Project 4 – Clustering

CS548 / BCB503 / CS583 Knowledge Discovery and Data Mining - Fall 2019

Prof. Carolina Ruiz

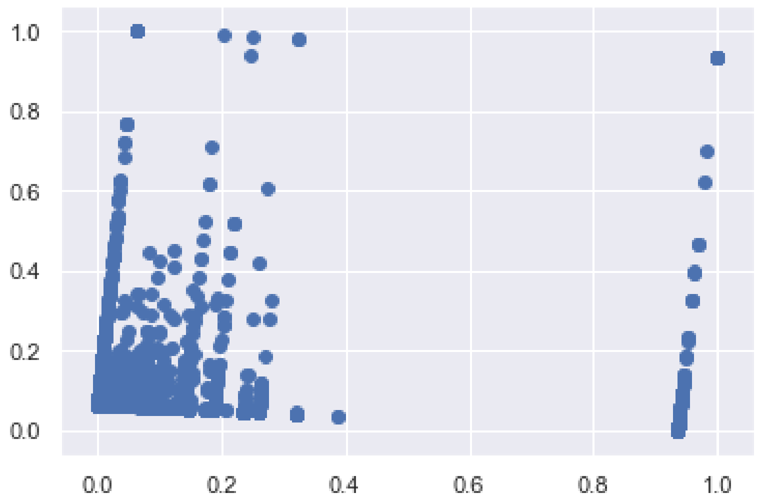
Students: Bhoomi Patel and Srinarayan Srikanthan

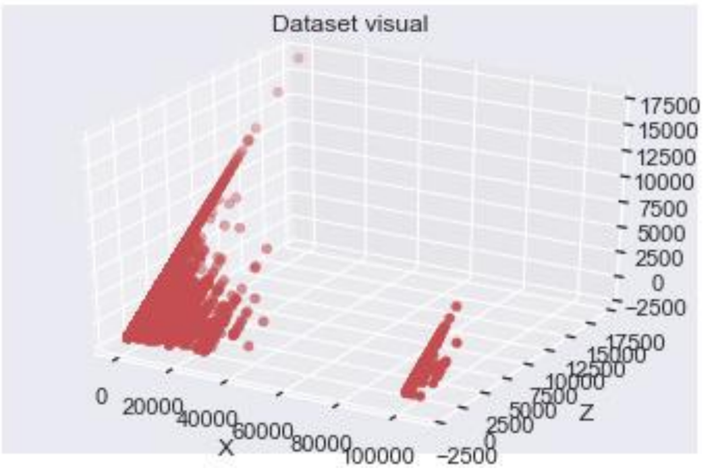
Dataset :	
● Dataset Description (not needed in this project)	
● Data Exploration	/05
● Initial Data Preprocessing (if any)	/05
Code Description:	/40
Experiments:	
Guiding Questions	/10
K-means - Sufficient & coherent set of experiments	/10
- Objectives, Parameters, Additional Pre/Post-processing	/10
- Presentation of results	/10
- Analysis of individual experiments’ results	/10
Hierarchical - Sufficient & coherent set of experiments	/10
- Objectives, Parameters, Additional Pre/Post-processing	/10
- Presentation of results	/10
- Analysis of individual experiments’ results	/10
DBSCAN - Sufficient & coherent set of experiments	/10
- Objectives, Parameters, Additional Pre/Post-processing	/10
- Presentation of results	/10
- Analysis of individual experiments’ results	/10
Quantitative Analysis of Results and Discussion	/45
Qualitative Analysis of Results, Discussion, and Visualizations	/45
Advanced Topic	/30
Total Written Report Project 4	/300 = /100

Dataset Description, Exploration, and Initial Preprocessing: (at most 1 page)

[05 points] Data Exploration: (e.g., comments on aspects of the dataset THAT ARE RELEVANT FOR CLUSTERING. This could include visualizations, issues with the data, and so on.)

The entire dataset when visualized in 2-d and 3-d space, there is a clear separation between the points after applying pca. Though the dataset was collected to classify the population into two buckets based on income, these separation in the points did not correlate to them.





The analysis for various guiding questions were performed on a sub-sample of the entire dataset for two primary reason. Firstly to not run into memory issues in computing distance between points, after PCA the points were very much concentrated in certain regions, in order to negate this effect of domination due to the concentration of points.

[05 points] Initial data preprocessing, if any, based on data exploration findings: (e.g., removing IDs, strings, necessary dimensionality reduction, converting attributes to numeric, scaling attributes if needed, and so on.)

Removed the entire column *instance weight*. Further there were missing values in columns state of previous residence, citizenship,country of birth self are represented as '?' and the column hispanic Origin has values NA. These values were converted to categorical values using the following: df [field]= df[field].replace('?', df[field].mode()[0]); df = df.fillna(df['hispanic Origin'].mode()[0]). Further using *sklearn.decomposition.PCA* to perform dimensionality reduction with *n_components=2* which helps in identifying patterns based on the correlation between features.

Code Description: Python Libraries and Functions you used and what parameters you experimented with. (At most 1.75 page.)

[05 points] Preprocessing Techniques for Clustering:

First, we used label encoding for *target* and *education* column using function *LabelEncoder()* and then, we *OneHotEncoded* all the nominal values using the function *pd.get_dummies(df) i.e.* OneHotEncoding with drop = 'first'. Later, scaled age using *scale()* function.

Libraries used: *pandas*, *sklearn.cluster.KMeans*, *sklearn.cluster.AgglomerativeClustering*, *sklearn.cluster.DBSCAN, sklearn.metrics.silhouette_score*

[05 points] K-means Clustering:

Apply the preprocessing techniques mentioned above. Then based on the guiding questions, use *KMeans* with *parameters n_clusters* (the no. of clusters to form), *init* (method for initialization), *n_init* (No. of times the k-means algorithm with different centroid seeds), *max_iter* (maximum number of iterations for single run), *random_state* (random number generation for centroid initialization), *algorithm* (KMeans algorithm to use). Using *fit* function fit the dataset and the using *predict* function compute closest cluster for each sample. Obtain SSE using *KMeans.interia_* and labels using *KMeans.labels_*. Finally compute *silhouette_score* with *parameters labels and dataset.*

[05 points] Hierarchical Clustering:

Apply the preprocessing techniques mentioned above. Then based on the guiding questions, use *AgglomerativeClustering* with *parameters n_clusters* (the no. of clusters to form), *affinity* (metrics to compute linkage), *linkage* (the distance to be usedl between sets of observations), *compute_full_tree* (stopping construction at n_clusters), *distance_threshold* (the threshold above which the clusters will not be merged). Using *fit_predict* function obtain cluster labels . Finally compute *silhouette_score* with *parameters labels and dataset.*

[05 points] DBSCAN:

Apply the preprocessing techniques mentioned above. Then based on the guiding questions, use *DBSCAN* with *parameters eps* (the distance between two samples), *min_samples* (no. of samples in a neighbourhood of a point), *metric* (the metric used to calculate distance), *algorithm* (the algorithm used to find nearest neighbors), leaf_size (leaf_size for the algorithm). Using *fit_predict* function fit the dataset and compute cluster labels for each sample. Finally compute *silhouette_score* with *parameters labels and dataset.*

[10 points] Quantitative Clustering Evaluation: including metrics listed on the project description and possibly others you used

Libraries used: *sklearn.metrics.silhouette_score, sklearn.metrics.cluster.adjusted_rand_score*, *sklearn.metrics.cluster.normalized_mutual_info_score, sklearn.metrics.cluster.adjusted_mutual_info_score*, *sklearn.metrics.homogeneity_score, sklearn.metrics.completeness_score, sklearn.metrics.v_measurescore*, *sklearn.metrics.cluster.contingency_matrix* *sklearn.metrics.silhouette_score* was used to measure internal indices for all the 3 types of clustering. Further, for K-Means, computed SSE by using *kmeans.interia_*. Relatives indices was computed by using adjusted_rand_score, normalized_mutual_info_score, adjusted_mutual_info_score and passing labels of two different experiments as parameters. External indices was computed using completeness_score, v_measure_score, homogeneity_score, contingency matrix and passing labels, target as parameters to the functions. For v_measure_score, a parameter beta is passed which is the ratio of weight attributed to homogeneity vs completeness

[10 points] Qualitative Clustering Evaluation: using Visualization, including MDS and at least one more visualization technique (e.g., heatmap of the correlation between proximity matrix and incidence matrix) you used

MDS:

Library used: *sklearn.manifold.MDS*, *matplotlib.pyplot.plt*.

Use *MDS* with *n_components=2* (no. of dimensions for dissimilarities), *dissimilarity* (the dissimilarity measure to use). Using *fit_transform*, convert the data into two dimensional space and obtain *X_new.* Use this *X_new* to visualize using *plt.*

HeatMap:

Library used: *sklearn.metrics.pairwise.pairwise_distances*, *seaborn.sns*

Append the labels obtained after clustering to the data and sort using the labels. Use *pairwise_distances* on the data to obtain a proximity matrix X. Thus, this way, the similarity matrix will be ordered with respect to the cluster labels. Use *sns.heatmap* with parameter X.

[10 points] Three Guiding Questions about the dataset domain that can be answered by Clustering methods (at most 1/4 page):

1. Are there any naturally occurring clusters based on the demography of the population?
2. Can we obtain clusters based on the female population?
3. Is there any natural separation between the Native and Immigrant population?

[40 points] Summary of Experiments with K-means. *At most 1 page.*

	Pre-process	**# clusters**	**# iterations**	**SSE**	**% of instances per cluster**	**Observations about experiment (e.g., observations from visualization, interpretation of centroids, analysis of similarity among instances in the same cluster)**	**Other Parameters used, Silhouette coefficient**
P1	OneHotEncoding, Scaling age	4	6	55876.7	38.60, 24.42, 18.60, 18.37	With the visualization, though 4 clusters were obtained, they were very close to each other and there were few overlapping points, which was reflected in the value of Silhouette Coefficient.	init=’kmeans++’, random_state=0. Silhouette coeff= 0.446
P1	OneHotEncoding, Scaling age	4	8	55881.5	38.61, 24.35, 18.85, 18.19	With initialization as random, the 4 clusters obtained were somewhat different from ‘kmeans++’. As the number of clusters increased, the SSE increased, thus proving that n = 4 is apt value for the no. of clusters	init='random', algorithm=’full’. Silhouette coeff= 0.46
P2	Filter only data with sex=Female	3	8	58691.8726	54.48, 42.95, 2.46	The clusters formed were compact, increasing the number of clusters resulted in clusters being clustered with some having only few instances in each of them.	algorithm=’auto’, random_state=0. Silhouette coeff= 0.3494
P2	Filter only data with sex=Female	3	8	58691.8729	54.32, 42.19, 3.17	By changing algorithm from full to auto, there was no evident change, but changing init from k-means++ to random changed the cluster labels of several points, though the visualization of the clustering was identical.	algorithm=”full”, random_state=0. Silhouette coeff= 0.3496
P3	Filter data based on the citizenship column	2	2	14657.1253	88.12, 11.88	On visualizing, we found that instead of 2, 6 different clusters were obtained which were very well separated from each other and hence the silhouette coefficient obtained for this was very high	init=’kmeans++’, algorithm=’auto’, random_state=0. Silhouette coeff= 0.9127
P3	Filter data based on the citizenship column	2	4	14413.7493	89.63, 10.37	With initialization as random, the algorithm performed bad i.e it formed two centroids on the same cluster which led to decrease in the silhouette score. Further, when we increased the no. of clusters, the SSE decreased, thus implying that actually 6 clusters are formed.	init=’random’’, algorithm=elkan, random_state=0. Silhouette coeff= 0.8778

[40 points] Summary of Experiments with Hierarchical Clustering (single link, complete link, average, Ward). *At most 1 page.*

	Pre-process	**# clusters**	**Link type**	**affinity**	**Time taken**	**% of instances per cluster**	**Observations about experiment (e.g., observations from visualization, analysis of nested clusters, analysis of similarity among instances in the same cluster)**	**Silhouette Coefficient**
H1	OneHotEncoding, Scaling age	20	single	euclidean	18 secs	79.7, 16.28, 3.66, 0.19, 0.045, 0.025, 0.02, 0.02, 0.01, 0.01, 0.01, 0.005, 0.005, 0.005, 0.005, 0.005, 0.005, 0.005, 0.005, 0.005	With n_clusters<20 and link type='single', the Silhouette Coefficient was going below zero, i.e negative, which indicated that the points are wrongly assigned to the clusters.	0.017
H1	OneHotEncoding, Scaling age	5	complete	cosine	30 secs	38.79, 33.15, 19.95, 5.615, 2.5	With complete linkage and affinity as cosine, the same guiding question gave good results with respect to Silhouette Coefficient, even with n_clusters = 5 stating that the clusters are somewhat separated to each other	0.25
H2	OneHotEncoding and filtering	6	ward	euclidean	20 sec	22.45, 21.68, 21.09, 15.33, 10.58, 8.83	Based on visualization from the dendogram it is evident that few points are very similar to each other but greatly different from the remaining points	0.50
H2	OneHotEncoding and filtering	6	complete	cosine	16 sec	39.56, 15.93, 15.33, 13.37, 10.57, 4.96	Changing the affinity to cosine and linkage to complete did not have a huge difference in the dendogram,but the distribution of the clusters changed, silhouette score went down	0.42
H3	OneHotEncoding and filtering	2	average	manhattan	16 secs	93.96, 6.04	As we increased the no. of clusters, the silhouette score increases. But with linkage='average', the results obtained weren't as good. WIth n_clusters = 6, the Silhouette Coefficient	0.88
H3	OneHotEncoding and filtering	2	ward	euclidean	18 secs	87.21, 12.79	Using euclidean distance with linkage='ward' increases the silhouette coefficient, thus indicating that linkage type ward is efficient enough to obtain well separated clusters.	0.90

[20 points] Summary of Experiments with DBSCAN *At most 1 page.*

	Pre-process	**Epsilon**	**minPts**	**# clusters**	**Time taken**	**% of instances per cluster**	**Observations about experiment (e.g., observations from visualization, analysis of core, border and noise points, analysis of similarity among instances in the same cluster)**	**Silhouette Coefficient**
D1	OneHotEncoding, Scaling age	0.16	4	2	11 secs	99.45, 0.55	For this guiding question,there was a region that was densely populated.Other methods separated this dense regions as number of clusters was specified, DBSCAN clustered all these points together	0.27
D1	OneHotEncoding, Scaling age	0.16	4	2	9 secs	98.86, 1.14	Changing the algorithm from auto to kd-tree and the distance metric to manhattan resulted only in the change of silhouette score, with clustering being identical	0.21
D2	OneHotEncoding and filtering	0.04	6	7	10 sec	24.49,22.32, 14.38,11.58, 10.05, 8.76, 7.06	Slight variation in the epsilon value obtained based on visualization using n-neighbors, had the drastic variation on the number of clusters that were being formed with,which implied that there were several points with distance between them in that range.	0.63
D2	OneHotEncoding and filtering	0.04	6	8	14 sec	22.91, 17.02, 13.72,11.86, 10.58, 10.52, 7.42, 6.12	Changing the distance metric to cosine distance greatly improved the silhouette score, and the clustering indicated that almost all the points were equally distributed among the clusters.	0.73
D3	OneHotEncoding and filtering	0.18	4	5	10 secs	87.215, 6.21, 3.03, 2.31, 1.25	Using ball-tree as the algorithm and manhattan as distance resulted in four clusters.Though this guiding questions target had two classes, choosing epsilon values from the visualizations did not yield two clusters.	0.98
D3	OneHotEncoding and filtering	0.18	4	5	9 secs	87.21, 6.075, 3.03, 2.31, 1.39	changing the algorithm provided no different clusterings, but changing the distance metric to cosin resulted in some points from the second cluster to be pushed on to the final cluster.	0.97

[45 points] Quantitative Analysis of Results and Discussion (at most 2 pages).

Include here: (1) Calculations or quantitative analysis you did to obtain good initial parameter values for K-means and DBSCAN; evaluation of clusterings using (2) internal indices, (3) relative indices and (4) external indices; and (5) other quantitative results across experiments and clustering methods. Explain your work.

(1) To obtain, good initial parameters for K-means and DBSCAN, we computed the best value for each of the guiding question using the elbow method which plots SSE against each value of k, for k-means and for DBSCAN, used *sklearn.neighbors.NearestNeighbors* where we fixed the no. of neighbors and based on the distance with the kth nearest neighbor, plotted the distance against dataset to obtain the best value of Epsilon.

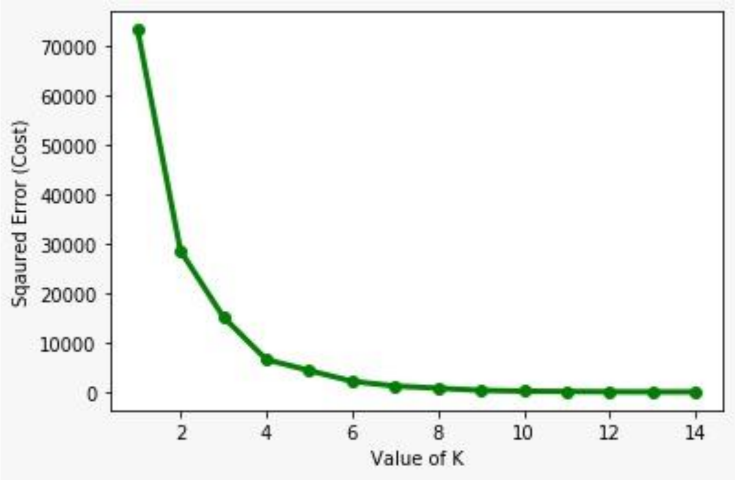


Fig 1.2: Eps for DBSCAN

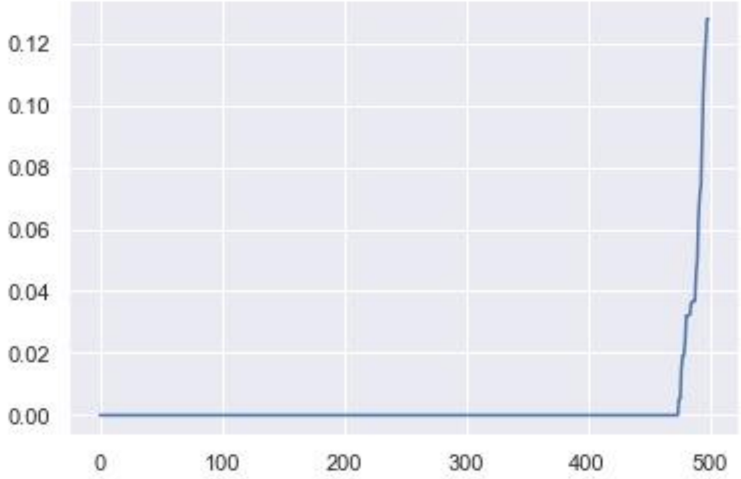


Fig 1.1: Elbow for kMeans

K-Means: Figure 1.1 represents an elbow graph for Guiding Question 1. As observed, there is an elbow in the graph after which the SSE starts decreasing in a linear manner. Hence, for the first guiding question, the value of k i.e no .of clusters was chosen 4 as the optimal value.

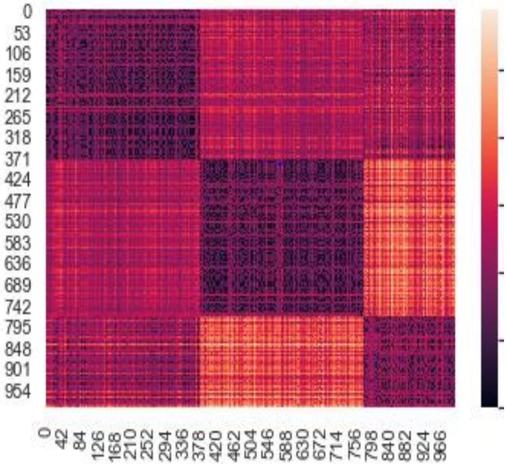
DBSCAN: Figure 1.2 represents a plot for Guiding Question 2. Here, the minPts i.e the no. of neighbors were fixed and values of epsilon were obtained. As minPts increased, the plot for epsilon started becoming a right angle saying there exists no elbow for such kth nearest neighbor. When the k value was fixed at 6, the epsilon value obtained was 0.04 as shown in the graph

(2) Internal Indices:

When the experiments were performed, it was observed that there is always a trade-off between SSE and Silhouette Coefficient.

K-Means: For this, used Sum of Squared Errors, heatmap and Silhouette Coefficient. The silhouette coefficient obtained for the 3rd Guiding Question was the best of all which implied that the clusters are well separated from each other. For other guiding questions the SSE and silhouette coefficient implied that though the clusters are formed, there are few data points which are overlapping and hence justified the valued obtained.

Agglomerative Clustering: Used heatmap and Silhouette Coefficient for measuring the internal indices. With heatmap (as shown below), it was observed that the data points which formed clusters were together and highly correlated. This algorithm performed badly on the 1st guiding question and was very sensitive to the change in the affinity and linkage. There was a drastic change for silhouette coefficient when the affinity and linkage values were changed.



DBSCAN Clustering: Again heatmap and Silhouette Coefficient was used for measuring the internal indices. DBSCAN was able to correctly predict 6 well separated clusters was guiding question 3, wherein the silhouette coefficient was very good.

(3) Relative Indices:

Relative indices between two clusters with the same clustering method: Here, relative indices when calculated on the 1st guiding question for kMeans as clustering method, the values obtained were: adjusted_rand_score = 0.86, normalized_mutual_info_score = 0.85, adjusted_mutual_info_score = 0.85

Relative indices between two different clustering methods: Here, relative indices was calculated on the 3rd Guiding Question using K-Means and Agglomerative Clustering. The values obtained were: adjusted_rand_score = 0.92, normalized_mutual_info_score = 0.89, adjusted_mutual_info_score = 0.91

(4) External Indices:

The 3rd Guiding Question was built with respect to target wherein the target used was *citizenship* column. When external Indices was computed using the clustering method used as DBSCAN. The various values obtained were: homogeneity_score = 0.76, v_measure_score = 0.36, completeness_score = 0.35

When the clustering method used was K-Means, the various values obtained were: homogeneity_score = 0.78, v_measure_score = 0.38, completeness_score: 0.37

With the above analysis it was observed that all the three external indices values obtained with different clustering were very similar in terms of homogeneity and completeness, thus stating that the data instances were associated to the similar clusters using different clustering methods.

When contingency matrix was obtained for the above, it was observed that the maximum data points were clustered correctly, i.e. the number of samples in true class and in predicted class were maximum for that number of samples in true class

(5) Other Quantitative results:

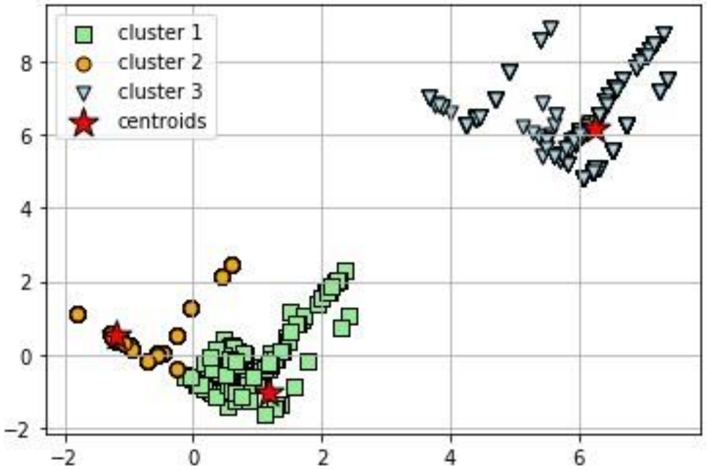
To emphasize on how well the clusters are separated with respect to target and how well they are defined, used *sklearn.metrics.fowlkes_mallows_score*, *sklearn.metrics.calinski_harabasz_score.* With this, we obtained high values for Guiding Question 2 and 3, For eg: For Guiding Question 3, values obtained were: fowlkes_mallows_score = 0.92, calinski_harabasz_score = 10753586.50 which further supported our conclusion that the clusters are well separated. Further for Guiding Question 1, when this measure was calculated with initialization method as *random*, the values obtained weren't good, thus concluding that the with random initialization, the clusters obtained weren't good in terms of separation as well in terms of readability.

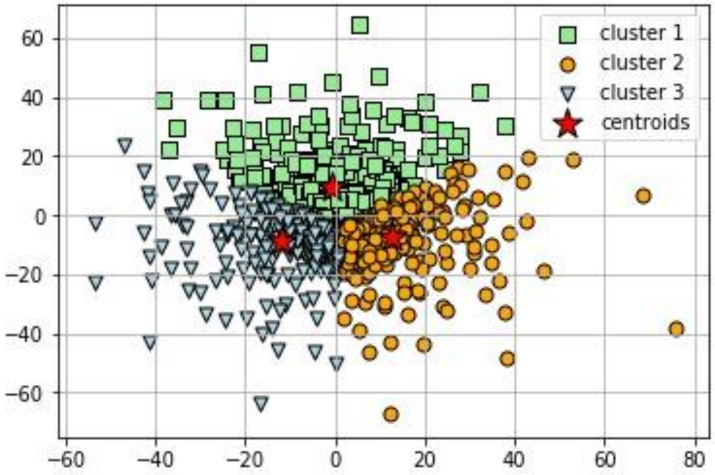
[45 points] Qualitative Analysis of Weka and Python Results on and Visualizations (at most 2 pages)

Include here (1) visualizations of the clustering using MDS and other visualization methods; (2) inspection of the actual clusters' members to find similarities among data instances in a cluster and dissimilarities with data instances in different clusters; (3) additional analysis of the results from the point of view of the dataset domain; and (4) answers that the experiments provided to your guiding questions.

(2) Inspection of the actual clusters' members to find similarities among data instances in a cluster and dissimilarities with data instance in different clusters:

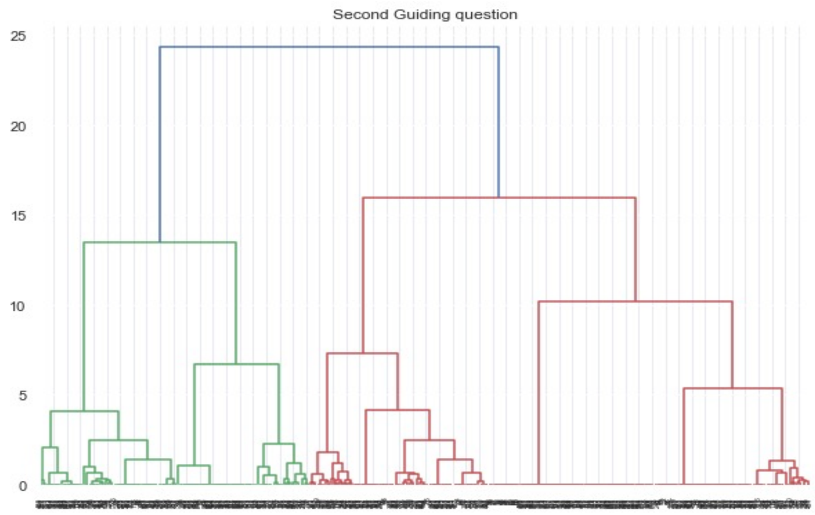
The similarity and dissimilarity among the data instance, can be computed based on the SSE obtained and the Silhouette coefficient. The image represents the visualization for the Guiding Question, obtained with K-Means Clustering. As seen, the data instances were clustered into three different sets, which implied that even though the SSE was high, there were well separated naturally occurring clusters. The Silhouette coefficient obtained started increasing as we changed the clustering technique from K-Means to Hierarchical and then to DBSCAN. With DBSCAN it was observed that the points were densely populated and with change in distance metrics, better clusters were formed. Similarly, for the data instances based on demographics, it was observed that the data instances within the cluster were not that similar to each other which was clearly depicted in the low Silhouette score obtained combined with the SSE value.

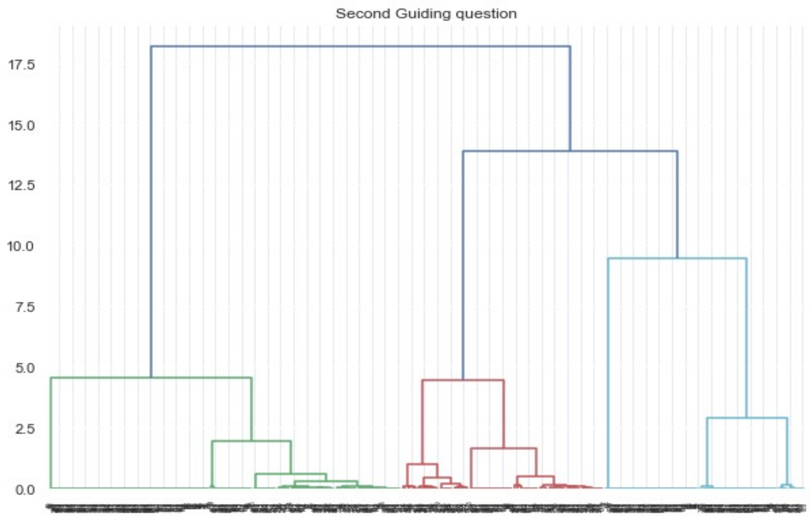




(3) Additional analysis of the results from the point of view of the dataset domain:

Though the initial visualization of the dataset on two and three dimension showed a clear separation in terms of two clusters, these were not actually in accordance with the target variable which had two different classes of income. For the third guiding question, which had two target classes, was actually well separated into 6 different clusters that were well separated and compact on all the three methods. This clearly proved that the points in each cluster were very similar to each other and different from other clusters.On analyzing the samples in each of these clusters. One interesting finding was that most of the immigrants were clustered into two clusters, with most of the members in the cluster having state of previous residence as California. For the second guiding question all the methods provided consistent number of clusters which suggested that there was a natural separation among the population that was being analyzed. Looking into the samples from one of the clusters, it was found that it had samples with income as more than 50000 and all had a qualification of at least a high school graduation. Going by this pattern, it was expected that even the male population would have such a clustering, but that was not the case. From this an interesting observation was made, women with a degree are more likely to make more money than a male.



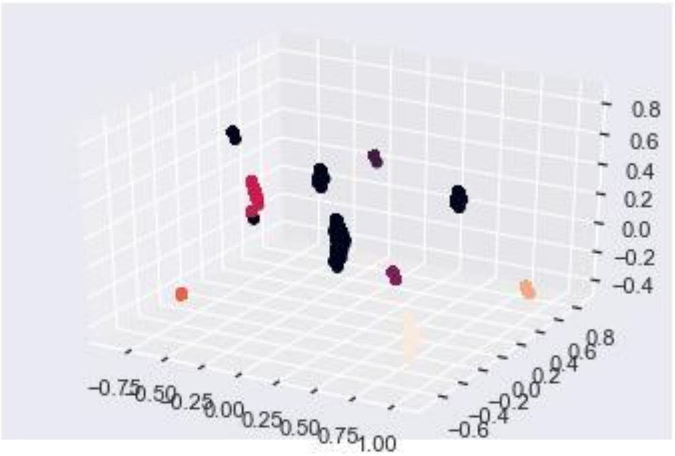


(4) answers that the experiments provided to your guiding questions:

Guiding Question 1: On running various experiments on the Guiding Question 1 with algorithms such as K-Means, Hierarchical and DBSCAN, it was observed that the clusters were not well separated from each other. Hence, stating that it was difficult to obtain clusters based on the demographics of the population. Thus, the population had mixed demographics which were not well clusterable.

Guiding Question 2: Here, we decided to see if there are any naturally occurring clusters for the female population. It was observed that the data points were densely populated and we obtained on average 7 clusters by varying the method of clustering. The best cluster was obtained with DBSCAN, wherein the data instances were highly similar to each other in their neighborhood.

Guiding Question 3: For Guiding Question 3, we thought of finding clusters based on citizenship as target. Further, we narrowed our dataset to concentrate just on two types of citizenship i.e Native-Born in the United States and Foreign born- Not a citizen of U S, and see if we obtained two naturally occurring clusters for them. It was observed that though the clusters were well separated, there were not divided into 2 but 6 different well separated clusters, thus implying that there were other attributes that contributed to a person’s citizenship status.



Note: For k-means and hierarchical clustering visualization is done after using PCA and MDS

Advanced Topic: CLIQUE algorithm for Subspace Clustering

[7 points] List of sources/books/papers used for this topic (include URLs if available):

- Lance R Parsons, Ehtesham Haque, Huan Liu, Subspace clustering for high dimensional data: a review, Published in SIGKDD Explorations 2004 (https://www.kdd.org/exploration_files/parsons.pdf)
- Jyoti Yadav, Dharmender Kumar, Subspace Clustering using CLIQUE: An Exploratory Study, IJARCET 2, February 2014 (http://ijarcet.org/wp-content/uploads/IJARCET-VOL-3-ISSUE-2-372-378.pdf)
- Madalina Ciortan, Subspace clustering, TowardsDataScience blog, April 7 2019 (https://towardsdatascience.com/subspace-clustering-7b884e8fff73)

[20 points] In your own words, provide an in-depth, yet concise, description of your chosen topic. Make sure to cover all relevant data mining aspects of your topic.

A high dimensional data consists of inputs having n number of features. And hence, to overcome this, we only have to rely on dimensionality reduction techniques. Feature selection is one of the possible solutions, but with that we cannot get rid of the redundant dimensions. A better idea is to use Subspace clustering wherein we find clusters from all the subspaces. To obtain such type of clusters, we use CLIQUE (Clustering In QUEst) algorithm which partitions each dimension into the same number of equal length intervals and also converts high dimensional data space into non-overlapping rectangular units. It does so by finding any number of arbitrary shape clusters in any number of dimensions using Depth First Search (DFS) algorithm. Moreover, this number is not predetermined by any parameter. The algorithm takes input as the number of bins or the grid size and the minimal density and works as follows:

1. Divide each dimension into equal interval i.e D_k
2. Fix minimum input
3. Find dense units i.e set of all k-dimensional dense units belonging to lower dimensional projections D_{k-1}
4. Determine high coverage subspaces
5. Identify clusters by computing $E = E - Cm$ where E is set of all dense units in Subspace S
6. Generate minimum clusters C
7. Remove all covers of cluster whose units are covered by at least another cover

The hyper rectangular clusters are defined by a Disjunctive Normal Form (DNF) which helps presents them in easily interpretable ways. The clusters could be overlapping which implies that instances can belong to more than one cluster. This is advantageous because clusters often exist in different subspace and thus represent different relationship. Though Clique Clustering is highly sensitive to the input parameters which can lead to very different results yet it is still an essential algorithm in the family of bottom-up space clustering.

[3 points] How does this topic relate to clustering?

In this course, we discussed various techniques of clustering such as k-Means, Hierarchical, etc. Further we even discussed Multidimensional Spacing as a technique to visual high dimensional data. Since CLIQUE clustering is also a technique to obtain clusters in high dimensional space, this topic is highly related to Clustering.

Authorship: The initial Data Exploration relevant to clustering was done by Sri whereas visualizing the dataset to obtain correlation was done by Bhoomi. Both members came up with different guiding questions and both built their code, ran experiments and then based on the results, merged the code to obtain the best output for the guiding questions. Mutually decided which would be the best part of the model to incorporate in the report and it was Clustering in High Dimensional Space concept that motivated Sri to choose the advanced topic. Based on the topic, Bhoomi came up with the concept of CLIQUE algorithm for Subspace Clustering.