Unsupervised Learning:
Comparison of Clustering and Dimension Reduction Techniques

Oded Lippmann 060516119
Gal Hagag 301363446

March 13, 2022

Abstract

In the field of data mining, clustering is one of the major issues. The goal of this unsupervised machine learning technique is to find similarities within the data points and group similar data points together. Those data points which weren't labeled in advance go through a data labeling process. Furthermore, we can remove in advance, before clustering, some of the categories and refer to them as external variables. This can help us study the link between the categories removed to all other categories in the data. For example, if we make a survey of leisure activity in the population, we can remove the category of "gender", and after clustering the remaining categories, we can analyze the link between leisure activities preferred by men to those preferred by women. This paper covers some unsupervised methods and statistics exams in order to analyze the US Census Data. The US Census Data is categorical data with 68 categories. At first, we're removing four columns – those will be referred to as external variables. We performed multiple corresponding analysis in order to reduce its dimension and project it to a space in R^{10} with numeric values where we can use cluster methods with a reasonable computing time. After the conversion, we compared the following methods to cluster the data: K-means, Gaussian mixture model, Agglomerative clustering, and Spectral clustering. We used different methods to determine the optimal number of clusters and tuned each algorithm hyperparameters to best fit the data and the external variables. We used DBSCAN and KNN to find anomalous data points in the data and examined whether they are associated with any of the external variables. Finally, we projected the data into 2 dimensions using several techniques in order to visualize data and the clusters. The main conclusion of our study is that in order to achieve good results, the clustering task needs to be divided into several steps where each step is based or tied to the previous ones. Each step may include different methods fitted to the relevant data analyzed, and the conclusion conducted along the way must be statistically verified. This work was implemented using Python scripts and external libraries. The code is available at Git.

1 Introduction

Clustering is one of the most widely used techniques for exploratory data analysis, with applications ranging from statistics, computer science, biology to social sciences, or psychology. In every scientific field which deals with empirical data, researchers attempt to get the first impression of their data by trying to identify groups of "similar behavior". There are many methods to implement clustering algorithms and it's important to implement them in a way that best classifies the data. We followed several steps along this analysis process in order to achieve the best clustering results: preprocess the raw data, tune the hyperparameters for the different clustering algorithms, anomaly detection and using dimension reduction algorithms in order to plot the clustering results. Throughout the work, we used statistical tests to compare the performance of different methods and algorithms to achieve the best clustering results. This work compared four main clustering algorithms: K-Means, Gaussian mixture model, Spectral clustering, and Agglomerative clustering which were implemented on US Census dataset. Before clustering the data, we removed 4 columns from the data – and treated them as external variables. Those will be later in use to examine the association between them and the clustered data. Section 2 of this work explains in detail and in chronological order the different methods we used throughout the work. Section 3 examines the main results. And finally, in sections 4 5 we conclude with a discussion that holds our main insights from this work.

2 Methods

2.1 Sampling data

US Census dataset consists of 2,458,285 rows and 68 columns with categorical ordinal data. For example, "dAge" column maps the age of an individual 0:[1-12], 1:[13-19], 2:[20-29], etc. In order to get practical running times and to have presentative results, we sampled 10,000 samples and used statistical tests to verify that our sample is actually a good representation of the entire data (Using statistic test was possible due to the ordinal data).

2.2 Removing columns

In unsupervised learning, this method allows us to examine the linkage between the external variables and the remaining data. In this work, the external variables were age, sex, Hispanic origin, and years of work. After clustering the data we used statistical tests to measure how well the external variables associated with the clusters.

2.3 MCA

The first challenge we've dealt with in this work was to cluster categorical data. straightforward clustering techniques will raise two main issues: (1) The data is categorical, hence we need to define the relevant distance or similarity metric to separate the data points from one another (for example Euclidean which is the most common distance metric won't fit in this case). (2) Clustering large datasets requires great computation power. On top of that, if we'll convert the categorical columns into One-Hot-Vectors (to overcome issue mentioned above), the result will be a sparse matrix with over 2.4 million rows and over 200 columns. To overcome those challenges, we implemented the Multiple Corresponding Analysis (MCA). MCA is a generalization of the PCA [AV07] when the variables to be analyzed are categorical instead of quantitative. MCA is obtained by using standard correspondence analysis on an indicator matrix (OH). At last, when applying MCA, we chose the reduced dimension by projecting the data to a reduced dimension of orthogonal axes while preserving high inertia (high variance).

2.4 Clustering Algorithms

We used four clustering algorithms: K-means, Gaussian mixture model (GMM), Agglomerative clustering, and spectral clustering. The main hyperparameter for all those algorithms is the number of clusters. In order to tune it, we used two methods.

2.4.1 Elbow method

for each number of clusters, we calculate the loss function to evaluate how well all the data points fitted in their cluster (for example sum of squared distance in K-means). The more clusters we use the better results we get (lower loss function). The elbow method is used in order to choose a point where diminishing returns (loss function) are no longer worth the additional cost (clusters). We chose the "elbow" of the curve as the cutoff point to determine the optimized number of clusters.

2.4.2 Silhouette score

silhouette score is used to evaluate the quality of clusters created using clustering algorithms and its value ranges from -1 to 1 [SN20]. The silhouette measures how similar an object is to its own cluster compared to other clusters. When determining the number of clusters, we look for two results: (1) What's the number of clusters that provides the best silhouette score. (2) A "good" Silhouette plot – most of the observations are positive and close to the Silhouette score and the clusters are similar in their density.

2.5 Hyperparameters Tuning

In the first part of this work, we tuned only the number of clusters in order to define for each clustering algorithm the optimal number of clusters. Later on, the number of clusters was given based on the external variable we analyzed (amount of unique values). We tunned the hyperparameters for each method by "learning" the patterns on a train set and validating the score on a validation set (cross validation of 5 sets to maximize AMI score). The algorithms were later compared with their optimal hyperparameters.

2.5.1 K-Means

The algorithms assume spherical clusters that are separable in a way so that the mean value converges towards the cluster center. Hyperparameters which were tuned: (1) Number of clusters. (2) Algorithm type: "full" (using EM) or "Elkan" (using triangle equality). (3) Different tolerance. (4) Different Methods of initializations.

2.5.2 Gaussian Mixture Model

It's a probabilistic model that assumes all the data points are generated from a mixture of a finite number of Gaussian distributions with unknown parameters. Hyperparameters which were tuned. (1) Number of clusters. (2) Covariance threshold. (3) Different covariance types. (4) Different non-negative regularizations were added to the diagonal of covariance.

2.5.3 Spectral Clustering

The algorithm makes use of the eigenvalues of the similarity matrix of the data to perform dimensions reduction before clustering in fewer dimensions [Lux04]. Hyperparameters which were tuned. (1) Number of clusters. (2) Number of neighbors. (3) Affinity which determines how to construct the affinity matrix.

2.5.4 Agglomerative Clustering

Common hierarchical clustering to group objects in clusters based on their similarity. Hyperparameters which were tuned. (1) Number of clusters. (2) Linkage - which distance to use between sets of observation. (3) Affinity which deters how to construct the affinity matrix.

2.6 Clustering Comparison Measures

Clustering comparison measures play an important role in cluster analysis. Most often, such measures are used for external validation (in our case the external variables), that is, assessing the goodness of clustering solutions according to a "ground truth" clustering. In our work, we used two different measures: V-measure and Adjusted mutual information (AMI), and referred to the external variables as the target clustering.

2.6.1 V-Measure

The V-measure is an external entropy-based cluster evaluation measure. It measures how successfully the criteria of homogeneity and completeness have been satisfied (as in [RH07]). V-measure compares a target clustering against an automatically generated clustering to determine how similar the two are. The V-Measure requires the calculation of two terms: (1) homogeneity: A perfectly homogeneous clustering is one where each cluster has data points belonging to the same class label. Homogeneity describes the closeness of the clustering algorithm to this perfection. (2) Completeness, A perfectly complete clustering is one where all data points belonging to the same class are clustered into the same cluster. V-measure is the harmonic mean between homogeneity and completeness:

$$V - Measure = \frac{(1+\beta) * homogeneity * completeness}{(\beta * homogeneity + completeness)}$$

We chose the default value for $\beta = 1$, which means the homogeneity and completeness share the same weight.

2.6.2 Adjusted Mutual Information (AMI)

AMI as proposed by [VEB10] is an adjustment of the Mutual Information (AMI) score to account for chance. It accounts for the fact that the MI is generally higher for two clusterings with a larger number of clusters, regardless of whether there is actually more information shared.

2.7 ANOVA Test

Analysis of Variance (ANOVA) is a statistical test to identify any statistical differences between the means of three or more independent groups. If we manage to reject the null hypothesis such difference exists and we continue with the Tukey test.

2.8 Tukey Test

Tukey's test is a single-step multiple comparison procedure and statistical test. It can be used to find means that are significantly different from each other. Tukey test is used to compare the means of every treatment to the means of every other treatment. that is, it applies simultaneously to the set of all pairwise comparisons and therefore can be efficient for our goal to determine the best cluster for each external variable, after rejecting the null hypothesis in ANOVA test.

2.9 Anomaly Detection

We've implemented two main methods to find anomalies in the data: (1) DBSCAN which is based on density and defines less dense observation as anomalies [EKSX96]. (2) KNN which is based on distance and defines observations that are far from the "group" as anomalies.

2.9.1 DBSCAN

There are two hyperparameters we tuned while using DBSCAN. (1) eps: represent the maximum distance between two samples for one to be considered as in the neighborhood of the other. (2) Number of neighbors: we fixed the number of neighbors to be 20 and plotted the distances of each point in ascending order of the distance. By using the elbow method, we optimized the "eps" value. Furthermore, we applied another method to find "eps": two graphs showing the mean silhouette score in relation to different "eps" values. Both methods gave us similar results.

2.9.2 KNN

In this method, we defined the number of neighbors - K. For each observation, the algorithm measures the average distance of the K nearest neighbors and if that distance is higher than the threshold we defined the observation as an anomaly.

2.10 Dimension Reduction

In order to visualize our results we used several methods for reducing the dimensions of the data into 2D:Kernel Principal Component Analysis (with radial-basis function), Laplacian Eigenmaps, Isomap, and TSNE.

3 Results

The objective of this work was to compare clustering techniques on a given data. First, we've explored how well each clustering method works with a different number of clusters (elbow test and Silhouette score). Later we used statistical measures to determine how well each clustering method fits the external classification defined by the external variables. In the end, we used different dimension reduction techniques for visualizing our results. We sampled 10K rows out of 2.46M and performed t-test to compare the mean and standard deviation of every column in the sample to the entire data (used seed values for reproducibility). That way we were convinced that the sample chosen is indeed representative of the entire data with a confidence of 95% ($p - value \geq 0.054$).

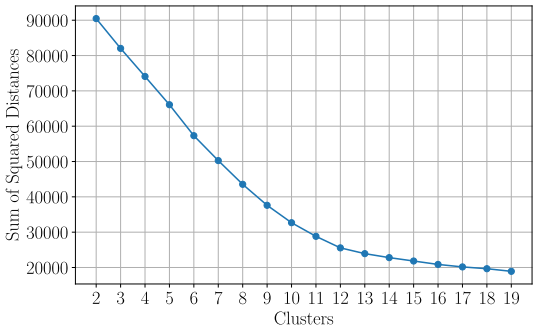


Figure 1: Elbow Method for Choosing Clusters

We converted the data to numeric by using MCA and reduced the number of dimensions into 10 dimensions after several trials (10, 15, 20, 50). It was decided once we didn't gain better results with higher dimensions and the fact that higher dimensions require longer computation time. In order to estimate the number of clusters, we used the elbow method on the K-means algorithm. The results were that the optimum number of clusters is between 7-9 as presented in Figure 1. We used the Silhouette score to verify that indeed this range of the number of clusters provides good results for all clustering algorithms as seen in Figure 2 and Figure 3

In Figure 2 we can see that most clustering methods get decent scores with 9 clusters and in Figure 3 we see that most clustering methods have "well-formed" Silhouette plots - similar in their density and average score (K-means and Agglomerative methods in particular).

Based on each external variable (assuming they represent the true label), we've tuned the hyperparameters for all clustering methods mentioned above. The best hyperparameters were chosen for each pair of methods and labels according to the AMI score using cross-validation of 5 sets.

As shown in Figure 4 we can see the results are similar (but not identical once reviewing the scores themselves). It is clear that the number of years of work ("iYearwrk") and age ("dAge") variables are much more associated with the clusters generated by the different methods as opposed to the gender ("iSex") and Hispanic origin ("dHispanic"). For each label, in order to compare the clustering methods, we performed an ANOVA test (based on 30 samplings) and Tukey test using a significance level of 5%.

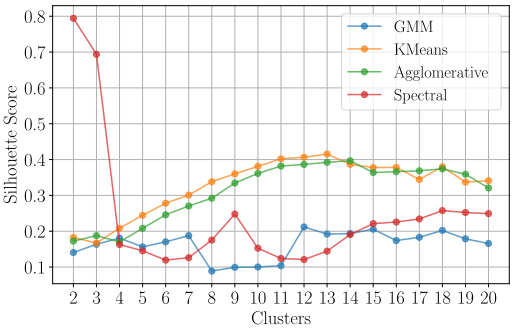


Figure 2: Silhouette Analysis for different number of clusters

The ANOVA test points out that not all algorithms perform the same ($p-value < 0.05$), therefore we executed the Tukey test (see Table 1) to identify the best clustering algorithm. As shown in [Table 1], in "iYearwrk" label we found that K-means and GMM get the best scores ($p-value = 0.001$) but it's unclear which of them perform best ($p-value = 0.3488$) and for "iSex" label GMM is best the

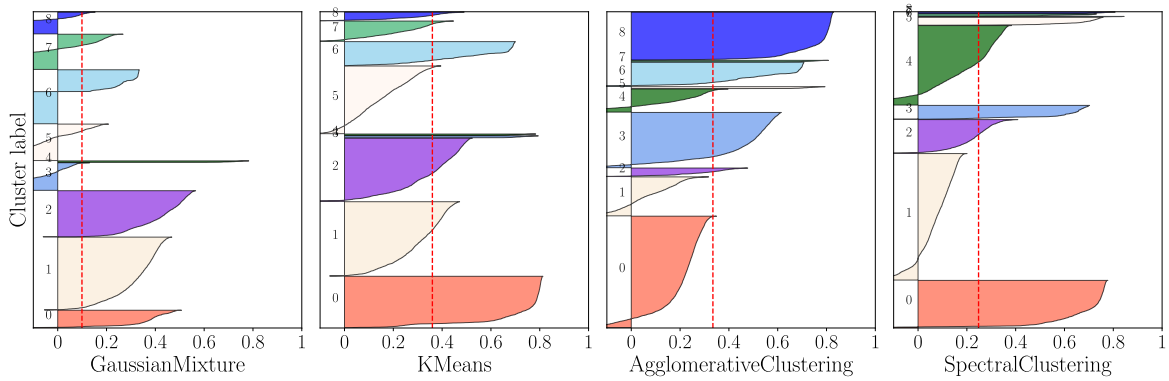


Figure 3: Silhouette scores for 9 clusters for each clustering method

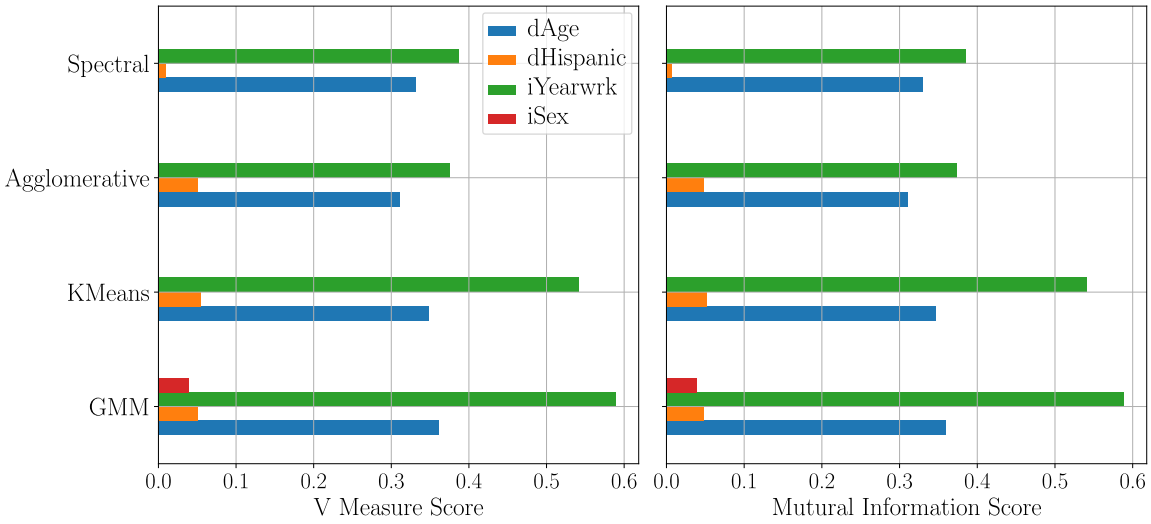
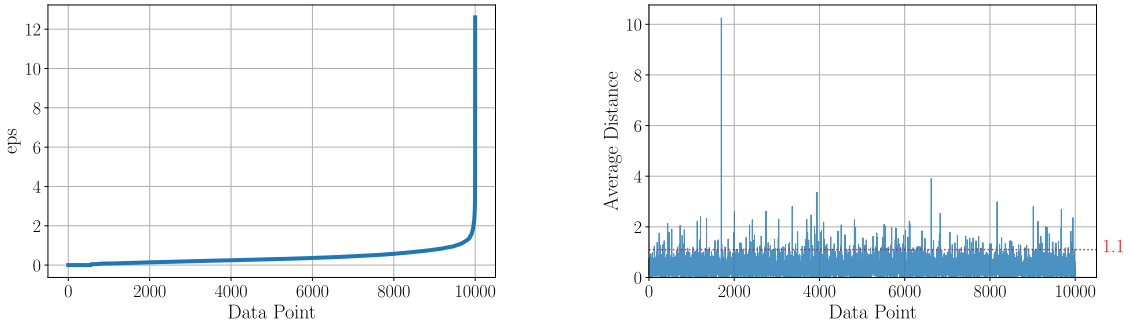


Figure 4: Assessing the clustering performance using two methods: V-measure score and AMI for each method. The number of clusters is with accordance to the number of unique values in the labels

iYearwrk			
Group1	Group2	Diff	p-value
Agglomerative	GMM	0.088	0.001
Agglomerative	KMeans	0.068	0.001
Agglomerative	Spectral	-0.018	0.615
GMM	KMeans	-0.025	0.348
GMM	Spectral	-0.106	0.001
KMeans	Spectral	-0.081	0.001

iSex			
Group1	Group2	Diff	p-value
Agglomerative	GMM	0.022	0.001
Agglomerative	KMeans	0.006	0.239
Agglomerative	Spectral	0.000	0.900
GMM	KMeans	-0.016	0.001
GMM	Spectral	-0.021	0.001
KMeans	Spectral	-0.005	0.335

Table 1: Comparison of different clustering techniques with respect to the two most associated external variables based on V-measure score



(a) Elbow method for ‘eps‘ in DBSCAN

(b) Average distance for 5 neighbors

Figure 5: Finding the optimal distance hyperparameters in DBSCAN and KNN

clustering method ($p - value = 0.001$).

In order to find anomalies within the data, we used 2 methods: DBSCAN and KNN. For the DBSCAN method, we fixed the number of neighbors to be 20 and tuned the "eps" (distance) by using the elbow method as seen in Figure 5. We found that the optimal "eps" value lies around the value of 1.4. Applying this value generated 375 outliers (from 10,000).

As for the KNN method, we set the number of neighbors to 5 and calculated for each data point the average distance between it and its neighbors Figure 6, then we set the threshold of being an anomaly to 1.1 which generated 269 outliers.

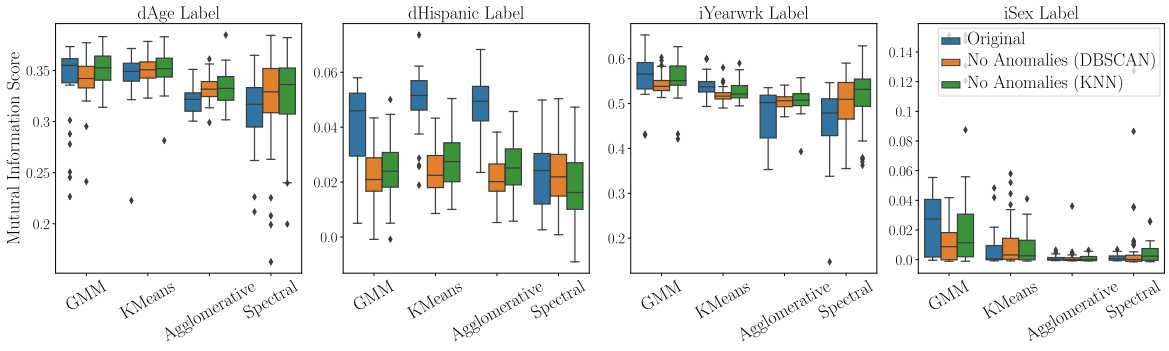
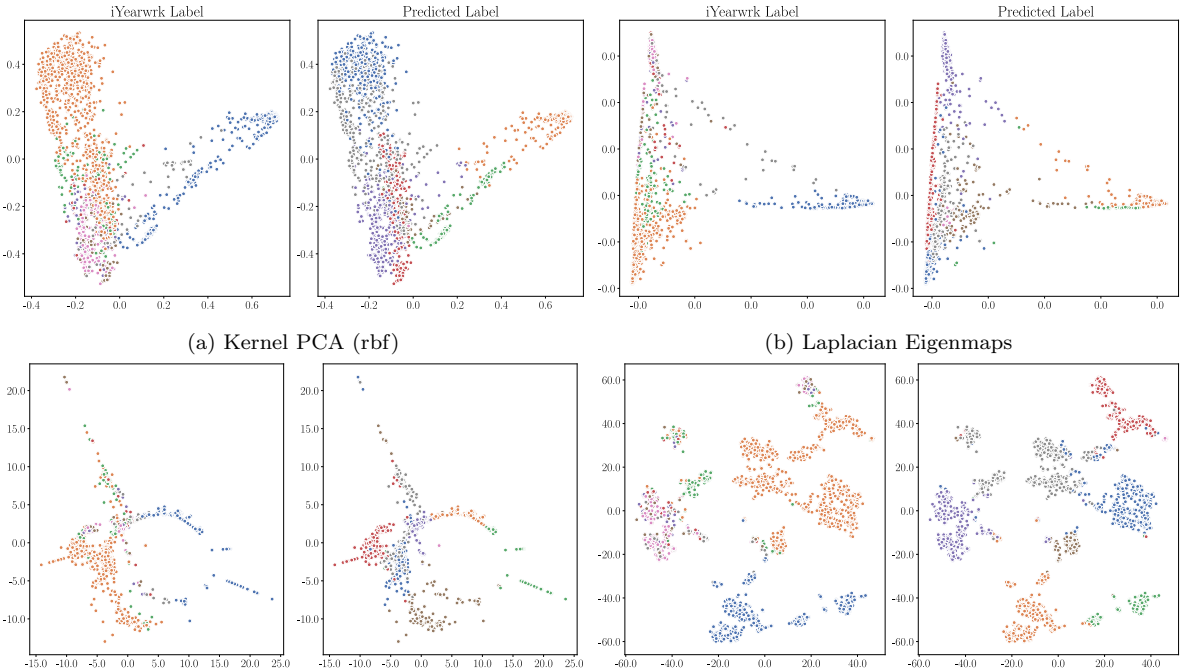


Figure 6: Algorithms scoring after filtering anomalies

We tested whether the anomalies are correlated with any of the external variables but the results were inconclusive. Anomalies found by DBSCAN were all uncorrelated with any of the external labels ($p - value > 0.3045$) whereas, the anomalies found by KNN were indeed statistically associated with all of them ($p - value < 0.0139$). We ran all the clustering algorithms again and compared the results with and without anomalies Figure 6. The results were inconclusive. For some algorithms, removing the outliers harmed the scores, for others benefited or had no significant effect. Because of this reason we continued our analyses on the original data. Several methods for dimension reduction were in use to enable us to visualize our clustered data in Figure 7. To compare these methods we used the clusters generated by K-means and use "iYearwrk" for representing the ground truth (as those outperformed any other sets)



(a) Kernel PCA (rbf)

(b) Laplacian Eigenmaps

(c) Isomap

(d) TNSE

Figure 7: Comparison of dimension reduction techniques: predicted clusters with K-means algorithm and 'iYearwrk' actual group

4 Discussion

The first challenge in this work was to deal with large categorical data. Using statistics tests we showed that 10,000 samples (0.4% of the data) are a satisfying representation of the data. In order to deal with the challenge of clustering big categorical dataset we tried a few methods: (1) using MCA and reducing the data to 10, 15, 20, and 50 dimensions or clustering the data without dimension reduction by converting the dataset to a matrix of OHE and using different metric (Gower or Jaccard). We conclude that reducing the dimension of data to 10 dimensions and converting it to numeric using MCA achieved a reasonable computing time and good results, showing that CA algorithms can produce good results when reduction of dimensions is applied before clustering due to the advantages of those algorithms to identifying patterns in data, expressing the data in such a way as to highlight their similarities and differences and compressing the data without much loss of information [SMZA11]. As to anomalies - We concluded that Anomalies found by DBSCAN were uncorrelated with any of the external labels, whereas those found by KNN were correlated with all of the external data. Despite this fact, removing the anomalies from the data was found to be inconclusive and in general, ineffective in improving our clustering results. "iYearwrk" label was found as the most linked external variable to the clusters produced by the different algorithms, in particular with K-means and GMM which get the highest scores. This can be interpreted as both of them use the same principles - GMM is an expansion of K-means.

5 Conclusions

In this work, we've dealt with many challenges in order to achieve good clustering results. We saw that the clustering techniques need to be chosen carefully with respect to the data that we cluster and the hyperparameters. In unsupervised learning, we are missing data labeling. As a way to deal with this absence, we can remove some variables in advance and later on use them to explain our data as part of the path of finding the best clustering algorithm and grouping the data together. Statistical tests were an important milestones of the analyses in order to verify our conclusions and continue to the next steps with confidence. In conclusion, clustering is a major challenge. In order to achieve good results, the clustering task needs to be divided into several steps where each step is based or connected to the previous ones. Each step needs to include different methods, where every method should be fitted to the data analyzed, and statistically verified.

References

[AV07] Hervé Abdi and Dominique Valentin. Multiple correspondence analysis. *Encyclopedia of measurement and statistics*, pages 651–657, 2007.

[EKSX96] Martin Ester, Hans-Peter Kriegel, Joerg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. volume 96, pages 226–231, 01 1996.

[Lux04] Ulrike Luxburg. A tutorial on spectral clustering. *Statistics and computing*, 17:395–416, 2004.

[RH07] Andrew Rosenberg and Julia Hirschberg. V-measure: A conditional entropy-based external cluster evaluation measure. pages 410–420, 2007.

[SMZA11] Rahmat Widia Sembiring, Jasni Mohamad Zain, and Embong Abdullah. Dimension reduction of health data clustering. *International Journal of New Computer Architectures and their Applications*, 1, 2011.

[SN20] Ketan Rajshekhar Shahapure and Charles Nicholas. Cluster quality analysis using silhouette score. pages 747–748, 2020.

[VEB10] Nguyen Vinh, Julien Epps, and James Bailey. Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance. *Journal of Machine Learning Research*, 11:2837–2854, 2010.