Course Project DRAFT

Data Mining Foundations

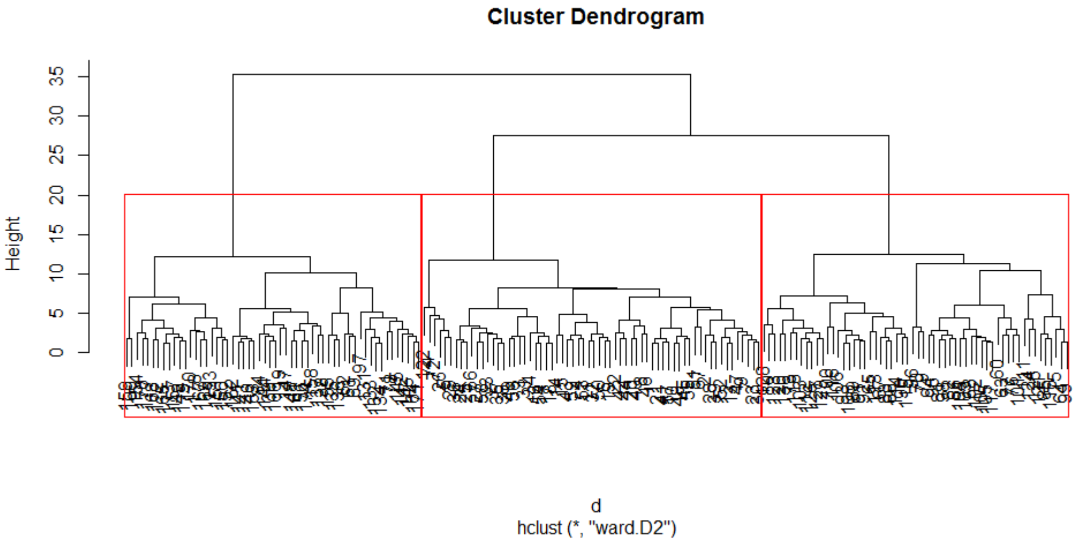
Masha Kinley

Executive Summary

The aim of this project is to apply three different clustering techniques on two different datasets with the purpose of demonstrating the absence of a perfect solution single clustering technique applicable to any problem and the need to apply multiple data mining methods to find the precise one in each case depending on the type of data.

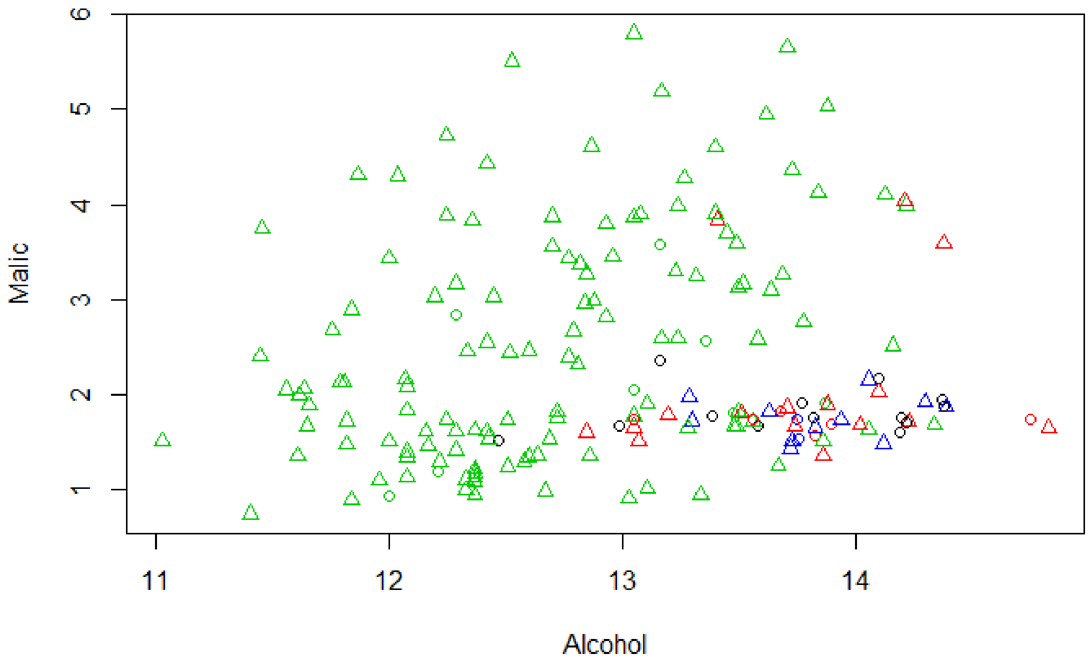
The clustering algorithms I will use are k-means clustering, density-based clustering (DBSCAN) and hierarchical clustering. In each case, all three algorithms will be applied to datasets with known number of clusters. The datasets chosen for this project are the wine dataset and a custom designed Clx set, the wine dataset has three clusters with some entries which could be misclassified and Clx set contains three non-convex clusters.

Hclust for wine



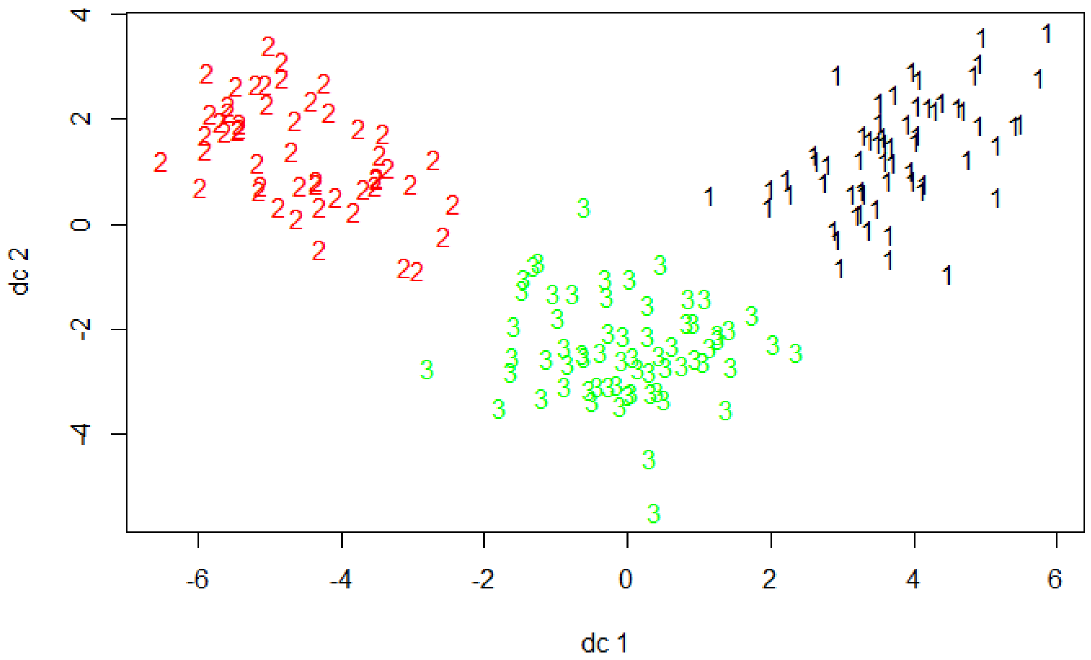
As seen in the hclust dendrogram above, although the hierarchical clustering method found the three clusters in this dataset it is difficult to interpret the results, due to the large number of individual data points in this set.

DBSCAN



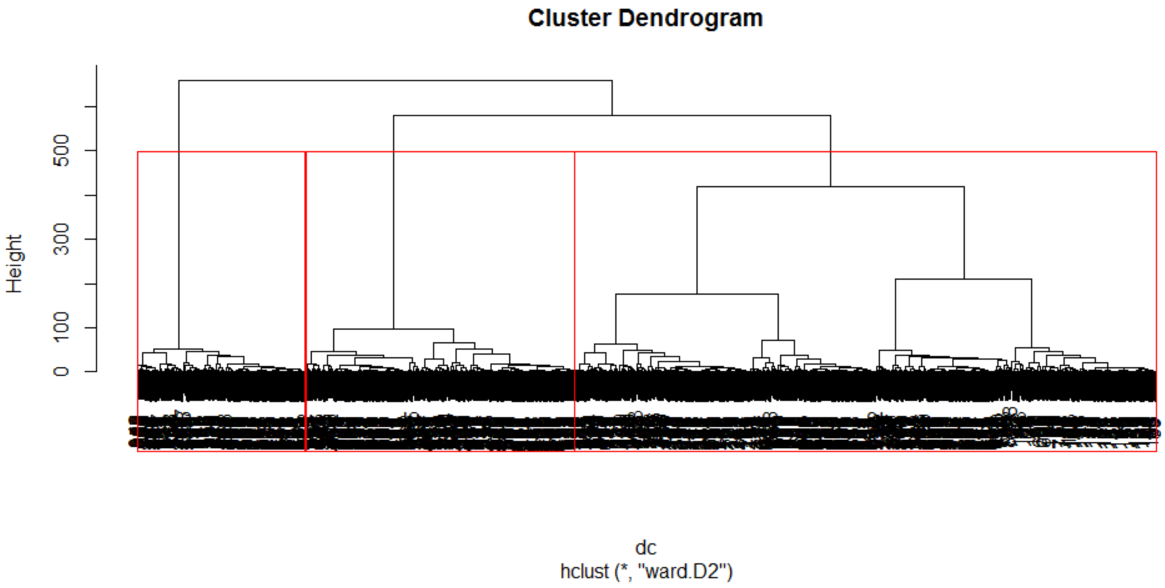
DBSCAN also identified the three clusters but due to difficulty finding the correct input parameters for this method and the close-clustered data in this set, it struggled to identify the clusters because they are somewhat uniform.

Kmeans for wine



Kmeans method seems to have identified the clusters clearly and, most importantly, for this type of data where each single point must belong to a cluster, this method is the obvious choice because it will classify each potential outlier as belonging to a cluster.

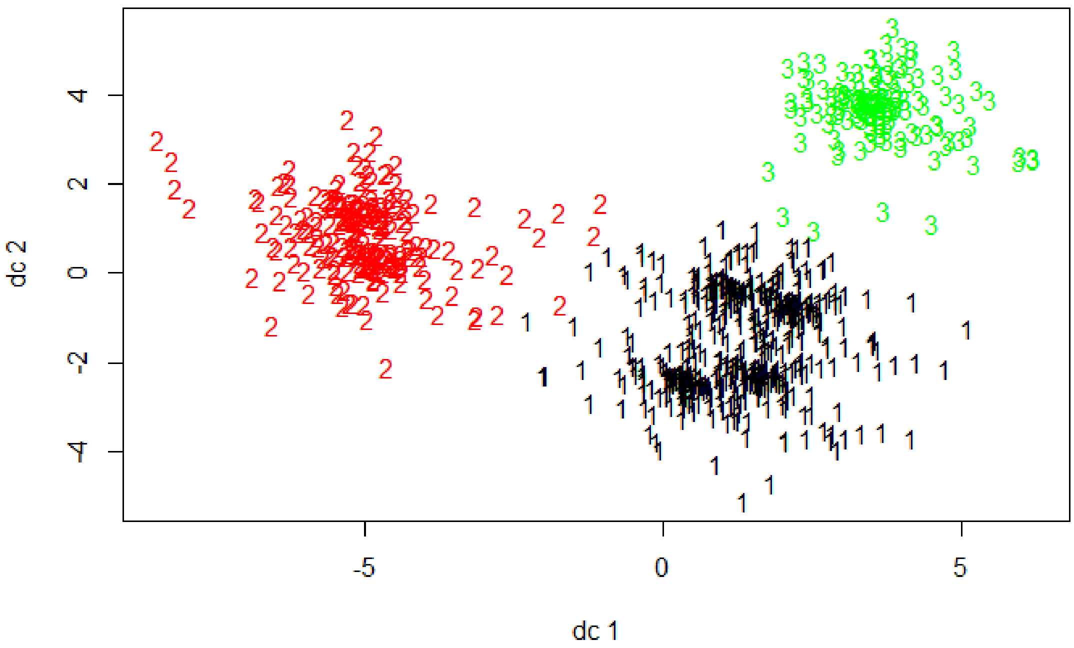
Hclust for Clx dataset



```
summary(clx)

    Attr_1          Attr_2          Attr_3          Class

 Min.   :-28.445   Min.   :-30.458   Min.   :-24.7382   class1:472

 1st Qu.: -6.864   1st Qu.: -1.970   1st Qu.: -7.4858   class3:137

 Median :  8.917   Median :  2.703   Median : -0.1708   class4:218

 Mean   :  8.646   Mean   :  4.497   Mean   :  0.1873

 3rd Qu.: 26.137   3rd Qu.:  9.949   3rd Qu.:  8.5707

 Max.   : 42.750   Max.   : 37.383   Max.   : 22.9934
```

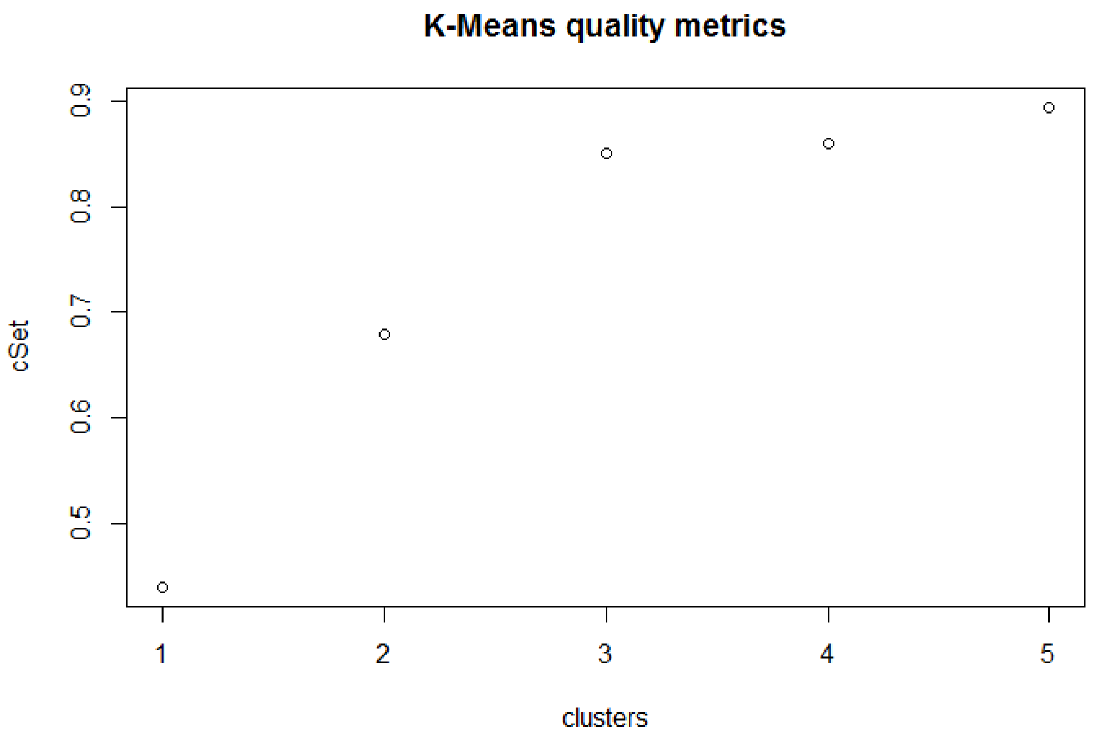
This dataset is large, and using hclust becomes cumbersome when interpreting results as seen from the dendrogram above. It should also be noted that the results can be interpreted as three clusters but the algorithm, in this case, finds four (dividing one of the three into a sub cluster).

Kmeans for Clx

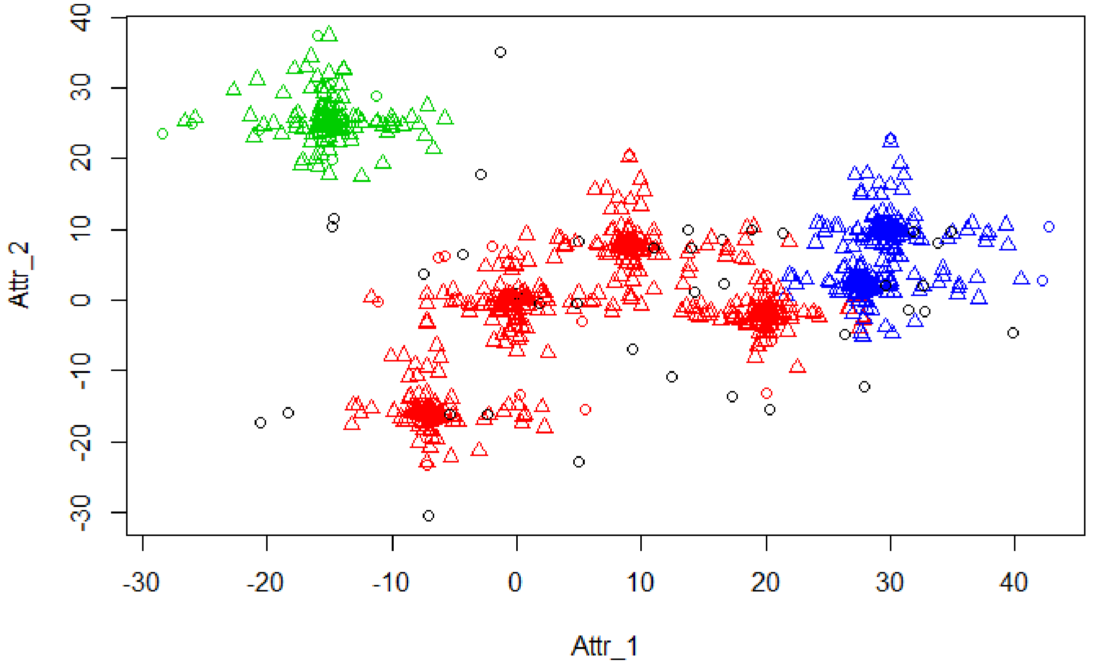


This method, although, being able to identify the clusters, is very slow when analyzing the data and it assigns every single point into the cluster, thus illuminating the outliers.

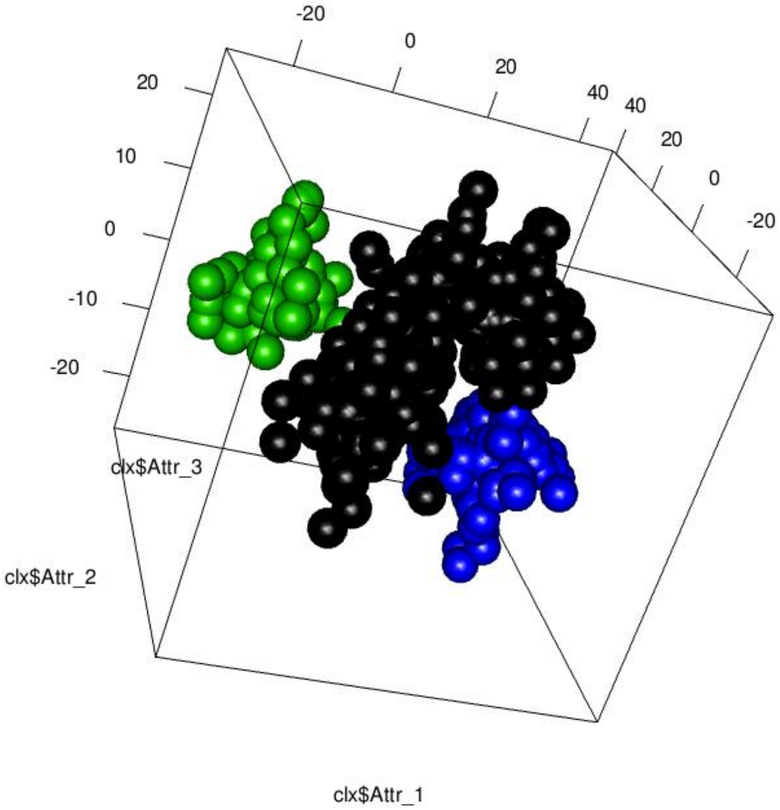
Further, the potential misinterpretation of the data can be noted in the quality metric plot below, the absence of clear inflection point indicates the potential confusion of the results as to the number of clusters.



DBSCAN for Clx



This method, appears to have produced the most prescise results, identifying both the clusters of the correct shape, as evidenced by the 3d rendering of the same set below and the potential outliers (seen as black dots).



Based on the overview of the results of this project, it is possible to conclude that even though this project involved only two datasets, though significantly different in nature, it is evidenced that applying the same methods to classify the data in these sets do not produce similar quality results, thus the method which is best for one type of data will not necessarily work for another.

Problem Description

Clustering is used in data mining to discover patterns in data and to enable analysis.

The aim of this project is to apply three different clustering techniques on two different datasets with the purpose of demonstrating the absence of a perfect solution single clustering technique applicable to any problem and the need to apply multiple data mining methods to find the precise one in each case depending on the type of data.

The clustering algorithms I will use are k-means clustering, density-based clustering (DBSCAN) and hierarchical clustering. In each case, all three algorithms will be applied to datasets with known number of clusters. The datasets chosen for this project are the wine dataset and a custom designed Clx set, the wine dataset has three clusters with some entries which could be misclassified and Clx set contains three non-convex clusters.

Analysis Technique

Clustering analysis is an important part of data mining. Clustering is the division of a dataset into multiple groups or clusters, with the aim of achieving a high degree of similarity between objects in the same cluster. Clustering identifies the dense and sparse regions in the dataset and reveals the patterns in the data. The three algorithms in this project differ in the methods used to process data during clustering.

K-Means

One of the first steps in building a K-Means clustering work is to define the number of clusters to work with. Subsequently, the algorithm assigns each individual data point to one of the clusters in a random fashion. The underlying idea of the algorithm is that a good cluster is the one which contains the smallest possible *within-cluster* variation of all observations in relation to each other. The most common way to define this variation is using the squared Euclidean distance according to the formula:

$$SS(k) = \sum_{i=1}^{n} \sum_{j=0}^{p} (x_{ij} - \bar{x}_{kj})^2$$

Where k is the cluster,x_{ij} is the value of the j^th variable for the i^th observation, and x_{kj}-bar is the mean of the j^th variable for the k^th cluster (Ng, 2016).

The k-means aims to minimize the sum of squared distances between all points and the cluster center.

Generally, the way K-Means algorithms works in the following way:

1. Begin by choosing a number of clusters, I began this experiment with two clusters and ended with eight clusters.
2. Each data point is randomly assigned to a cluster
3. Each cluster's centroid (mean within cluster) is calculated.
4. Each data point is assigned to its nearest centroid (iteratively to minimize the within-cluster variation) until no major differences are found.

The algorithm eventually converges to a point, although it is not necessarily the minimum of the sum of squares. The algorithm stops when the assignments do not change from one iteration to the next (Rego, 2015).

The format of the K-means function in R is kmeans(*x*, *centers*) where *x* is a numeric dataset (matrix or data frame) and *centers* is the number of clusters to extract. The function returns the cluster memberships, centroids, sums of squares (within, between, total), and cluster sizes. Then a plot of the total within-groups sums of squares against the number of clusters in a K-means solution is generated, in this project this plot is named "K-Means Quality Metrics". A bend in the graph can suggest the appropriate number of clusters (Galili, 2015).

The weakness of K-Means is that the algorithm has no concept of outliers, so all points are assigned to a cluster even if they do not belong in any. In the domain of anomaly detection, this causes problems as anomalous points will be assigned to the same cluster as “normal” data points. The anomalous points pull the cluster centroid towards them, making it harder to classify them as anomalous points. Another problem is that K-Means will only identify spherically-shaped clusters (Nandi, 2015).

DBSCAN

DBSCAN (Density-Based Spatial Clustering of Applications with Noise) is the most well-known density-based clustering algorithm. Unlike K-Means, DBSCAN does not require the number of clusters as a parameter. Instead it discovers the number of clusters of any arbitrary shape based on the data. The idea ε-neighborhood is fundamental to DBSCAN to approximate local density. Assume a point *p* and its neighborhood of radius ε, the *mass* of the neighborhood can be defined as the number of data points contained within such neighborhood, and the *volume* of the neighborhood is volume of the resulting shape of the neighborhood thus defining the density at the point *p* of the given neighborhood. Clustering analysis of the entire dataset is achieved by calculating the local density approximation for all points in the given dataset and grouping of points that are *nearby* (contained in the same neighborhood) and which have similar local density approximations identifies such points as belonging in the same cluster.

The algorithm has two parameters:

1. *ε*: The radius of the neighborhoods around a data point *p*.
2. *minPts*: The minimum number of data points we want in a neighborhood to define a cluster.

Using these two parameters, DBSCAN categories the data points into three categories:

1. Core Points: A data point p is a core point if Nbhd(p,ε) [ε-neighborhood of p] contains at least minPts ; |Nbhd(p,ε)| >= minPts.

2. Border Points: A data point *q is a border point if Nbhd(q, ε) contains less than minPts data points, but q is reachable from some core point p.

3. Outlier: A data point o is an outlier if it is neither a core point nor a border point.

The steps to the DBSCAN algorithm are:

1. Pick a point at random that has not been assigned to a cluster or been designated as an *outlier*. Compute its neighborhood to determine if it's a *core point*. If yes, start a cluster around this point. If no, label the point as an *outlier*.

2. Once we find a *core point* and thus a cluster, expand the cluster by adding all *directly-reachable* points to the cluster. Perform "neighborhood jumps" to find all *density-reachable* points and add them to the cluster. If an an *outlier* is added, change that point's status from *outlier* to *border point*.

3. Repeat these two steps until all points are either assigned to a cluster or designated as an *outlier*.

The weakness of DBSCAN is the time and computing power it requires as well as the complexity of identifying the initial parameters. When the data density is not uneven, the quality of clustering is very poor. Input parameters are sensitive and Eps, MinPts parameters are difficult to determine (Han, Kamber, & Pei, 2012).

Hierarchical clustering (hclust)

Hierarchical clustering involves creating clusters that have a predetermined ordering from top to bottom. The algorithm builds a hierarchy from the bottom-up, and like DBSCAN it does not require the number of clusters as a parameter.

The steps for hierarchical algorithm are:

1. Put each data point in its own cluster.

2. Identify the closest two clusters and combine them into one cluster.

3. Repeat the above step till all the data points are in a single cluster.

Different techniques can be used to determine which clusters are closest, the following two are used most often:

- Complete linkage clustering: Find the maximum possible distance between points belonging to two different clusters.

- Mean linkage clustering: Find all possible pairwise distances for points belonging to two different clusters and then calculate the average.

In hierarchical cluster analysis dendrogram graphs are used to display results and visualize how clusters are formed.

One weakness of hierarchical clustering is the dendrogram display of the results, because for very large datasets with multiple clusters such displays could be impractical and difficult to examine (Kodali, 2016).

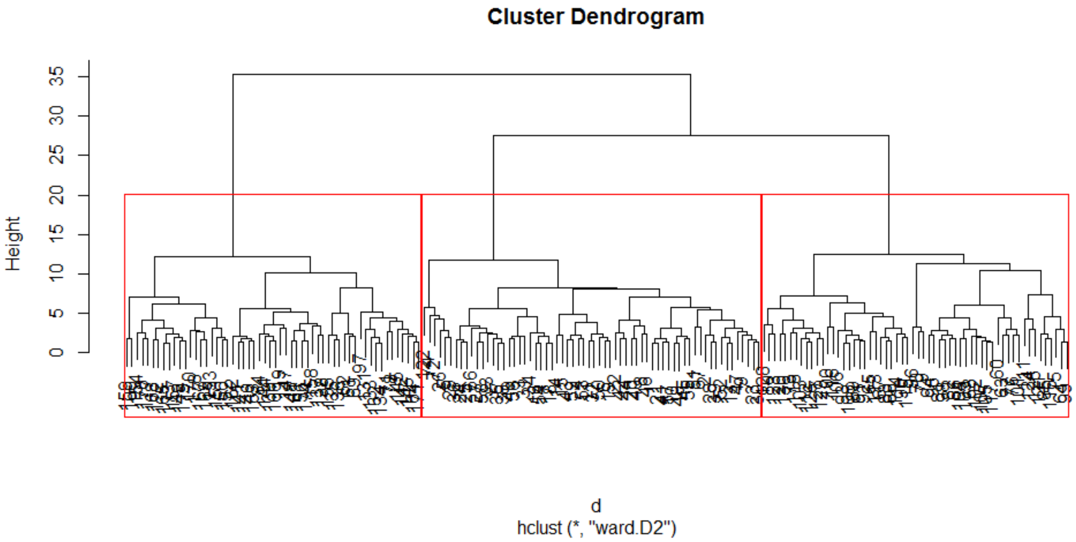
Assumptions

The assumption made in this project is that both data sets are large enough to be a good representative sample for this type of data.

Results

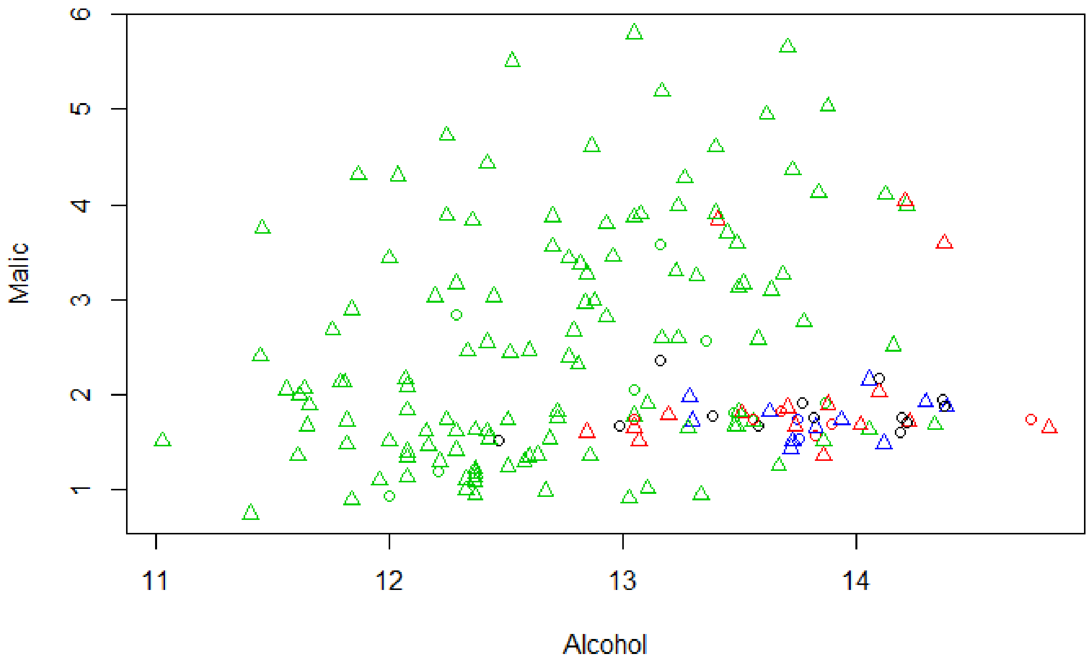
When applied to wine data set the algorithms produced the following results.

Hclust for wine



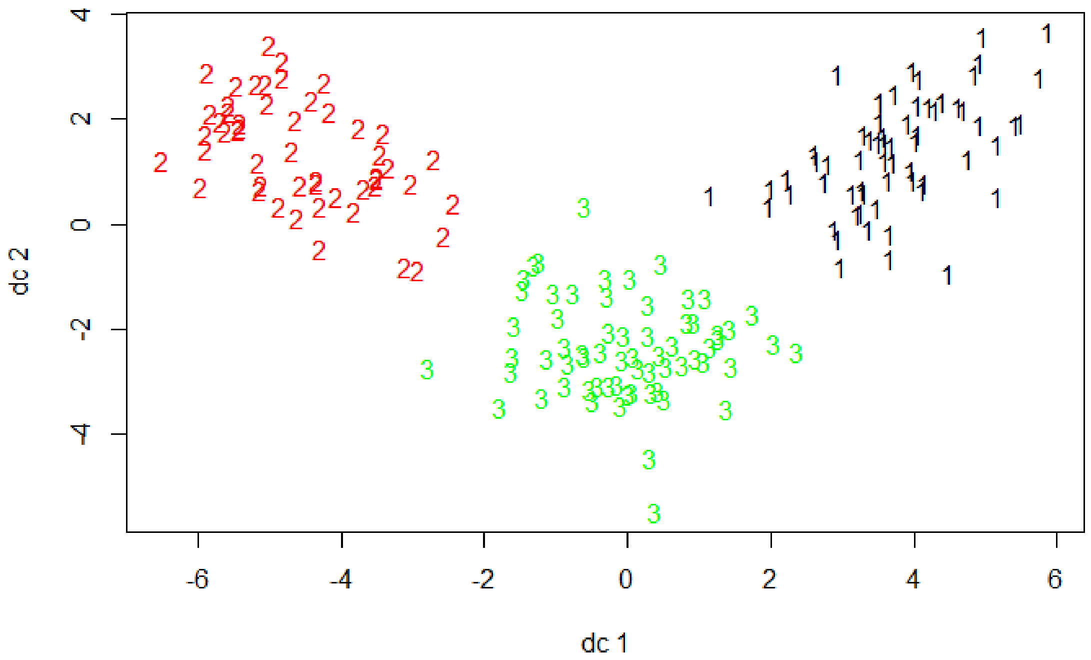
As seen in the hclust dendrogram above, although the hierarchical clustering method found the three clusters in this dataset it is difficult to interpret the results, due to the large number of individual data points in this set.

DBSCAN



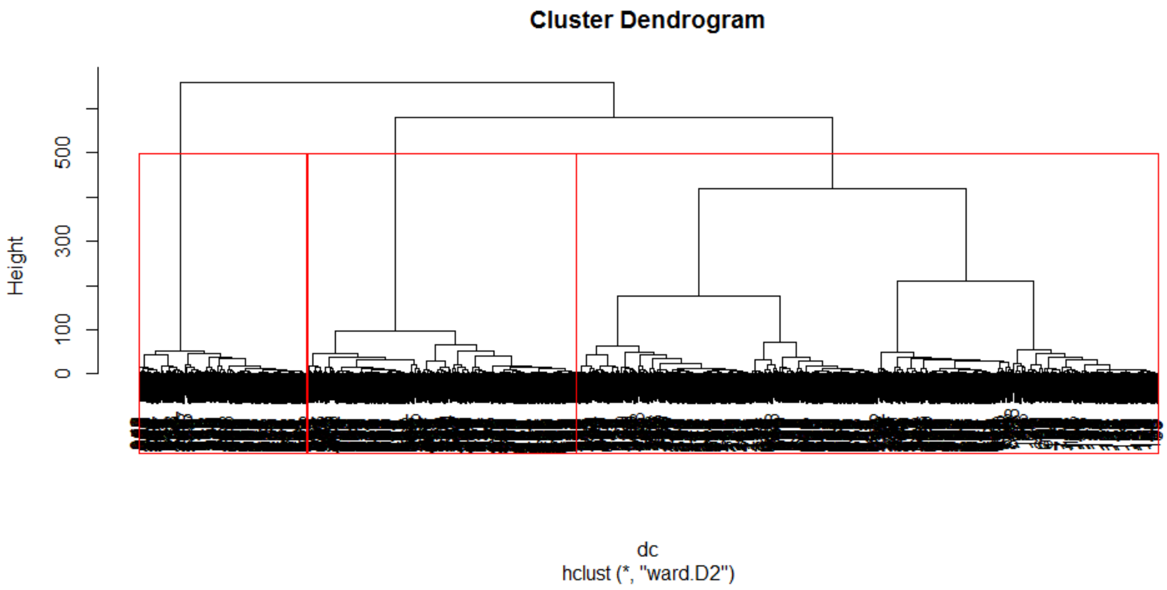
DBSCAN also identified the three clusters but due to difficulty finding the correct input parameters for this method and the close-clustered data in this set, it struggled to identify the clusters because they are somewhat uniform.

Kmeans for wine



Kmeans method seems to have identified the clusters clearly and, most importantly, for this type of data where each single point must belong to a cluster, this method is the obvious choice because it will classify each potential outlier as belonging to a cluster.

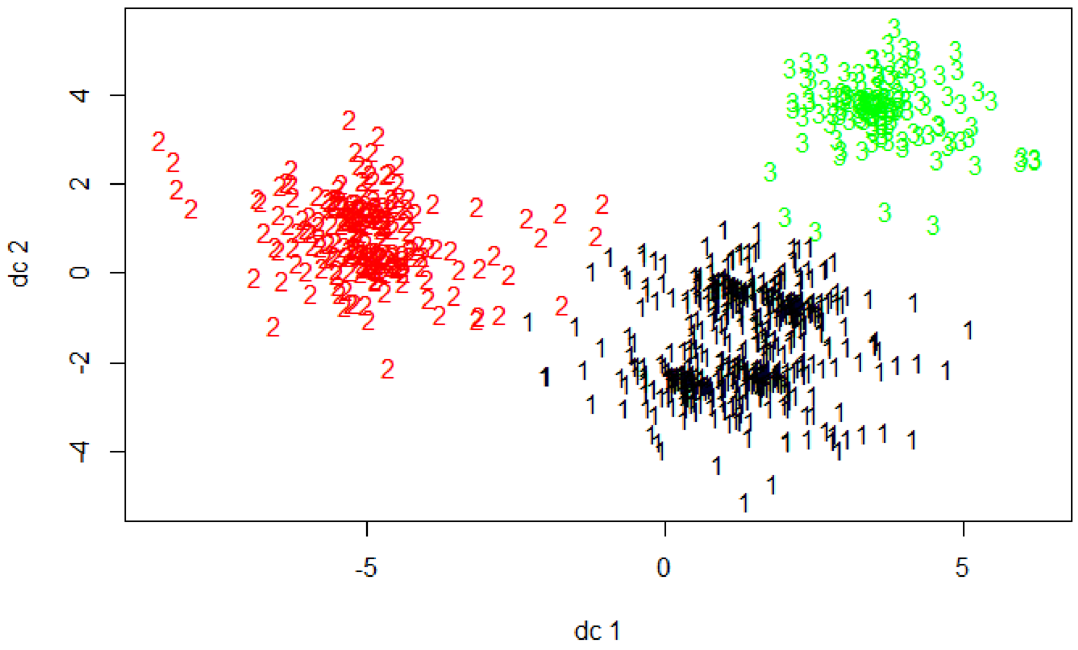
Hclust for Clx dataset



```
summary(clx)
    Attr_1           Attr_2           Attr_3           Class
 Min.   :-28.445   Min.   :-30.458   Min.   :-24.7382   class1:472
 1st Qu.: -6.864   1st Qu.: -1.970   1st Qu.: -7.4858   class3:137
 Median :  8.917   Median :  2.703   Median : -0.1708   class4:218
 Mean   :  8.646   Mean   :  4.497   Mean   :  0.1873
 3rd Qu.: 26.137   3rd Qu.:  9.949   3rd Qu.:  8.5707
 Max.   : 42.750   Max.   : 37.383   Max.   : 22.9934
```

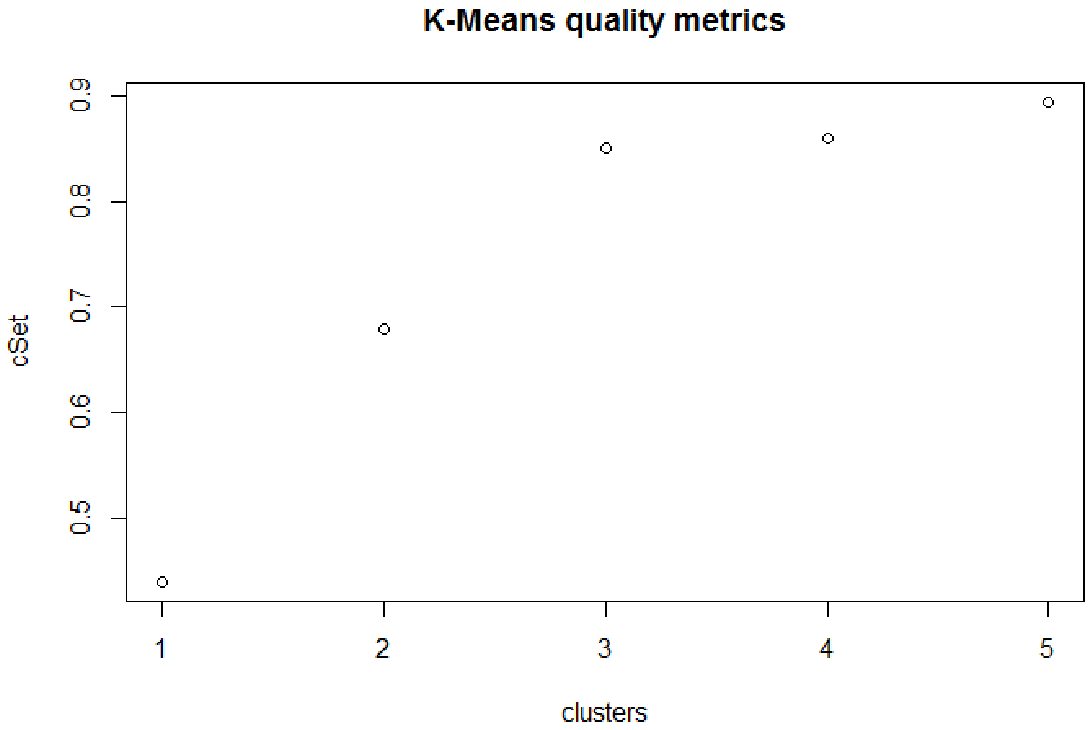
This dataset is large, and using hclust becomes cumbersome when interpreting results as seen from the dendrogram above. It should also be noted that the results can be interpreted as three clusters but the algorithm, in this case, finds four (dividing one of the three into a sub cluster).

Kmeans for Clx

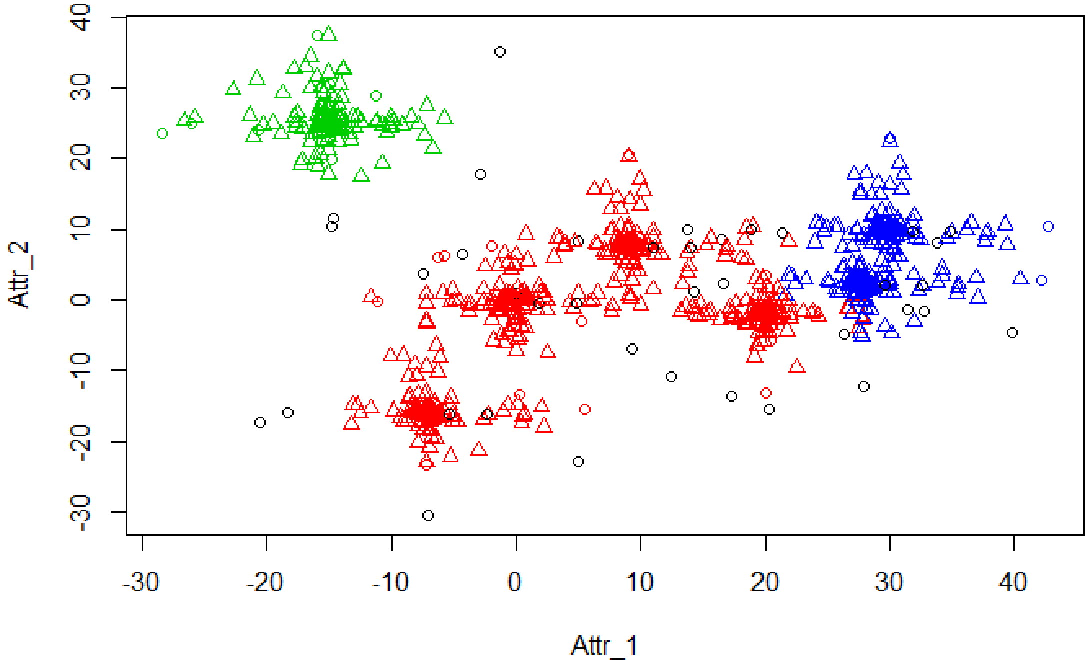


This method, although, being able to identify the clusters, is very slow when analyzing the data and it assigns every single point into the cluster, thus illuminating the outliers.

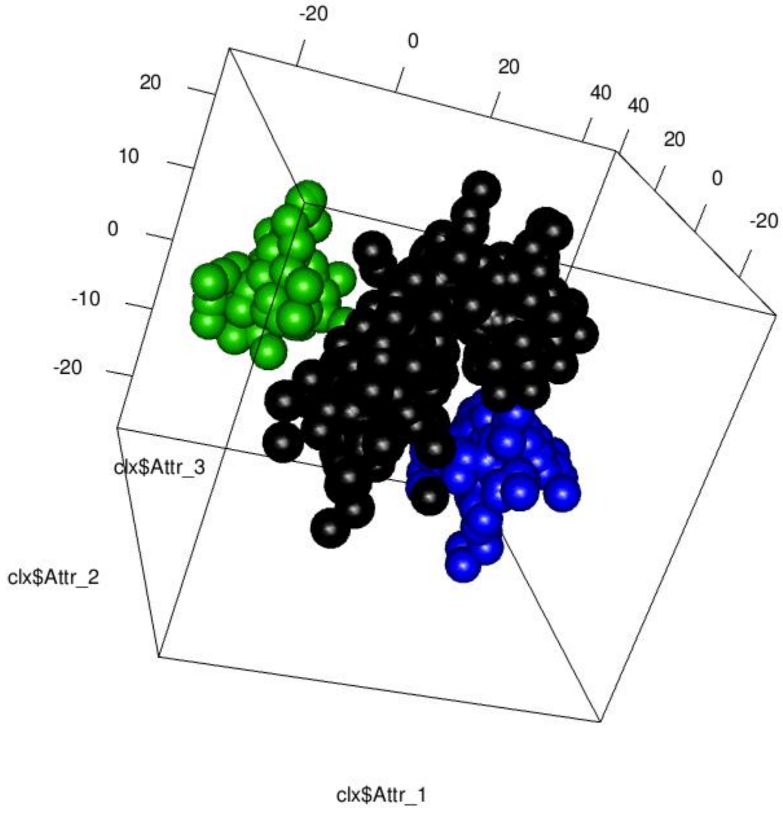
Further, the potential misinterpretation of the data can be noted in the quality metric plot below, the absence of clear inflection point indicates the potential confusion of the results as to the number of clusters.



DBSCAN for Clx



This method, appears to have produced the most prescise results, identifying both the clusters of the correct shape, as evidenced by the 3d rendering of the same set below and the potential outliers (seen as black dots).



Based on the overview of the results of this project, it is possible to conclude that even though this project involved only two datasets, though significantly different in nature, it is evidenced that applying the same methods to classify the data in these sets do not produce similar quality results, thus the method which is best for one type of data will not necessarily work for another.

Issues

I was only familiar with one algorithm of the three I chose for this project, as a result it was difficult for me to work through the results and identify which of the errors were actual issues and which happened due to my unfamiliarity with the methods. I had difficulty identifying the correct input parameters for DBSCAN.

References

Galili, T. (2015, August). *K-Means Clustering .* Retrieved from R-Statistics : https://www.r-statistics.com/2013/08/k-means-clustering-from-r-in-action/

Han, J., Kamber, M., & Pei, J. (2012). *Data Mining Concepts and Techniques .* Waltham: Morgan Kaufman .

Kodali, T. (2016, January 22). *Hierarchical Clustering in R.* Retrieved from R-Bloggers : https://www.r-bloggers.com/hierarchical-clustering-in-r-2/

Nandi, M. (2015, September 9). *Topology and Density-based Clustering.* Retrieved from Dominodatalab.com: https://blog.dominodatalab.com/topology-and-density-based-clustering/

Ng, A. (2016, August). *The k-Means Algorithm.* Retrieved from Quora: https://www.quora.com/What-is-the-k-Means-algorithm-and-how-does-it-work

Rego, F. (2015, July). *RPubs - Example of K-Means.* Retrieved from RPubs : https://rpubs.com/FelipeRego/K-Means-Clustering