COSC 757 Data Mining Assignment 3

Mary Snyder

Department of Computer & Information Sciences  
College of Science & Mathematics  
Towson University

msnyde8@students.towson.edu

**ABSTRACT**

In this paper, I will be exploring a dataset to become more familiar with data clustering through the COSC 757 Data Mining Assignment 3.

**Categories and Subject Descriptors**

H.2.8 **[Database Management]** Database Applications – *Data mining*

**Keywords**

Clustering; Multivariate; Cluster Analysis; Clustering Approaches; Partitioning Approach; k-means; k-medoids; Hierarchical Approach; AGNES; Single Linkage; Complete Linkage; Density-based Approach; DBSCAN; Accuracy; Error Rate; Sensitivity; Specificity; Precision; Recall; F Measure;

# INTRODUCTION

## Dataset

I chose a dataset from the UCI Machine Learning Repository classified for the task of Clustering. This AAAI 2013 Accepted Papers dataset comprises the metadata for all the main track only papers accepted for the 2013 AAAI conference. The dataset contains information regarding the paper’s title, abstract, and keywords of varying granularity. There are 150 instances with no missing values and contains 5 attributes: Title, Keywords, Topics, High-Level Keyword(s), and Abstract. The Topics and High-Level Keywords attributes were used for clustering.

## Objective of Analysis

The objective of clustering, or cluster analysis, is to find similarities among the data between the characteristics found in the data and using those similarities to group the data into clusters. Many times, it is used to gain insight into data distribution, but it may also be used as a pre-processing step for other algorithms.

Good cluster methods produce high quality clusters with high intra-class similarity or cohesion within clusters. They also have low inter-class similarity or distinction between clusters. The quality of the method depends on its measure of similarity, the implementation, as well as how well it is able to discover hidden patterns within the data. Similarity/dissimilarity is expressed in terms of distance, while quality is more subjective.

# METHODOLOGY

## Preprocessing

The dataset contains two different keyword attributes as well as the title and abstract information (see Figure 1). Since in most cases the title would be a unique value, I eliminated it as a possible clustering attribute. Similarly, the abstract contained a description of the paper that would mostly likely be unique, so it was eliminated from the clustering attribute selection as well. There are two keyword attributes, which varied in their degree of granularity, one more simplistic (Keywords) and one more categorical (High-Level Keyword(s)). I chose to eliminate the more simplistic Keywords attribute in favor of the more categorical High-Level Keyword(s) in hope this would produce better clustering results. The remaining attribute Topics also seemed fairly categorized, so I chose to pair it with the High-Level Keyword(s) for the analysis (see Figure 2).

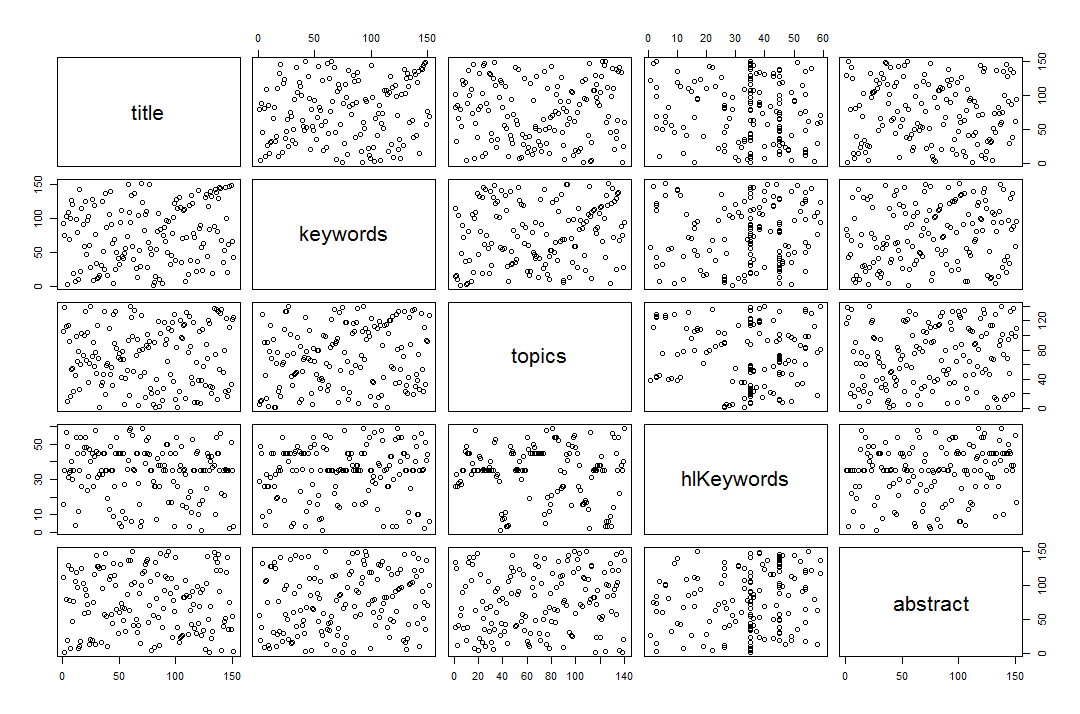


Figure 1. AAAI 2014 Accepted Papers Dataset Attributes

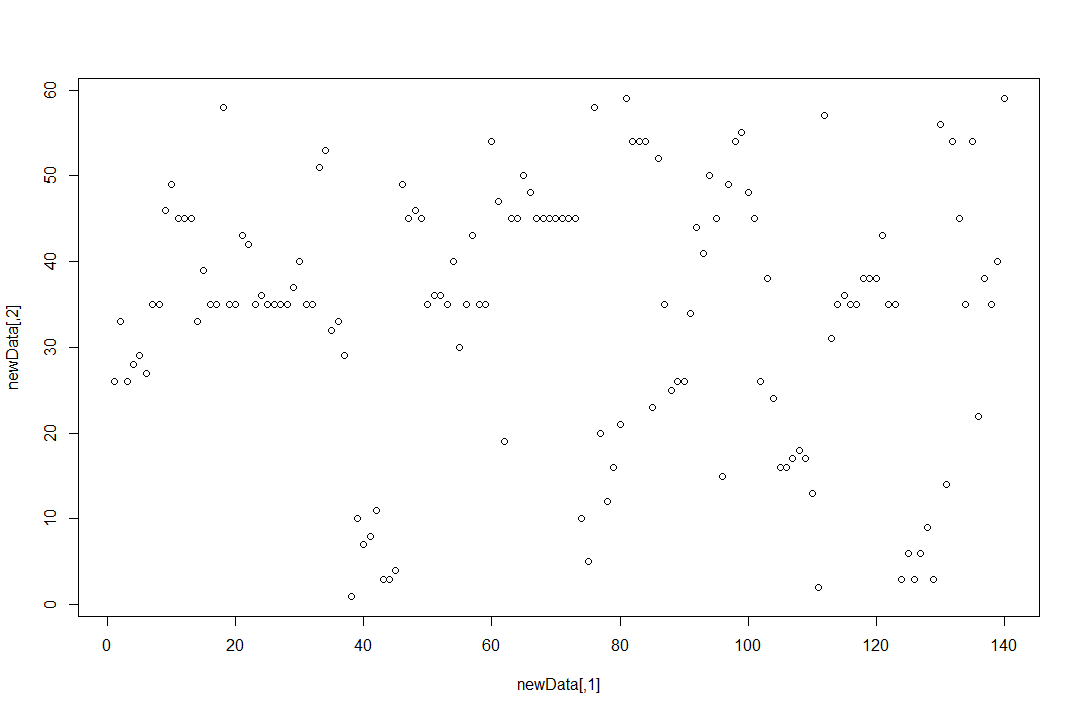


Figure 2. Topics and High-Level Keyword(s) for Clustering

## Clustering Approaches

### Partitioning Approach

The partitioning approach toe clustering constructs various partitions and then evaluated them by some criterion. The data set is divided, or partitioned, into *k* clusters, so as to optimize the chosen partitioning criterion and such that the sum of squared distances in minimized or

#### K-means

K-Means in a partitioning approach to clustering in which each cluster is represented by the center of the cluster. There are four steps for the k-means algorithm:

1. Partition the dataset into *k* nonempty subsets
2. Compute the seed points as the mean point or centroid of the clusters current partitioning
3. Assign each object to the cluster with the nearest seed point
4. Repeat from step 2 until the assignments do not change

K-means is often considered a greedy algorithm, but it is efficient running at *O*(*tkn*) where *n* is the number of instances, *k* is the number of clusters, and *t* is the number of iterations. K-means also has a few weaknesses including the number of clusters *k* needs to be specified ahead of time, it is sensitive to noisy data and outliers, and it can only be applied to objects in a continuous n-dimensional space.

#### K-medoids

K-Medoids is a partitioning approach to clustering in which each cluster is represented by one of the objects in the cluster. It is similar to the k-means approach, but instead of taking the mean value of the object cluster as the seed point, medoids, or the most centrally located object in a cluster, are used. In addition, while k-means only applies to objects in a continuous n-dimensional space, k-medoids can be applied to a wide range of data. K-medoids again suffers from weaknesses similar to k-means such as the number of clusters *k* must be specified ahead of time. K-medoids also does not scale well for large datasets due to the computational complexity of the algorithm.

The Partitioning Around Medoids (PAM) algorithm is used for k-medoids clustering. The algorithm works similarly to k-means except for the reassessment, which is completed as follows:

Start from the initial set of medoid and iteratively replace one of the medoids with one of the non-medoids to determine if it improves the total distance of the resulting cluster

### Hierarchical Approach

The hierarchical approach to clustering decomposes the set of data into a hierarchy using some criterion. It uses a distance matrix as the clustering criteria. Unlike the k-means and k-medoids partitioning approaches, this method does not require the number of clusters *k* to be provided; however, it does need a termination condition to be specified.

The hierarchical approach is not without its weaknesses. In creating the hierarchy a previous step can never be undone. The approach also has a time complexity of *O(n2)*, where *n* is the total number of objects, so the approach does not scale well.

#### AGNES

Agglomerative Nesting (AGNES) uses linkage and a dissimilarity matrix to cluster data. Nodes with the highest/lowest (depending on the linkage) criteria of dissimilarity are combined into a cluster, progressing in an ascending fashion, with all nodes eventually in one cluster.

##### Single Linkage

Single linkages uses the smallest distance between an element in one cluster and an element in another:

##### Complete Linkage

Conplete linkages uses the largest distance between an element in one cluster and an element in another:

### Density-based Approach

The density-based approach to clustering is based on specified connectivity and density functions. Some of the advantages to density-based approach over other approaches is it can handle noise, cluster discovered can be of arbitrary shape, and only one scan of the data is needed.

Density-based clustering work with two parameters:

Eps: Maximum radius of the neighborhood

MinPts: Minimum number of points in an Eps-neighborhood (NEps) of the point

Density-based approach also uses the concepts of density-reachable and density-connected. A point *p* is defined as density-reachable from a point *q* if there is a chain points such that *pi+1* is directly density-reachable from *pi*. A point *p* is defined as density-connected to a point *q* if there is a point *o* such that both, *p* and *q* are density-reachable from *o*.

#### DBSCAN

Density-Based Spatial Clustering of Applications with Noise (DBSCAN) clusters data into maximal sets of density-connected points. In spatial databases with noise, clusters discovered through DBSCAN will be of arbitrary shape. The DBSCAN algorithm is as follows:

Arbitrarily select a point *p*

Retrieve all points density-reachable from *p* w.r.t. *Eps* and *MinPts*

If *p* is a core point, a cluster is formed

If *p* is a border point, no points are density-reachable from *p* and DBSCAN visits the next point of the dataset

Continue until all of the points have been processed

# RESULTS

## Partitioning Approaches

For both partitioning methods, a *k* value for the number of partitions needs to be specified ahead of time. To determine the best value I used the Elbow method, which looks at the sum of squared error (SSE) within groups as a function of the number of clusters (see Figure 3). Looking for the bend of elbow in the plot gives a good indication of a value for *k*. In this case, 8 or 10 clusters seemed to be good bend/elbow locations so I used both those values for *k* in my analysis.

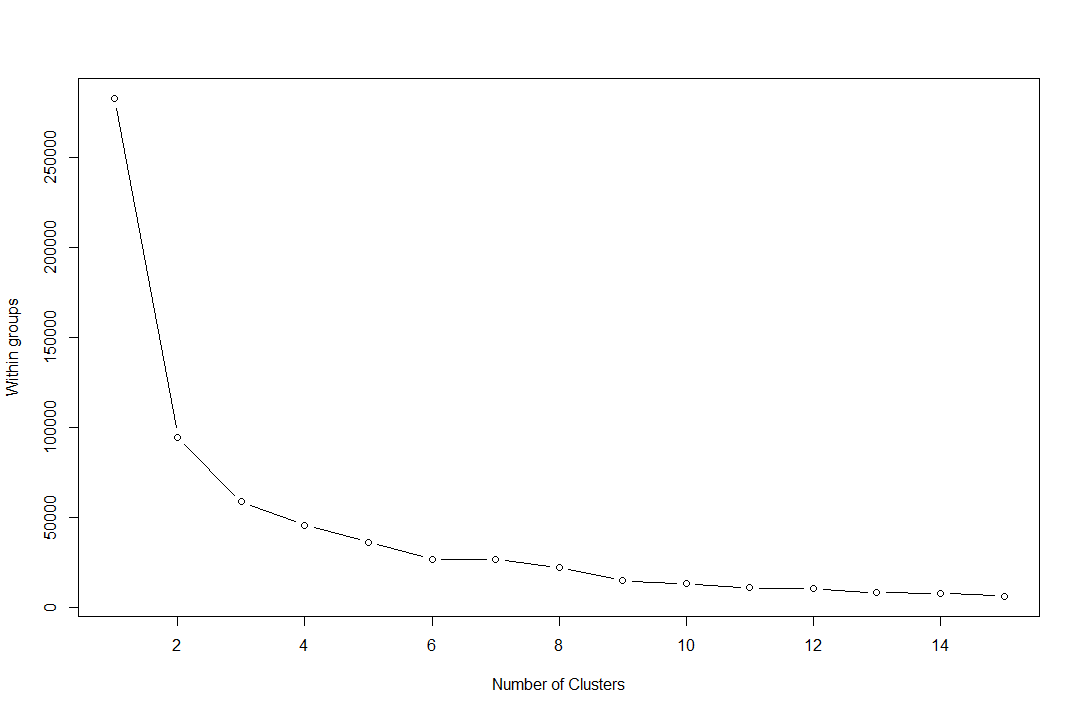


Figure 3. Number of Clusters to Determine Best *k* Value

### K-means

The Decision Tree from constructed from the training set was more complicated than I had expected (see Figure 1). When I looked at the below Figure 2 of the relative error and complexity point (CP) it the complexity of the tree made more sense. As the size of the tree grew, the CP continued to decrease as well as the relative error. One interesting part of the resulting tree was even with the increase in tree size, the algorithm still did not determine a great way to classify balanced scales.

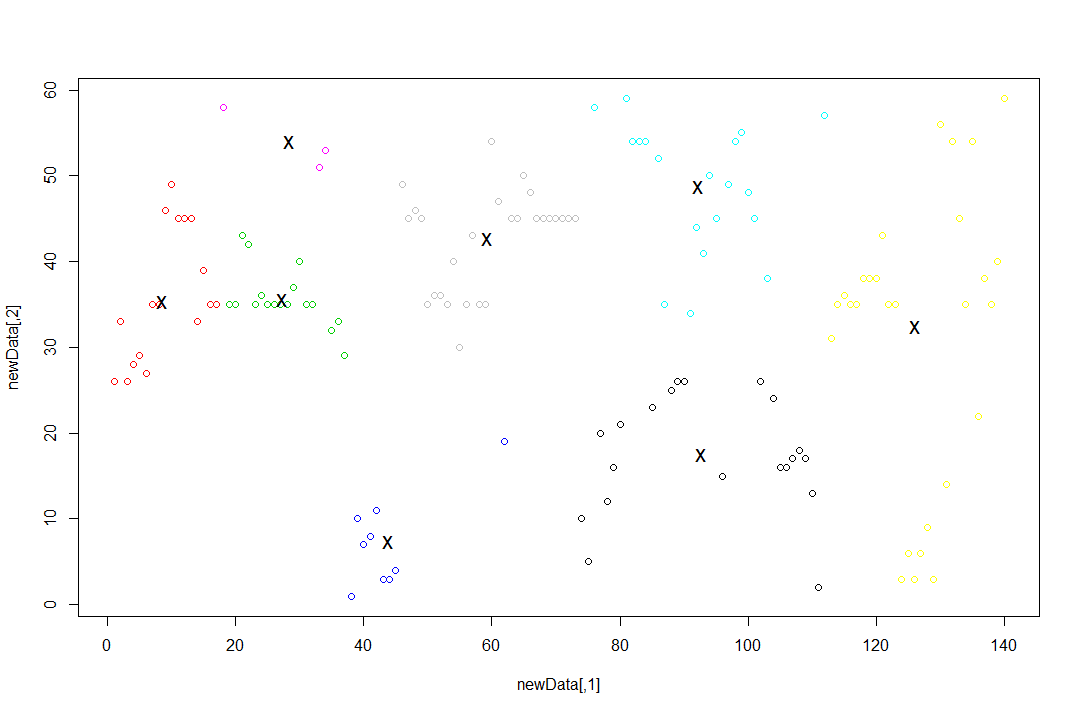


Figure 4. K-means for *k*=8

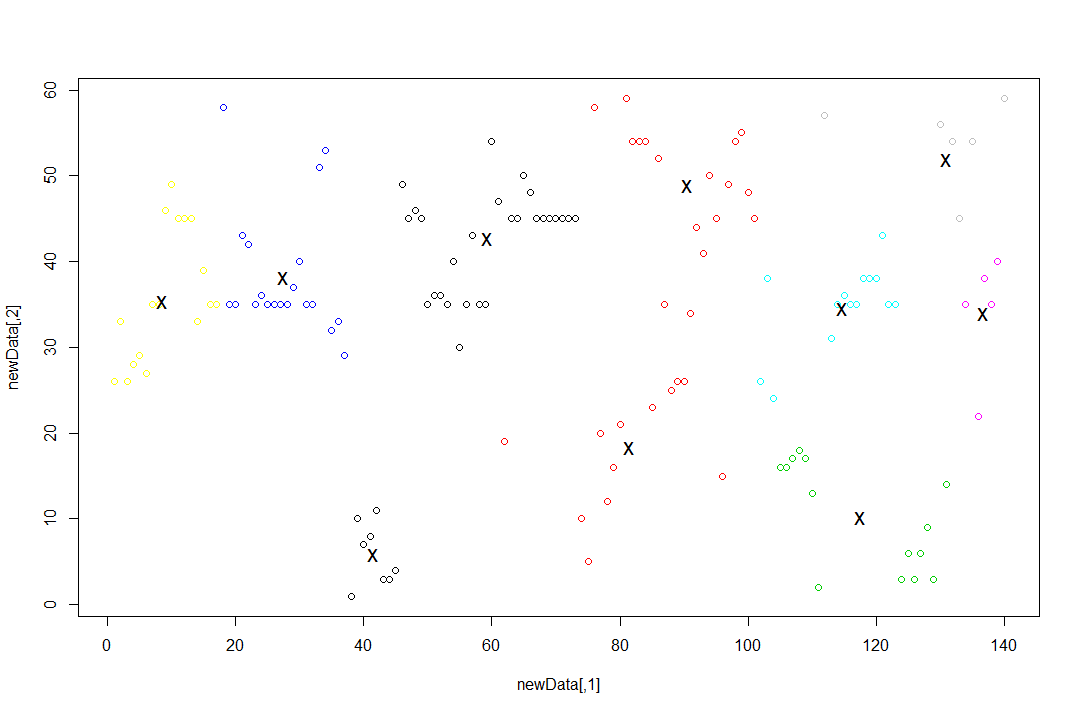


Figure 5. K-means for *k*=10

### K-medoids

It would make sense that this would lead to some error in classifying both left and right tipped scales as well; however, the numbers show those were not the only classification errors. There were additional errors in classifying left-tipped and right-tipped scales in addition to the balance classification errors.

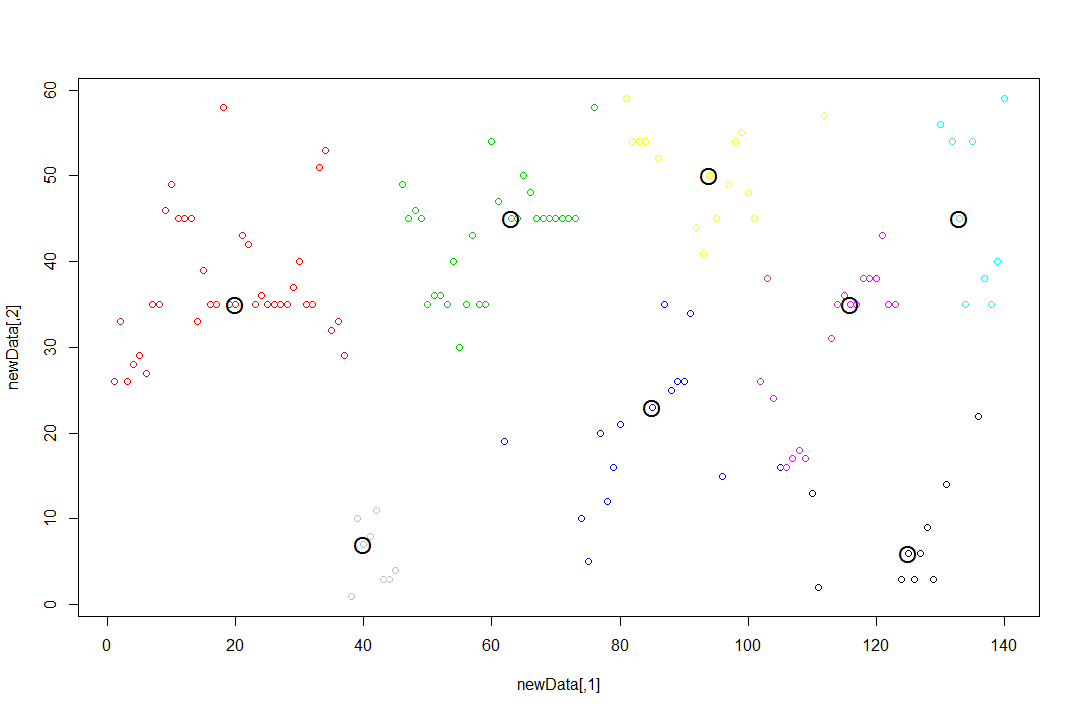


Figure 6. K-medoids for *k*=8

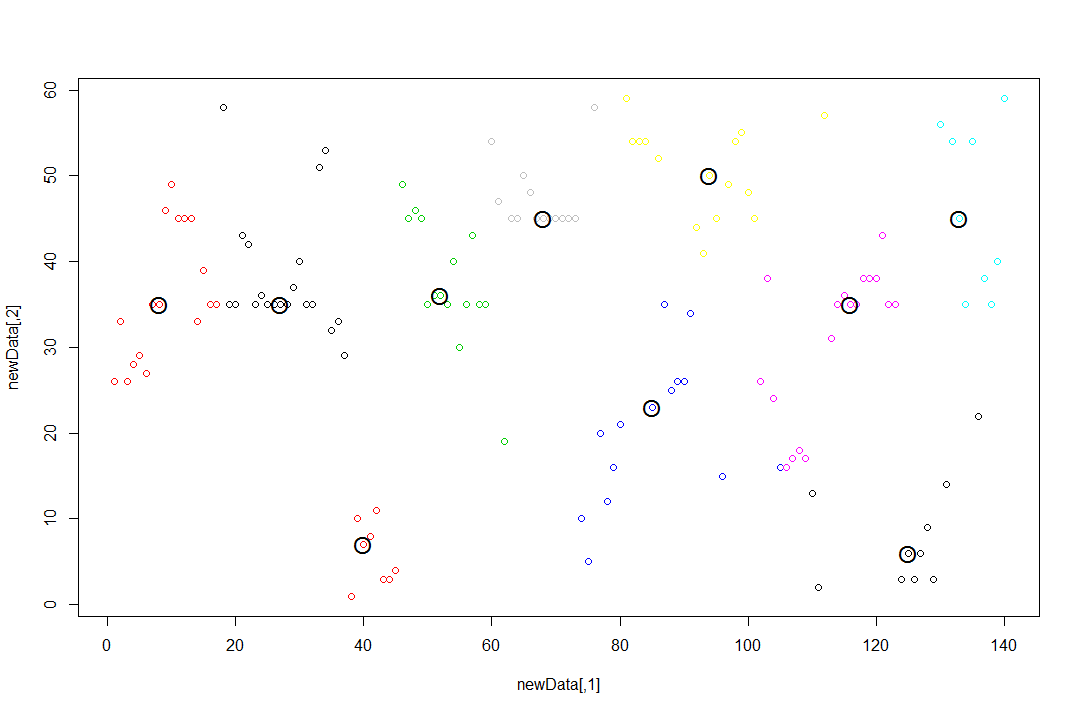


Figure 7. K-medoids for *k*=10

## Hierarchical Approaches

### Single Linkage

The Random Forest classification while similar to Decision Tree classification did produce different results. While there were still no true positive values for a balanced scale (see Table 3 below), there was lot higher percentage of true positives and conversely a lower percentage of false negatives/false positives overall.

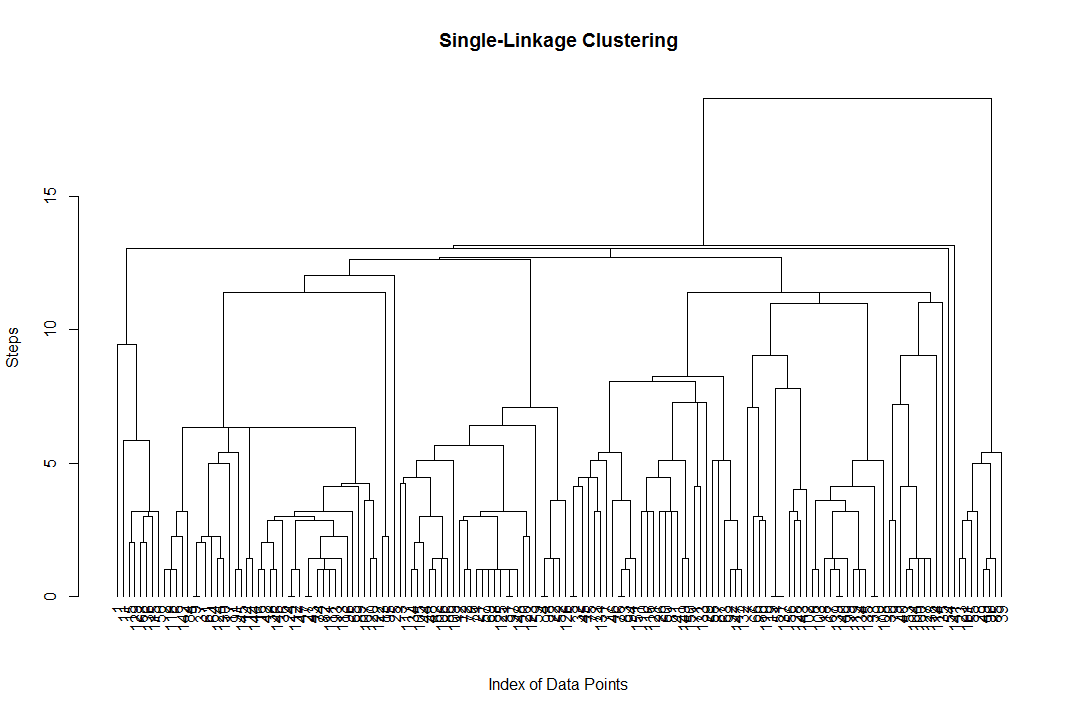


Figure 8. Single-Linkage Clustering

### Complete Linkage

As shown below, each of the four attributes used in the classification (right-weight, right-distance, left-weight, left-distance) have a fairly equal importance.

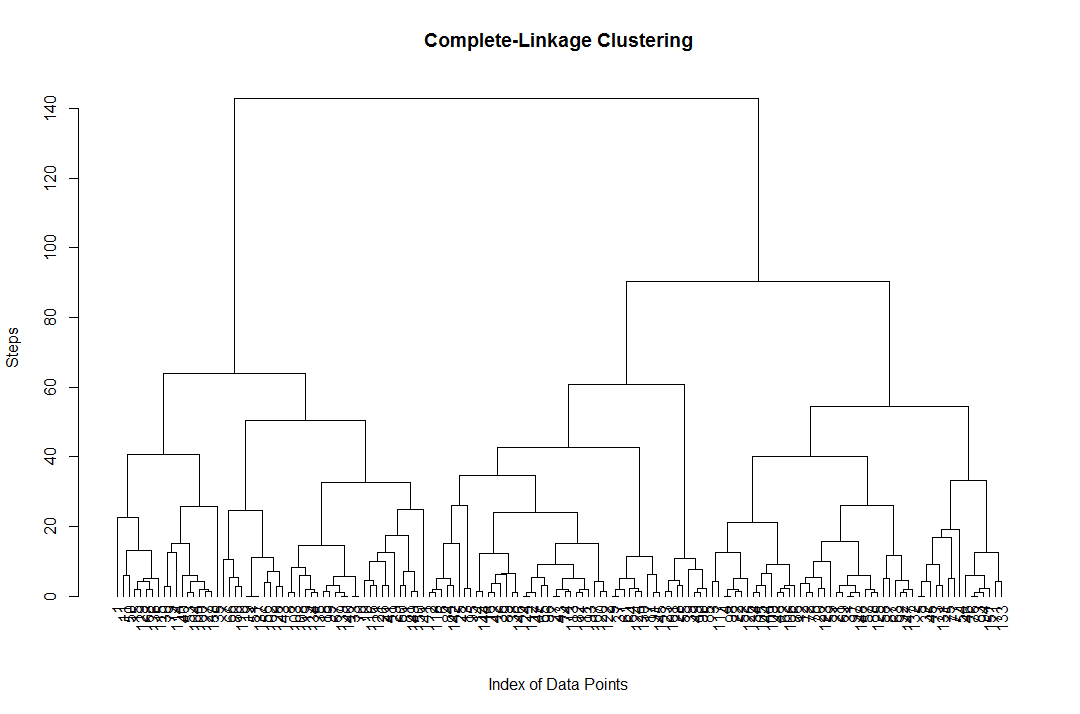


Figure 9. Complete-Linkage Clustering

## Density-based Approach

### DBSCAN

The Random Forest classification while similar to Decision Tree classification did produce different results. While there were still no true positive values for a balanced scale (see Table 3 below), there was lot higher percentage of true positives and conversely a lower percentage of false negatives/false positives overall.

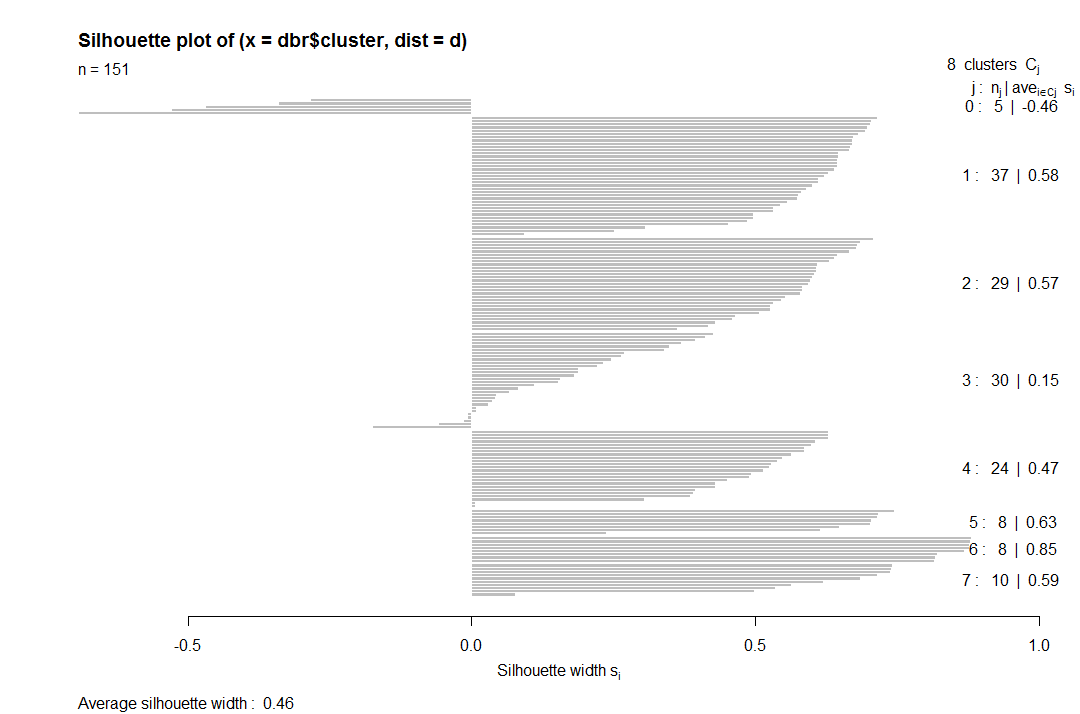


Figure 10. DBSCAN Silhouette Plot for ??=11.1

# CONCLUSIONS

## Evaluation Metrics

### Accuracy and Error Rate

### Sensitivity and Specificity

### Precision and Recall

## Decision Tree Classification Evaluation Metrics

## Naïve Bayes Classification Evaluation Metrics

## Random Forest Classification Evaluation Metrics

# REFERENCES

1. Larose, D. and Larose, C.D. *Discovering Knowledge in Data: An Introduction to Data Mining*. Wiley-Interscience, 2014.
2. Lichman, M. (2013). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science.
3. Moran, K.H., Wallace, B.C., and Brodley, C.E. Discovering AAAI Keywords via Clustering with Community-sourced Constraints. *AAAI Conference on Artificial Intelligence (AAAI), 2014*.