Unsupervised Learning Exam Report

Filippo Monaco
Università degli Studi di Milano-Bicocca
Id: 840089

Marco Picione
Università degli Studi di Milano-Bicocca
Id: 827116

***Abstract*—This project report provides an overview of our approach to clustering the given dataset. We cover data pre-processing, our choice of clustering models and their optimization, and conclude with a supervised evaluation of their performance.**

Link to the Colab: https://colab.research.google.com/drive/1vGTI7CGN8tt_gISq5jXuRKxqz9wT58a-?usp=sharing

1. Introduction

The objective of this project is to develop a robust clustering solution for the provided dataset, with the goal of trying to predict whether a person might suffer from a certain condition (Diabetes, Hypertension, or Stroke), given a series of medical information.
Most of the report will focus on the data pre-processing we performed in order to clean up and simplify the dataset (section 2), while in section 3 we will focus on the choice of models we used to cluster the dataset, and their unsupervised optimization. Finally, in section 4, an evaluation of the models is presented, using a supervised approach.

2. Data Pre-processing

Before clustering the data using the previously mentioned algorithms, some data exploration and pre-processing is requiried in order to have a clear picture of what the data actually is, and how to best manage it. Specifically, this is done to perform feature extraction, anomaly detection, and apply normalization. This section is structured as follows: in subsection 2.1 the provided data is properly explored, in section 2.2 univariate considerations are made. Follows section 2.3 in which scaling and mixed data are examined, and finally in section 2.4 outliers removal is dealt with.

2.1. Dataset Exploration

The dataset is taken from the Diabetes Prediction (TFUG Chd Nov 2022) Kaggle challenge [1]. It is composed of 40108 objects, 18 attributes each. Table 1 summarizes each feature by providing their observed variability range and data type.

As one can see, the attributes have different types: as properly explained in section 2.3 the analysis has to deal with such variability.

Variable	Range	Type
Age	1-13	Ordinal
Sex	0-1	Nominal
HighChol	0-1	Nominal
CholCheck	0-1	Nominal
BMI	12-98	Ratio
Smoker	0-1	Nominal
HeartDiseaseorAttack	0-1	Nominal
PhysActivity	0-1	Nominal
Fruits	0-1	Nominal
Veggies	0-1	Nominal
HvyAlcoholConsump	0-1	Nominal
GenHlth	1-5	Ordinal
MentHlth	0-30	Ratio
PhysHlth	0-30	Ratio
DiffWalk	0-1	Nominal
Diabetes	0-1	Nominal
Hypertension	0-1	Nominal
Stroke	0-1	Nominal

TABLE 1: Variables with their observed range and their type.

After ensuring that no missing value is present, the presence of 2456 duplicates was noticed: since we are dealing with medical data and none of the attributes indicates a specific patient (an ID for example), we decided not to remove them.

Since the proposed analysis is costly in terms of resources (RAM) and computational time, all the following analysis is performed only on a randomly sub-sampled subset (75%) of the original data. Furthermore, the attributes “Diabetes”, “Hypertension” and “Stroke” are used as target variables for supervised evaluation of the obtained solutions, and thus were not considered during clustering.

2.2. Univariate considerations

Before clustering the data, a box plot and the distribution for each univariate attribute is plotted in order to search for outliers and eventual attributes that could be removed. Follow some specific considerations for specific attributes:

Cholesterol As one can see from figure 1, the "CholCheck" attribute has a highly unbalanced distribution and the information it carries is redundant when combined with "HighChol". Indeed, for one to have a correct value of HighChol, they would necessarily need to have done a CholCheck. For this reason it was decided to completely discard that attribute, whilst making sure to also remove all the objects that presented value 0 for that attribute.

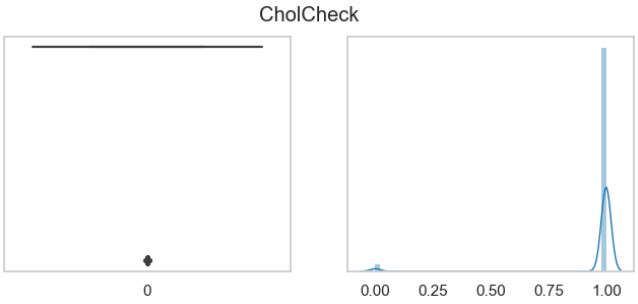

Figure 1: Box plot (left) and distribution (right) of attribute "CholCheck".

Heavy Alcohol Consumption We also wanted to highlight the distribution of the HvyAlcoholConsump attribute, as shown in figure 2. From the image, we can see that most points have value 0, and all others are considered outliers. One might be inclined to remove these points, as we did previously with the attribute related to Cholesterol. However, in this case, we believe that this feature can be useful to predict our target variables. Additionally, the dataset does not have a correlated variable (as with CholCheck and HighChol).

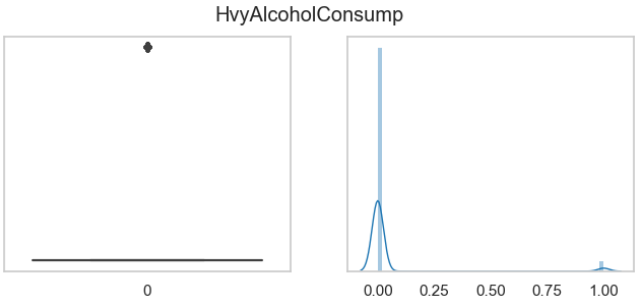

Figure 2: Box plot (left) and distribution (right) of attribute "HvyAlcoholConsump".

BMI In figure 3, we can see the boxplot associated to the "BMI" variable, which shows the presence of several outliers. The objects associated with those values (BMI $\geq$ 46) were removed, since, giving context to the attribute, they were miss-calculated or just reported incorrectly (values above 50 should be medically impossible).

Stroke Even if it is a target variable, it will be useful for section 4 to look at the distribution of the Stroke label. From

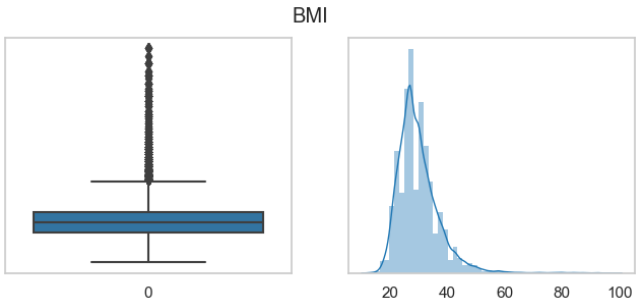

Figure 3: Box plot (left) and distribution (right) of attribute "BMI".

figure 4 we can indeed see how most points have label 0, whilst only a small fraction having value 1. This will be used to justify later on the results found from clustering (spoiler alert!), but we thought it best to include in this section.

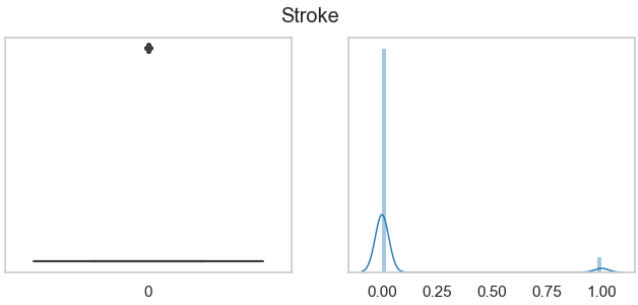

Figure 4: Box plot (left) and distribution (right) of target variable "Stroke".

2.3. Scaler and Gower Distance

Before running the cluster algorithms, in order to have all the attributes varying in the same range (between 0 and 1), the MinMaxScaler was applied.
Furthermore, since we are dealing with mixed data (both categorical and non-categorical), one needs to make sure to use the correct metric for computing the distance between points. In this case, a specific metric is needed for each of the different attribute types (nominal, ordinal, and ratio). The solution to this problem is given by Gower's distance [4]: when considering the similarity between two objects i and j, each of their attribute is pairwise-compared using the relative metric. The total distance is then obtained by performing a weighted average of all the distances. Figure 5 shows the obtained distance matrix.

2.4. Outlier detection

Considering all of the fourteen attributes left and the previously obtained distance matrix, we look for outliers inside

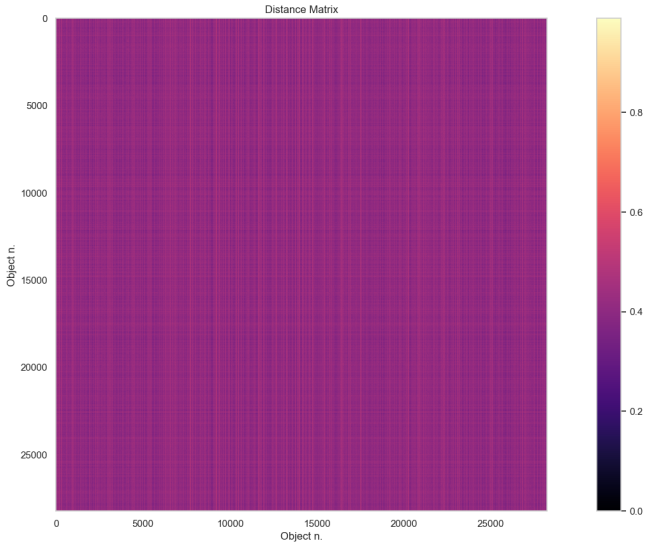

Figure 5: Gower distance matrix of the dataset before removing outliers through anomaly detection.

the dataset. In particular, using the k-nearest neighbors algorithm, we look for sparse data: for each point in the dataset, the distance between itself and its four nearest neighbors is computed. Then, the average of these distances is assigned to each point. The reasoning behind this is that sparser points will have a greater distance to their neighbors compared to regular points and thus will be marked as outliers. In order to correctly choose a threshold to distinguish between outlier and inlier points, the data is first sorted according to the average distance previously mentioned, and then plotted. Finally, by using the elbow-metohd, the threshold point is chosen. Figure 6 shows the obtained curve as well as the distribution of points according to the average distance to their corresponding neighbors.

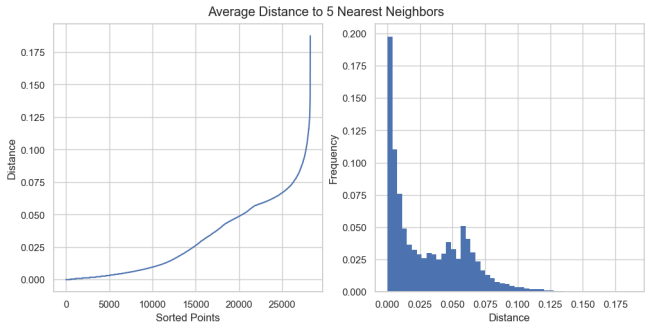

Figure 6: Average distance to four nearest neighbors, sorted (left), and corresponding distribution (right).

The cutting point was visually chosen to be 0.75 [a.u.]: all points with an average distance to their four nearest neighbors greater than this value are marked as outliers and therefore discarded from further analysis.

Considering all the steps performed during the preprocess, the clustering is performed on 26270 objects with 14 attributes each.

3. Clustering Models

The clustering algorithms proposed are hierarchical (3.1) and k-prototype (3.2), a version of k-means that is able to deal with mixed data.

3.1. Hierarchical Clustering

The Hierarchical Clustering algorithm organizes data in a tree structure called dendrogram, by first considering all the data objects as separate clusters, then merging the two most similar ones, until only one remains. The edge length between two nodes represents the dissimilarity of the merged clusters. This metric is computed using the distance matrix and a merging method (hyper-parameter). In this case, the method "Complete" was chosen: two clusters are compared according to the maximum distance of two elements inside different clusters. Figure 7 shows the obtained dendrogram.

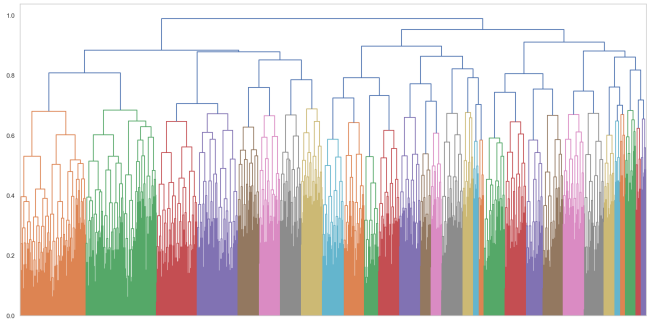

Figure 7: Dendrogram obtained with "Complete" as merging method.

In order to find the best cutting point and thus the best number of clusters, several solutions were evaluated according to unsupervised performance metric: cohesion, separation, and silhouette scores. Figure 8 shows the obtained curves.

As one can see from the plot, the silhouette scores drops rapidly as soon as the number of clusters increases; since this score takes implicitly into account a point-wise definition of cohesion and separation, the best number of clusters for the hierarchical approach is set to 2.
Figure 9 shows the dendrogram with cutting point at 0.973.

The interpretation of clusters will be properly described in section 4.

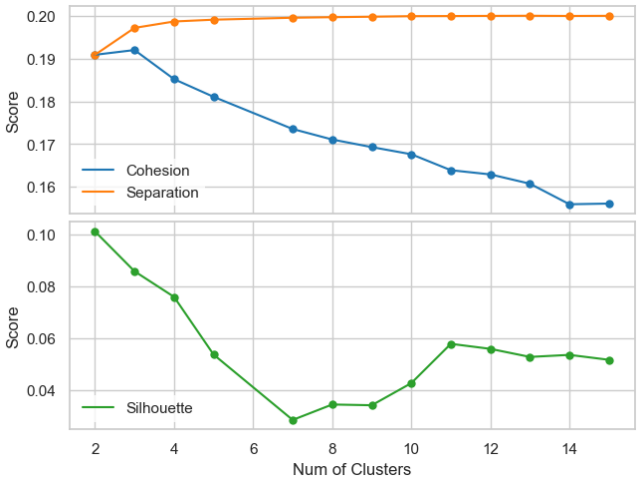

Figure 8: Cohesion, separation, and silhouette scores for hierarchical clustering in function of number of clusters.

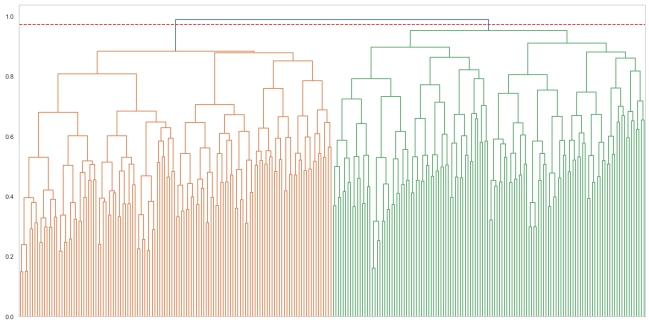

Figure 9: Dendrogram cut at 0.973 in order to obtain two clusters.

3.2. k-Prototype Clustering

K-prototype algorithm works similarly to the k-means++ algorithm, updating the centroids at each iteration according to the current partition of data. The idea of k-prototype is to extend the k-means++ algorithm in order also consider mixed data: the similarity between two points is calculated according to different metrics related to the attribute type. Additionally, the data points themselves are not actually needed to use k-prototypes: the distance matrix is enough. As k-means++, k-prototype takes as arguments the number of initialization, and the initialization method. To tune those parameters, we evaluated the inertia for different values of both initialization method and number of initializations. Figure 10 shows the mean and the variance of the inertia of the two methods, repeating the clustering twenty times for each number of initialization.

As one can see, "Huang" initialization method reaches both a smaller average value of cost (inertia) and variance, which finds its minimum for number of initializations in 6. In general, one can expect that inertia will approach a

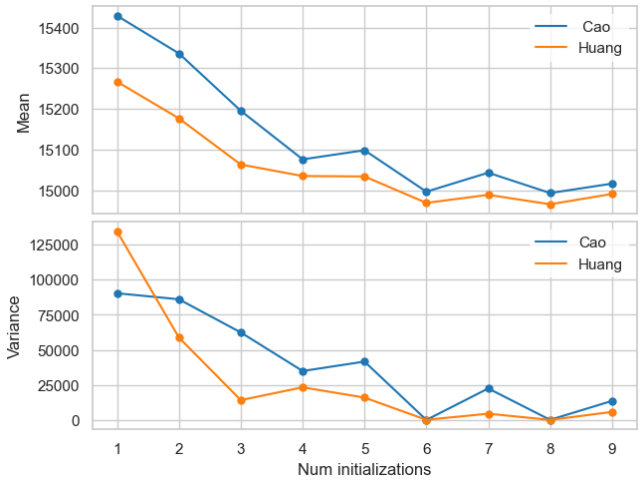

Figure 10: Mean and variance vs the number of initialization.

constant value as the number of iterations increases: the following fluctuation are due to the fact that the cluster centres initialization is not deterministic, which is why this analysis was performed. The idea is then to find the lowest number of initializations so that the solution is optimal whilst also not taking too long to compute.

Finally, choosing "Huang" as initialization method, 6 as number of initializations, and performing again the clustering, cohesion, separation and silhouette scores were computed. Figure 11 shows the obtained curves along with the inertia curve.

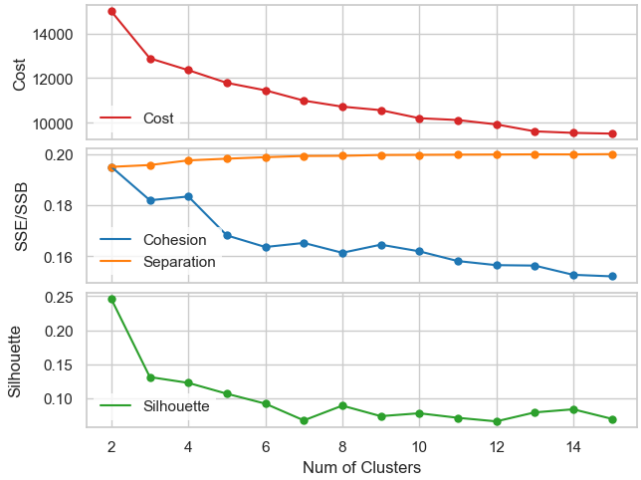

Figure 11: Cohesion, separation, and silhouette scores for k-prototype clustering in function of number of clusters.

As one can see, no elbow point was found in the cost curve, and therefore the optimal number of cluster was chosen according to the silhouette score and set to two.

4. Evaluation

After performing the clustering, the next step is the interpretation of results. This section is structured as follows: in subsection 4.1 a supervised evaluation of the obtained clustering solutions is provided; follows a cluster interpretation in subsection 4.2, and finally feature selection is described in paragraph 4.3.

4.1. Supervised evaluation

Using attributes "Diabetes", "Hypertension" and "Stroke" as true labels, a supervised evaluation was performed. The following tables summarizes the obtained results for Rand and Fowlkes-Mallows scores for both hierarchical (Table 2) and k-prototype (Table 3) algorithms.

	Rand	Fowlkes-Mallows
Diabetes	0.52	0.52
Hypertension	0.52	0.52
Stroke	0.51	0.67

TABLE 2: Obtained scores for hierarchical algorithm

	Rand	Fowlkes-Mallows
Diabetes	0.52	0.61
Hypertension	0.50	0.60
Stroke	0.70	0.82

TABLE 3: Obtained scores for k-Prototype algorithm

Since both scores indicate a better solution when closer to 1, we can conclude that both hierarchical and k-prototypes clustering is not optimal, as we can see from tables 2 and 3.

The reason for Stroke being that much better compared to the other 2 target variables, is due to its distribution in the dataset, as previously mentioned in section 2.2. Since most points have label 0, and the clustering solution found by k-prototypes is very unbalanced (as we will see in the next section), it makes sense for it to have a higher score.

4.2. Clusters interpretation

In this section, an interpretation of the clusters found by both models is provided. Figure 12 shows the distribution of labels assigned by Hierarchical and k-prototype algorithm to each data point. We can clearly see how k-prototypes found two clusters of vastly different sizes, while the hierarchical model divided the data set in two balanced subsets.

Even if the found solution is not optimal, an interpretation of the clusters is provided.

In order to see if the clusters separate the response variable, precision and recall were evaluated for each target variable using the clustering solution provided by k-prototype algorithm. Table 4 shows the obtained results.

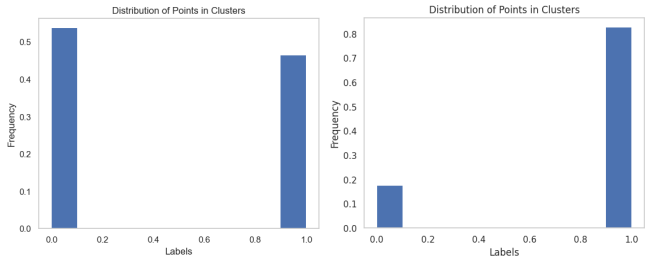

Figure 12: Distribution of labels inside the two clusters for hierarchical clustering (left) and k-prototype (right)

	Precsion	Recall
Diabetes	0.45	0.74
Hypertension	0.50	0.76
Stroke	0.04	0.56

TABLE 4: Precision and Recall scores for each target variable for the k-prototype solution.

This result can also be visually inspected in figure 13 where the true negative, false negative, true positive, and false positive counts are reported.

Again, due to the non optimal clustering solution found, there is no remarkable result.

A possible explanation of this might be the fact that the used attributes are not sufficiently related to the response variables and thus we cannot interpret the obtained clustering as related to a particular disease.

4.3. Feature selection

If a good clustering solution was found, a selection of features could have been made by looking at the distribution of the variables inside the two clusters. Figures 14 and 15 show a boxplot for each attribute inside the two clusters for both used clustering algorithms.

By visually looking at the boxplots is possible to exclude from the analysis the attributes that have a similar distribution in both clusters. The rational of this is that they are not related to the obtained clustering solution. Unfortunately, in the case of this work, such an interpretation is not possible due to the non optimal clustering solution found. In spite of everything, if it weren't like this looking for example at figure 14 one can think to exclude the attributes "Sex", "HighChol", "Smoker","HearthDiseasorAttack", "Fruits", "Veggies" and "HvyAlcoholConsump".

Furthermore, always just in case of a good clustering solution, one may be interested in finding the attributes related to a particular response disease. In order to do so, one could think of selecting data inside the cluster more related to that disease and performing only on those data a further clustering analysis.

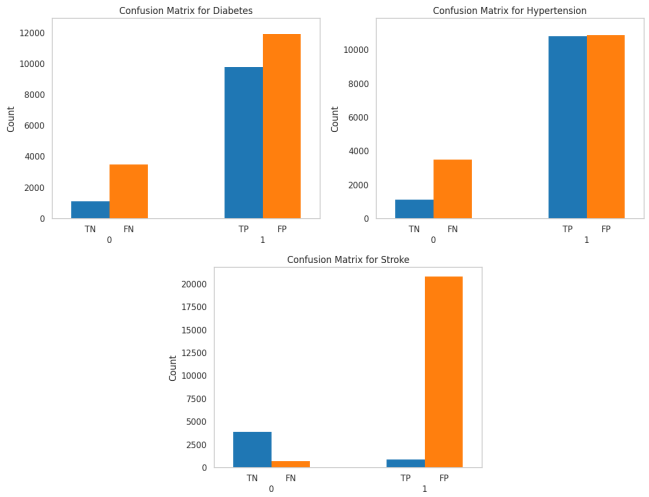

Figure 13: True negative, false negative, true positive and false positive for each target disease

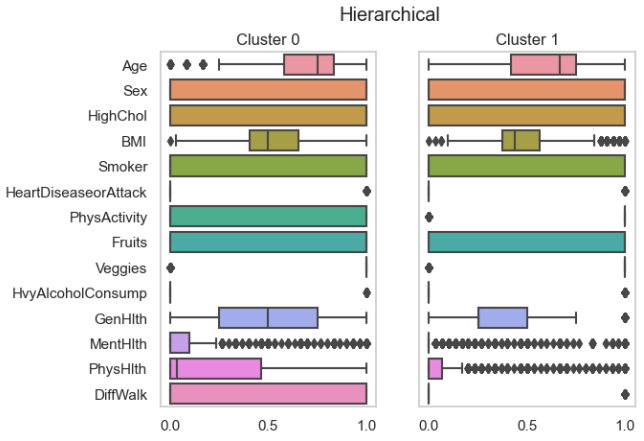

Figure 14: Box plots for each attribute inside both clusters for hierarchical algorithm.

5. Conclusion

In this work a medical dataset was analyzed in an unsupervised fashion in order to predict the response of three diseases. Even if we did not exhaust every possible optimization, and the found clustering solution was not optimal, we can confidently say that we are proud of this project, and that we learned a lot from it.

6. Take home message

"If you know the dataset and you know yourself, you need not fear the result of a hundred clustering solutions. If you know yourself but not the dataset, for every solution found you will also face an interpretation nightmare. If you know neither the dataset nor yourself, you will succumb before even opening the .xlsx file.".

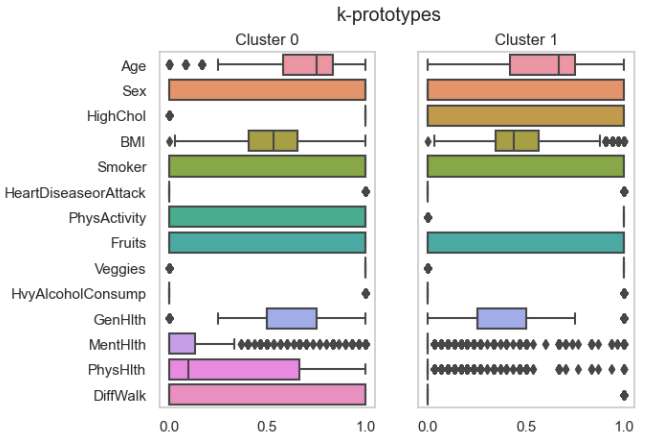

Figure 15: Box plots for each attribute inside both clusters for k-prototype algorithm.

- Sun Tzu, probably

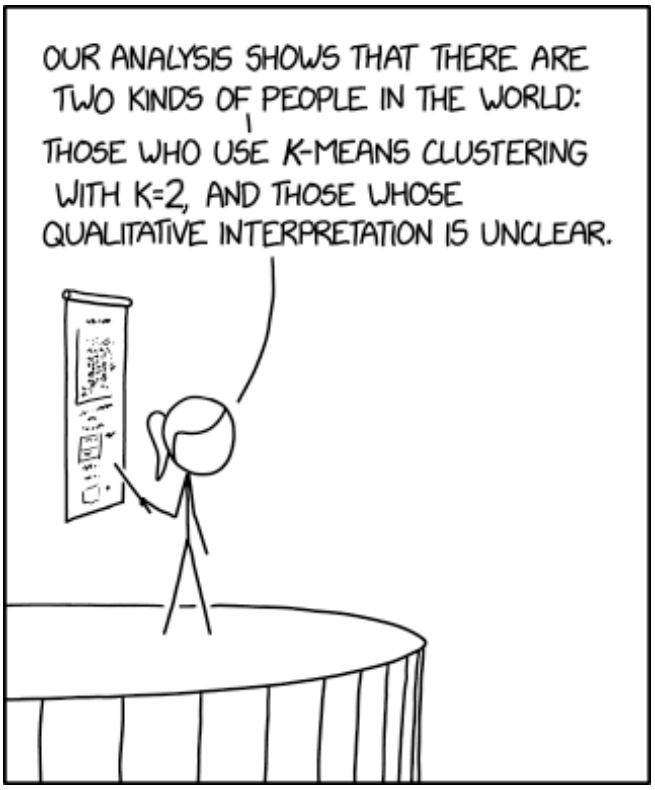

Figure 16: (This comic would have been funny if the solution we found made sense).

Acknowledgments

The authors would like to thank themselves for the hard work done.

References

[1] Kaggle: data science competition platform, where the dataset was taken from [https://www.kaggle.com/competitions/diabetes-prediction-tfug-chd-nov-2022/overview]

[2] scikit-learn: tools for predictive data analysis [https://scikit-learn.org/stable/modules/classes.html]

[3] scipy: Python library for scientific computing, data manipulation, numerical optimization, and more. https://pypi.org/project/scipy/]

[4] wwwjk366/gower: Python library for computing Gower's Distance https://github.com/wwwjk366/gower]

[5] kmodes: Python library for k-modes and k-prototypes implementation https://pypi.org/project/kmodes/]

[6] Out Past Lab Sessions: much of the code used in this project was taken (and obviously re-adapted) from our past weekly assignments [https://drive.google.com/drive/folders/1cjYkmQqQOqEh25NsYcwi2EQarrK3AXtJ?usp=sharing]

We confirm that this report is entirely original and does not contain any form of plagiarism. We did not make us of ChatGPT or any other Natural Language Processing models.