Unsupervised Learning - Final Project

Reut Levi (ID. 318648938), Oshrit Shtossel (ID. 208982553)

March 2022

Abstract

In this project, we apply several clustering algorithms on the US Census Data. We find the most appropriate clustering algorithm using supervised and unsupervised methods. We also find which external variable is best associated with clusters. We further analyze the anomalies in the data and test whether anomalies are associated with any of the external variables. Finally, we propose a smart visualization that presents the real clustering and a specific clustering algorithm on the same plot. We discovered that K-means algorithm is the best for this task and that the best associated external variable is iYearwrk. The code is available at `https://github.com/ReutLevi44/Reut_and_Oshrit_Unsepervised/tree/master/Reut_and_Oshrit_Unsepervised`.

1 Introduction

Unsupervised learning is a field that uses machine learning algorithms to analyze and cluster unlabeled datasets. These algorithms discover hidden patterns or data groupings without the need for human intervention [3]. During the project, we use three methods of this field: clustering, dimension reduction and anomaly detection. Clustering is the task of dividing the population or data points into a number of groups such that data points in the same groups are more similar to other data points in the same group than those in other groups. In simple words, the aim is to segregate groups with similar traits and assign them into clusters [2]. Dimension reduction, is the transformation of data from a high-dimensional space into a low-dimensional space so that the low-dimensional representation retains some meaningful properties of the original data, ideally close to its intrinsic dimension. Working in high-dimensional spaces can be undesirable for many reasons; raw data are often sparse as a consequence of the curse of dimensionality, and analyzing the data is usually computationally intractable (hard to control or deal with). Dimension reduction into 2 or 3 dimensions allows us to visualize the data [9]. Anomaly detection is a step in data mining that identifies data points, events and observations that deviate from a dataset's normal behavior. Machine learning is progressively being used to automate anomaly detection [4].

In this project, we apply one hot on the categorical data. Then, we reduce the dimension by the MCA algorithm because of the high dimensionality and sparsity of the data. We further apply different clustering algorithms and evaluate their performance by silhouette and mutual information. We use statistical tests for comparison between the different algorithms such as ANOVA and T-test. We also relate the external variables to the clusters. Finally, we suggest a smart visualization to present the real labels vs predicted labels.

2 Methods

2.1 US Census Data

The data was collected as part of the 1990 census in USA. There are 68 categorical attributes and over 2,458,285 samples [1].

Data preprocessing Since the data is categorical, we have converted the data to one hot vectors using the pandas library [6]. In addition, in order to avoid memory problems, we have sampled randomly 20,000 samples from the data, we repeated the sampling 10 times and reported all our measures as an average over the 10 sampling groups.

2.2 Multiple Correspondence Analysis (MCA)

MCA is a package for python, intended to be used with pandas. MCA is a feature extraction method, essentially PCA for categorical variables [8]. Since our data is a high dimensional categorical data, we applied the MCA dimension reduction algorithm to check whether reducing the data dimension would help the clustering algorithms to converge.

2.3 Train test split

We have divided the data into two groups, the first group was used for training, meaning finding the best hyperparameters (number of clusters, dimension for reduction). The second group was used as an external test, all the results were reported on sampling from this group.

2.4 Statistical tests

- To test whether there is a difference between all the algorithms performances, we used a *one-way ANOVA* [5].
- To test whether the 2 best algorithm are different in their performance, we used a *paired t-test on two related samples of scores* [7].

2.5 Finding the optimal number of clusters

In order to find the most appropriate combination of number of clusters and the dimension of the MCA reduction, we applied a grid search running on different number of clusters (between 2 to 15), and on different number of dimensions (10, 50, 100, 200, 300, and without MCA).

We measured the quality of the clustering by two different metrics:

- **K-means Loss** This metric is unique to the K-means algorithm. We have calculated the loss for each combination and created a heatmap of these values. Then, we first chose the dimension according to the lowest column and created the elbow function of the number of clusters vs the loss value of the chosen dimension. Since the loss is biased towards high number of clusters, we defined the best number of clusters as the one that reduced the loss to 50% from the beginning loss.
- **Silhouette score** We used this generic measure for all the clustering algorithms. We are aware of the problem that there is a bias towards low number of clusters in this measure, however, we chose the best combination according to the highest silhouette score in the heatmap.

2.6 Anomaly detection

In order to detect anomalies in the data, three methods were applied.

- **K-means** The data is first clustered according to the K-means algorithm. Then, point that has a significant distance from the centroid of its respective cluster (score) is suspected to be an anomaly. In our case, the significant distance is defined as a higher distance than $mean(scores) + 3 \cdot std(scores)$, where std is the standard deviation.
- **GMM** The data is first clustered according to the GMM algorithm. Then, for each point in the data we assigned a score of the log likelihood of each point in the data. As the score increases, the sample is more likely to be a non-anomaly point and as the score decreases, the sample is more likely to be an anomaly point. The cutoff is set to $mean(scores) - 3 \cdot std(scores)$. A point with a score that is lower than this cutoff is an anomaly point.
- **One class SVM** This method constructs hypersphere to encompass a large majority of the instances, and classifies points outside the margins as anomalies. The score of this algorithm is defined as the distance of the point from the center of the hypersphere. As this distance increases, the more likely the point is an anomaly, and as the distance decreases, the more likely the point is an integral part of the data. The default is binary one class SVM that divides the data into two equal groups. In comparison, we set the cutoff of the scores such that the percentage of the anomalies will be 1% of the data.

3 Results

3.1 Choosing the best number of clusters

First, we chose the best number of clusters for each algorithm separately. Since the data is categorical, we reduced the dimension of the data using the MCA algorithm (see Methods). Therefore, we optimized simultaneously the dimension as well as the number of clusters by grid search. We measured the clustering quality by the silhouette score for all the algorithms, and for the K-means algorithm, we have also checked the loss function values. The best dimension for all the algorithms apart from DBSCAN is 10 with the lowest K-means loss (Fig. 1A) and the highest silhouette score (Fig. 1C, D, F). However, the best dimension for the DBSCAN algorithm is 200 (Fig. 1E). We also checked all these measures without dimension reduction on the original one hot data, but the results were extremely poor (in different scales from the results with the dimension reduction). Therefore, we did not include the results of 'without MCA' in the heatmaps. After fixing the dimension, we chose the most appropriate number of clusters by choosing the highest score of the silhouette values according to the heatmaps, or using the elbow function on the best dimension for the K-means loss by choosing the number that reduced the initial loss by more than 50%. For K-means and hierarchical algorithms, the best number of clusters is 8 (Fig. 1B, C, D), in comparison to DBSCAN and GMM algorithms that the best number is 2 (Fig.1E, F). All these analysis were done on 10 cross validations that were sampled from the training set (20,000 samples each).

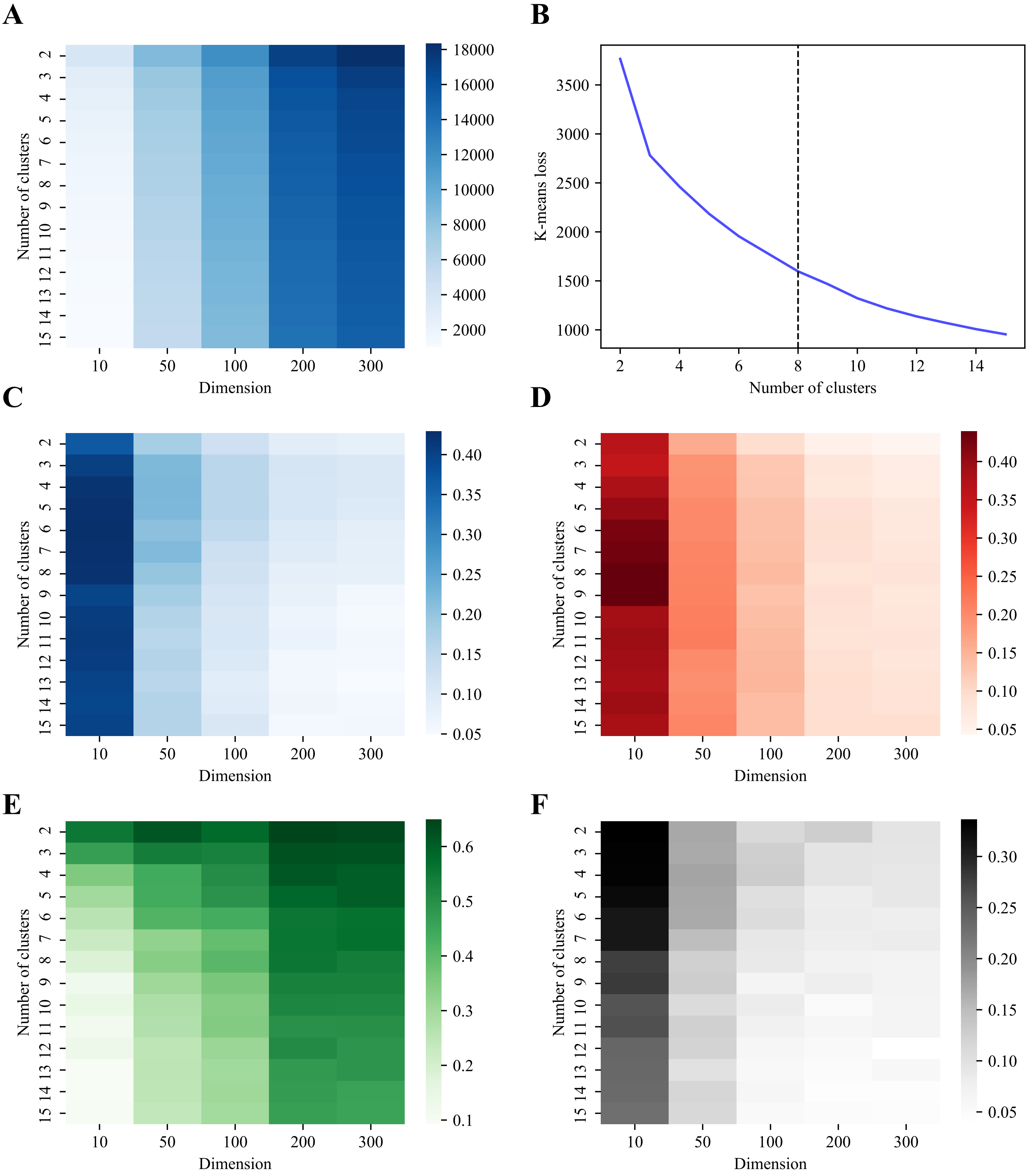

Figure 1: Optimal number of clusters. **A.** Heatmap of K-means loss values as a function of number of clusters and dimension. **B.** The elbow method on the best dimension chosen by A (10). The x-axis represents the number of clusters, while the y-axis represents the K-means loss values. The black vertical line represents the chosen number of clusters which reduced the initial loss by more than 50%. **C, D, E, F.** Heatmap of silhouette scores as a function of number of clusters and dimension for the K-means algorithm (**C**), hierarchical algorithm (**D**), DBSCAN algorithm (**E**) and the GMM algorithm (**F**).

3.2 Best clustering algorithm

For each algorithm, we used the optimal combination of dimension and number of clusters that we have found in the previous section. We evaluated the clustering quality in an unsupervised way by calculating the silhouette score and in a supervised way by calculating the mutual information (MI) with the external variables. We sampled 50 groups of 20,000 samples each from the external test set. For each group, we measured the silhouette score as well as the mutual information with each of the external variables (separately). Then, an ANOVA test was applied between the different algorithms, and a paired T-test was performed between the two algorithms with the highest scores.

According to silhouette measure, the ANOVA test yeilds a significant result of $p < 4.09e-129$. The best algorithm is DBSCAN with an average score of 0.648, it significantly outperforms the hierarchical algorithm with $p < 7.1e-59$ of the paired T-test (Fig. 2A). According to the mutual information score with the most informative external variable, iYearwrk, the picture is quite different. The best algorithm is the K-means algorithm with an average mutual information of 0.97, and DBSCAN is extremely bad. These results are significant with $p < 3.61e-165$ of the ANOVA test, and $p < 7.53e-18$ in the paired T-test between the two best algorithms (Fig. 2B). For the p-values of the other external variables see Table 1. Because of this gap in the evaluation, we have defined a new score which considers both the silhouette score and the mutual information score by averaging the two scores. While taking into account the two measures together, the best algorithm is K-means, in other words, the K-means algorithm is more stable than DBSCAN ($p < 2.37e-137$ of the ANOVA test and $p < 1.77e-14$ of the paired T-test, Fig.2C). An illustration of the silhouette score of the best algorithm (K-means) is presented in Fig. 2D.

	ANOVA test p-value	**T-test p-value**
dAge	3.32e-138	3.03e-26
dHispanic	6.69e-173	4.14e-17
iYearwrk	3.61e-165	7.53e-18
iSex	3.11e-96	0.01

Table 1: ANOVA and paired T-test p-values. The ANOVA test was performed for each external variable separately between all the clustering algorithms and the paired T-test was applied between the two best clustering algorithms.

3.3 Association with external variables

To associate the external variable with the clustering algorithm, we used mutual information score. We reran each algorithm with the same optimal dimension as before, but with the number of clusters equals to the real number of categories of each external variable (2 for the iSex variable, 8 for the iYearwrk and dAge variables and 10 for the dHispanic variable, Fig. 2E). For each clustering algorithm, we applied an ANOVA test between the four external variables ($p < 9.68e-282$) for the best algorithm, K-means. Then, we applied a paired T-test between the two best external variables (iYearwrk and dAge) and got $p < 9.38e-53$. Therefore, iYearwrk is the external variable that best associates with the K-means clustering and its mutual information is 0.97.

	ANOVA test p-value	**T-test p-value**
K-means	9.68e-282	9.38e-53
Hierarchical	6.95e-240	7.43e-46
DBSCAN	1.2e-55	0.001
GMM	1.34e-232	2.5e-41

Table 2: ANOVA and paired T-test p-values. The ANOVA test was performed for each algorithm separately between all the external variables and the paired T-test was applied between the two best external variables.

3.4 Anomaly detection

We have applied several anomaly detection algorithms:

- **K-means** We clustered the data using the K-means algorithm with the optimal combination of number of clusters and dimension. Each point got a score during the clustering of its distance from the centroid of the cluster it belongs to. We defined a cutoff on the scores as explained in the Methods section. We got 1.507% from the data as anomalies (Fig. 3A).

- **GMM** We clustered the data using the GMM algorithm with the optimal combination of number of clusters and dimension. Each point got a score of its log likelihood. We defined a cutoff such that all the samples with the scores that are lower than the cutoff are defined as anomalies (see Methods), which are 0.44% of the data (Fig. 3B).

- **One class SVM** At first, we used the binary class SVM on the 10-dimensional data, which divides the data into two groups. The anomaly group that is assigned as 1 and the non-anomaly group that is assigned as -1. The problem with the binary one class SVM is that it defines too much samples as anomalies by dividing the data into two equal groups (Fig. 3C). Therefore, we defined a new cutoff by using the original scores of the one class SVM (see Methods). By using the new cutoff, we decreased significantly the amount of anomalies in the data (from 50% to 1%, Fig. 3D).

We further tried to explain the anomalies according to the external variables. We calculated the mutual information between the anomaly labels and the external variables, we have repeated on the analysis for all the algorithms. All the mutual information scores are extremely low. Therefore, we believe the external variables do not explain well the anomalies. However, the most explainable variable is dAge with the binary one class SVM.

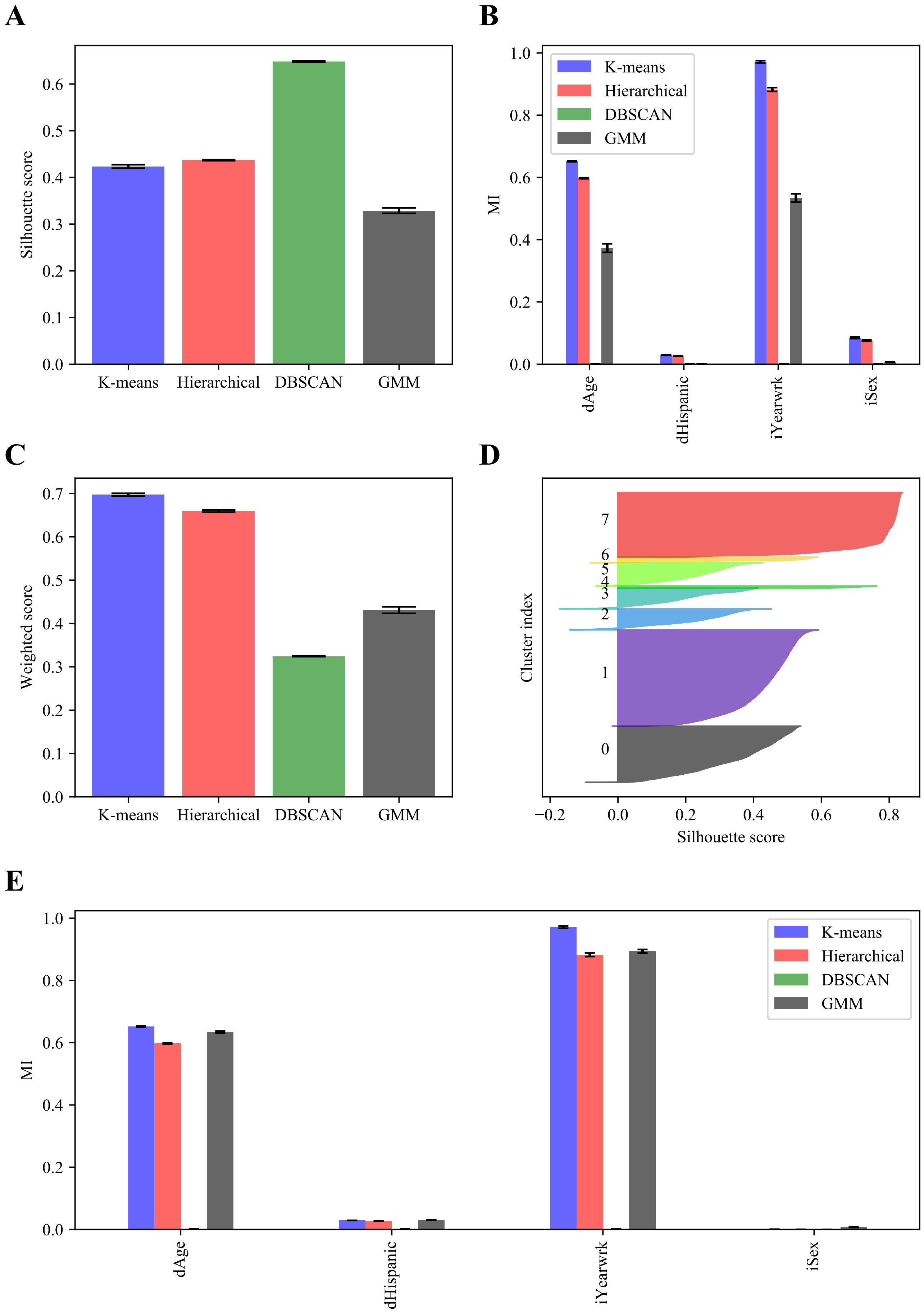

Figure 2: Best clustering algorithm and most associative external variable. **A.** Silhouette score per clustering algorithm on the best combination of dimension and number of clusters. **B.** Mutual information between each algorithm's predicted labels and each of the external variables. **C.** Weighted score that is defined as the average between the silhouette score and the mutual information. **D.** Silhouette visualization of the best algorithm, K-means, according to the weighted score. **E.** Mutual information between each algorithm's predicted labels and each of the external variables, where the number of clusters is set to the real number of categories per each external variable. The black errorbars represent the standard error of each bar.

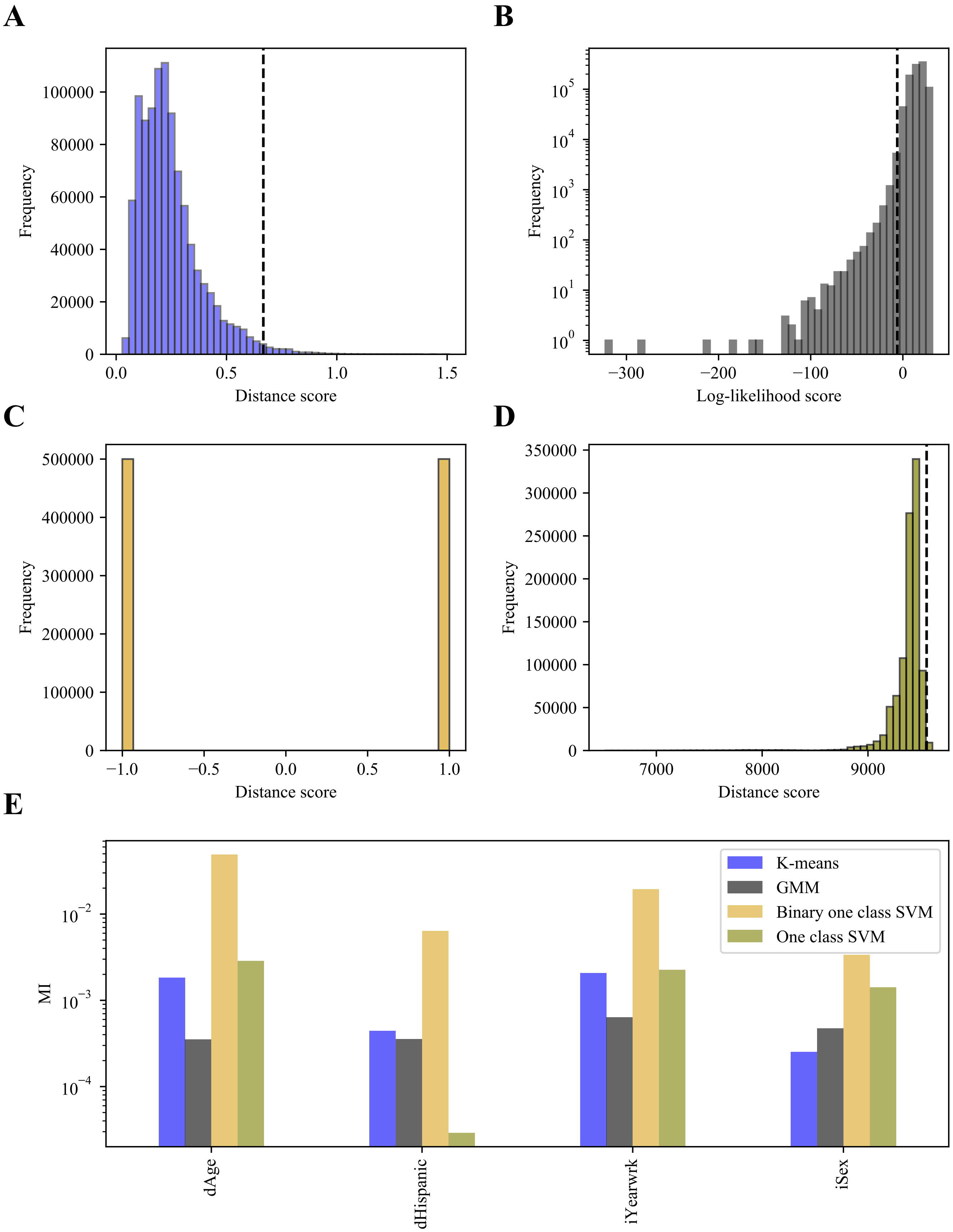

Figure 3: Anomaly detection. **A, B, C, D.** Scores histograms for K-means algorithm (**A**), GMM algorithm (**B**), where the y-axis is in a log scale, binary one class SVM (**C**) and one class SVM (**D**). The black line represents the anomaly or non-anomaly cutoff. **E.** Mutual information between each external variable and each anomaly detection algorithm.

3.5 Visualization

To visualize the data, we used the 10-dimensional data that we got by the MCA algorithm. Then, we applied TSNE for the dimension reduction into 2 dimensions. We presented both the real labels of the most associative external variable, iYearwrk, and the labels of the best algorithm K-means. We plotted the points of the data twice, one big points with light colors for the representation of the K-means clustering and one small points with dark colors for the representation of the external variable (Fig. 4). As we can see from the result, the K-means clustering is quite neat. Some of the clusters are consistent between K-means and the real labels, whereas some of the clusters are inconsistent. For example, sometimes K-means divides one cluster in the real data into some different clusters (like the purple cluster), and sometimes the K-means clustering contains points from many clusters of the real data (like the green cluster).

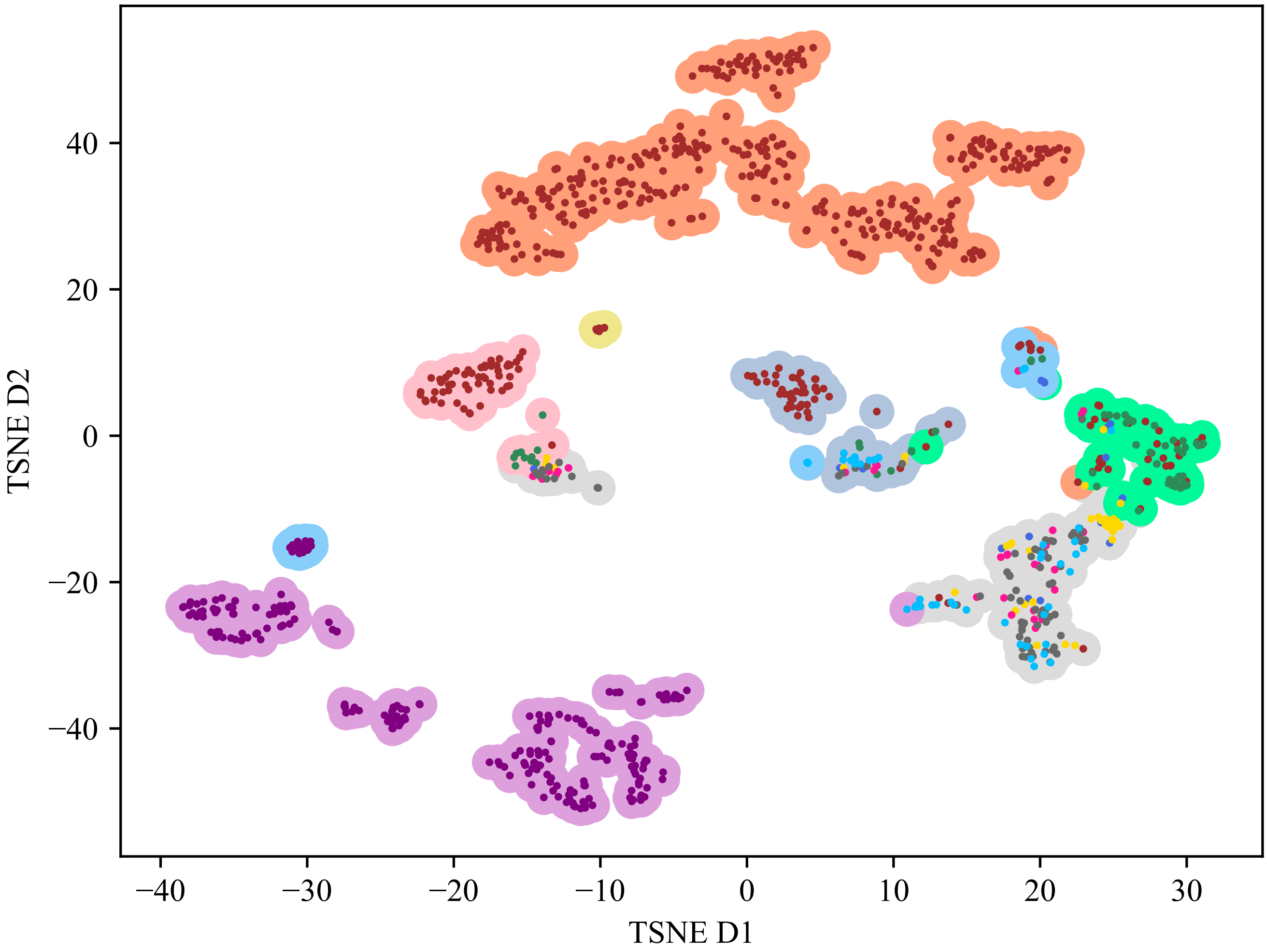

Figure 4: Visualization for real labels for iYearwrk external variable and the best clustering algorithm, K-means. The data was reduced into 2 dimensions by the TSNE algorithm. The big points with the light colors represent the K-means clustering, while the small dark points represent the real clustering of the external variable, iYearwrk.

4 Discussion

In this project we implemented supervised and unsupervised algorithms on the US Census data. We found what is the optimal combination for each clustering algorithm. The optimal number of clusters was divided into two groups, one of the K-means and the hierarchical algorithms with 8 clusters (consistent both on silhouette and K-means loss), and the second of the DBSCAN and GMM algorithms with 2 clusters. We tend to consider the 8 clusters as more reliable since the silhouette score has a bias for preferring a low number of clusters. As expected, DBSCAN got the highest silhouette score, followed by the K-means and the hierarchical algorithms. Surprisingly, the 2 clusters GMM got the worst silhouette score. In addition, we measured the quality of the clusters by mutual information with the external variables. There is a gap between the best algorithm according to the silhouette and the best algorithm according to mutual information. Therefore, we propose our weighted score that considers the two scores equally. However, this score can be further enlarged with different weights for each score according to one's priors. The score we have suggested gives a higher score for consistent algorithms that get good scores according to both measures. When we calculated the mutual information with the external variable, we performed two approaches, one for evaluating the best algorithm, where we used the number of clusters we optimized separately for each algorithm. Second for finding the best associative external variable, where we used the number of clusters equals to the real number of clusters of each external variable. This change has improved the mutual information of iYearwrk, dAge and dHispanic but unexpectedly has worsen the mutual information of iSex.

Moreover, we detected the anomalies in the data by using K-means, GMM and one class SVM algorithms. To handle the problem that binary one class SVM separates the data into two equal groups, we have improved the algorithm by using its scores and by defining manually the cutoff that determines the percentage of anomalies (we set the cutoff such that 1% of the data will be considered as anomaly). We tried to explain the anomalies by the external variables, but the relation measure by the mutual information is really poor. The highest mutual information is of the binary one class SVM what seems at first quite surprising. However, we assume it occurs because it divides the data into two equal groups, what enables a higher mutual information.

Finally, we proposed a nice visualization that represents the real labels besides the predicted labels on the same plot. It is interesting to reproduce the visualization with 3 dimensions, it may show separation between different clusters that seem scrambled and may be separated in a higher dimension.

References

[1] US Census Data, 1990.

[2] Clustering. https://www.analyticsvidhya.com/blog/2016/11/an-introduction-to-clustering-and-different-methods-of-clustering/, 2016.

[3] Unsupervised Learning. https://www.ibm.com/cloud/learn/unsupervised-learning, 2020.

[4] Anomaly Detection. https://www.anodot.com/blog/what-is-anomaly-detection/, 2022.

[5] Anova. http://www.biostathandbook.com/onewayanova.html, 2022.

[6] Get dummies pandas. https://pandas.pydata.org/docs/reference/index.html, 2022.

[7] Paired T-test. https://www.statisticssolutions.com/free-resources/directory-of-statistical-analyses/paired-sample-t-test/, 2022.

[8] Michael Greenacre and Jorg Blasius. *Multiple correspondence analysis and related methods*. Chapman and Hall/CRC, 2006.

[9] Laurens Van Der Maaten, Eric Postma, Jaap Van den Herik, et al. Dimensionality reduction: a comparative. *J Mach Learn Res*, 10(66-71):13, 2009.