Projection and Clustering of the US Census Income Dataset

TEAM CENSUS, Middle East Technical University, TR
RAHEEM HASHMANI and DENIZ GERMEN

We attempt to analyze a sample of the U.S. Census Bureau Income Dataset by projecting its data onto 2-dimensions and forming clusters, both to match the original data labeling and to determine additional, unlabeled clusters. We perform various linear and nonlinear projections, which helps us visualize potential clusters, numerous clustering methods for various number of clusters, and validation tests to determine the quality of the generated clusters. We find that while 2 clusters seem to be the optimal number, the clustering does not occur with respect to the given data labeling, but rather to an unknown, complex relationship within the dataset. We conclude that this dataset is too complex for standard clustering algorithms and that more sophisticated ones are necessary to explore the relationships present within this dataset.

CCS Concepts: • **Mathematics of computing → Exploratory data analysis**; **Dimensionality reduction**; **Cluster analysis**; • **Computing methodologies → Model verification and validation**.

Additional Key Words and Phrases: PCA, MDS, t-SNE, UMAP, Hierarchical, k-means, k-medoids, SOM.

1 INTRODUCTION

In this report, we analyze the U.S. Census Bureau Income Dataset taken from Data Science Dojo [15].

This dataset was chosen because it has a large number of datapoints, 14 different variables, 2 labels with the possibility of having more than 2 clusters, and because of personal interests in how various factors might affect the average income of an American citizen.

We first conduct a preliminary analysis where we try to find more information about the raw data, such as the data types of the various variables and the number of unique values they contain. Then, in order to project and visualize our dataset on 2 dimensions, we perform Principal Component Analysis (PCA), multiple versions of Multidimensional scaling (MDS), t-distributed Stochastic Neighbor Embedding (t-SNE), and Uniform Manifold Approximation and Projection (UMAP) to try and visualize our dataset in 2 dimensions. Following this, we try to determine the number of possible clusters within the dataset using multiple hierarchical clusterings, k-means and k-medoids clusterings, and k-means clustering applied on a Self-organizing Map (SOM). While the labels tell us there are 2 clusters, we feel this might be a generalization given the 48,842 datapoints, and strive to see if more good clusters can be formed. Finally, we perform evaluations on the the formed clusters to test their stability and internal and external validity.

Towards the end of the report, we discuss and analyze our results and give a brief conclusion of our findings. Additional results, plots, and codes are available at https://github.com/RKHashmani/IncomeDataVisualization [6].

2 DATA

In this dataset, the U.S. Census Bureau surveyed 48,842 people and recorded 15 attributes including their ages, working class, years of education, and income, among other things. This dataset in its current form was compiled by Ronny Kohavi [9] and is designed to be used for training and testing various classifiers, with 14 attributes as features and the last attribute, income level, as the class, being either above or below $50,000 USD (50K).

This dataset has previously been used for various statistical analysis purposes, such as clustering aggregation [5] and anomaly detection [12].

Authors' addresses: Team Census, Middle East Technical University, Universiteler Mh., Dumlupinar Blv. No:1, Ankara, Cankaya, TR, 06800; Raheem Hashmani, hashmani.raheem@metu.edu.tr; Deniz Germen, deniz.germen@metu.edu.tr.

Table 1. Column labels, their type, number of unique values, range (if applicable), and representation type for the entire dataset.

Column Label	Type	Unique Values (Range if Applicable)	Representation
age	Discrete	73 (17 - 90)	Ratio
workclass	Qualitative	9	Nominal
fnlwgt	Discrete	21648 (12285 - 1490400)	Ratio
education	Qualitative	16	Ordinal
education-num	Discrete	16 (1-16)	Ratio
marital-status	Qualitative	7	Nominal
occupation	Qualitative	15	Nominal
relationship	Qualitative	6	Nominal
race	Qualitative	5	Nominal
sex	Qualitative	2	Nominal
capital-gain	Discrete	119 (0 - 99999)	Ratio
capital-loss	Discrete	92 (0 - 4356)	Ratio
hours-per-week	Discrete	94 (1 - 99)	Ratio
native-country	Qualitative	42	Nominal
income	Qualitative (Classes)	2	Ordinal

Due to computational limitations, we sample 5,000 datapoints and use that as our sample dataset for most of this paper. Since many of the methods we use have complexities of $O(n^2)$ or $O(n^3)$, which require a lot of memory and processing time, 5000 datapoints were deemed to be a good balance between processing time and detailed results.

2.1 Preliminary Analysis

In the preliminary analysis, we took a peak at the raw data, discerned the the type of data and the format that it is represented in, and determined the number of unique values and the ranges for each variable. The results of this is shown in Table 1. As we can see, there are 14 features/variables and 2 classes in the *income* column. In addition, a class imbalance was detected, with there being approximately 3.2 times more "<=50K" classes then there were ">50K" classes.

2.2 Data Preparation

Due to our dataset being very large (containing 48,842 datapoints), having mixed data types (discrete and qualitative data types), and containing many outliers (mostly in the ">50K" class), it was necessary to conduct data cleaning and preparation measures. Additionally, we removed two variables from each of the datapoints as well. *fnlwgt* told us the approximated number of people in USA that would fit that particular datapoint's set of variables. This did not contribute to the dependent variable (*income*) and was added for statistical purposes, and was therefore removed for our case. *education* was merely the categorical version of the *education-num*, and was thus superfluous.

To deal with the mixed data types, we used label encoding to assign numbers to the qualitative values. One-hot encoding was attempted, and while the processing time increased, the results were largely unchanged, and thus one-hot encoding was not used. Datapoints with incomplete information or that were duplicates of other datapoints were discarded. Without the latter process, the distance matrices would contain negative or zeroed values, which causes errors for many of the MDS algorithms (such as Sammon's Non-Linear Mapping).

To remove outliers, multiple methods were considered, including the Interquartile (IQR) range method, the Z-score method, the Mahalanobis distance method and Cook's distance method (the latter two being famous for multivariate data). Ultimately, the IQR method showed the best results. It takes the difference between the 75th percentile and the 25th percentile (the interquartile range) of each column (variable) and removes datapoints where a variable's value is 1.5× IQR greater than the third quartile or less than the first quartile. One understandable outcome of this was that almost all the outliers had nonzero *capital-gain* and *capital-loss* values, which represent gain/loss by investments. Thus, after the outlier removal process, all the remaining datapoints had zero values for those 2 variables. We therefore decided to remove them as well, leaving only 10 variables, plus the class column. Figure 1 shows boxplots for the original, pre-sampled dataset before and after the outlier removal, with scaling done to better unify the ranges across the variables.

Finally, to deal with the large dataset size, 5000 samples were randomly chosen such that there would be a 50:50 class balance and the final column, *income*, was removed as it was the "class" column.

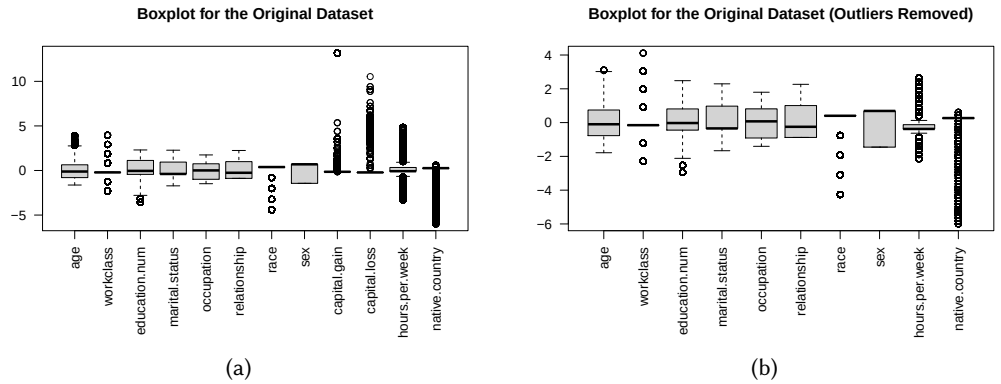


Fig. 1. Boxplots for the original, pre-sampled dataset (a) before and (b) after the outlier removal. Scaling was performed in order to unify the ranges across the variables. We can see that the range has decreased, owing to the extreme values-having outliers being removed.

3 METHODS

3.1 Projection

We first apply Principal Component Analysis (PCA) and 4 different Multidimensional Scaling (MDS) algorithms to better visualize our dataset. For the former, an online tutorial was used to get experience with properly coding PCA in R [10]. PCA was done for both the complete, original dataset and the sampled dataset. In addition, Scree Plots were plotted in order to to get a better understanding of the PCA. For MDS, Classical multidimensional scaling (Torgerson's MDS) of a data matrix, Sammon's Non-Linear Mapping, Kruskal's Non-metric Multidimensional Scaling, and Symmetric Smacof were used, with another online tutorial being used as a general practical guide [4]. Following this, we then apply nonlinear dimensionality reduction techniques, namely Uniform Manifold Approximation and Projection (UMAP) and t-distributed Stochastic Neighbor Embedding (t-SNE), using [13] and [16] as practical guides, respectively. Hyperparameter tuning was performed using grid search and the best resulting plots were selected.

3.2 Clustering

For clustering, hierarchical, k-means, k-medoids, and k-means on SOM clustering is performed.

For hierarchical clustering, we perform Agglomerative Nesting (AGNES) hierarchical clustering with 6 different linkages and Divisive Analysis (DIANA) clustering. An online tutorial was used to get a better understanding of the functions [11].

For AGNES, we use 6 different linkages: Group Average, single-link (MIN), complete-link (MAX), Ward's method, weighted (Unweighted Pair Group Method with Arithmetic Mean, UPGMA), and Generalized Average (Flexible UPGMA). All of these behave slightly differently and help us learn more about our dataset and why a particular method worked better than another. Along with using DIANA, we will determine the best 2 hierarchical clustering methods using their agglomerative (for AGNES) and divisive (for DIANA) coefficients, with values closer to 1 being preferred. We will then make cutoffs at 2 and 5 clusters, because of the original labeling and intuition gained from the PCA, respectively. Finally, we plot colored dendrograms to better visualize the clusterings.

For k-means and k-medoids clustering, we first try to estimate the number of clusters, k, to select and additionally use k = 4 (based on previous testings) and k = 5 (estimated clustering from PCA). An online tutorial [1] and the documentation [14] was used to get a better understanding of how to implement the related functions. The clara() function was selected for k-medoids as it is suited for relatively large datasets like ours and the R implementation uses randomized sampling similar to CLARANS [14].

To estimate the number of clusters, we use two methods, the the elbow method and the silhouette method, the latter plots the average silhouette width vs. different number of clusters, with a silhouette being the value of how well each object lies within its cluster.

For each of the estimated clusters and for k = 4 and k = 5, we randomly initialize k-means clustering 100 times each and find the clustering with the best results. Additionally, using the results of the hierarchical clustering, we find 4 more k-means clusterings (k = 2 and k = 5 for the top linkage and DIANA) using the hierarchical clustering centers as initialization points. Similarly, we perform k-medoid clustering for the selected k values as well. Finally, we perform another PCA and project all the clusterings onto the first 2 principal components.

For the Self-organizing Map (SOM) clustering, we first create a SOM from our dataset using a toroidal shape, which helps with distributing errors and prevents edge effects that push extreme values to the edge [7]. A grid search was conducted to find the correct hyperparameters for the SOM. k-means clustering for the aforementioned k values is then performed on the SOM's neuron grid to detect clusters within the SOM. An online guide was used to learn implementation [7].

3.3 Validation of Clusterings

To validate our clusterings, we use various stability, internal, and external validation tests. Another online tutorial was followed to better learn the relevant functions [8].

For stability tests, we perform the nonparametric bootstrap, Average Proportion of Non-overlap (APN), Average Distance (AD), Average Distance between Means (ADM), and Figure Of Merit (FOM) stability tests.

The nonparametric bootstrap method uses the model we have already fitted to our dataset (the cluster model) to generate more samples with the same distribution as the real data, with the "nonparametric" version making no assumption about how the data is distributed. It then uses this simulated data to calculate standard errors, construct confidence intervals, and perform hypothesis testing in what is known as "bootstrapping" [3]. Using the clusterboot() function, we calculate a clusterwise Jaccard bootstrap mean for each clustering, with larger values being preferred.

APN values are within the interval $[0, 1]$, with values closer to 0 indicating consistent clustering results. It measures the average proportion of datapoints that are not placed in the same cluster. AD, ADM, and FOM all have ranges from $[0, \infty]$, with smaller values being preferred. AD measures the average distance between datapoints within the same cluster. ADM measures the average distance between cluster centers for datapoints within the same cluster, with our implementation using Euclidean distance as the distance measure. Finally, FOM measures the average intra-cluster variance of a removed datapoint, where the clustering is based on the remaining datapoints [2].

For internal validation, we perform the Connectivity, Silhouette Width, and Dunn Index validation tests. Connectivity measures the extent to which datapoints are placed within the same cluster as their nearest neighbor. Its values range from $[0, \infty]$, with smaller values being preferred. The Silhouette Width is a non-linear combination measurement of the compactness and separation of the clusters. It averages each datapoint's Silhouette value, which measures the degree of confidence of that datapoints assignment to its particular cluster, and has values within the range of $[-1, 1]$, with values closer to 1 being better. Finally, the Dunn Index is also a non-linear combination measurement of the compactness and separation of the clusters, but it takes the ratio of the smallest distance between datapoints in different clusters to the largest intra-cluster distance. It has values in the range of $[0, \infty]$, with larger values being better.

Finally, for external validation, we choose k = 2 sized clusters to match the 2 classes in the original dataset and compute the Rand Index. The Rand Index measures the agreement between two clusterings which, in our case, is the original cluster labels and our k=2 clusters. We use a corrected version which measures the same similarity, adjusted for chance, with a range from -1 (no agreement between the clusters) to +1 (a perfect agreement) [8].

4 RESULTS

4.1 Projection

4.1.1 Principal Component Analysis.

We first generate a covariance matrix of the entire dataset and use it to calculate the eigenvectors and corresponding eigenvalues of the covariance matrix. The eigenvalues can be seen in Table 2. We then apply the PCA, the summary of which is shown in Table 3. The process is then repeated for the sampled dataset, and the first 2 principal components for both are plotted and shown in Figure 2. Additional data such as the eigenvectors and the scree plot can be found at [6].

Table 2. All 14 eigenvalues for our full dataset.

1.115221e+10	5.553198e+07	1.622496e+05	1.885317e+02	1.494532e+02	6.042144e+01
1.800758e+01	1.604838e+01	5.024804e+00	2.436235e+00	1.960878e+00	1.900012e+00
6.843967e-01	1.387255e-01				

4.1.2 Multidimensional Scaling.

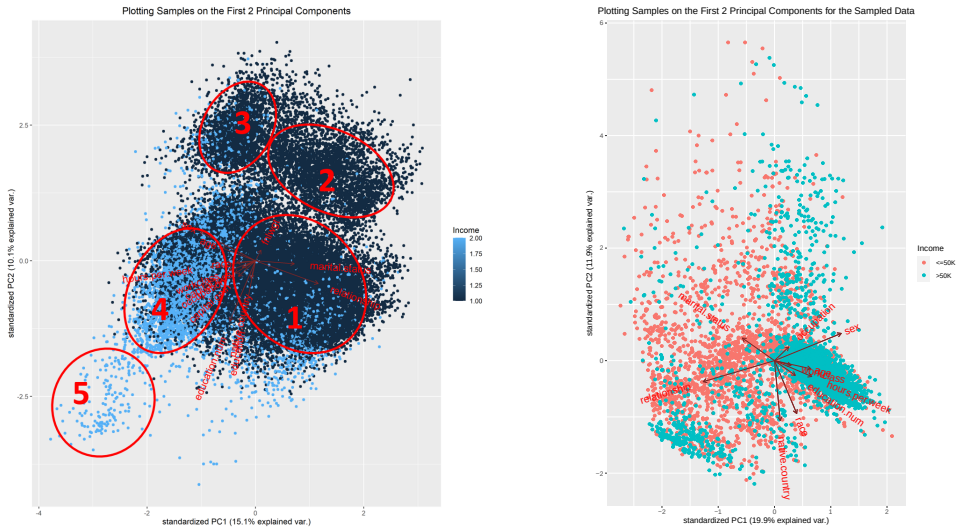
The results of our 4 MDS processes are shown in Figure 3. Figures 3b, 3c, and 3d show the results when the Classic Torgerson's MDS, Figure 3a, is used as an initialization. The randomly initialized MDS process did not show any meaningful results, and were thus not added to the final paper. However, they can be viewed on our GitHub repository [6].

4.1.3 Nonlinear Dimensionality Reduction.

The results of our Nonlinear Dimensionality Reduction, after performing hyperparameter tuning, are shown in Figure 4.

Table 3. Summary of the PCA analysis on the full dataset, depicting the importance of its components.

	PC1	PC2	PC3	PC4	PC5	PC6	PC7
Standard deviation	1.4551	1.1892	1.11738	1.06042	1.04282	1.01388	0.97489
Proportion of Variance	0.1512	0.1010	0.08918	0.08032	0.07768	0.07343	0.06789
Cumulative Proportion	0.1512	0.2522	0.34143	0.42175	0.49943	0.57285	0.64074
	PC8	**PC9**	**PC10**	**PC11**	**PC12**	**PC13**	**PC14**
Standard deviation	0.96297	0.92427	0.919	0.86081	0.82801	0.76782	0.62150
Proportion of Variance	0.06624	0.06102	0.0604	0.05293	0.04897	0.04211	0.02759
Cumulative Proportion	0.70698	0.76800	0.8284	0.88133	0.93030	0.9724	1.00000



(a) Entire Dataset (b) Sampled Dataset

Fig. 2. Plotting the datapoints of (a) the entire dataset and (b) the sampled dataset on the first 2 principal components. From (a) we can make out roughly 5 different clusters. From (b), we can make out 4 dense regions: 3 parallel "legs" and one region connected to them in a perpendicular fashion.

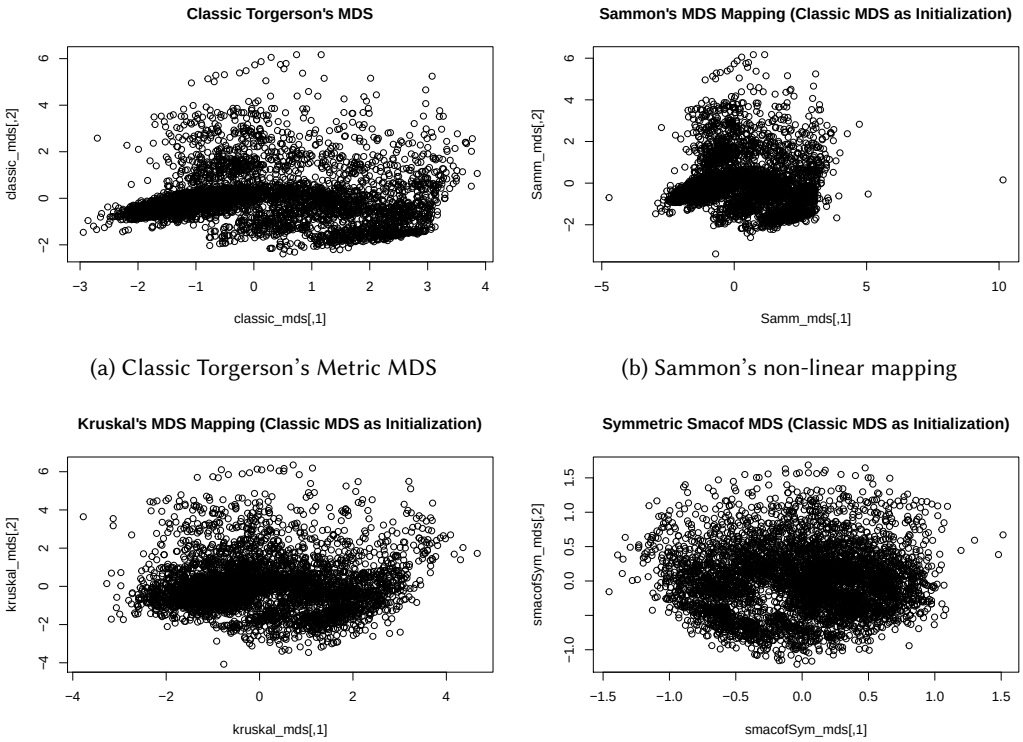
4.2 Clustering

4.2.1 Hierarchical Clustering.

For hierarchical clustering, we create a scaled distance matrix of our sampled dataset and perform all 7 hierarchical clustering methods. Their respective coefficients can be seen in Table 4. The top 3 are Ward's method, Generalized Average (GAverage), and DIANA, however, only the results and further processes of Ward's method and DIANA will be shown in this paper. The rest can be accessed from our GitHub repository [6]. DIANA was chosen over GAverage because DIANA's validation test results were higher overall. The selected hierarchical method's dendrogram is cut to make 2 and 5 clusters, which will then be used to initialize k-means clustering. Figure 5 shows their colored dendrograms, depicting these cluster choices.

4.2.2 K-Means Clustering.

Figure 6 shows the results of the elbow and silhouette methods, the former suggesting k = 12 and



(a) Classic Torgerson's Metric MDS

(b) Sammon's non-linear mapping

(c) Kruskal's non-linear mapping

(d) Symmetric Smacof non-linear mapping

Fig. 3. MDS Plots using (a) the Classic Torgerson's Metric MDS as the initialization for (b), (c), and (d). (a) and (b) roughly show 3 clusters forming, but (c) and (d) are mostly globular

Table 4. Agglomerative coefficients for the hierarchical clustering methods. Note, the DIANA method has a divisive coefficient instead, but it serves a similar purpose. Values closer to 1 are desired.

	Average	Single	Complete	Ward	Weighted	GAverage	DIANA
Coefficients	0.8950	0.8368	0.9249	0.9934	0.9073	0.9761	0.9351

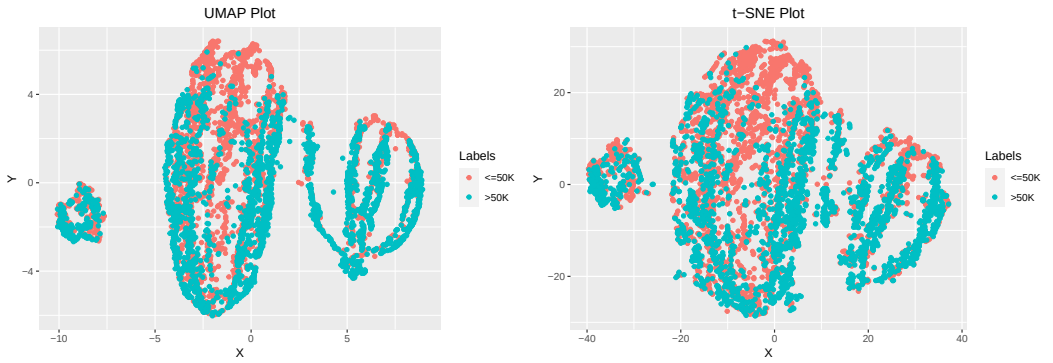
the latter suggesting k = 2. While k-means clustering was applied for randomly initialized k = 2, 4, 5, and 12, only k = 2 and k = 12 are shown (in order to space space), the rest can be found at [6]). A PCA is produced and the randomly initialized and hierarchical clustering-initialized k-means clusterings are projected onto the first 2 principle components, shown in Figure 7.

4.2.3 K-Medoids Clustering.

The results of the k-medoid clustering is shown in Figure 8.

4.2.4 Self-organizing Map Clustering.

The SOM with their k-means clustering is shown in Figure 9.



(a) UMAP Plot (b) t-SNE Plot

Fig. 4. UMAP and t-SNE plots. Both look similar: they contain helical shapes projected onto 2 dimensions and show 3 distinct clusters, but with each cluster containing a mix of both *income* classes.

4.3 Validation

The results of all of the validation tests are shown in Table 5. The Clusterwise Jaccard bootstrap stability test, Table 5a, was performed on all of the randomly initialized k-means clusterings and k-medoid clusterings, the other stability and internal validation tests, Table 5b, were performed on all of the hierarchical, k-means, k-medoids, and SOM clusterings, and the external validation test, Table 5c, was performed on all of the k = 2 clusterings (excluding the SOM clustering, which did not have a direct way to check external validity). Plots for the stability and internal validation tests can be found at [6].

5 ANALYSIS AND DISCUSSION OF RESULTS

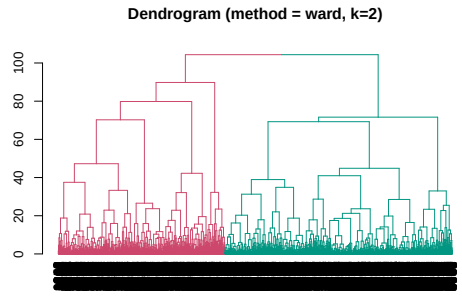
5.1 PCA

From the eigenvalues, because they are decreasing and the first 3 eigenvalues are considerably larger than the rest, we predicted that the PCA will give us decent results and some structure to visualize.

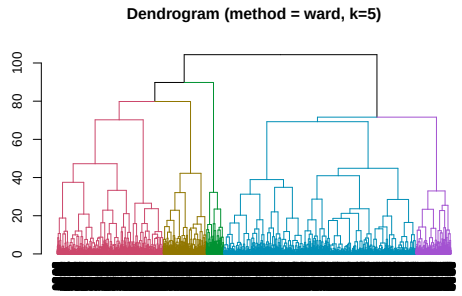
Upon applying PCA, we saw from the summary of the PCA analysis, Table 3, that it would take the first 8 principal components to encompass 70% of the cumulative variance. However, since our goal is to project them, we chose to focus on the first 2 components, which contain about 25% of the variance.

From Figure 2, we can see the shape that the datapoints take when plotted across the first 2 principal components. From the arrows, we can understand that higher values in a particular variable causes a datapoint to go in the direction of the arrow on the PCA plot. This seemingly also works for categorical data such as “relationship” where a label of 6 (Wife) would place the datapoint away from a label of 1 (Husband).

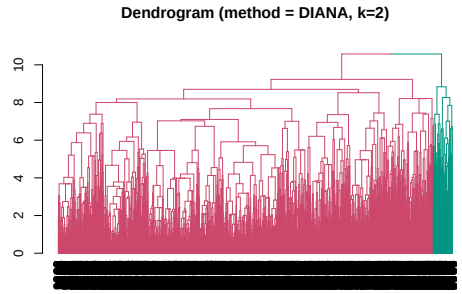
Interestingly, while the dataset only contains 2 official classes, “<=50K” (Income 1) and “>50K” (Income 2), we can see more clusters being formed. Figure 2a shows the possible clusters circled. Clusters 1, 2, and 3 are primarily “<=50K”, but they seem to form 3 sub-clusters on their own, possibly proving that “<=50K” can be further broken down into 3 clusters. Clusters 4 and 5 seems to be primarily “>50K”, but Cluster 5 is sparser. While this could indicate that the datapoints within are anomalies, they are still relatively dense compared to some of the other visible outliers. As such



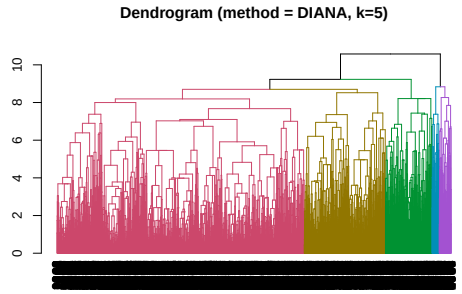
(a) Ward's Method, Clustering = 2



(b) Ward's Method, Clustering = 5

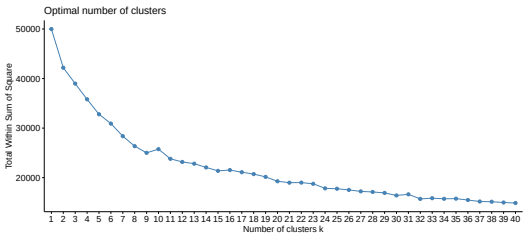


(c) DIANA, Clustering = 2

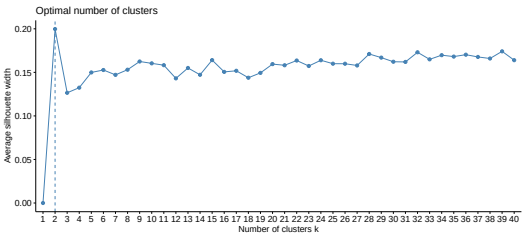


(d) DIANA, Clustering = 5

Fig. 5. Dendrograms of the hierarchical clusterings using Ward's method and Divisive Analysis (DIANA) clustering, colored to represent clusters of 2 (the original clustering) and 5 (predicted clustering from PCA).



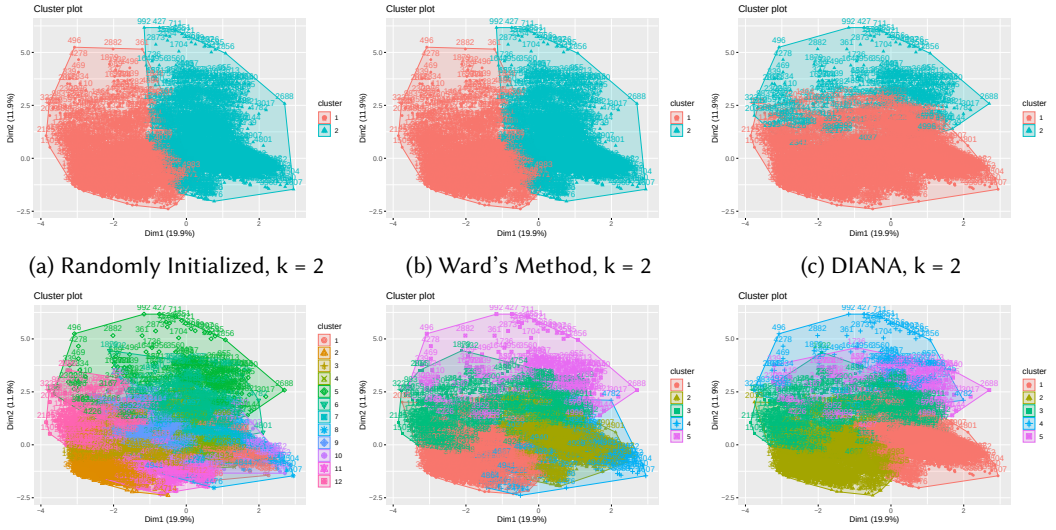
(a) Total Within Sum of Squares vs. k



(b) Average silhouette width vs. k

Fig. 6. Plots to help estimate the number of clusters, k, to select for k-means clustering, using (a) the elbow method and (b) the silhouette method. From (a), we can see an elbow form at k = 12, and from (b), we can see k = 2 is the optimal cluster number.

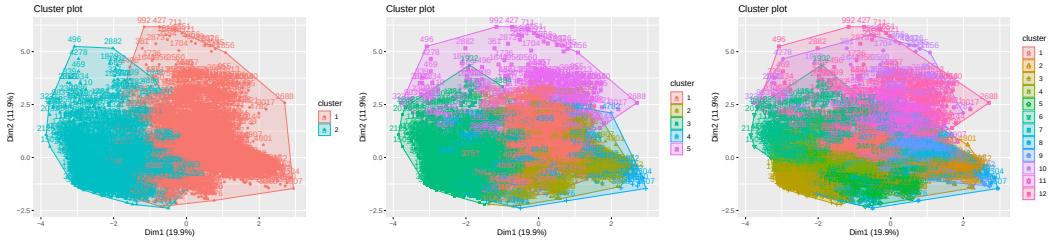
cluster 5 can be considered an additional clustering, possibly proving that ">50" can be further broken down into 2 sub-clusters. PCA on the sampled dataset, Figure 2b, has less discernable clusters, but roughly 4 dense regions can be identified towards the bottom of the plot, with two



(a) Randomly Initialized, k = 2 (b) Ward's Method, k = 2 (c) DIANA, k = 2

(d) Randomly Initialized, k = 12 (e) Ward's Method, k = 5 (f) DIANA, k = 5

Fig. 7. Results of the k-means clustering for 2 randomly initialized cluster options and 4 hierarchical clustering-initialized options. There is excessive overlap for all of the cluster options except k = 2. Note: larger versions of these plots are available online [6].



(a) k = 2 (b) k = 5 (c) k = 12

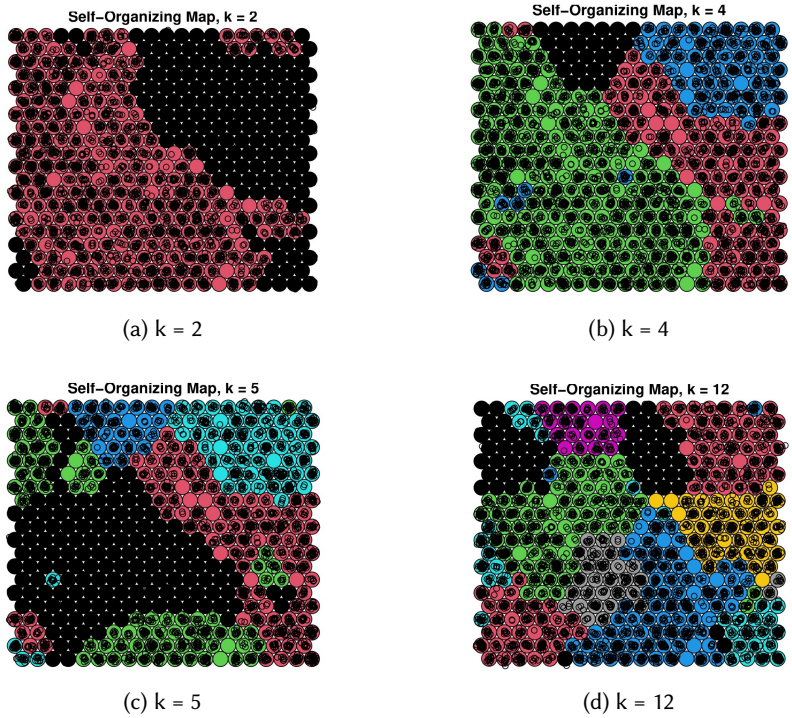
Fig. 8. Results of the k-medoids clustering for 3 k values. (a) seems to be balanced with a bit of overlap, but the other 2 have large overlaps. More clustering result plots can be seen at [6].

ovals on opposite ends being filled with the ">50K" class and the center and perpendicular ovals being filled with the "<=50K" class.

5.2 MDS

From the Classic Torgerson's MDS, Figure 3a, we can make out roughly 3 horizontal clusters, with a smaller one forming on top. However, for the rest of the MDS plots using it as an initialization, Figures 3b, 3c, 3d, as well as the randomly initialized ones (not shown in this paper) do not give any clear cluster shapes other than vague outlines of 2 or 3 overall clusters.

Unlike PCA, MDS did not give us very useful visual information about our datasets. This could partly be because, while PCA was able to use all 48,842 datapoints, MDS was only able to use 5000 samples due to computation time limitations. In addition, Kruskal's MDS might not have been



(a) k = 2 (b) k = 4

(c) k = 5 (d) k = 12

Fig. 9. Results of the SOM clustering for all 4 k values. Certain colors in each of the SOMs are scattered across the map. Note: the black color also represents a cluster.

very effective in any case due to it being a non-metric MDS algorithm, which specializes in ranked datasets (our dataset did not contain many ranked variables).

5.3 Nonlinear Projection

Both the UMAP, Figure 4a, and t-SNE, Figure 4b, plots show similar structure. We can see 3 distinct clusters that seem to be circling around within their confines. These 3 clusters seem to primarily have the ">50K" datapoints overlaid on top of the "<=50K" datapoints. The t-SNE plot seems to reflect the t-SNE method's instability on complex datasets, seeing as how its clusters are not as tightly bound. While there is no clear division between the 2 given classes, both nonlinear projections show 3 different possible clusters which are a combination of various datapoints from both the given classes. It can be argued that we get a better representation of the dataset using nonlinear projection methods, giving evidence that our dataset is highly complex.

5.4 Hierarchical Clustering

From Table 4, we can see that Ward's method, GAveraged, and DIANA have good cluster distinction. From Ward's method, we can understand that using the squared error when two clusters are merged to judge similarity is a good fit for our dataset. This method is less susceptible to noise, so the fact that it has the highest coefficient could be an indication that our sampled data is still noisy even after outlier removal.

Table 5. Tables summarizing the results for the validations tests. From the Clusterwise Jaccard bootstrap, (a), and other stability and internal validity tests, (b), k = 2 seems to be the ideal number of clusters, however, the external validity tests, (c), show poor scores.

(a) The Clusterwise Jaccard bootstrap mean for k-mean and k-medoid clusterings. There is one value per cluster, i.e. k = 12 has 12 values. Larger values are preferred.

		Clusterwise Jaccard Bootstrap Means						
k-means	k = 2	0.8838	0.9370					
	k = 4	0.6790	0.4836	0.2922	0.5393			
	k = 5	0.6053	0.5731	0.5072	0.3451	0.5267		
	k = 12	0.4350	0.6284	0.8123	0.8348	0.5325	0.8279	...
k-medoids	k = 2	0.9750	0.9360					
	k = 4	0.6647	0.7236	0.7292	0.7227			
	k = 5	0.6472	0.6863	0.6295	0.7018	0.3733		
	k = 12	0.9239	0.7929	0.8224	0.8770	0.7542	0.7057	...

(b) Results for the Stability and Internal Validation Tests.

		Score	Method	Clusters
Stability	APN	0.0236	DIANA	2
	AD	3.0501	K-means	12
	ADM	0.0855	DIANA	2
	FOM	0.9387	Ward's Method	12
Internal	Connectivity	113.87	DIANA	2
	Dunn	0.0709	DIANA	5
	Silhouette	0.3346	DIANA	2

(c) Summary of all the k = 2 clustering methods and their Rand Index when compared to the original labeling. Values closer to 1 are preferred.

Clustering Method		Rand Index
Hierarchical	DIANA	0.0003
	Ward's Method	0.0855
	GAveraged	0.0002
k-means	Random-Init	0.1009
	Ward-Init	0.1009
	GAveraged-Init	0.0003
	DIANA-Init	0.0003
k-medoids		0.0708

Comparing their colored dendrograms, Ward's method's k = 2 cut dendrogram, Figure 5a, seems to be more balanced and close to reflecting the sampled dataset's 50/50 class ratio, whereas DIANA's dendrogram, Figure 5c, seems to be highly imbalanced. Both of their k = 5 dendrograms, Figures 5b and 5d, seem to have varied proportions of clustering, with one dominant cluster in both.

5.5 k-means and k-medoids

All of the k-means, Figure 7, and k-medoids, Figure 8, plots for k = 2 seem to have almost equally balanced clusters, with DIANA and the k-medoid clustering having more overlap between the two clusters. All of their k = 5 clustering show a lot of overlap, with an excessive amount overlap for the randomly initialized k = 12 k-means and k-medoid clusterings, Figures 7d and 8c, respectively. One thing to note is that, since the outliers have been removed, the k = 2 clusterings no longer focus on the division between outliers and non-outliers. As such, the k = 2 clustering results seem to reflect the actual 50/50 nature of the sampled dataset.

5.6 Self-Organizing Map

The expectation when applying k-means on the neurons of the SOM was that clear borders between the neurons would be seen. However there is an unexpected scattering of clusters, with clusters being spread across the map. Given the nature of SOMs, similar neurons should be neighbouring each other, causing their clusters to be together as well, which is not what is observed in any of the SOM plots in Figure 9. While one possible reason for this could have been incorrect hyperparameters, an extensive hyperparameter search was applied and scattering of clusters was present in all of the results, indicating that this is either a feature of the dataset, or that a more complex SOM algorithm might be needed.

5.7 Validation of Clusterings

From the stability tests, for the nonparametric bootstrap method, Table 5a, we can see that k = 2 had the highest clusterwise Jaccard bootstrap mean for both k-means and k-medoids, indicating that k = 2 clustering is the most stable one given this test. For APN and ADM tests, Table 5b, the DIANA hierarchical clustering with cutoff at k = 2 was the most stable clustering, with values very close to 0. However, for the AD and FOM tests, the k = 12 clusterings were more stable, with the randomly initialized k-means and Ward's method hierarchical clustering, respectively. It should be noted that the AD score is relatively quite high, suggesting that, while the k = 12 k-means clustering is relatively stable, it is not objectively stable.

For all of the internal validation tests, Connectivity, Dunn Index, and Silhouette Width, the DIANA hierarchical clustering had the best scores, with k = 2 for Connectivity and Silhouette width, and k = 5 for the Dunn Index. However, the Connectivity score is very high, giving us an indication that very few datapoints are placed within the same cluster as their nearest neighbor.

Finally, for external validation tests, Table 5c shows that almost all of the clustering options have poor Rand Index scores that are very close to 0. This tells us that there is a poor agreement between these k = 2 clusterings and the original labeling of our dataset. The largest value is only 0.1009, achieved by the Ward's method-initialized and randomly initialized k-means clusterings. This result was unexpected. With outliers, our k = 2 clusterings would focus mostly on the outliers, explaining these poor scores. However, now that the outliers are gone and the k = 2 clustering projections in Figures 7 and 8a seem to be almost evenly divided, the low scores come as a surprise. This indicates that our clusterings have found a different way to separate the dataset, unrelated to the income, and suggests that our dataset is highly complex. This is further backed by the results of the nonlinear projection, which show clearly distinct clusterings, but without clustering according to the given *income* label.

It should be noted that other clustering analysis done on the same dataset either used a restricted number of variables, such as categorical data only [5] or used highly complex algorithms such as Probability Density Estimators [12].

6 CONCLUSION

In conclusion, this report performed a preliminary analysis, sampled the dataset such that the *income* class was balanced, performed PCA on the entire dataset, performed PCA, multiple MDS, UMAP, and t-SNE projections on the sampled dataset, performed multiple hierarchical, k-means, and k-medoid clustering on the sampled dataset and k-means on the dataset's SOM, and, lastly, measured cluster stability and validity, both internal and external.

While MDS did not give us a very good visual description of our dataset, PCA on the entire dataset and PCA and nonlinear projection on the sampled dataset was valuable because we were able to identify 5 (from the entire dataset) and 4 (from the sampled dataset) potential clusters.

Hierarchical, k-means, and k-medoids clustering indicated that k = 2 was the better clustering, as the other cluster options had heavy overlaps. The silhouette method supported this by suggesting that k = 2 was the best cluster option, while the elbow method suggested the unsuccessful k = 12. k-means applied to the SOM proved unsuccessful as well. The lack of cohesion between cluster estimation algorithms and the unsuccessful k-means SOM attempt could be attributed to the complex nature of our dataset.

Validation of our various clusters reinforced the idea that k = 4, 5, and 12 might not be good fits for our sampled dataset. While our k = 2 clusters seem to have the best scores for stability and internal validation, particularly the DIANA hierarchical clustering, they all scored poor Rand Index scores when compared to the original clustering labels. This suggests that some other property was detected by these cluster methods, resulting is relatively good k = 2 clusters, but not with respect to the original *income* class.

As a result of our analysis, we can conclude that the U.S. Census Bureau Income Dataset is quite complex and that traditional clustering methods are insufficient in understanding details necessary to cluster based on the given *income* class. This paper can be used as a basis to begin more advanced clustering procedures with a larger sample size in order to better understand the complex relationship between the variables within this dataset.

ACKNOWLEDGMENTS

To Professor M. Volkan Atalay, who always takes time out to help his students. Without his CENG 574 lectures, many of the concepts used in this paper would be foreign to us.

REFERENCES

[1] Bradley Boehmke. 2017. K-means Cluster Analysis. https://uc-r.github.io/kmeans{_}clustering
[2] Guy Brock, Vasyl Pihur, Susmita Datta, and Somnath Datta. 2020. *clValid , an R package for cluster validation.* Technical Report. Department of Bioinformatics and Biostatistics, University of Louisville.
[3] Cosma Shalizi. 2011. The Bootstrap. https://www.stat.cmu.edu/{~}cshalizi/402/lectures/08-bootstrap/lecture-08.pdf
[4] Gaston Sanchez. 2013. 7 Functions to do Metric Multidimensional Scaling in R | Visually Enforced. http://www.gastonsanchez.com/visually-enforced/how-to/2013/01/23/MDS-in-R/
[5] Aristides Gionis, Heikki Mannila, and Panayiotis Tsaparas. 2007. Clustering aggregation. *ACM Transactions on Knowledge Discovery from Data* 1, 1 (mar 2007), 4. https://doi.org/10.1145/1217299.1217303
[6] Raheem Hashmani and Deniz Germen. 2021. IncomeDataVisualization. https://github.com/RKHashmani/IncomeDataVisualization
[7] Jeff Bowman. 2020. Tutorial: Self Organizing Maps in R. www.polarmicrobes.org/tutorial-self-organizing-maps-in-r/
[8] Kasia Kulma. 2017. Cluster Validation In Unsupervised Machine Learning. https://kkulma.github.io/2017-05-10-cluster-validation-in-unsupervised-machine-learning/
[9] Ron Kohavi. 1996. Scaling up the Accuracy of Naive-Bayes Classifiers: A Decision-Tree Hybrid. In *Proc. of the Second Int. Conf. on Knowledge Discovery and Data Mining* (Portland, Oregon) *(KDD'96).* AAAI Press, 202–207.
[10] Luke Hayden. 2018. PCA Analysis in R - DataCamp. https://www.datacamp.com/community/tutorials/pca-analysis-r
[11] Manish Pathak. 2018. Hierarchical Clustering in R. www.datacamp.com/community/tutorials/hierarchical-clustering-R
[12] Dan Pelleg and Andrew Moore. 2004. *Scalable and Practical Probability Density Estimators for Scientific Anomaly Detection.* Ph.D. Dissertation. Carnegie Mellon University. https://dl.acm.org/doi/book/10.5555/1023559
[13] R-Bloggers. 2019. Running UMAP for data visualisation in R. https://www.r-bloggers.com/2019/06/running-umap-for-data-visualisation-in-r/
[14] R-core. 2016. kmeans function. https://www.rdocumentation.org/packages/cluster/versions/2.1.0/topics/clara
[15] Rebecca Merrett. 2019. Census Income. code.datasciencedojo.com/datasciencedojo/datasets/tree/master/CensusIncome
[16] Saurabh Jaju. 2017. Guide to t-SNE machine learning algorithm implemented in R & Python. https://www.analyticsvidhya.com/blog/2017/01/t-sne-implementation-r-python/