STAT30270 – STATISTICAL MACHINE LEARNING (OL)

STUDENT NO: 18201501

ASSIGNMENT – 2

Introduction:

The famous US Congress voting day at 1984 was taken for which the unsupervised classification is about to be conducted. The dataset holds 435 observations of US Congressman classified as Republican and democrat casting their votes based on the 16 attributes information's as mentioned below Figure1. Our motive is to perform clustering the given dataset into set of k-groups by scrutinizing the best value of k, by the most commonly used unsupervised machine learning algorithm namely K-Means clustering and K-Medoids. Thereby comparing both clustering to see how well the results agree by investigating the cluster stability and similarity between the two clusters using Silhouette and Rand Index measures.

Attribute	Attribute Information	Attribute	Attribute Information
1	Class name	9	Aid to Nicaraguan contras
2	Handicapped infants	10	Mx missile
3	Water project cost sharing	11	Immigration
4	Adoption of the budget resolution	12	Synfuels corporation cutback
5	Physical fee freeze	13	Education spending
6	el salvador aid	14	Superfund right to sue
7	Religious groups in schools	15	Crime
8	Anti satellite test ban	16	Duty free exports
17	Export administration act south Africa		

Fig1. Attribute information's on which Congressman (Democrat / Republican) express their agreement and disagreement

Data Modelling:

From the 435 observations of the US Congressman voted, 267 are found to Democrat and 168 to be Republican sharing 16 attributes as tabulated above with 45.2% Democrat and 58.2% Republican. Votes casted has been classified as 'Y', 'N' and '?' that represents the 'Yes', 'No' and Missing attributes which is neither 'Y' nor 'N'. On manipulating the '?' values across each attribute on one hand, is was found that attribute 17 (Export administration act south Africa) was having the maximum values of '?' with 104 observations and attribute 3 (Water project cost sharing) accounting second maximum with 48 observations. On other hand, attribute 11 (Immigration) expressed minimum values of 7 observation having '?' values.

Because, all the values of the attributes from 2 to 17 were represented in 'Y' and 'N', transformed the values to binary format with 1's and 0's. On considering, the congressman who failed to agree or disagree with the attribute information were treated as the ones who disagree to the same. Hence, the '?' value was modified and considered as 'N' for analysing the dataset.

K-Means clustering:

K-Means clustering is an iterative and partitioning clustering algorithm. It classifies the cluster into k clusters. It classifies the object into multiple groups considering that the data points within the cluster are quite similar, whereas the data points from different clusters are quite dissimilar.

Once, k-value (namely the number of clusters) is elected, we embark by randomly generating k cluster centers and calculate the centroids for each cluster and label the data point to the closest centers using distance measure (Euclidean distance by default). Continue the process and reassign each observation to its closest centroids. The algorithm stops, when no data points are moved between the groups. (Algorithm has converged to its local minimum)

Important challenge of selecting the number of clusters in K-Means is done by calculating the total within-cluster sum of square, which is the measure of goodness of the model and is expected to be minimum. On plotting the cluster values from the range of 1 to 10 against the total withiness sum of square, offers a way of selecting the right value of clusters (K) which is popularly known as elbow plot.

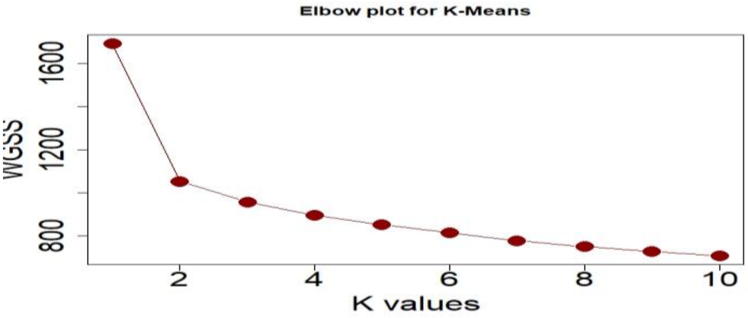
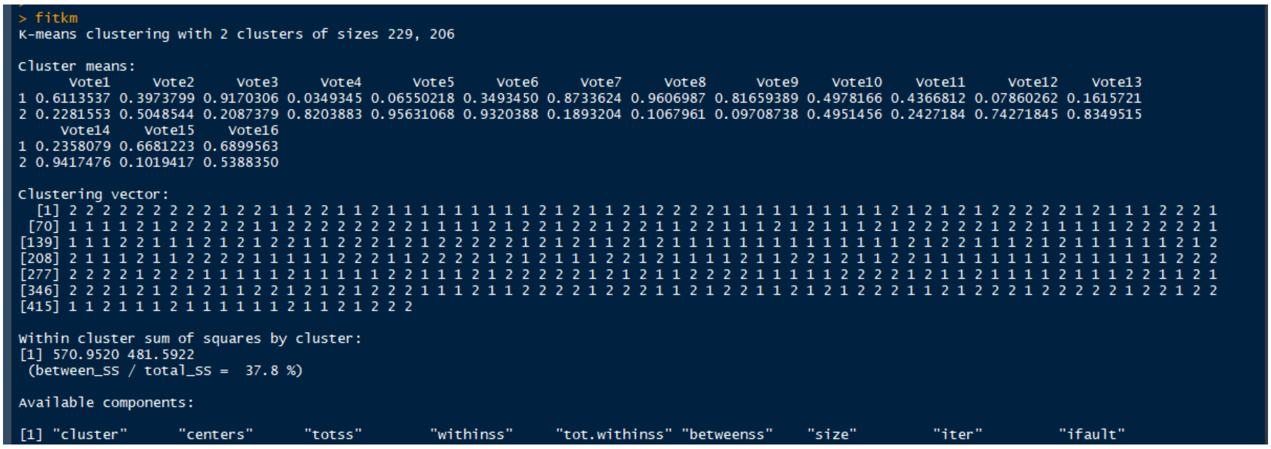


Fig2. Elbow plot for K-Medoids

Based on the Elbow plot, the k cluster values 2 and 3 are considered to be appropriate number of clusters because both 2 and 3 have location of bend in the plot. kmeans function in R is used for this computation with nstart value of 1000, which will try 1000 different starting assignments and select the best results corresponding to the one with local minima value. On one hand, analysing with the cluster value of 2, elects two centers from each column attribute and performs the k-means algorithm till the algorithm converges. Once done, we result with the cluster means, center and the clusters to which each data points belongs. Below is the result of k-means with cluster value of 2.

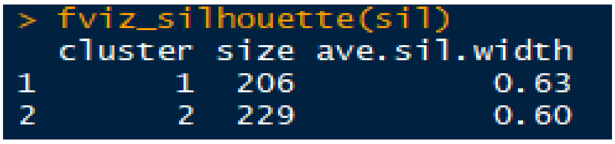


```
> fitkm
K-means clustering with 2 clusters of sizes 229, 206

Cluster means:
      Vote1     Vote2     Vote3     Vote4      Vote5     Vote6     Vote7     Vote8      Vote9    Vote10    Vote11     Vote12    Vote13
1 0.6113537 0.3973799 0.9170306 0.0349345 0.06550218 0.3493450 0.8733624 0.9606987 0.81659389 0.4978166 0.4366812 0.07860262 0.1615721
2 0.2281553 0.5048544 0.2087379 0.8203883 0.95631068 0.9320388 0.1893204 0.1067961 0.09708738 0.4951456 0.2427184 0.74271845 0.8349515
     Vote14    Vote15    Vote16
1 0.2358079 0.6681223 0.6899563
2 0.9417476 0.1019417 0.5388350

Clustering vector:
  [1] 2 2 2 2 2 2 2 2 2 1 2 2 1 1 2 2 1 1 2 1 1 1 1 1 1 1 1 1 2 1 2 1 1 2 1 2 2 2 2 1 1 1 1 1 1 1 1 1 1 2 1 2 1 2 1 2 2 2 2 2 1 2 1 1 1 2 2 2 1
 [70] 1 1 1 1 2 1 2 2 2 2 2 1 1 2 2 2 2 2 2 2 2 1 1 1 1 2 1 2 2 1 2 2 1 2 2 1 1 2 2 1 1 1 2 1 2 1 1 1 2 1 2 2 2 2 2 1 2 2 1 1 1 1 1 2 2 2 2 2 1
[139] 1 1 1 2 2 1 1 1 2 1 2 1 2 2 1 1 2 2 2 1 2 1 2 2 2 2 2 1 2 1 2 1 1 2 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 2 2 1 1 1 2 1 2 1 1 1 1 1 1 2 1 2
[208] 2 1 1 1 2 1 1 2 2 2 2 1 1 1 1 1 2 2 2 1 1 2 2 2 2 1 2 1 2 1 1 1 2 2 1 2 1 1 1 1 2 1 1 2 2 1 2 1 1 2 2 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 2 2 2
[277] 2 2 2 2 1 2 2 2 1 1 1 1 1 2 1 1 1 1 1 2 2 1 1 1 2 1 2 2 2 2 2 1 2 1 2 1 1 2 2 2 2 1 1 1 1 1 2 2 2 2 1 2 1 1 2 1 1 1 1 2 1 1 1 2 2 1 1 2 1
[346] 2 2 2 1 2 1 2 1 2 1 1 2 2 1 2 1 2 1 2 2 2 1 1 1 2 1 1 2 2 2 2 1 2 2 2 1 1 2 1 2 2 1 1 2 1 2 1 2 2 2 1 1 2 1 2 2 2 1 2 2 2 2 2 1 2 2 1 2 2
[415] 1 1 2 1 1 1 2 1 1 1 1 1 1 2 1 1 2 1 2 2 2

Within cluster sum of squares by cluster:
[1] 570.9520 481.5922
 (between_SS / total_SS =  37.8 %)

Available components:

[1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss" "betweenss"    "size"         "iter"         "ifault"
```

Fig3. Clustering solution of K-Means clustering

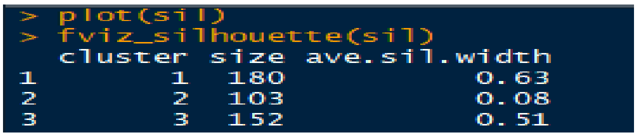
The US Congress voting data results into two clusters with size 229, 206 with cluster means of 0.611 and 0.2281 for the first attribute and corresponding two cluster centers for the remaining 15 classified attributes in dataset. The local minimum value / within cluster sum of squares is calculated to be 37.8%. On the other hand, on classifying the cluster with cluster value of 3, resulted in 3 clusters with size 180, 152 and 103 with cluster means 0.166, 0.701 and 0.475 for the first attribute and a within cluster sum of square value of 43.4% which is found to be greater than k-means with cluster value of 2. The measure of goodness of the model is revealed by the minimum value of within cluster sum of square. Hence, k=2 is found to better clustering for clustering the US congressman.

Here comes in the play of Silhouette, when you need to additionally confirm which number of clustering (K) does better with K-means algorithm. Silhouette plot is the measure to assess the coherence of cluster to identify is the cluster is very well clustered or wrongly clustered, whereas Silhouette value close to ‘1’ represents well clustering, ‘0’ represents data points lies on boundary of two clusters and ‘-1’ represents data point should be assigned to different cluster. Below are the plots of silhouette representing for K-means clustering using the k values of 2 and 3. Firstly, with 3 clusters, average silhouette width is 0.46 with cluster 1, cluster 2 and cluster 3 width 0.63, 0.08, and 0.51 respectively, which makes evident that data points of cluster 2 lies on the boundary of two clusters and has average silhouette width of 0.08. Secondly, with 2 clusters, average silhouette 0.63 with cluster one width 0.63 and cluster width 0.60. On comparing both k-values, k=2 reveals better clustering compared to k=3 with US Congressman data.



```
> fviz_silhouette(sil)
  cluster size ave.sil.width
1       1  206          0.63
2       2  229          0.60
```

Fig4. Silhouette width for 2 clusters for K-Mean clustering



```
> plot(sil)
> fviz_silhouette(sil)
  cluster size ave.sil.width
1       1  180          0.63
2       2  103          0.08
3       3  152          0.51
```

Fig5. Silhouette width for 3 clusters for K-M clustering

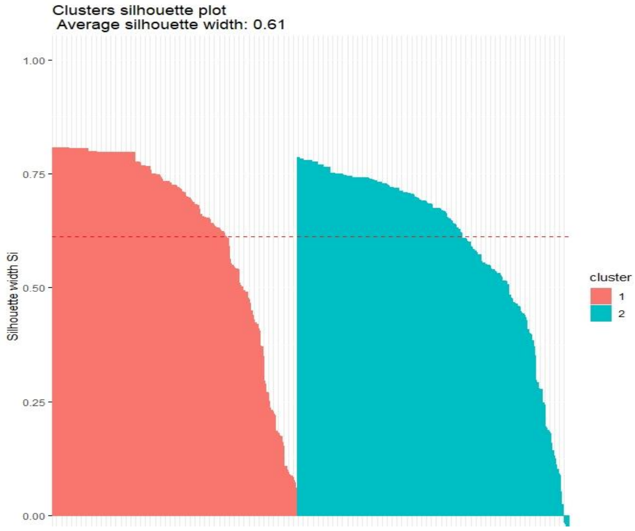


Fig6. Silhouette plot for k=2

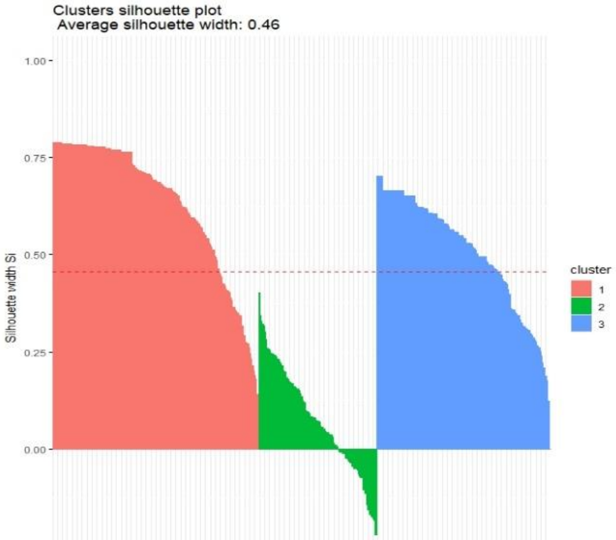


Fig7. Silhouette plot for k=3

K-Medoids:

Clustering algorithm similar to K-Means clustering but each cluster is represented by the data point in the cluster namely Medoids. Medoid is a data point that has minimum average dissimilarity between all other data points of that cluster. K-Medoids is better clustering method than K-means as it is less sensitive to outliers because it replaces the cluster means by cluster Medoids.

K-Medoids uses PAM(Partition Around Medoids) function to perform clustering returns the value of cluster Medoids, updates on the cluster each data points is classified and the its objective function (namely build and swap). After identifying the set of K-Medoids, clusters are constructed with attribute observation nearby to its Medoids. Every time, objective function is calculated when an selected data point is swapped by non-medoid data point is swapped. Objective function is the sum of dissimilarity of all the data points to its nearby Medoids. Algorithm stops, when objective function reaches minimal value.

There are many methods for dissimilarity calculations in k-medoids clustering such as Euclidean (for Euclidean distance between two vectors), Manhattan (Absolute distance between vectors), Binary (For Binary data). Since our data is of the binary format, I made use of the distance measure 'binary' which gave a swap value of 0.342. Even though the input data is binary, even with the distance measure Euclidean and manhattan K-Mediods was able to classify the data points with swap value of 1.771 and 3.537 respectively. It is exciting to inter that, though it was theoretically wrong use to distance measure other than 'binary' it was successful in classifying the data points practically. Finally it is evident that binary distance is the better means of clustering the binary data because of its minimum value of swap.

On the other hand, the minimal value of swap helps in identifying the choice of the number of cluster. Hence, on verifying the elbow plot with cluster values against the swap value for cluster k =1 to 10. It resulted in the same fashion as k-means.

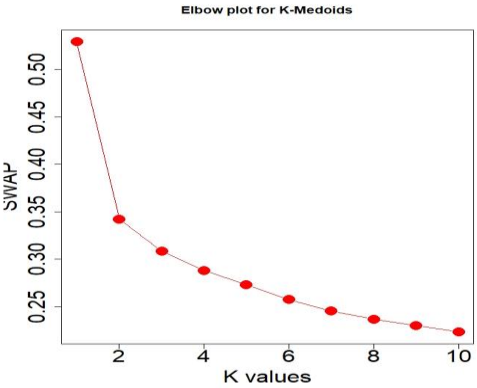


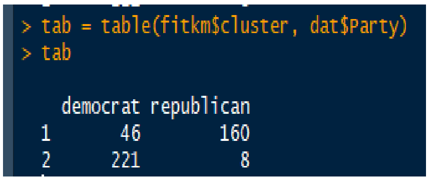
Fig8. Elbow plot for K-Medoids

From the elbow plot of K-Medoids, it seen that both k=2 and k=3, shows location of bend in the plot. So, we can classify the K-Medoid clustering with both values of k and chose the best after scrutinizing after measuring its similarity with the K-Means algorithm. With 2 cluster, we obtain two Medoids 234 and 26 with minimum sum of dissimilarity represented by objective function with build 0.3686 and swap value of 0.3421. Similarly with 3 cluster, we get three Medoids 234, 294 and 338 with same value of build and swap as 0.3083.

Using Silhouette plot, it is inferred that with 2 clusters and 3 clusters, the average silhouette width is 0.42 and 0.29 respectively. Thus, Silhouette reveals 2 clusters performs better clustering when compared to 3 clusters in K-Medoid clustering with US congressman data.

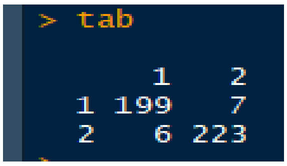
Misclassification:

The rate of misclassification of the clustering algorithm is constructing a cross tabulation with K-means clustering values and K-Medoids clustering values. From the table mentioned below, this is evident that out of 206 data points in cluster 1, 7 data points of cluster 1 are misclassified to cluster 2 and out of the 229 data points of cluster 2, 6 data points of cluster 2 are misclassified to cluster 1. Hence, the misclassification rate is 2% (0.029).



```
> tab = table(fitkm$cluster, dat$Party)
> tab

   democrat republican
 1       46        160
 2      221          8
```

Fig9. Disagreement between Congressman



```
> tab

       1   2
  1  199   7
  2    6 223
```

Fig10. Cross tabulation between K-means and K-Medoids.

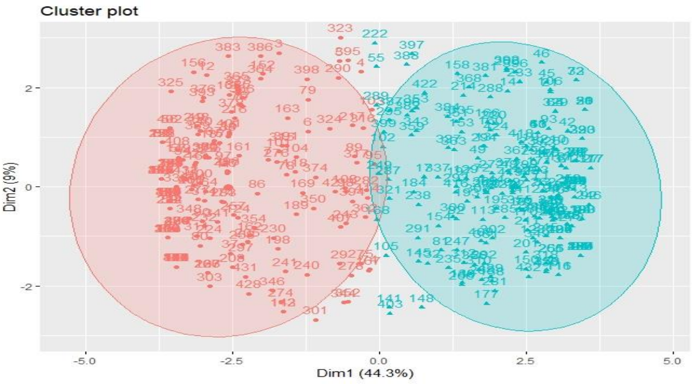


Fig11. Visualization of K-Means with 2 clusters

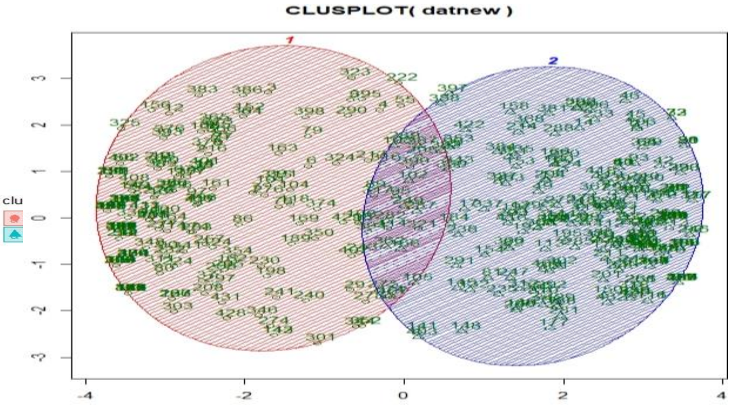


Fig12. Visualization of K-Medoids with 2 clusters

Rand-Index:

Rand-Index is used to identify the group agreement between the clustering algorithms. It ranges between 0 and 1, whereas near ‘0’ refers to little agreement and near ‘1’ refers to strong agreement. In our case, rand index value is 0.94 that intimates strong agreement between the two clustering algorithms. Adjusted Rand-Index does not the overlapping between the cluster values, it considers if the pair of elements lies on same cluster or different clusters. Hence, it is always less than Rand-Index value. Adjusted Rand-Index value for the data is 0.88, reveals higher agreement between the clustering algorithms.

Conclusion:

- Elbow plot assisted in making decision about number of cluster to chosen for clustering the agreement of democrats and republicans to the issues mentioned in Figure.1.
- Distance measure “Binary” helped in evaluating the dissimilarity measure between the data points since our dataset holds agreements in binary form.
- Silhouette guided in identifying the well clustering in k-Means and K-Medoids.
- Rand index value revealed the strong agreement of 94% between the clustering algorithms K-Means and K-Medoids.
- There is 12% disagreement between democrats and republicans on agreement about the 16 attribute information’s.