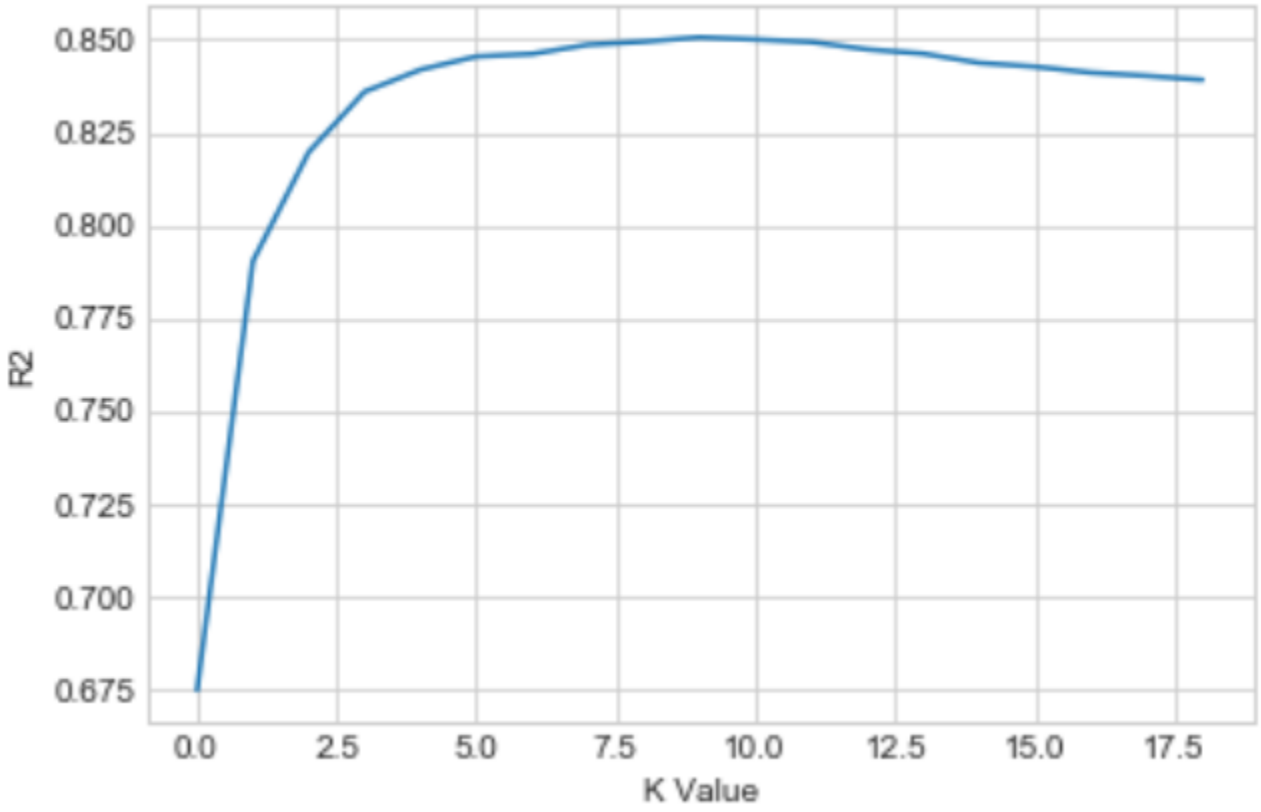


From all the above investigation we can decide that, to get the best accuracy from distance-weighted k-NN we can chose Euclidean distance and n value to be 2 or 3 and the k value to be 10 for which the accuracy comes to 92.8

All the supporting data points for the graph are present in the excel workbook - Results.xlsx

Part3-Developing k-NN for Regression Problems

a) The distance weighted k-NN for regression is implemented and is available in the ipython notebook Part3 ML Assignment.



The above graph shows the R2 value for every K value and from that we can take the k value to be 10 for which the R2 is maximum of 0.85065

b) The problem with equally weighing all the features is that it will create bias towards features with higher values. Which means that one feature might contribute more to the predicted value.

   This can be avoided by feature selection and identifying which feature contributes more to the output