Report on Partitioning Clustering and Energy Forecasting

Name: Hasanur Rahman Mohammad
Student ID: w1780941

Module Code: 5DATA002W
Tutor: Mahmoud Aldraimli
Seminar Group: 5CS01

Contents

1 Partitioning Clustering

1.1 Pre-Processing the data

For this task we were given a vehicle.xmls file containing **846** samples, with **19** different attributes including the **'Class'**. However, as the goal is to perform k-means clustering on the data, an unsupervised learning algorithm, it is required to remove the **'Class'** column as the model will classify the data on its own. I also removed the 'Sample' column as it will affect the next pre-processing tasks, scaling and outlier removal.

When it comes to the order, I chose to remove the outliers first as they seemed to negatively affect the clustering results if I scaled the data before removing them. To find the outliers I found the **z-score** for each of the samples and then removed any samples with a **z-score** than **3** and less than **-3**.

1.2 Finding the best k using: Nblust, Elbow method, Gap statistics and sillhoutte methods

1.2.1 Nblust

As shown below, Nbclust says the best number of clusters is 3. Considering the original number of classes is 4 I believe that this is a good result.

```
* Among all indices:
* 6 proposed 2 as the best number of clusters
* 12 proposed 3 as the best number of clusters
* 1 proposed 6 as the best number of clusters
* 1 proposed 8 as the best number of clusters
* 1 proposed 11 as the best number of clusters
* 1 proposed 12 as the best number of clusters
* 2 proposed 15 as the best number of clusters

                   ***** Conclusion *****

* According to the majority rule, the best number of clusters is  3
```

1.2.2 Elbow Method

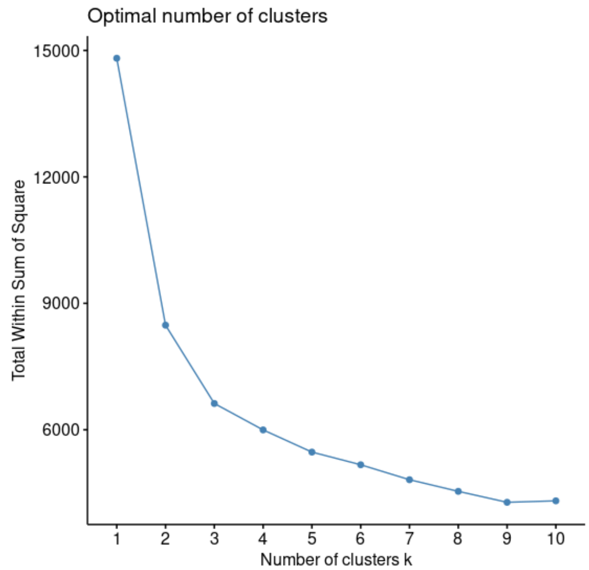

Figure 1: Elbow method plot

The Elbow method uses the **WCSS(within-cluster sums of squares)** which measueres how close data points are in respect of their cluster centers. Based on the plot above, the reccomended number of clusters is **3** as that is where the results begin to flatten out slowly indicating that increasing the clusters anymore will not result in any increase in performance.

1.2.3 Gap Statistics

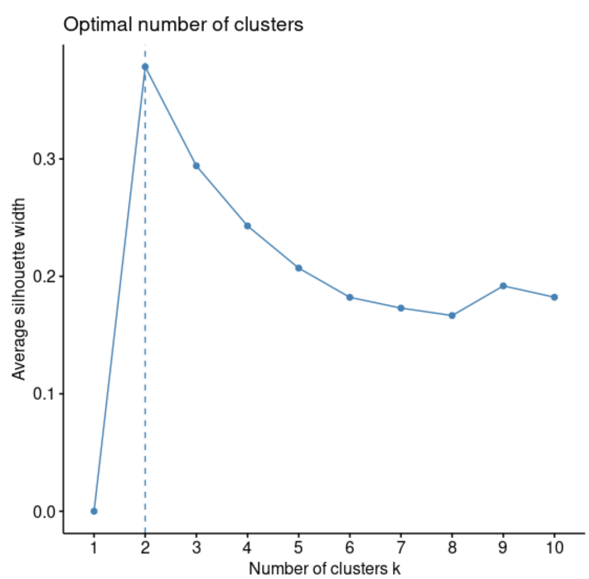

Figure 2: Gap statistics plot

The Gap statistics also uses the **WCSS** to calculate the best number of clusters to use. However, the reccomended number of clusters in this case is **2**, knowing that the orignal data set has **4** possible classes, we can conclude that this result is worse than what we got with the elbow method which was **3**.

1.2.4 Sillhoutte Method

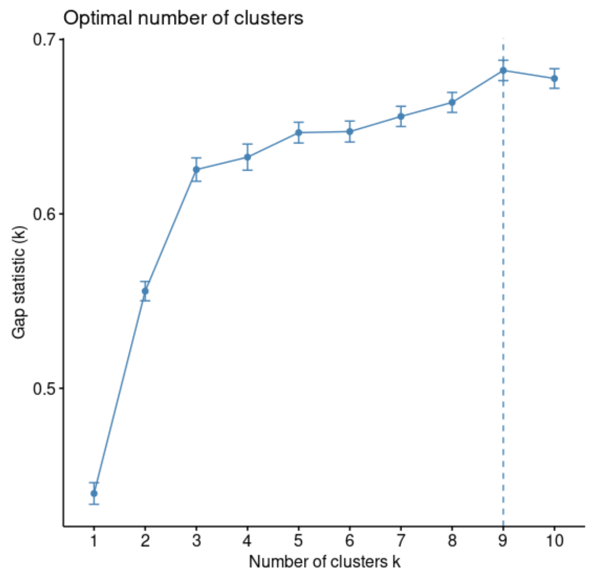

Figure 3: Sillhoutte method plot

The sillhoutte plot shows how similar a data point is to its own cluster using the **sillhuotte score**, this is a value that ranges from -1 and 1, with values closer to -1 meaning the data point should be in another cluster and the closer the value is to 1 meaning the current cluster is a good fit for the data point This is where things get interesting, based on the plot above **9** is the reccomended number of clusters. This is significantly higher than any of the other results from the other evaluation methods, I made to sure to run the model several times checking if there were errors with the code, but it gave **9** as the ouput everytime. This is by far the worst result as the orignal data set has **4** classes

However as shown later in the report, after running the evaluation tools for the data that had **PCA** done on it. The results for the sillhoutte plot were a lot more controlled and matched the other evaluation methods as well. This led me to believe that having a data set that is too multi-demensional led to an extreme result for the sillhoutte plot.

1.3 K-means Clustering investigation

Using the results from the evaluation methods, I decided to go for **k=3**, as both **Nbclust** and the **Elbow Method** gave a result of the best **k** being **3**. Below you can see the plot made from the clustering, without looking at the output data you can see a clear distinction between the clusters where there is no overlapping

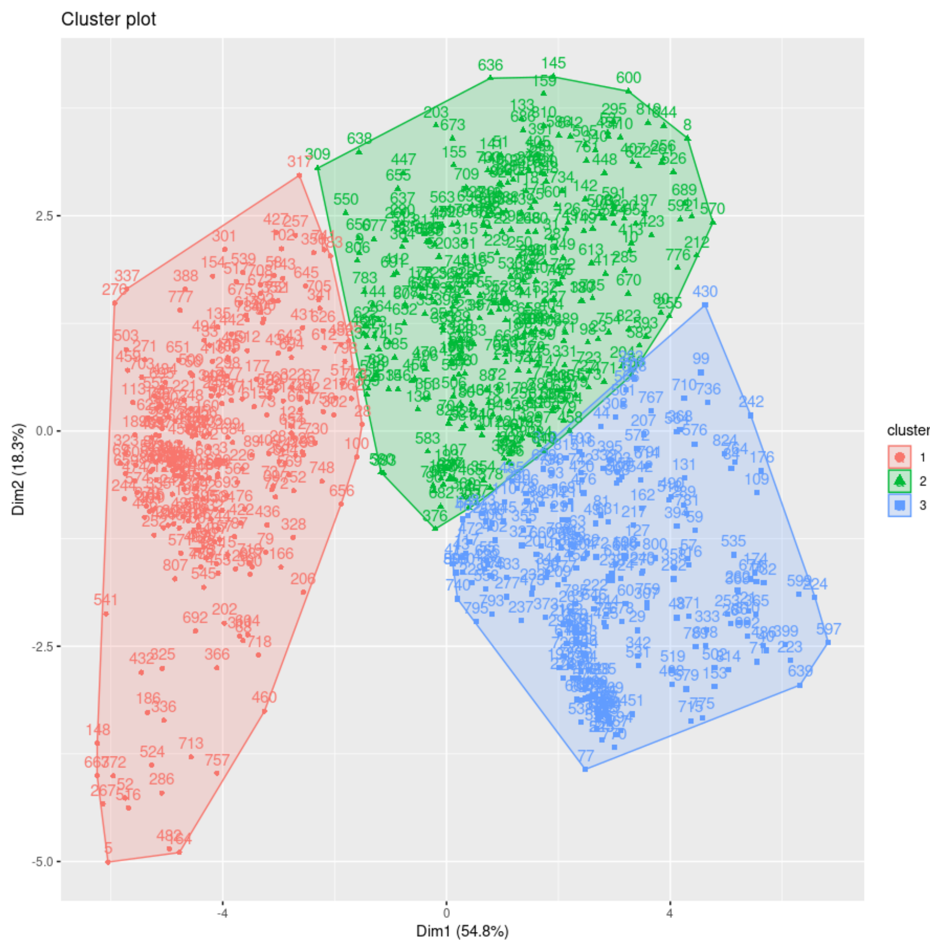

Figure 4: Clustering plot

Below are the kmeans output for the clustering attempt with **k=3**. First of all you can see that the sizes of each cluster is evenly distributed which means there isn't a cluster that has too many or too little data samples.

```
K-means clustering with 3 clusters of sizes 256, 331, 237

> kmeans_data$centers
        Comp       Circ     D.Circ       Rad.Ra Pr.Axis.Ra   Max.L.Ra    Scat.Ra
1  1.1672551  1.1913560  1.2226654  1.061855474  0.2398399  0.6675158  1.3141094
2 -0.2324797 -0.5226347 -0.2851558 -0.002041173  0.3625937 -0.1440161 -0.4446806
3 -0.9361458 -0.5569412 -0.9224294 -1.144132376 -0.7654747 -0.5198934 -0.7984081
       Elong Pr.Axis.Rect Max.L.Rect Sc.Var.Maxis Sc.Var.maxis     Ra.Gyr  Skew.Maxis
1 -1.2220251    1.3199740  1.1102132    1.2689258    1.3291991  1.0980640 -0.08461041
2  0.3064563   -0.4736786 -0.4874626   -0.3936680   -0.4533218 -0.5482611 -0.66286263
3  0.8919890   -0.7642435 -0.5184154   -0.8208477   -0.8026391 -0.4203797  1.01716369
   Skew.maxis  Kurt.maxis  Kurt.Maxis    Holl.Ra
1  0.16667482  0.27331007  0.01515673  0.2044549
2 -0.06083852 -0.01874875  0.75780956  0.6641968
3 -0.09506837 -0.26903603 -1.07474720 -1.1484793
```

```

> kmeans_data$cluster
  [1] 2 2 1 2 1 2 2 2 2 2 2 2 2 2 1 3 2 1 1 3 3 2 2 1 2 3 1 1 3 2 2 2 1 2 2 3 1 3 1 3 3
 [42] 2 3 3 3 3 2 3 2 1 2 1 2 2 3 1 3 1 3 3 3 2 3 3 1 2 1 1 1 2 3 2 1 2 3 1 3 3 1 2 3 2
 [83] 2 3 2 3 1 2 1 2 3 1 3 3 1 3 2 2 3 1 1 1 3 3 2 2 2 3 3 3 2 1 1 3 2 3 3 2 2 2 3 2 2
[124] 1 1 2 3 1 3 2 3 2 2 3 1 3 2 1 2 2 2 2 1 2 2 1 2 1 2 3 2 2 3 1 2 2 1 1 2 1 3 3 1 1
[165] 2 1 2 2 2 2 2 3 1 3 2 3 1 2 2 2 1 2 1 2 2 1 2 3 1 3 3 3 2 2 1 1 2 2 2 3 3 1 2 2 2
[206] 1 3 2 3 1 3 2 1 3 1 3 3 2 1 2 1 3 3 3 3 1 2 3 2 3 1 3 2 2 3 1 3 3 2 2 1 3 3 1 3 2
[247] 2 1 2 2 1 1 3 2 2 2 1 3 3 2 2 3 3 2 2 2 1 2 3 3 1 2 2 3 3 1 3 2 2 3 1 3 3 2 2 1 2
[288] 1 3 2 2 1 2 2 2 3 2 1 1 1 1 1 2 2 1 3 3 3 2 3 1 1 3 1 2 3 1 3 2 2 2 1 1 3 1 1 3 1
[329] 2 2 2 3 3 1 1 1 1 2 2 2 1 3 2 3 1 2 2 1 2 1 1 1 2 2 3 1 2 3 3 2 2 2 2 2 3 1 1 3 3
[370] 1 3 1 3 1 2 2 2 2 1 3 2 2 2 2 2 2 2 1 2 1 2 1 2 3 3 2 2 2 3 3 2 3 1 2 2 3 2 3 1 2
[411] 3 2 2 1 2 1 2 1 1 3 3 1 2 3 3 2 1 1 3 3 1 1 3 1 1 1 2 2 2 2 2 1 3 3 2 1 2 2 1 2 3
[452] 1 3 3 1 1 2 2 1 1 1 3 1 1 2 2 3 1 1 2 2 3 3 1 2 3 1 1 2 3 1 1 2 1 3 3 1 1 1 3 3 1
[493] 1 1 2 2 1 3 2 1 3 3 1 3 2 2 3 2 1 2 1 1 2 3 2 1 1 3 3 2 1 2 1 1 2 2 2 2 2 3 3 2 2
[534] 1 3 3 2 3 1 2 1 3 3 1 1 2 1 2 2 2 1 2 3 2 1 2 2 3 1 1 1 1 2 3 3 3 1 1 1 2 1 3 2 1
[575] 3 3 3 2 3 2 2 2 2 2 2 2 1 2 2 1 2 2 2 3 1 3 3 2 3 2 2 3 3 1 1 3 2 3 1 2 2 1 2 3 1
[616] 3 1 3 3 2 3 2 1 1 2 1 2 2 3 2 3 1 2 1 3 2 2 2 3 2 3 2 1 2 1 3 2 2 2 2 1 2 3 1 2 1
[657] 2 2 1 3 1 3 2 2 2 3 1 2 3 2 3 1 2 2 1 3 2 3 2 2 3 2 1 1 2 2 1 1 2 3 2 1 1 1 1 2 1
[698] 2 2 1 1 2 1 2 1 2 3 1 2 3 1 1 1 2 3 3 1 1 1 2 1 2 2 1 2 3 2 3 2 1 2 3 2 2 2 3 1 3
[739] 3 3 1 3 1 1 3 2 2 1 2 3 1 1 3 2 2 1 1 1 3 1 2 1 1 3 3 1 3 1 2 3 2 1 1 2 3 2 1 1 2
[780] 2 3 2 2 1 3 2 1 3 3 1 3 2 3 3 3 2 1 1 2 3 1 2 1 1 3 2 1 3 3 2 2 1 3 3 3 2 2 2 2 2
[821] 2 1 2 3

> kmeans_data$tot.withinss
[1] 6624.09

> kmeans_data$betweenss
[1] 8189.91

Within cluster sum of squares by cluster:
[1] 2191.909 2735.763 1696.418
 (between_SS / total_SS =  55.3 %)
```