# CIS 6930: Introduction to Data Mining

# PROJECT 2: CLUSTERING

Raheen Mazgaonkar

UFID: 4714-4316

November 7, 2017

## INTRODUCTION

Clustering is an unsupervised technique of data mining in which data is grouped based on their characteristics. In this project, different clustering techniques, particularly k-means, hierarchical, density-based and graph-based clustering were studied. Methods for implementing the same in R were studied. These methods were implemented for dataset 1. Further, best possible clustering technique was analyzed for dataset 2 and the same was implemented.

## CLUSTERING METHODS

### 1) HIERARCHICAL CLUSTERNG

In hierarchical clustering algorithms, clusters are not disjoint subsets of original data. Instead, they are a set of nested clusters that are organized as a tree. For hierarchical clustering there are two strategies: agglomerative and divisive. Agglomerative hierarchical clustering is a bottom up approach in which each data point starts as its own cluster and each cluster is eventually merged to form a single cluster eventually. Divisive hierarchical clustering is a top down approach in which we start with a single cluster containing of all the data points and split the clusters recursively. A hierarchical clustering is often displayed graphically using a tree-like diagram called a dendrogram. Basic steps followed in agglomerative hierarchical clustering is as follows:

i) Compute proximity matrix.
ii) Repeat the following until one cluster is remaining
    a) Merge the closest two clusters.
    b) Update the proximity matrix to reflect the proximity between the new cluster and the original clusters.

In my implementation, hclust method of stats package was used to implement agglomerative hierarchical clustering. For this method, we need to pass the distance matrix which was calculated based on Euclidean distance. For proximity measure, Ward's method has been used. In Ward's method the proximity between two clusters is defined as the increase in the squared error that results when two clusters are merged. Thus, this method uses the same objective function as is used by the K-means clustering. Hence, for better comparison between the two algorithms ward proximity measure was used. The hclust method continues merging till a single cluster is formed. In order to form 8 clusters, we need to cut the final tree. For this cutree method is used by passing k = 8.

Further to compute accuracy of the predicted labels with respect to given labels, external_validation using adjusted rand index method was used. Adjusted Rand index measures the similarity of the two assignments, ignoring permutations and with chance normalization. Hence, we don't have to explicitly calculate mapping between given labels and predicted labels to compute accuracy. Finally, the clustering results was plotted in 3D using scatter3D wherein each data point was assigned color according to the cluster they were assigned to.

## 2) K-MEANS

K-means is a prototype based clustering technique. K-means defines a prototype in terms of a centroid, which is usually the mean of a group of points. Basic algorithm for k—means is as follows:

i) Select k points as initial centroids.
ii) Repeat the following until centroids do not change
    a) Form K clusters by assigning each point to its closest centroid.
    b) Re-compute the centroid of each cluster.

One of the major concerns, in k-means clustering is selecting the optimal k and deciding which points should be selected as initial centroids so as to get best clusters.

In my implementation, I used kmeans method of package stats. In this method number of desired clusters was passed as 8 so that we can compare the results with the given cluster labels. As mentioned above k-means randomly selects initial centers. The selection of these centers may affect the accuracy of the result, hence we set nstart parameter to 25. By this, it selects 25 different combinations of centers and then selects the centers with lowest within cluster variation.

The result achieved by this clustering was then compared with the given labels using external_validation using adjusted rand index method. Adjusted Rand index measures the similarity of the two assignments, ignoring permutations and with chance normalization. Hence, we don't have to explicitly calculate mapping between given labels and predicted labels to compute accuracy. Finally, the clustering results was plotted in 3D using scatter3D wherein each data point was assigned color according to the cluster they were assigned to.

## 3) DENSITY BASED

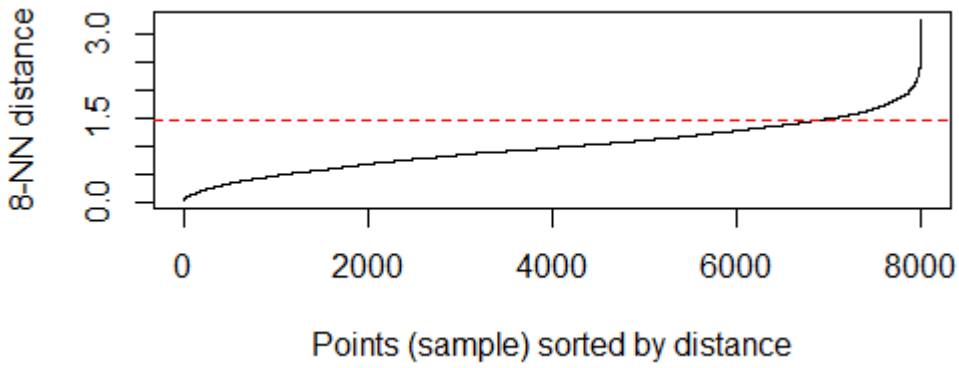
Density-based clustering locates regions of high density that are separated from one another by regions of low density. DBSCAN is a simple and effective density-based clustering algorithm. given a set of points in some space, it groups together points that are closely packed together (points with many nearby neighbors), marking as outlier points that lie alone in low-density regions (whose nearest neighbors are too far away). Basic DBSCAN algorithm is as follows:

DBSCAN requires two parameters: ε (eps) and the minimum number of points required to form a dense region[a] (minPts). It starts with an arbitrary starting point that has not been visited. This point's ε-neighborhood is retrieved, and if it contains sufficiently many points, a cluster is started. Otherwise, the point is labeled as noise. Note that this point might later be found in a sufficiently sized ε-environment of a different point and hence be made part of a cluster.

If a point is found to be a dense part of a cluster, its ε-neighborhood is also part of that cluster. Hence, all points that are found within the ε-neighborhood are added, as is their own ε-neighborhood when they are also dense. This process continues until the density-connected cluster is completely found. Then, a new unvisited point is retrieved and processed, leading to the discovery of a further cluster or noise.

Selection of MinPts and Eps: The k-dist is computed for all data points for some k and sort them in increasing order and then plo the sorted values. The eps at which we see a sharp change is the optimal eps for minpts as k

For implementing density based clustering dbscan method of fpc package was used. To determine optimal MinPts and Eps, we used kNNdistplot function to plot kNN distance for each point in sorted order, the nearest neighbor distance at which this plot bent was used as eps value. Different values of k and eps were tried in order to get 8 clusters so that we can compare the achieved clusters with given labels. By this method we found optimal MinPts as 8 and Eps as 1.463 (See figure) .




External_validation with adjusted rand index method was used to compare the given and predicted labels. Adjusted Rand index measures the similarity of the two assignments, ignoring permutations and with chance normalization. Hence, we don't have to explicitly calculate mapping between given labels and predicted labels to compute accuracy.

Finally, the clustering results was plotted in 3D using scatter3D wherein each data point was assigned color according to the cluster they were assigned to.

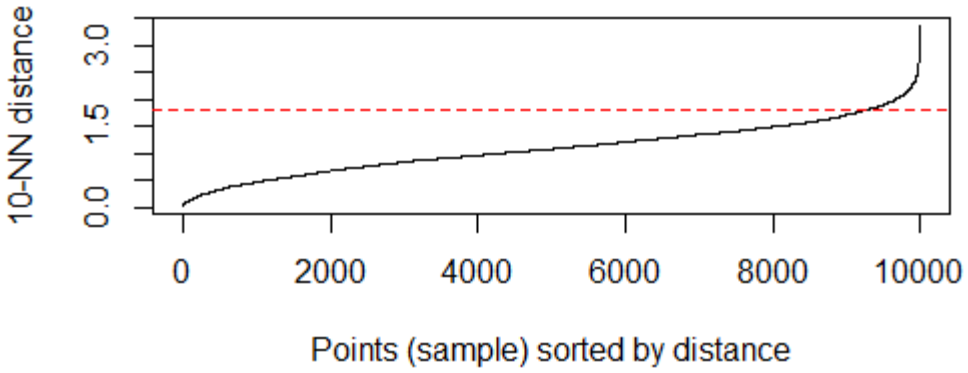
## 4) GRAPH BASED

For graph based clustering, shared nearest neighbor algorithm was used. SNN Clustering algorithm is a combination of Jarvis-Patrick algorithm and DBSCAN with SSN Similarity and SSN Density. The basic algorithm for SNN is as follows:

i. compute the similarity matrix
ii. sparsify the matrix by keeping only k most similar neighbors for each data point
iii. construct the SSN graph (use the Jarvis-Patrick algorithm)
iv. find SSN density of each point p: in the KNN list of p count q s.t. $sim(p,q) \geqslant \epsilon$
v. find the core points: all points with SSN density greater than min_pts are the core ones
vi. form clusters from the core points

vii. all non-core points not within ϵ from the core ones are discarded as noise
viii. align non-noise non-core points to clusters

For implementing shared nearest neighbor, sNNclust method of dbscan package. For this method we need to explicitly specify values of k, Minpts and Eps. Different values of k and Minpts were tried in order to get 8 clusters in order to facilitate comparison between predicted and given labels. Eps value for corresponding Minpts was found by plotting Minpts and nearest neighbor distance. By this method appropriate values were found to be k=10, Minpts= 1.6 and Eps= 10 (See figure below).



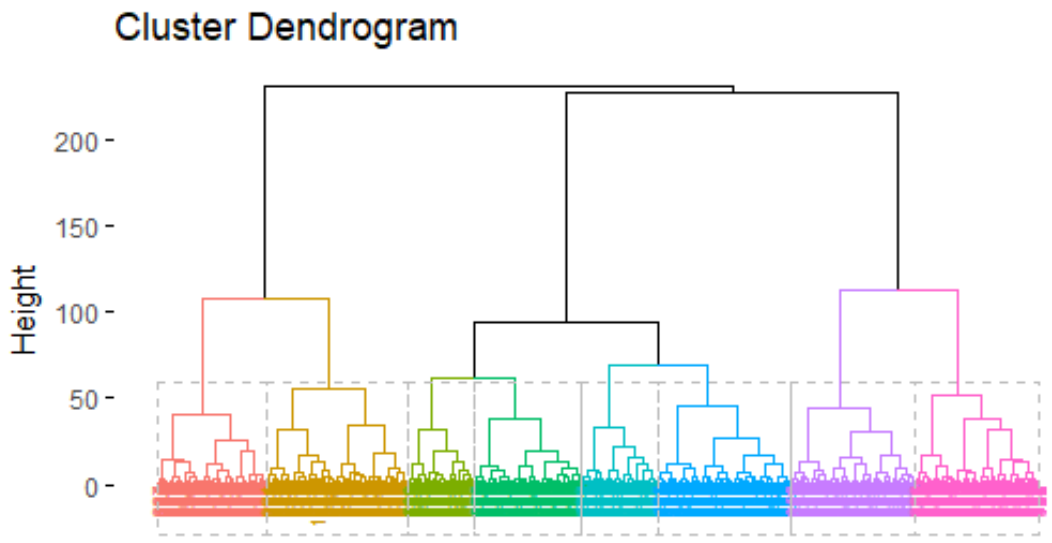

## OBSERVATION & ANALYSIS FOR DATASET 1

### PRE-PROCESSING

While scanning through dataset1 it was observed that there are no missing values. Hence, there is no need to omit any values. Also, the data in all the columns fall in a single range. Hence normalization is not required as well.

### 1) AGGLOMERATIVE HIERARCHICAL CLUSTERING

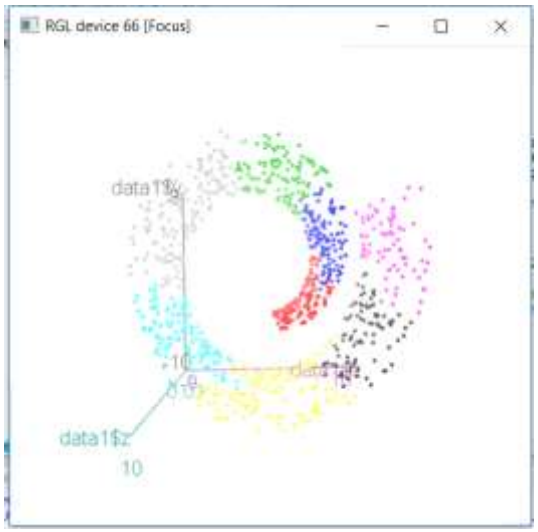
Since agglomerative hierarchical cluster doesn't depend on random selection of initial points we do not need to run it multiple times. Then dendrogram obtained by running hierarchical clustering using Euclidean distance and Ward's proximity measure was as follows:




The accuracy with agglomerative hierarchical clustering was as follows:

```
---------------------------------------
purity                          : 0.167
entropy                         : 0.9755
normalized mutual information   : 0.0122
variation of information        : 5.8901
normalized var. of information  : 0.9939
---------------------------------------
specificity                     : 0.8699
sensitivity                     : 0.1305
precision                       : 0.1244
recall                          : 0.1305
F-measure                       : 0.1274
---------------------------------------
accuracy OR rand-index          : 0.7781
adjusted-rand-index             : 3e-04
jaccard-index                   : 0.068
fowlkes-mallows-index           : 0.1274
mirkin-metric                   : 221698
---------------------------------------
```

The resultant clusters were as follows:

Given Clustering Hierarchical Clustering

## 2) K-MEANS

Since k-means works by initially selecting centroids randomly, to calculate accuracy, k -means was run 5 times using 5 different seed values. The results obtained with k=8 and nstart = 25 were as follows:

i) Seed = 123

```
[1] "Results of K-Means clustering"

---------------------------------------
purity                          : 0.166
entropy                         : 0.9733
normalized mutual information   : 0.0131
variation of information        : 5.8808
normalized var. of information  : 0.9934
---------------------------------------
specificity                     : 0.8688
sensitivity                     : 0.1316
precision                       : 0.1245
recall                          : 0.1316
F-measure                       : 0.128
---------------------------------------
accuracy OR rand-index          : 0.7773
adjusted-rand-index             : 5e-04
jaccard-index                   : 0.0684
fowlkes-mallows-index           : 0.128
mirkin-metric                   : 222450
---------------------------------------
```

ii) Seed = 6930

```
[1] "Results of K-Means Clustering"

---------------------------------------
purity                          : 0.163
entropy                         : 0.9754
normalized mutual information   : 0.0127
variation of information        : 5.8882
normalized var. of information  : 0.9936
---------------------------------------
specificity                     : 0.8696
sensitivity                     : 0.1305
precision                       : 0.1242
recall                          : 0.1305
F-measure                       : 0.1273
---------------------------------------
accuracy OR rand-index          : 0.7778
adjusted-rand-index             : 1e-04
jaccard-index                   : 0.068
fowlkes-mallows-index           : 0.1273
mirkin-metric                   : 221932
---------------------------------------
```

iii) Seed = 2156

```
[1] "Results of K-Means Clustering"

---------------------------------------
purity                          : 0.163
entropy                         : 0.9754
normalized mutual information   : 0.0127
variation of information        : 5.8882
normalized var. of information  : 0.9936
---------------------------------------
specificity                     : 0.8696
sensitivity                     : 0.1305
precision                       : 0.1242
recall                          : 0.1305
F-measure                       : 0.1273
---------------------------------------
accuracy OR rand-index          : 0.7778
adjusted-rand-index             : 1e-04
jaccard-index                   : 0.068
fowlkes-mallows-index           : 0.1273
mirkin-metric                   : 221932
---------------------------------------
```

iv) Seed = 1414

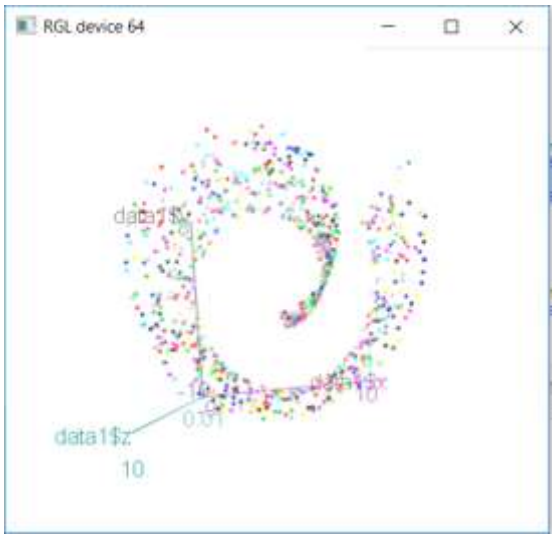
```
[1] "Results of K-Means Clustering"

---------------------------------------
purity                          : 0.163
entropy                         : 0.9754
normalized mutual information   : 0.0127
variation of information        : 5.8882
normalized var. of information  : 0.9936
---------------------------------------
specificity                     : 0.8696
sensitivity                     : 0.1305
precision                       : 0.1242
recall                          : 0.1305
F-measure                       : 0.1273
---------------------------------------
accuracy OR rand-index          : 0.7778
adjusted-rand-index             : 1e-04
jaccard-index                   : 0.068
fowlkes-mallows-index           : 0.1273
mirkin-metric                   : 221932
---------------------------------------
```

v) Seed = 729

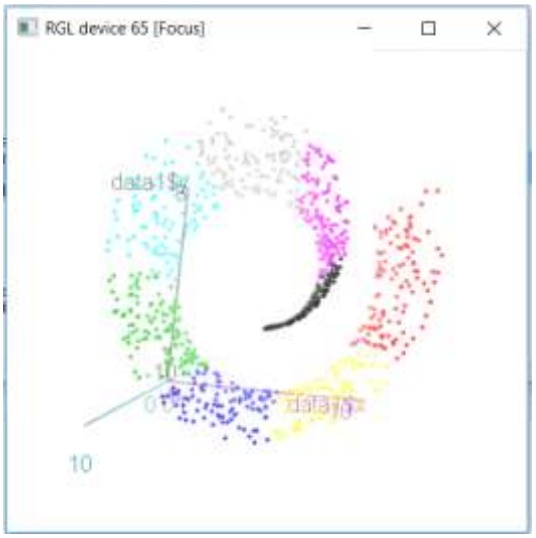
```
[1] "Results of K-Means Clustering"

---------------------------------------
purity                          : 0.166
entropy                         : 0.9733
normalized mutual information   : 0.0131
variation of information        : 5.8808
normalized var. of information  : 0.9934
---------------------------------------
specificity                     : 0.8688
sensitivity                     : 0.1316
precision                       : 0.1245
recall                          : 0.1316
F-measure                       : 0.128
---------------------------------------
accuracy OR rand-index          : 0.7773
adjusted-rand-index             : 5e-04
jaccard-index                   : 0.0684
fowlkes-mallows-index           : 0.128
mirkin-metric                   : 222450
---------------------------------------
```

Thus, average accuracy is 77.674. Final seed was chosen to be 6930 since we got maximum accuracy with it.

The resultant clusters were as follows:




Given Clustering



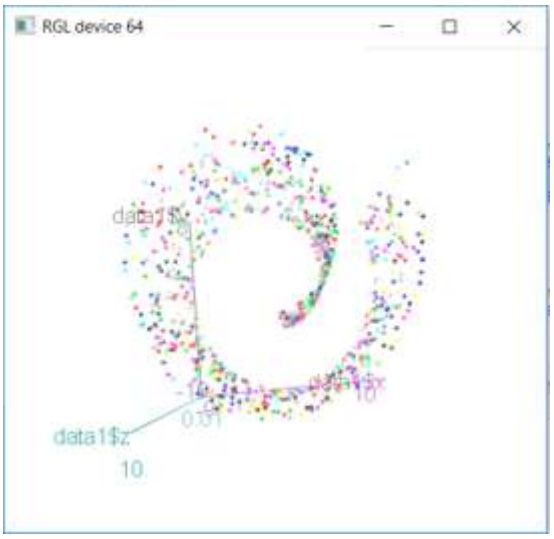

K-means Clustering

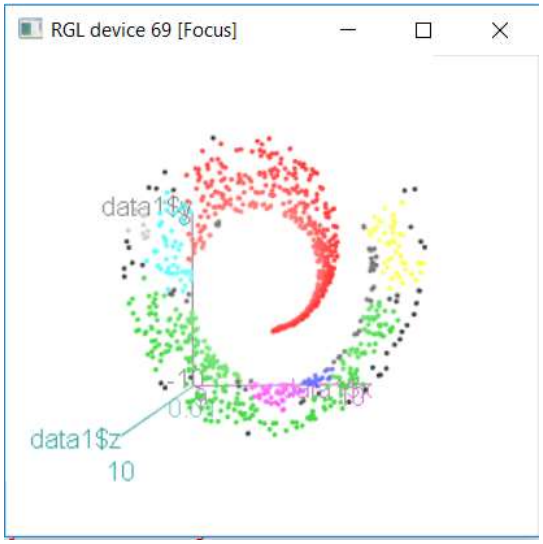
## 3) DBSCAN

Since dbscan doesn't depend on randomly selecting initial centers, it was run just once. The accuracy values obtained with Minpts as 8 and Eps as 1.463 as were as follows:

```
----------------------------------------
purity                          : 0.164
entropy                         : 0.6978
normalized mutual information   : 0.0208
variation of information        : 5.0399
normalized var. of information  : 0.9895
----------------------------------------
specificity                     : 0.6861
sensitivity                     : 0.3152
precision                       : 0.1246
recall                          : 0.3152
F-measure                       : 0.1785
----------------------------------------
accuracy OR rand-index          : 0.64
adjusted-rand-index             : 7e-04
jaccard-index                   : 0.098
fowlkes-mallows-index           : 0.1981
mirkin-metric                   : 359596
----------------------------------------
```

There clusters obtained were as follows:




Given Clustering

Density Based Clustering

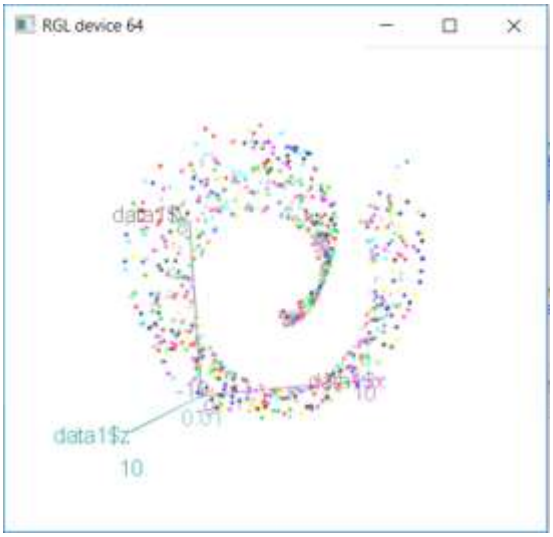
From the above images we see, that unlike k-means and hierarchical clustering, some of the points were assigned as noise points (indicated by black color ).

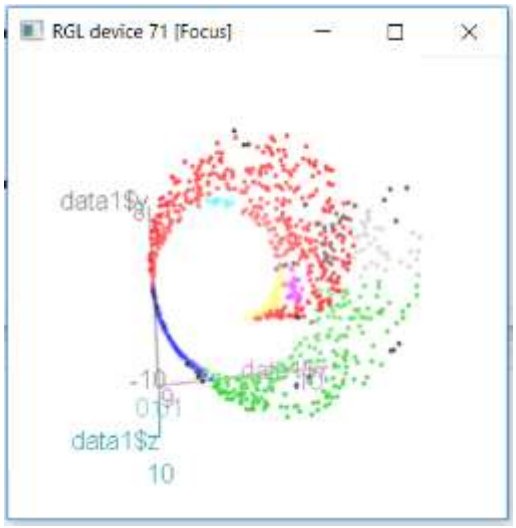
## 4) SHARED NEAREST NEIGHBOUR

Since shared nearest neighbor doesn't depend on randomly selecting initial centers, it was run just once. The accuracy values obtained with k =10, minPts = 1.6 and Eps= 10 were as follows:

| | |
|---|---|
| purity | : 0.165 |
| entropy | : 0.7573 |
| normalized mutual information | : 0.0183 |
| variation of information | : 5.2232 |
| normalized var. of information | : 0.9907 |
| specificity | : 0.7208 |
| sensitivity | : 0.2826 |
| precision | : 0.1254 |
| recall | : 0.2826 |
| F-measure | : 0.1738 |
| accuracy OR rand-index | : 0.6664 |
| adjusted-rand-index | : 0.0022 |
| jaccard-index | : 0.0951 |
| fowlkes-mallows-index | : 0.1883 |
| mirkin-metric | : 333262 |

There clusters obtained were as follows:




Given Clustering

Graph Based Clustering

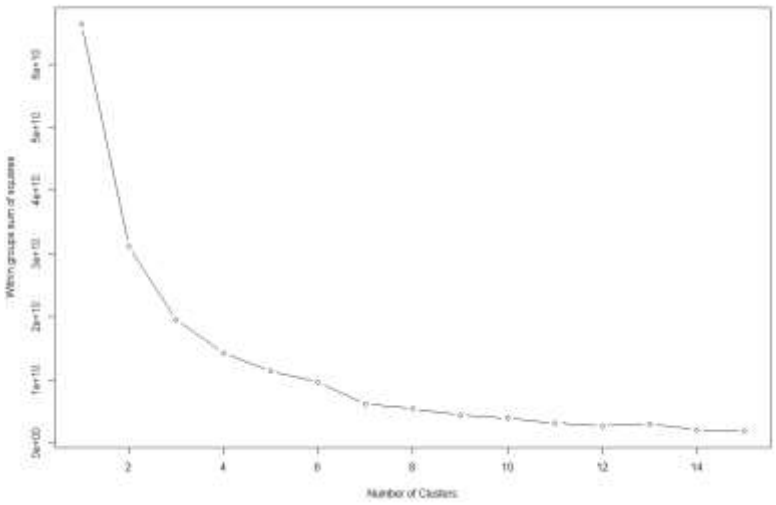
From the above images we see, that unlike k-means and hierarchical clustering, some of the points were assigned as noise points (indicated by black color).

## OBSERVATION AND ANALYSIS FOR DATASET 2

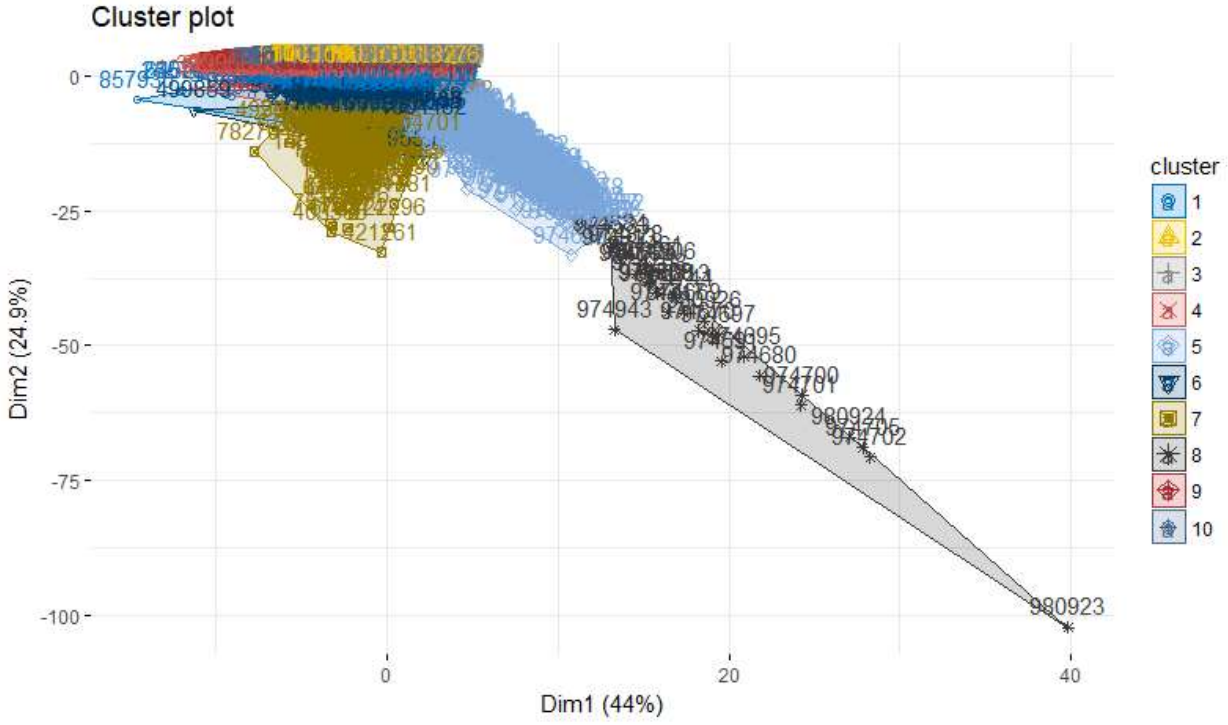
In the second section, we were required to perform clustering on a large dataset (more than 1 million points). Since in the previous section we got maximum accuracy using k-means and hierarchical clustering, it seemed to be an obvious choice for clustering. However, the most optimal clustering technique depends on the data. Hence, we look at that 4 clustering techniques anew.

From literature review, it was found that BIRCH (Balanced Iterative Reducing and Clustering using Hierarchies) algorithm and CURE (Clustering Using REpresentatives) which are hierarchical clustering algorithms work efficiently on large datasets. However, both of them are not supported in R anymore. On running the methods used in part one on dataset2 it was found that k-means algorithm worked best for the given dataset. Hence, k-means was used for clustering. For deciding the optimal number of cluster elbow method was used. In this method, we compare the sum of

squared error (SSE) for a number of cluster. SSE is defined as the sum of the squared distance between each member of a cluster and its cluster centroid. For this, we plot SSE for a sequence of k values. The value of k for which we get there is a drop in SSE value is selected as number of clusters. For our data SSE/k plot for k = 2 to 15 was as follows:



From the plot above we see that as number of clusters increase, the SSE reduces. At k = 10, the reduction in SSE slows. Hence, for this dataset we shall seek 10 clusters. The plot achieved on clustering is as follows:



For evaluating our clustering method, we use BSS/TSS ratio. BSS is the total weighted distance of various cluster centroids to the global mean of data and TSS is the total distance of data points from global mean of data. For any dataset, TSS value is constant. Ideally, clusters should be tight and homogenous, so BSS value should be higher. Hence, the higher the BSS/TSS ratio, the better the clustering is. For our clustering, the following values were found:

```
[1] "Total Sum of Squares (TSS)"
[1] 66329045576
[1] "Between Sum of Squares (BSS)"
[1] 62675316058
[1] "BSS/TSS ratio"
[1] 0.9449151
```

Hence, ratio is 0.9449 which is considered to be good.

## CONCLUSION

For dataset1, various clustering methods were studied and implemented. Of these it was found that hierarchical clustering was the most accurate with an accuracy of 77.81%, closely followed by k-means with an average accuracy of 77.674%. Graph-based and density based clustering gave lower accuracy 66.64% and 64% respectively. Further, on observing the plots it was seen that for our data set. K-means and hierarchical clustering gave more well separated clusters in comparison to graph based and density based clusters. Moreover, graph-based and density based clustering could not assign some points to individual clusters and assigned them as noise points. Overall, due to better accuracy and more well-defined clusters, for our dataset k-means is the best clustering algorithm.

For dataset2, clustering methods for large datasets were reviewed. Of these, it was found that k-means was the most suitable clustering technique for our dataset hence the same was implemented. To find the optimal k elbow method was used. By this we found that the most appropriate value of k was 10. To evaluate the resultant clusters, BSS/TSS ratio was used. The fit obtained by clustering is considered to be good when this is closer to 1. For my clustering, the ratio was 0.9449. Hence, we this can be considered as a very good fit.

## References


i. https://www.statmethods.net/advstats/cluster.html
ii. https://www.rdocumentation.org/packages/stats/versions/3.4.1/topics/kmeans
iii. https://www.r-bloggers.com/k-means-clustering-in-r/
iv. https://en.wikipedia.org/wiki/K-means_clustering
v. http://scikit-learn.org/stable/modules/clustering.html#clustering-performance-evaluation
vi. https://www-users.cs.umn.edu/~hanxx023/dmclass/cluster_survey_10_02_00.pdf
vii. https://stackoverflow.com/questions/15376075/cluster-analysis-in-r-determine-the-optimal-number-of-clusters/15376462#15376462
viii. http://www.mattpeeples.net/kmeans.html

ix. http://www.sthda.com/english/wiki/amazing-interactive-3d-scatter-plots-r-software-and-data-visualization
x. https://www.dezyre.com/data-science-in-r-programming-tutorial/k-means-clustering-techniques-tutorial
xi. https://stats.stackexchange.com/questions/82776/what-does-total-ss-and-between-ss-mean-in-k-means-clustering
xii. https://en.wikipedia.org/wiki/Hierarchical_clustering
xiii. https://en.wikipedia.org/wiki/DBSCAN
xiv. http://mlwiki.org/index.php/SNN_Clustering