**1st Objective One (Partitioning Clustering)**

**Analyzing data with given classification**

Since the data set is given with a classification field, at first, we can analyze the data with some selected columns in order to get a basic idea about the data and the given classification.

Graphical user interface, application

Description automatically generated

Four types of vehicles are shown in different colors. When observing these graphs, specially Graph1 and Graph4, we can see that data are squeezed towards a particular axis. Also, there are significant differences in the scale of some of the features. For example, scale difference between, “Sc.Var.Maxis” and “Max.L.Ra” is huge which impacts to any data algorithm badly. Also, there are clear set of outliers too.

These are clear indication to normalize or scale data before applying any kind of algorithm to the given data set.

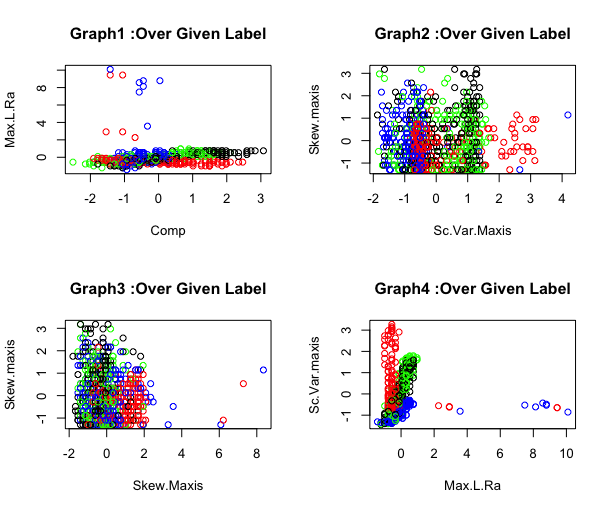
Even though, the provided data set is labeled one, after analyzing above graphs, there is no sign of identifying clear data clusters by comparing only two features at a time. Also, it is hard to say that these data set can be labeled into 4 classes, because there is huge overlapping of data points.

**program-01 –** source code for above graph

Text

Description automatically generated

Let’s draw the same four graphs after applying normalization to the given data set.



With the above graph, we can see that the data set is properly normalized into an acceptable scale. So, therefore, before applying any algorithm to the given data set, it is important to scale or normalized the data set. But there are still outliers to be quantile. Let’s analyze it later.

**program-02** – source code for above graph

Graphical user interface, text, application

Description automatically generated

**Outlier Detection**

First, let’s run the summary function on the given data set before doing any processing on the data set.

The following is the five-number summary output.

A picture containing text, receipt

Description automatically generated

Circ Feature Analysis

Q1 = 40.00

Q3 = 49.00

IQR (Inter Quantile Range) = Q3 – Q1 = 49.0 – 40.00 = 9

Q1 - 1.5(IQR) = 40.00 – 1.5 x 9 = 26.5

Q3 + 1.5(IQR) = 49.00 + 1.5x9 = 62.5

The minimum value of ‘Circ’ feature is 33.00 which is bigger than then 26.5. And also, the maximum value of ‘Circ’ feature is 59.00 which is below the 62.5. So, there are no outliers for ‘Circ’ feature values for entire data set.

Sc.Var.maxis Feature Analysis

Q1 = 318.2

Q3 = 587.0

IQR (Inter Quantile Range) = Q3 – Q1 = 587.0 – 318.2 = 268.8

Q1 - 1.5(IQR) = 587.00 – (1.5 x 268.8) = 183.80

Q3 + 1.5(IQR) = 587.00 + (1.5 x 268.8) = 990.20

The maximum value of ‘Sc.Var.maxis’ is 1018.00 which is bigger than 990.20 which is an outlier.

As you can see in highlighted, the values for ‘Circ’ field is almost equally distributed. There are not outliers. The ‘Min’ and ‘Max’ values are reasonably close to the ‘Median’ and ‘Mean’ values. This is a good quality of a feature of a data set when applying an algorithm.

But ‘Sc. Var.maxis’ feature has significant outliers. It’s ‘Max’ values are significantly different. When an algorithm compares two data points of ‘Sc.Var.maxis’ features, it will result a huge difference which negatively impact to the final result.

We will further check for outliers by generating boxplots for each of the feature under given classification.

|  |  |
| --- | --- |
| Chart, scatter chart  Description automatically generated | Chart  Description automatically generated |
| Chart  Description automatically generated | Chart  Description automatically generated |
| Outliers | |

If you see the above four graphs for each given classification, we can see outliers for many of the given features. Also, there are some features which has no a single outlier. So therefore, it is important to quantile the dataset before applying any algorithm to the given data set in order to get more accurate result.

**program-04** – source code for the above graph

Graphical user interface, text, application, email

Description automatically generated

**Removing Outliers**

Let’s look at the five-number summary out for the whole data set after removing outliers.

A picture containing text, receipt

Description automatically generated

Now you can see that the maximum value of ‘sc\_var\_maxis\_2’ (‘Sc. Var.maxis’) is below the upper outlier limit. So, this indicates, outliers of the data set is properly adjusted.

Further, the following graphs were generated again after removing outliers.

|  |  |
| --- | --- |
| Chart  Description automatically generated | Chart  Description automatically generated |
| Chart  Description automatically generated | Chart, box and whisker chart  Description automatically generated |

**program-04** – source code to remove outliers.

Graphical user interface, text, application

Description automatically generated

**Normalizing dataset**

In our program above, ‘vehicle\_quantiled’ field contains the data set after removing outliers. Since, we already identified that our data set is needed to normalize, let’s normalized the data set before degerming the number of clusters required. We can R’s scale function for this.

Text, letter

Description automatically generated

**Determining Number of Clusters**

We use **NbClust** package in order to determine the most appropriate number of clusters for the given data set. We will use two distance measures with **NbClust** in order to find out the number of clusters for K-Mean algorithm. The results of both analysis are given bellow.

Graphical user interface

Description automatically generated with low confidence

Euclidean distance measure

Chart, line chart, histogram

Description automatically generated

Text, letter

Description automatically generated

Manhattan distance measure

Chart, histogram

Description automatically generated

Text

Description automatically generated

Both the analysis suggests that the most appropriate number of clusters in order to perform the K-Mean algorithm for the given data set is 2.

**Perform K-Mean**

Even though the data set is given with 4 labels, according to the analysis, we should be able to identify two clear clusters, not 4 clusters. Let’s apply the K-Means for both scenario and analyze the output.

K-means with 4 number of clusters

Graphical user interface, text, application

Description automatically generated

K-means with 2 number of clusters

The k-means summary

Text

Description automatically generated with medium confidence

Let’s compare the cluster results with original data set’s classification field.

Graphical user interface, text, application

Description automatically generated

I did relabel ‘bus’, ‘van’, ‘opel’ and ‘saab’ as 1,2 3,4 respectively in order to compare with the cluster results with original data set. Here, we can see that, K-means has successfully identified two clusters even though original data set is classified into 4. One cluster with all the ‘van’s and the other cluster with other 3 types. All ‘van’s are properly identified as a single cluster and it has been unable to distinguish all other three types. There are so many overlapping when compare with original classification.

Let’s plot same two features after applying the K-means using scaled data set. As you can see bellow, two clear clusters can be identified.

Chart

Description automatically generated

**2nd Objective (MLP)**

**Determining input vectors**

Since the given data set is time series single feature value, in order to perform MLP-NN, we need to create different input vectors. We are going to create 10 new input vectors by using given date ordered GBP values.

First 8 vectors will be created using lag() function. This will create 8 input vectors with last 8 days GBP values. Among those 8 fields, the latest day values will be on left side in the data set.

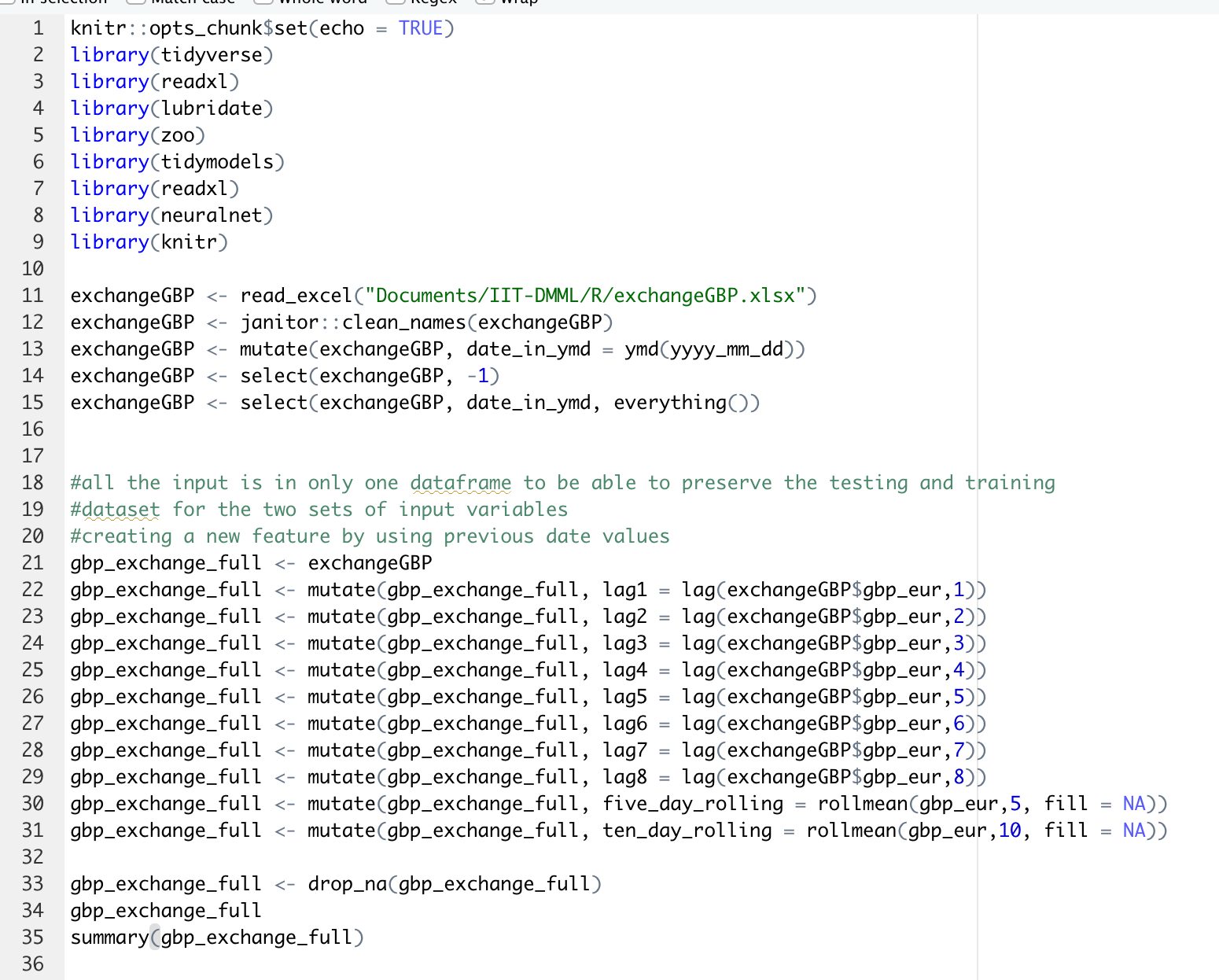
Last 2 vectors are created by getting the mean values of nearest 5 rows and 10 rows respectively for a given GPB value.

So, the final input data set will be as follows.

Text

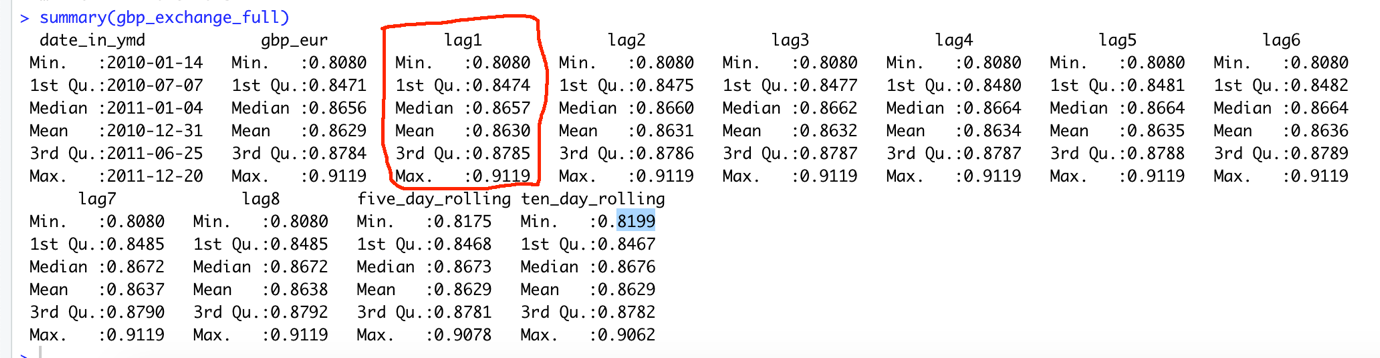
Description automatically generated

The following peace of code shows the imported libraries and how initial set of vectors were created.



**Outlier Detection**

Let’s see the summary of the data set before proceeding in order to see for any outliers.



Lag1 Outlier Analysis

Q1 = 0.8474

Q3 = 0.8785

IQR (Inter Quantile Range) = Q3 – Q1 = 0.8785 – 0.8474 = 0.0311

Q1 - 1.5(IQR) = 0.8474– (1.5 x 0.0311) = 0.80075

Q3 + 1.5(IQR) = 0.8785+ (1.5 x 0.0311) = 0.92515

As per above analysis, both ‘Min’ and ‘Max’ values are within the range for ‘lag1’ summary analysis. Also, the ‘Min’ and ‘Max’ values are same for the first 8 vectors which are within 0 and 1 range which is acceptable for ‘logistic’ activation function for MLP-ANN.

Let’s further see a graphical view for ‘lag1’ data set. The following graph shows the variation of GBP for whole data set for a single input vector.

Graphical user interface, chart, line chart

Description automatically generated

The same graph was generated for all the input vectors for whole data set as follows.

Chart

Description automatically generated with low confidence

Let’s see the same graph after applying the normalization for the data set. The following graph shows the result of ‘lag1’ after applying normalization.

Chart, line chart

Description automatically generated

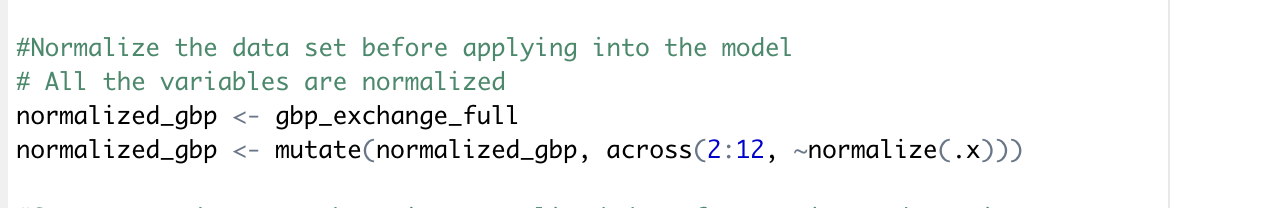
As you can see above, there is no significant different of the graph even after normalizing the data set. Based on above all analysis, we can decide that the normalization the given data set is not necessary before applying MLP-ANN model for the given specific data set.

If we use ‘logistic’ activation function, having all the values in between 0 and 1 will result more accurate output. When analyzing the given data set, there is not high requirement to perform normalization because, almost all the values are in between 0 and 1. And also there are no significant outliers. But for better accuracy, we can normalize the data set before applying MLP-ANN, because it does not impact the final output negatively. If there is an impact, it always be a positive impact to the final output.

The pieces of code to generate graph and normalization is as follows.

Text

Description automatically generated



Text

Description automatically generated

Testing with two layers and different combination of nodes and capture the results of all performs operations.

Get the lag values for last 10 days

**Network Architecture**

We are going to use two hidden layers with 50 different combination of nodes by using two activation function, ‘logistic’ and ‘tanh. After getting summary details of all these, best network structure will be selected.

The following tables shows the best fit out of 4 different network structure results. Each execution will output 50 results. Out of those 50, the best results are highlighted bellow table.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Structure | Input Vector | Hidden Layers | Nodes Layer1 | Nodes Layer2 | Activation Function | RMSE | MAE | MAPE |
| 1 | Lag1 | 2 | 2 | 4 | logistic | 0.00449 | 0.00344 | **0.397** |
| 2 | Lag1 + lag2 | 2 | 4 | 2 | logistic | 0.000119 | 0.0000948 | **0.0109** |
| 3 | Lag1 | 2 | 7 | 4 | tanh | 0.00448 | 0.00343 | 0.396 |
| 4 | Lag1 + lag2 | 2 | 8 | 4 | tanh | 0.000210 | 0.000156 | 0.0179 |

Structure 1 Result

Table

Description automatically generated

Structure 2 Result

Table

Description automatically generated

Structure 3 Result

Table

Description automatically generated

Structure 4 Result

Table

Description automatically generated

If we consider structure 1 and structure3, we can notice that error values are little bit higher than the structure 2 and structure4. So, having more than one input vector will produced much better results rather than having a single input vector.

As per the above analysis, structure 2 will give the best result, since it has the lowest error factor values.

Model with the lowest RMSE will be the best.

Need to specify how many hidden layers are going to be used and how many nodes or neurons are going to be under one layer. In our case, we are going to use two layers with different combination of nodes.

**3rd Obojective (SVR Model)**

**Determining input vectors**

Same steps followed to create and determine input vectors.