THE DIABETES DATASET
Final Project June - July 2023
Unsupervised Learning
Master degree in Artificial Intelligence for Science and Technology

Rebecca Casati
Student ID: 859789

Giulia Denti
Student ID: 845053

***Abstract*—The aim of this project is to cluster the Diabetes dataset using three target variables: Diabetes, Hypertension and Stroke. Since the data are of mixed-types, the solution that has been found to deal with them was to compute the distance matrix with Gower's distance, which will be used as metric in the following analysis. No dimensionality reduction was performed because it was found that the FAMD and UMAP methods were ineffective. Anomaly detection was performed via the k-Nearest Neighbor algorithm, which allowed to identify approximately 2.000 outliers. Clustering was performed using three methods: Hierarchical, Density-Based (DBSCAN) and k-Prototypes. With some hyper-parameter tuning, it was found that the best solutions were those given by Ward's method for the Hierarchical and the k-Prototypes, both with three clusters. Lastly, some unsupervised (Silhouette coefficient and Dunn index) and supervised measures (Rand score, ARI and Fowlkes and Mallows index) were computed in order to evaluate the clustering solutions found, which showed that none of them were good clustering or at least meaningful.**

1. Introduction

This paper reports the results obtained after developing the code for the exam project of the Unsupervised Learning course of the Master Degree in Artificial Intelligence for Science and Technology held by the University of Milano Bicocca, the University of Milano Statale, and the University of Pavia.
The aim of this project is to perform an unsupervised analysis of a smaller variant of the Diabetes Dataset. In particular, clustering of the data objects is performed without considering three target variables that are nevertheless available and have been used to perform a supervised evaluation, in addition to the unsupervised one, of the clustering solutions found.
The report is divided into sections that deepen the nature of the dataset and its preprocessing, the anomaly detection that has been performed, the different clustering algorithms used, and a final evaluation of their performance.

2. Data

The Diabetes dataset [1] is a collection of medical information of 40.108 patients. The dataset was synthetically cured from the original data source using CTGAN (Conditional Tabular Generative Adversarial Network). The information for each patient was one target variable, Diabetes, and 17 feature variables, which may be indicators of diabetes. The variables are as follows:

- age: categorical discrete variable which express the age of the patient (e.g. 1 = 18-24, 9 = 60-64, 13 = 80 or older).
- sex: categorical binary variable: 0 = female, 1 = male.
- HighChol: high colesterol. Categorical binary variable: 0 = no, 1 = yes.
- CholCheck: colesterol check in 5 years. Categorical binary variable: 0 = no, 1 = yes.
- BMI: Body Mass Index. Numerical discrete variable.
- Smoker: smoked at least in their entire life. Categorical binary variable: 0 = no, 1 = yes.
- HeartDiseaseorAttack: had Coronary Heart Disease (CHD) or Myocardial Infraction (MI). Categorical binary variable: 0 = no, 1 = yes.
- PhysActivity: physical activity in past 30 days, not including job. Categorical binary variable: 0 = no, 1 = yes.
- Fruits: consume fruit one or more times per day. Categorical binary variable: 0 = no, 1 = yes.
- Veggies: consume Vegetables 1 or more times per day. Categorical binary variable: 0 = no, 1 = yes.
- HvyAlcoholConsump: for adult male: more than 14 drinks per week; for adult female: more than 7 drinks per week. Categorical binary variable: 0 = no, 1 = yes
- GenHlth: general health. Categorical discrete variable: 1 = excellent, 2 = very good, 3 = good, 4 = fair, 5 = poor.
- MentHlth: Days of poor mental health. Numerical discrete variable: scale 0-30 days.
- PhysHlth: Physical illness or injury days in past 30 days. Numerical discrete variable: scale 0-30 days.

TABLE 1. NUMBER OF DATA OBJECTS PER CLASS FOR THE THREE TARGET VARIABLES.

	1	0
Diabetes	20.489	19.619
Stroke	2.534	37.574
Hypertension	21.952	18.156

- DiffWalk: serious difficulty walking or climbing stairs. Categorical binary variable: 0 = no, 1 = yes.
- Hypertension: Categorical binary variable: 0 = no, 1 = yes.
- Stroke: Categorical binary variable: 0 = no, 1 = yes.
- Diabetes: Categorical binary variable: 0 = no, 1 = yes.

Instead of treating the dataset as it was originally meant to be treated, that is to perform clustering for a binary classification problem in order to determine whether a person has diabetes or not, the clustering of the data objects was performed by taking into consideration only the first 15 feature variables, and by keeping aside the last three (Hypertension, Stroke and Diabetes) as target variables.

Figure 1. Distribution of the variables.

2.1. Inspection of the Dataset

Some inspection of the dataset was performed to understand at a deeper level how it is composed and to consequently deal with it in the most appropriate way.

First of all, it was checked if there were some missing values, but none were found.

A second useful information involves knowing whether the dataset is balanced with respect to the target variables. It was found that it is actually balanced for the Diabetes and Hypertension variables, but not at all for the Stroke variable: only 6% of the people in the dataset did have a stroke. This imbalance in the Stroke variable represents a problem in the classification, since the clustering of the features will have trouble separating the few samples that had a stroke (Stroke = 1) from the other samples, thereby missing a cluster and class in the final prediction.

Table 1 reports the number of data objects per class for the three target variables.

Finally, the distribution of each variable was plotted [2], as shown in Figure 1. It can be seen that some of them are unbalanced, and in particular, the Stroke histogram shows the previously mentioned prevalence of data objects with 0 as Stroke value.

2.2. Data Preprocessing

Some preprocessing was applied to the data: after removing the target variables from the dataset, and since there were not missing values, the main thing that has been done was to differentiate the categorical variables from the numerical ones, which are just BMI, MentHlth, and PhysHlth.

This was done because it was necessary to apply scaling to the numerical variables [2], whereas the categorical variables were converted to string objects. For scaling, the MinMax scaler was applied, which transforms the features by scaling them to the range [0, 1].

Dimensionality reduction. Since the dataset has high dimensionality, which means that it has a high number of features (15 feature variables), it could be useful to apply dimensionality reduction to better visualize and interpret the data.

First, Factor Analysis of Mixed Data (FAMD) [3] was applied. This statistical analysis is similar to Principal Component Analysis (PCA), but, unlike PCA, it can work with categorical and numerical data simultaneously.

FAMD was first applied by projecting the data into a new space of the same dimension as the starting space to evaluate the explainability obtained from each component. The value of the cumulative variance obtained with all 15 components was only 64%; however, ideally, this value should be as close as possible to 100%, so that very little information would be lost in the dimensionality reduction step.

The lack of explainability through this method makes it impossible to perform dimensionality reduction, which can be attributed to the fact that FAMD searches for linear relationships between the variables; however, if the variables of the given dataset are not actually correlated (since the dataset has many features, it may be possible that there are some noisy variables that are not relevant to the study) or have a more complex relationship than a linear one, it makes it difficult for FAMD to perform well.

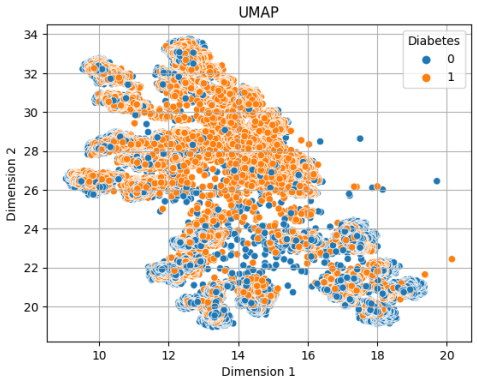


Figure 2. Dimensionality reduction with UMAP.

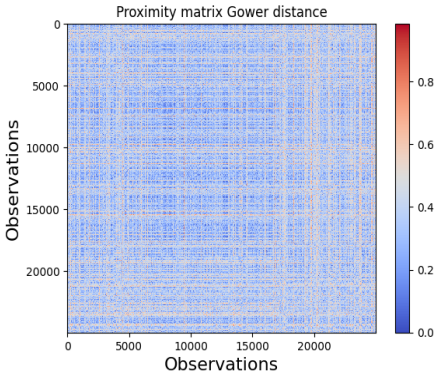


Figure 3. Matrix of Gower's distances between the data objects. This matrix reported here is just 25.000×25.000, as an example.

Given the poor performance of the linear reduction method (FAMD), a technique that is able to find nonlinear relationships was performed; among others, it was chosen to use the Uniform Manifold Approximation and Projection for Dimension Reduction (UMAP) [4]. This method computes the embedding of numerical data and the embedding of categorical values and then merges the two results [5].
The validation of this method is performed by looking at the plot of the data points in the new space and determining if there are any evident clusters, because otherwise this dimensionality reduction is not useful in order to study the distribution of the data. In Figure 2 is shown the plot of a UMAP reduction in two dimensions; it can be seen that there is no clear division in clusters. As a second check, the true labels were plotted to see if there was a separation between points belonging to different classes, but, even in this case, no obvious data separation was visualized (this check was performed with all three target variables, both together and alone. Figure 2 shows only Diabetes). Even with higher embedding dimensions, the plot did not show any useful information regarding the clustering nature of the dataset.

Given the failure of these two techniques, it has been chosen not to perform dimensionality reduction on the dataset.
The difficulty in reducing this dataset could be a sign that it will be difficult to understand the relationship between different data objects, which could also affect the effectiveness of cluster algorithms, leading to the possibility that it might be difficult to perform some clustering and find a good clustering solution.

2.3. Distance

Because each instance of the dataset contains both numerical and categorical features, it is not possible to use a single distance metric for all feature variables to compute the distance between each data object. The solution to this problem is to compute a distance matrix using Gower's distance [6], [7].
Gower's distance is a distance measure that can be used to calculate the distance between two entities whose attributes have a mixture of categorical and numerical values, which is the current case. It uses the concept of Manhattan distance for continuous variables, and Dice distance for categorical variables, in order to measure the similarity between objects using the following mathematical formula:

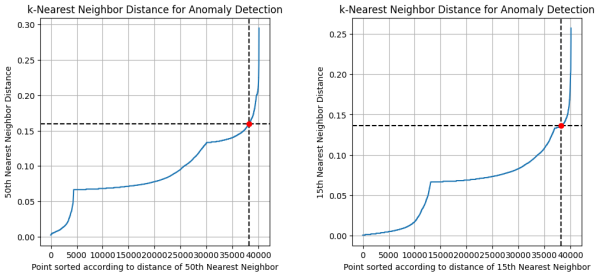
$$S_{ij} = \frac{\sum_k^n \omega_{ijk} S_{ijk}}{\sum_k^n \omega_{ijk}} \tag{1}$$

where S is the similarity value that needs to be computed, and in particular S_{ijk} is the contribution of the k-th variable, while the ω_{ijk} value depends on the fact if the comparison is valid or not for the k-th variable, taking values 1 or 0.

3. Anomaly Detection

An important and necessary step that must be addressed while preparing the data is to consider the presence of anomalies. If found, it is important to delete these anomalies from the data, because they can affect the quality of the clustering solution.

The approach used to detect such points was the k-th Nearest Neighbors [8]. This algorithm labels the samples that have a higher distance from their neighbors as anomalies, which means that isolated points or small groups of points are most likely to be considered outliers. The distance matrix was used for this computation.
For every sample, the distance taken into consideration was the one from the 50-th nearest neighbor, which is plotted in Figure 4a, where on the x-axis there are the points sorted considering their 50-th nearest neighbor distance increase. The plot was used to detect the outliers with the elbow method by picking the elbow point and discarding all the points with a distance from their 50-th nearest neighbor higher than the one corresponding to the selected point.



(a) 50th NN (b) 15th NN

Figure 4. Distance from the k-th Nearest Neighbor to find the anomalies. The elbow point is shown in red.

Indeed, from the elbow point on, the distance of the points from the 50-th nearest neighbor increases at a higher rate.

To check the validity of this method, the same was repeated by choosing 15 as the k-th nearest neighbor, and the plot of this investigation is shown in Figure 4b. With this method, selecting as elbow point a point that gave the exact number of outliers as those found with k = 50, it was found that the points identified as anomalies in the two cases were exactly the same.

In total, 1870 anomalies were found, which are 4,6% of the total. These points were removed from the dataset, scaling was applied, and the distance matrix was re-computed.

4. Clustering

Three clustering algorithms were used to find the best solution: Hierarchical clustering, DBSCAN, and k-Prototypes clustering.

4.1. Hopkins statistic

Before performing the clustering, the Hopkins statistic was computed [9], [10]. The Hopkins statistic is an unsupervised measure that is useful for better understanding the distribution of the data, and in particular to determine if there are inherent clusters in the data; otherwise, the clustering methods would not be effective and meaningful. The Hopkins statistic assesses the clustering tendency of a dataset by measuring the probability that it is generated by a uniform data distribution, that is, it tests the spatial randomness of the data.

The Hopkins statistic was computed both with the original dataset and with the clear (without anomalies) scaled one; the values found were:

- $H = 0.828$ for the original dataset.
- $H = 0.856$ for the clear and scaled dataset.

Both of these values are close enough to 1; therefore, it is safe to assume that the dataset actually has a natural tendency to form clusters. In addition, these values are quite similar to each other, but the value for the dataset without anomalies is slightly higher, indicating that removing the outliers makes it easy for the clustering algorithms to separate the data.

4.2. Hierarchical clustering

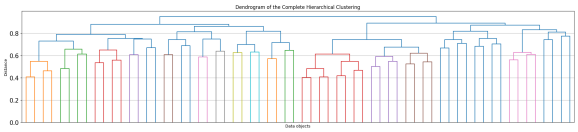
Hierarchical Clustering was first applied [11]. This algorithm produces a set of nested clusters organized as a hierarchical tree. Therefore, the clustering solution found with this algorithm can be visualized with a dendrogram, which is a tree-like diagram that records the sequences of merges or splits.

To perform some hyper-parameter tuning, the clustering was performed (using the precomputed distance matrix) more than once, changing from time to time different types of algorithms. The ones that were used are all agglomerative types of clustering with the following inter-cluster proximity definitions:

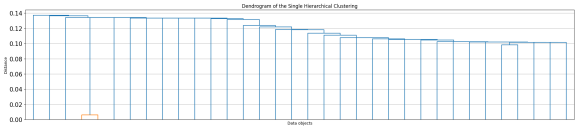
- complete (or maximum) linkage: the proximity of two clusters is based on the two most distant points in the different clusters. The dendrogram that corresponds to this clustering solution is shown in Figure 5a.
- single (or minimum) linkage: the proximity of two clusters is based on the two closest points in the different clusters. The dendrogram that corresponds to this clustering solution is shown in Figure 5b.
- Ward's method: the similarity of two clusters is based on the increase in squared error when two clusters are merged. The dendrogram that corresponds to this clustering solution is shown in Figure 5c.

For each of these methods, the Silhouette score [12] was computed while trying different numbers of clusters to find the optimal value. Figure 5d shows the results for each clustering algorithm in different colors. It is clear that Ward's method is the best, among the ones that have been tried, because the Silhouette score value is the highest and closest to 1 for every number of cluster values. Indeed, the Silhouette score is a metric used to compute the goodness of a clustering technique, and its value ranges from -1 (the clusters are assigned incorrectly) to 1 (the clusters are well separated and clearly distinguishable); this measure is further explained in Section 5.1.
Figure 5d shows also the elbow points [13] found for each clustering algorithm, which represent the optimal number of clusters in each case (three for single linkage and Ward's method and five for complete linkage). Since it was found that Ward's method yields the best clustering results, by looking at the Silhouette score, the actual optimal number of clusters chosen is 3. However, it is important to note that the number of clusters is not a hyper-parameter needed from the algorithm before clustering; it is just a choice of when to cut the clustering linkages after the computation.

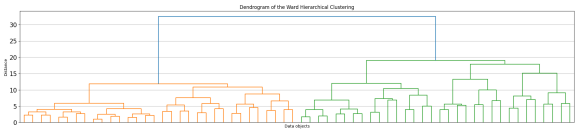
An explanation for why Ward's method and complete linkage perform better than single linkage could lie in the fact that the first two methods are less susceptible to noise with respect to single linkage, which, on the contrary, is really sensible to noise. This also provides an important



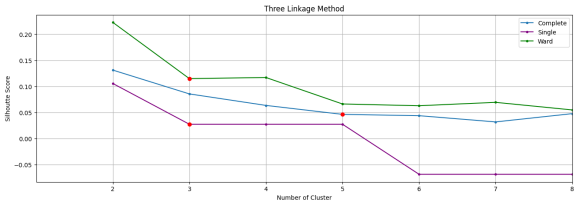
(a) complete linkage



(b) single linkage



(c) Ward's method



(d) Silhouette score

Figure 5. dendrogram of the Hierarchical clustering with different linkage methods: complete (a), single (b) and ward (c). These dendrograms are cut at a certain level in order to show more clearly the last cluster's linkages. In (d) it is shown the trend of the Silhouette score with the number of clusters with the elbow point (in red) found for the optimal number of clusters for each algorithm; different colors correspond to different linkage methods (blue = complete, purple = single, green = ward).

information about the dataset, which, as it turns out, contains a lot of noisy data, that were predictable given the high dimensionality of the dataset.
However, both Ward's method and complete linkage are biased towards globular clusters, while in the dataset taken into consideration the natural clusters may have a less regular shape (given also the high dimensionality), which could be the reason for the low value of the Silhouette score even for the best performing hierarchical clustering solution.

As previously said, it was found to be 3 the optimal number of clusters with this method. The data objects in these three clusters are divided as follows: 8.834 elements in the first cluster, 11.724 in the second cluster, and 17.680 in the third one.
To interpret the clusters related to the meaning of the target variables, it could be assumed that a clustering algorithm would divide the data into:

- Diabetes 1, Hypertension 0
- Diabetes 1, Hypertension 1
- Diabetes 0, Hypertension 0
- Diabetes 0, Hypertension 1

because these are all possible combinations of the outputs of Hypertension and Diabetes, whereas the third target variable, Stroke, as already mentioned in Section 2.1, is not considered because it is unbalanced in the dataset. However, since the clusters found by the algorithm are three and not four, it is possible to suppose that the algorithm may struggle to differentiate all those cases because it finds only three clusters out of those four.

4.3. DBSCAN

DBSCAN [14] is a density-based clustering method. This algorithm finds dense regions of objects and low-density regions; then, the objects belonging to high-density areas are grouped into clusters, whereas the points belonging to low-density regions are labeled as noise points and removed from the clustering.

DBSCAN was not able to properly cluster the points; indeed, it estimated only one cluster and a high number of noisy data, exactly 9.574. This last measure is in agreement with what was already inferred in Section 4.2, leading to the conclusion that the dataset contains many noisy points.

A reason of why density-based clustering, as DBSCAN, have troubles in performing the clustering of this dataset can be the fact that they do not work well with high-dimensional data, and in particular when the natural clusters of the data have different densities and/or overlap. Indeed, the overlap of clusters is probably the most relevant problem of the given dataset owing to the high number of features and the fact that most of them are categorical.

4.4. k-Prototypes clustering

The third type of clustering performed is k-Prototypes clustering [15], [16], [17].
This clustering algorithm is a variant of k-means clustering, which works only for continuous variables; on the contrary, k-Prototypes clustering is more suited for real-world problems, where the datasets have not only continuous but also categorical variables, as in the case dealt with in this project. In particular, k-Prototypes is a mixture of k-means (which clusters continuous variables) and k-modes (which clusters categorical variables). In this case, the distance matrix is not required to apply the algorithm.

To tune the hyper-parameters, the clustering algorithm was performed eight times, varying the number of clusters that the algorithm needed to create, from one to eight clusters. The results are shown in Figure 6, from which it is possible to identify the elbow point [13] that corresponds to the optimal number of clusters, which in this case is three, as in the case of hierarchical clustering.

The three clusters found using this method can have the same interpretation as the three clusters found using the hierarchical method.

The population of these three clusters id divided as: 5.960 elements in one cluster, 14.087 elements in the second one, and 18.191 elements in the third cluster.

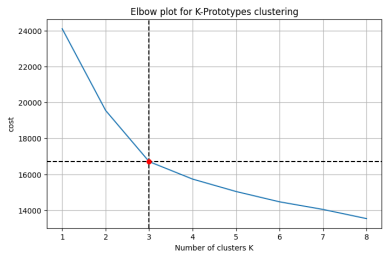


Figure 6. Elbow plot for the k-Prototypes clustering. On the y-axis there is the cost of the clustering solution, which is defined as the sum distance of all points to their respective cluster centroids.

5. Evaluation

After performing the clustering of the data, some evaluation of the goodness of the clustering solutions was performed to compare them and find the best clustering algorithm for this dataset. Both unsupervised and supervised measures have been reported, since the true labels were available.

Some unsupervised evaluation has already been performed and reported previously: the Silhouette score was computed in order to find the best clustering solution and the optimal number of clusters for the Hierarchical method. The cost (which is the measure of the overall distance between the data points and the centroids in the clusters) was computed in order to find the optimal number of clusters using the k-Prototypes method.

Because with the third clustering approach that was tried, DBSCAN, only one cluster was found, this method is considered unsuccessful and therefore not even considered in the following evaluations. The methods that were compared were Ward's method as hierarchical clustering, with three clusters, and k-Prototypes with three clusters.

5.1. Unsupervised performance measures

Average Silhouette Coefficient. The first unsupervised measure that was computed was the Average Silhouette Coefficient [12], already mentioned in Section 4.2.
This measure computes the average between the Silhouette coefficient of all the data samples, where the Silhouette coefficient for one point i is defined as:

$$S_i = \frac{b - a}{max(a, b)} \tag{2}$$

where:

- a is the average distance of the point i from the points of its cluster
- b is the minimum of the average distance of the point i from all the points in any of the other clusters

The values of the Silhouette Coefficient for a single point range from -1 to 1. A negative value means that the average distance of the point from the other points in its cluster (a) is greater than the minimum average distance from the points in another cluster (b), meaning that the sample has been assigned to the wrong cluster; a value near 0 indicates overlapping clusters, while 1 is the best possible value.

The values of the Average Silhouette Coefficient found for the two clustering solutions are:

- $S = 0,115$ for the Hierarchical clustering solution
- $S = 0,141$ for the k-Prototypes clustering solution

Both of these values are close to 0, suggesting that the clusters found in both cases overlap. This hypothesis adds to the fact that DBSCAN clustering cannot perform well, because it has a problem with overlapping clusters. Nevertheless, even though both values are low, the Silhouette coefficient found for the k-Prototypes solution is slightly higher than the other one. However, the difference is so small that it would not be safe to assume that this method is actually better than the hierarchical one.

Dunn Index. The second unsupervised measure performed is the Dunn Index [18], [19].
The Dunn index computes the minimum distance between two clusters and divides it by the diameter of the largest cluster as follows:

$$D = \frac{\min_{1 \leq i < j \leq q} d(C_i, C_j)}{\min_{1 \leq k \leq q} diam(C_k)} \tag{3}$$

where, if in total there are q clusters:

- $d(C_i, C_j) = \min_{X \in C_i, Y \in C_j} d_{MT}(X, Y)$ is the distance between two clusters C_i and C_j, with d_{MT} distance for mixed-type data, which in this case is the Gower distance
- $d(C_k) = \max_{X,Y \in C_k} d_{MT}(X, Y)$ is the diameter of a cluster C_k, i.e. a measure of its intra-cluster distance

The value of this index ranges between 0 and $+\infty$, where a higher value indicates a better clustering solution.

The results found are:

- $D = 0,0262$ for the hierarchical clustering solution
- $D = 0,0009$ for the k-Prototypes clustering solution

These values are very low and close to 0, which means that, in both cases, the clusters found overlap, as already found with the Silhouette coefficient. This is because the distance between the two closest clusters is small compared with the dimensions of the clusters.
In addition, the two values were very similar; however, in this case, the value found for the hierarchical solution was the highest.

5.2. Supervised performance measures

Rand Index. The first supervised measure computed was the Rand Index [20]. The Rand Index R is a measure used

to compare the clustering solutions two by two, and it is computed as:

$$R = \frac{a+d}{a+b+c+d} = \frac{a+d}{M} \tag{4}$$

where:

- a is the number of pairs of elements that belong to the same cluster in both the two clustering methods
- b is the number of pairs that for one clustering method are in the same clustering, while for the other are in different clusters
- c is the same of b but with the clustering solutions inverted
- d is the number of pairs that belongs to the different clusters in both the two clustering solutions
- $M = \binom{m}{2} = \frac{m(m-1)}{2}$ is the overall number of pairs, if m is the number of objects in the dataset

The Rand Index is always a value between 0 and 1, where 0 means that the two clustering solutions are in strong disagreement in the clustering of pairs of objects, and 1 means that the two clusterings give the exact same result in terms of clustering the data.

In this case, the Rand Index was computed for both the hierarchical and the k-Prototypes solutions by comparing the three clusters found in both cases with different hypotheses of the natural clusters of the dataset. Regarding the combinations of the target variables, only Diabetes and Hypertension were considered, since both clustering solutions will have trouble dealing with Stroke because of its imbalance, as already mentioned. In addition, two and four clusters are taken into consideration as the ground truth (not only three as the ones that have been found) because the clustering solutions may split one cluster or merge two different ones. The results are reported in Table 2.

All the values of the Rand Index are close to 0,5 and no value outperforms the others, which means that there is no clear answer to what the best clustering algorithm is. In addition, the similarity of the results does not allow us to understand which is the actual predicted target among all the ones for which the Rand Index was computed.

Adjusted Rand Index. Given the conclusions found after computing the Rand Index, a doubt that could arise is the fact that the clustering of the data objects into particular clusters may be casual; therefore, the Adjusted Rand Index ARI [21] was computed. Indeed, the Adjusted Rand Index takes into account the fact that some agreement between the two clustering methods can occur by chance, and it adjusts the Rand Index to consider this possibility.
It is computed starting from the Rand Index and the expected value E of the Rand Index for random clustering solutions as:

$$ARI = \frac{R-E}{max(R)-E} = \frac{R-E}{1-E} \tag{5}$$

The ARI value ranges from -1 to 1: the higher this value, the closer the two clusters are to each other. In particular, 1 indicates perfect agreement between the two clustering solutions, 0 indicates random agreement, and -1 indicates that the two are completely different.
In this case, the measure was computed for the same targets as those used for the Rand Index. Table 3 presents the results.

These results confirm the hypothesis that the clustering made with both the hierarchical method and the k-Prototypes method is casual; indeed, all values are close to 0.

Fowlkes and Mallows. A third supervised measure was computed: the Fowlkes and Mallows score FM [22]. This score measures the similarity between two sets, and its value lies between 0 and 1: the higher this value is, the higher the similarity between the two sets is. Its formula is the following:

$$FM = \sqrt{\frac{a}{a+b} \times \frac{a}{a+c}} \tag{6}$$

where a, b and c represent the same thing as the variables in Equation 4.

The results for this measure are reported in Table 4. All these results are just below or around 0,5, this means that the clusterings compared (Hierarchical/k-Prototypes with the ground truth) are not that similar, leading to the same conclusions already obtained with the Rand Index.

6. Conclusion

In conclusion, the analysis performed does not allow to obtain meaningful clustering results for the given dataset. Out of the three clustering methods that have been performed, only the Hierarchical and k-Prototypes obtained reasonable results; however, none of them was able to obtain a good clustering solution.
Many trials have been conducted to improve the results, both in terms of hyper-parameter tuning for the clustering algorithm and pre-processing of the data, without finding any solution.
The reason behind the failure in clustering the given dataset, therefore, can be imputed to the preliminary condition that has been observed in the analysis: the consideration of the variables Hypertension and Stroke as targets and not as feature variables, as they were originally intended to be considered in the Kaggle competition. Indeed, these variables would carry important information about the value of the variable Diabetes, which is the original target.

If one would like to improve the analysis of the given dataset following the direction of this project, some changes could be attempted.
First, in this study it was chosen not to perform any dimensionality reduction because the approaches that have been tried (FAMD and UMAP) were found to be ineffective, but it could be clever to try and implement an autoencoder adjusted for this particular dataset in order to perform dimensionality reduction; this method could succeed because

TABLE 2. VALUE OF THE RAND INDEX FOR DIFFERENT CLUSTERING SOLUTION.

Target	$R_{hierarchical}$	$R_{k-Prototypes}$
(D=1), (D=0)	0,534	0.530
(H=1), (H=0)	0,528	0,517
(S=1), (S=0)	0,403	0,429
(D=1 & H=1), (D=1 & H=0), (D=0 & H=1), (D=1 & H=0)	0,589	0.572
(D=0 & H=0), (D=0 & H=1), (D=1 & (H=1 *or* H=0)	0,561	0,542
(D=1 & H=0), (D=1 & H=1), (D=0 & (H=1 *or* H=0)	0,562	0,560
(D=1 & H=0), (D=0 & H=0), (H=1 & (D=1 *or* D=0)	0,553	0,533
(D=1 & H=1), (D=0 & H=1), (H=0 & (D=1 *or* D=0)	0,564	0,556

TABLE 3. VALUE OF THE ADJUSTED RAND INDEX FOR DIFFERENT CLUSTERING SOLUTION.

Target	$ARI_{hierarchical}$	$ARI_{k-Prototypes}$
(D=1), (D=0)	0,069	0.060
(H=1), (H=0)	0,058	0,035
(S=1), (S=0)	0,018	0,030
(D=1 & H=1), (D=1 & H=0), (D=0 & H=1), (D=1 & H=0)	0,068	0.053
(D=0 & H=0), (D=0 & H=1), (D=1 & (H=1 *or* H=0)	0,065	0,036
(D=1 & H=0), (D=1 & H=1), (D=0 & (H=1 *or* H=0)	0,072	0,077
(D=1 & H=0), (D=0 & H=0), (H=1 & (D=1 *or* D=0)	0,063	0,029
(D=1 & H=1), (D=0 & H=1), (H=0 & (D=1 *or* D=0)	0,063	0,058

TABLE 4. VALUE OF THE FOWLKES AND MALLOWS INDEX FOR DIFFERENT CLUSTERING SOLUTION.

Target	$FM_{hierarchical}$	$FM_{k-Prototypes}$
(D=1), (D=0)	0,466	0.474
(H=1), (H=0)	0,460	0,461
(S=1), (S=0)	0,576	0,601
(D=1 & H=1), (D=1 & H=0), (D=0 & H=1), (D=1 & H=0)	0,370	0.369
(D=0 & H=0), (D=0 & H=1), (D=1 & (H=1 *or* H=0)	0,416	0,410
(D=1 & H=0), (D=1 & H=1), (D=0 & (H=1 *or* H=0)	0,425	0,440
(D=1 & H=0), (D=0 & H=0), (H=1 & (D=1 *or* D=0)	0,426	0,418
(D=1 & H=1), (D=0 & H=1), (H=0 & (D=1 *or* D=0)	0,408	0,417

it would have fewer problems with the high-dimensionality of the data and in dealing with complex relations between the variables.
In addition, the metric used to compute the distance between the data points could be changed instead of using Gower; for example, an intuitive distance for mixed-type data is the following [18]:

$$d_{MT}(X,Y) = \sum_{j=1}^{l}(x_j - y_j)^2 + \lambda \sum_{j=l+1}^{m} \delta(x_j, y_j) \qquad (7)$$

where $\lambda > 0$ and *Simple Matching*:

$$\delta(x_j, y_j) = \begin{cases} 0 \text{ if } x_j = y_j \\ 1 \text{ if } x_j \neq y_j \end{cases} \qquad (8)$$

Another aspect that could be tried is to manage the variables in a different way than the one reported here, for example by converting all the variables to categorical; indeed, the variables that were treated as numerical in this report (BMI, MentHlth, PhysHlth) could be interpreted as categorical because they are not continuous. By doing this, it would also be possible to perform other types of clustering, such as k-modes.

Acknowledgments

The authors would like to thank Professor Fabio Antonio Stella and Doctor Giulia Cisotto, who held the course of Unsupervised Learning, for their explanations of the subjects addressed in this project.

The authors hereby declare that the contents of this paper are original and not generated by language models and all references and sources used have been properly cited.

References

[1] Kaggle Competition: https://www.kaggle.com/competitions/diabetes-prediction-tfug-chd-nov-2022.

[2] Postance, B. (June 2023). "Clustering Mixed Data." bpostance.github.io. https://bpostance.github.io/posts/clustering-mixed-data/.

[3] Halford, M. "Prince". https://github.com/MaxHalford/prince.

[4] McInnes, L. "UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction". https://umap-learn.readthedocs.io/en/latest/

[5] Eoghan Keany. (November 2021). "The Ultimate Guide for Clustering Mixed Data". https://medium.com/analytics-vidhya/the-ultimate-guide-for-clustering-mixed-data-1eefa0b4743b

[6] Analytics Vidhya. (June 2023). "Concept of Gowers Distance and Its Application Using Python." Medium. https://medium.com/analytics-vidhya/concept-of-gowers-distance-and-it-s-application-using-python-b08cf6139ac2.

[7] PyPI. "Gower." Python Package Index. https://pypi.org/project/gower/.

[8] scikit-learn. (2023). "Neighbors-based learning - sklearn.neighbors." scikit-learn Documentation. https://scikit-learn.org/stable/modules/neighbors.html.

[9] Rohanadagouda. "Unsupervised Learning using K-prototype and DBSCAN." Kaggle. https://www.kaggle.com/code/rohanadagouda/unsupervised-learning-using-k-prototype-and-dbscan#Hopkins-Statistics.

[10] Deore, S. (May 2022). "Really, what is Hopkins Statistic?" Medium. https://sushildeore99.medium.com/really-what-is-hopkins-statistic-bad1265df4b.

[11] SciPy. (2023). "Hierarchical Clustering - scipy.cluster.hierarchy." SciPy Documentation. https://docs.scipy.org/doc/scipy/reference/cluster.hierarchy.html.

[12] scikit-learn. (2023). "silhouette_score - sklearn.metrics." scikit-learn Documentation. https://scikit-learn.org/stable/modules/generated/sklearn.metrics.silhouette_score.html.

[13] PyPI. "kneed." Python Package Index. https://pypi.org/project/kneed/.

[14] scikit-learn. (2023). "DBSCAN - sklearn.cluster.DBSCAN." scikit-learn Documentation. https://scikit-learn.org/stable/modules/generated/sklearn.cluster.DBSCAN.html.

[15] PyPI. "kmodes." Python Package Index. https://pypi.org/project/kmodes/.

[16] Huang, Z. "Extensions to the k-Means Algorithm for Clustering Large Data Sets with Categorical Values". Data Mining and Knowledge Discovery 2, 283–304 (1998). https://doi.org/10.1023/A:1009769707641.

[17] Zazueta, Z. A. (November 2021). "K-Prototypes Clustering: For When You're Clustering Continuous and Categorical Data." Medium. https://zachary-a-zazueta.medium.com/k-prototypes-clustering-for-when-youre-clustering-continuous-and-categorical-data-6ea42c2ab2b9.

[18] Aschenbruck, R., and Szepannek, G. (2020). "Cluster validation for mixed-type data". Archives of Data Science, Series A, 6(1), 02.

[19] Rizzo, D. (June 2023). "Text clustering with K-means and TF-IDF." GitHub Gist. https://gist.github.com/douglasrizzo/cd7e792ff3a2dcaf27f6.

[20] scikit-learn. (2023). "rand_score - sklearn.metrics." scikit-learn Documentation. https://scikit-learn.org/stable/modules/generated/sklearn.metrics.rand_score.html.

[21] scikit-learn. (2023). "adjusted_rand_score - sklearn.metrics." scikit-learn Documentation. https://scikit-learn.org/stable/modules/generated/sklearn.metrics.adjusted_rand_score.html.

[22] scikit-learn. (2023). "fowlkes_mallows_score - sklearn.metrics." scikit-learn Documentation. https://scikit-learn.org/stable/modules/generated/sklearn.metrics.fowlkes_mallows_score.html.