Diabetes, Stroke, Hypertension: A Clustering Approach to Prediction

Antonio Balordi
Matr. Num. 842971,
MSc Artificial Intelligence for Science and Technology
Email: a.balordi@campus.unimib.it

Daniele Rossetti
Matr. Num. 843111,
MSc Artificial Intelligence for Science and Technology
Email: d.rossetti15@campus.unimib.it

***Abstract*—How can some of the most relevant diseases be prevented? We discuss Diabetes, Stroke and Hypertension. Is it even possible to find risk factors for which it is possible to perform a prediction? These are some of the questions that health officials are investigating. Owing to the development of data analysis techniques, there are more instruments than before.**

In this report, we investigate the possibility of implementing a clustering algorithm on our dataset for exactly this scope. We sought to elucidate the primary risk factors associated with these diseases and develop a predictive framework to inform proactive interventions and personalised healthcare strategies.

1. Introduction

Established in 1984 the *Behavioral Risk Factor Surveillance System* (BRFSS) [1] monitor over all the 50 states and collects continuously data via medical interview, creating one of the biggest health survey system in the world.
The prevalence of chronic diseases, such as Stroke, Hypertension and Diabetes, is one of the most significant challenges for public health, which has a relevant impact on lifestyle and represents a burden to the economy. The ability to predict, or at least assess, the risk of these diseases is fundamental for health officials, since an early diagnosis can lead to lifestyle changes and more effective treatment.

We propose a novel approach to the prediction task based on clustering analysis, trying to identify the features that are the most relevant and to assess the risk factors.
Traditional approaches to prediction suffer from a high failure risk owing to the elevated number of different features that influence the final outcome. By approaching the problem with cluster analysis, we aim to identify the risk factors that could result in the highest accuracy.
We hypothesized that some risk factors are shared among these diseases and patients with multiple conditions.
Using cluster analysis, we are searching for hidden patterns and relationships between risk factors and developing a predictive framework that can be used to improve the quality of life of patients.

1. https://www.cdc.gov/brfss/index.html

Our project endeavors to advance the current understanding of these chronic diseases and to provide insights into preventive measures. The final objective is to minimize the risk of these diseases, improve patient outcomes, and alleviate the burden on the public health system.

2. Dataset Description

The dataset consisted of eighteen variables, including demographic information, health information, and lifestyle choices. Specifically, the variables were: age, sex, high-cholesterol status, cholesterol check history, Body Mass Index (BMI), smoking history, presence of heart disease or heart attack, level of physical activity, fruit and vegetable consumption, heavy alcohol consumption, self-perceived general health, mental health status, physical health status, difficulty walking or climbing stairs, hypertension status, stroke history, and diabetes status.
Age was categorized into fourteen different levels. Sex is represented as a binary variable, with 0 indicating female and 1 indicating male. High cholesterol status was represented as a binary variable, with 0 indicating the absence of high cholesterol and 1 indicating its presence. Cholesterol check history was also binary, with 0 indicating no cholesterol check in the past 5 years and 1 indicating a cholesterol check within the past 5 years. The ”Smoker” variable indicates whether at least 100 cigarettes have been smoked in the subject’s lifetime, assigning a value of 1 if true and 0 otherwise. The variable concerning alcohol consumption determines whether the subject drinks, with a value of 1 if male consumed more than 14 drinks per week, or if female consumed more than 7 drinks per week. The ”general health” parameter asked subjects to self-describe their overall health status, with 1 indicating excellent, 2 very good, 3 good, 4 fair, and 5 poor. The ”mental health” variable records the number of days of poor mental health on a scale from 1 to 30 days. The ”physical health” variable records the number of days of physical illness or injury in the past 30 days on a scale from 1 to 30.
The first observation was that all variables were categorical, except for BMI, which was a numerical variable. One-hot encoding has already been applied to all the categorical variables.

The target variables were diabetes, stroke, and hypertension. In this case, we can observe, as in Fig. 1, that while diabetes and hypertension are balanced, stroke exhibits strong asymmetry.

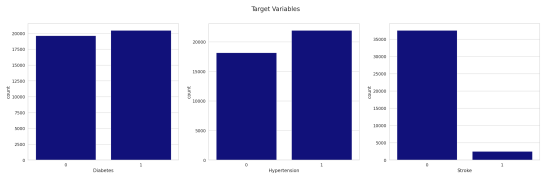

Figure 1. Total number of elements divided by category for each target variable.

3. Pre-Processing

When performing data analysis on a dataset, the first crucial phase to consider is pre-processing, which involves several steps. The primary objective was to encode data to enhance the performance of the model. The underlying idea is that, by cleaning the data, we can improve its quality and subsequently enhance the model's performance.
During the preprocessing stage, we encountered an issue related to BMI. We observed that the maximum value of this variable was 98, which does not align with the medical interpretation. Considering that the highest obesity class is typically identified by a BMI of over 40, we concluded that some values were randomly generated without appropriate validation. To address this problem, we decided to remove all observations with a BMI over 50.
After pruning the variables, we obtain the resulting distribution, as shown in Fig 2.
An important observation is related to the variable types in

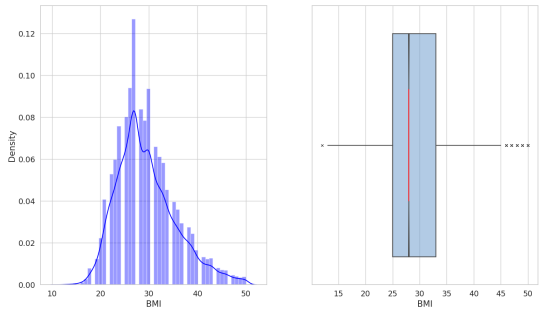

Figure 2. BMI Distribution on the left and BMI Boxplot on the right

the dataset. We noticed that all variables except BMI were categorical, resulting in a dataset with mixed data types.
Initially, our plan was to maintain the data in a mixed format and to implement strategies accordingly. For instance, one strategy could involve performing data analysis using Gower Distance instead of Euclidean distance. Gower Distance was specifically designed to handle mixed data.
However, in our case, we adopted a different approach. Since all variables, except for BMI, were categorical, we chose to transform the BMI variable into a categorical variable as well. This transformation is relatively straightforward because BMI is widely used globally and there are established guidelines for its categorization. Specifically, we referred to the CDC Guidelines [2], which divide BMI into the following categories:

Category	BMI
Underweight	< 18.5
Healthy range	18.5 to 25
Overweight range	25 to 30
Class 1 Obesity	30 to 35
Class 2 Obesity	35 to 40
Class 3 Obesity	> 40

TABLE 1. CATEGORIZATION FOR BMI

The second part of preprocessing involves studying the correlation matrix. For this analysis, we utilized the Pandas library [3], which provides useful tools for exploring the relationships between variables with the "corr" function already implemented in it. We computed the correlation matrix using the Pearson correlation coefficient, which assigns a score ranging from -1 to 1. A score close to -1 or 1 indicates a strong negative or positive correlation, respectively, whereas a score of 0 indicates no correlation between the variables. This approach allowed us to assess the relationships between the variables in our dataset.
The resulting matrix is shown in Figure 3.
From the plot, it is apparent that the correlation between

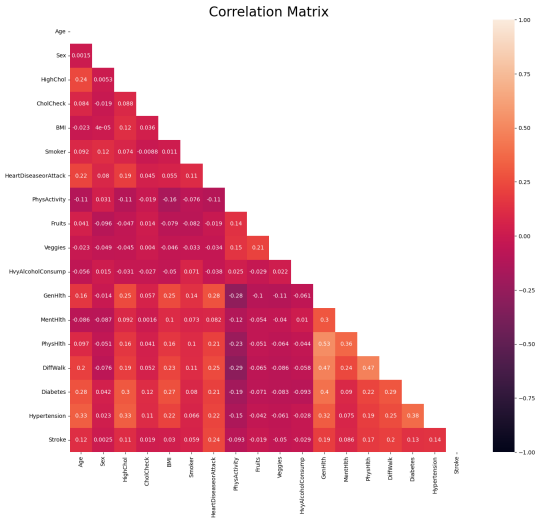

Figure 3. Correlation Matrix

variables is generally low. The highest correlations were observed between variables such as the General Health Score and the number of days of Physical and Mental Health. Another interesting correlation was found between the two

2. https://www.cdc.gov/obesity/basics/adult-defining.html
3. https://pandas.pydata.org/docs/index.html

target variables: Hypertension and Diabetes. This can be explained by considering that Hypertension is a known risk factor for Diabetes [4].

3.1. Dissimilarity Measure

As we will show after, one of the most important features that will be implemented for the clustering of this dataset is the dissimilarity measure, for this reason we believe that is worthy to describe it with a higher mathematical rigour.
We compute it in two different cases:
First suppose that our dataset contains k variables all categorical. Then the dissimilarity is $d(i, j)$ between the two objects i, j is defined as:

$$d(i,j) = \sum_{j=1}^{m} \delta_{i,j}^{(f)}$$

Where the indicator $\delta_{i,j}^{(f)}$ is 0 if the two objects for the variable f are the same and 1 otherwise.
Secondly, suppose to have mixed data type variables. Then the dissimilarity matrix is defined as:

$$d(i,j) = \frac{\sum_{f=1}^{p} \delta_{i,j}^{(f)} \delta_{i,j}^{(f)}}{\sum_{f=1}^{p} \delta_{i,j}^{(f)}}$$

The description of the dissimilarity Matrix is obtained from Kaufman and Rousseeuw, 1990 [2].

4. Outlier Detection

At this stage of the analysis, we address the issue of outlier detection. Given the categorical nature of the data, we opted to employ the K-nearest neighbor (KNN) method using suitable metrics. This method can be employed for detecting outliers in categorical data by evaluating the distance to its nearest neighbors. Unusual categories that deviate significantly from the majority can be identified as outliers. The first metric employed was the Hamming distance. We selected the 150th nearest neighbor as the observation point. We decided to inspect the distance plot in Fig. 4 and manually choose which distance would classify the points as outliers. In this case, we chose to exclude points that were farther than 0.38 in distance. Using this criterion, we identified 1123 outliers. To confirm the nature of the identified outliers, we repeated the procedure using the City Block metric. By examining Fig 5, we excluded all points with a distance greater than 12. As a result, we discovered 1241 outliers. Finally, we compared the outliers identified by both the methods to determine their consistency. To assess the performance, we utilized the Rand score, which considers all pairs of samples and counts the pairs that are assigned in the same or different clusters. The obtained Rand score of 0.921 serves as a strong indicator, indicating that the identified points are outliers. This high score reinforces the validity and reliability of our findings. After confirming that both

4. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC59153551/

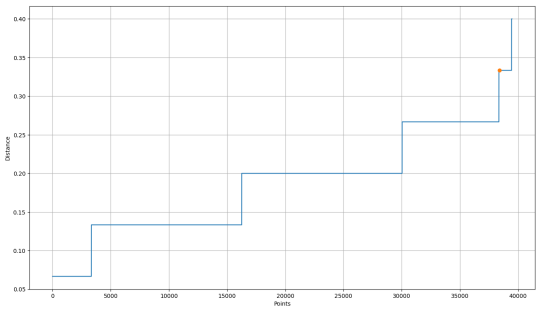

Figure 4. Sorted distances from the 150 nearest neighbors using the Hamming distance.

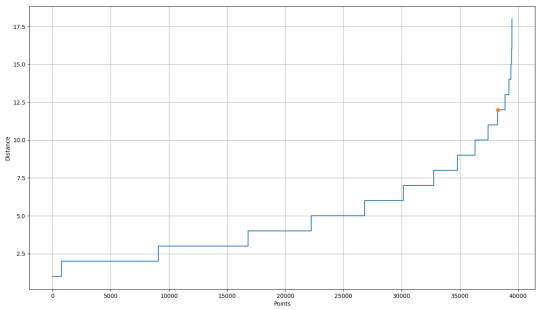

Figure 5. Sorted distances from the 150 nearest neighbors using the City Block distance.

methods were consistent, we selected the outliers identified using the City Block metric. In Fig. 6 you can observe the comparison of outliers found between data points 2000 and 3000, where a value of 1 represents non-outliers and a value of -1 represents outliers.

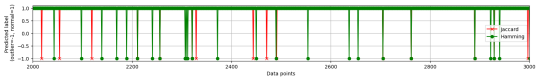

Figure 6. Match on outlier detection between NN with Hamming and with City Block distance

5. Clustering

The primary aim of this report is to implement various clustering techniques on the dataset and therefore assess their effectiveness in predicting three different pathologies, as mentioned previously.

As consequence, this section will exclusively be dedicated to the examination of three employed clustering algorithms, and in particular these algorithms are the following:

- K-Prototype, a clustering method similar to K-Means that has been studied to treat mixed-data type
- K-Modes, similar to K-means but works for only categorical data
- Hierarchical Clustering.

The choice of which algorithm to use is one of the first task that we need to deal each time an unsupervised clustering

task is approached and depends on a lot of different factors. Since the clustering methods can, approximately, be grouped in two main groups we decided to use one algorithm for each group.
First we have the *partitioning methods* which constructs k clusters and data must satisfy the requirements of partition where k is user-defined. The final aim of these algorithms is to uncover a structure that is already present in the data. In this category falls the K-Means algorithm (and the algorithms that derive from it).
The second category are the *Hierarchical methods*, algorithms that deal with all the different values for k in a single run. When implementing the hierarchical method we used an *Agglomerative* approach, starting from all the objects apart and trying to unifying them. They are usually used for classification tasks or similar.
For each clustering algorithm we will perform unsupervised evaluation and hyper-parameter tuning (referred to the optimal number of clusters) to study the best combination possible for each algorithm.
In particular we decided to use three different metrics for all the different clustering methods:

- Elbow Method: this method is applied to the cost function for the different clustering algorithms
- Calisnki-Harabasz Score: is a measure based on the internal dispersion of clusters and the dispersion between clusters. We are searching for the highest value possible.
- Silhouette Score: represents how well the clusters are separated. Is defined in the interval [-1, 1] and a value of 1 represents the highest separation possible for the data inside each cluster.

This methods are all inserted in the framework of internal validation which is an unsupervised method for evaluating a clustering algorithm. In fact, by using internal validation we do not need to access any external information.
For the final evaluation, however, we will use external validation since in our case we also have the Ground Truth. A summary of all the possible metrics is available in [3]

5.1. Hopkins Statistics

Clustering is a powerful algorithm with a wide range of applications in various scenarios. In the literature, there is a focus on assessing the *validity* of clustering, which involves developing metrics to compare the clustering results with ground truth labels to determine whether the model is able to accurately group the data. However, before entering the actual study, we strongly believe it is important to consider whether our data are truly clusterable; otherwise, the results obtained by our algorithm would be biased and not significant.
In a study by [1], it was noted that clustering algorithms impose a clustering structure on a dataset, even if the underlying data may not possess such a structure. This realization has led to the development of a research field known as "Clustering Tendency." It aims to investigate the clustering tendency of data without any prior knowledge of the clusters.
The examination of the clustering tendency in our data allows us to assess whether it is possible to identify clusters that naturally emerge or if the data are more randomly distributed. While several algorithms have been developed for this purpose, no single test has consistently outperformed the others.

Therefore, we have chosen to implement the Hopkins Statistic [5] due to its ease of use and interpretability.

Computation of the Hopkins statistic has been extensively discussed in various studies [5]. Here, we present the final equation for computing the Hopkins statistic, avoiding all the different steps needed to reach the result, because is not the objective of this report.
If we define X = x_i with i=1 to n as a collection of n patterns in a d-dimensional space, such that $x_i = x_{i1}, x_{i2}, ...$ In addition, let Y = y_j with j=1 to m be m sampling origins placed randomly in the d-dimensional sampling window. Distance u_j is defined as the minimum distance from y_j to its nearest pattern in X, and w_i is the minimum distance from a randomly selected pattern in X to its nearest neighbor.
The Hopkins Statistic is defined as follows:

$$H = \frac{\sum_{j=1}^{m} u_j^d}{\sum_{j=1}^{m} u_j^d + \sum_{j=1}^{m} w_j^d} \tag{1}$$

For data with a pattern that corresponds to well-aggregated data, the sampling origin to pattern nearest-neighbor distances should be larger than the randomly selected inter-pattern nearest-neighbor distances. Therefore, H should be greater than 0.5, which is almost equal to 1.0.
These considerations are obtained from [6].
We implemented the code for the Hopkins Statistic in Python using a pre-existent code [5].
The Hopkins Score was 0.77, meaning that we could reject the null hypothesis and that the data were clusterable.

5.2. K-Modes

K-Means is a popular and efficient clustering algorithm with a robust implementation in Python. This is also relatively easy to understand from a theoretical perspective. The K-Mode algorithm was first introduced by Huang in 1998 [7], along with other clustering algorithms that will be discussed later in this report. This algorithm has been developed exclusively to work with datasets that contain only categorical variables; for this reason we will need to perform the categorization of the BMI variable first.

While the traditional K-Means algorithm uses distances to compute clustering, in our case, we used dissimilarities. Specifically, we counted the total number of mismatches between the data points within the same category.
Because we are working with a dissimilarity matrix, we

5. https://sushildeore99.medium.com/really-what-is-hopkins-statistic-bad1265df4b

must modify the cost function to make it suitable for the elbow method. The updated cost function is as follows:

$$P(W,Q) = \sum_{l=1}^{k}\sum_{i=1}^{n}\sum_{j=1}^{m} w_{i,l}\delta(x_{i,j}, q_{l,j})$$

where $w \in W$ and $q_{l,j} \in Q$. W is a partition matrix and Q is the set of objects in the same object domain.
The K-Mode algorithm was run using the Huang initialization parameter.
We run the algorithm while keeping all the hyper-parameters stable and changing the number of clusters from two to eight.
Using the previously defined metrics, we can show the plot of the three metrics in Fig: 7.
According to both the elbow method using the Cost Function and the Silhouette Score, the evaluation metrics for the K-Mode algorithm suggest that the optimal number of clusters is two. However, the Calinski-Harabasz Score indicates that three clusters may be optimal, while two clusters perform the worst among the different possibilities. Despite this discrepancy, because two out of the three metrics agree on two clusters as the optimal choice, we will proceed with two clusters for the K-Mode algorithm. We will not repeat the evaluation metrics on this single measure, as the final section will focus on the supervised evaluation to compare the performance of the best clustering algorithm.

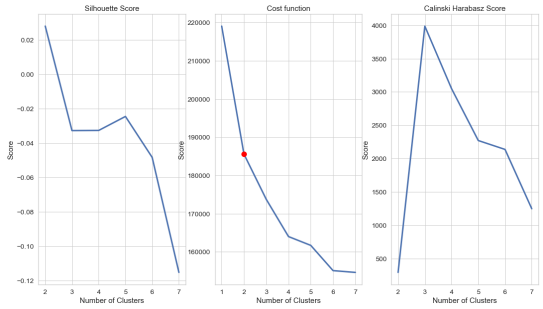

Figure 7. Metrics for K-Mode Evaluation

To visualize the clustering results, we utilize the *pairplot* function from the seaborn library [6]. We specifically focused on the three variables that exhibited the highest correlation. These variables were chosen based on their ability to provide meaningful information, considering that they cover a broader range of values rather than being binary in nature. The pairplot allows us to examine the relationships and distributions between these variables within each cluster. The results are shown in Fig: 8.

5.3. K-Prototypes

The next phase after implementing the K-Mode algorithm is the implementation of the K-Prototypes algorithm. This clustering algorithm is specifically designed to handle datasets with mixed-type attributes.

6. https://seaborn.pydata.org/generated/seaborn.pairplot.html

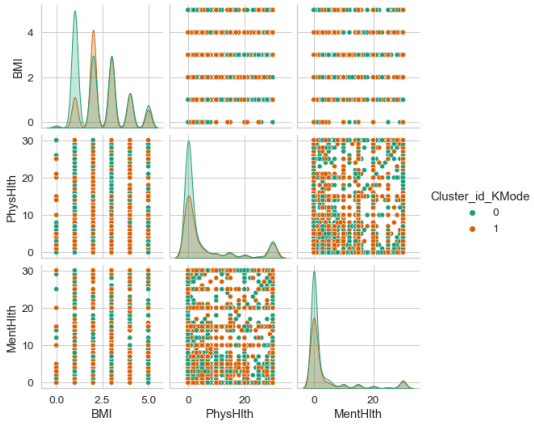

Figure 8. Clustering with K-Mode and two clusters

The initial implementation of the K-Prototypes is similar to that of the K-Modes algorithm developed by Huang [7]. However, the key difference is in the calculation of distances. K-Prototypes combine Euclidean distance for numerical data with a simple matching dissimilarity measure for categorical attributes. A correction factor $gamma$ was introduced to ensure a balanced consideration of both types of attributes.
For the implementation of the K-Prototypes, we utilized the same Python library, specifically the kmode library, as used for the K-Modes algorithm.
We briefly report the results obtained as a function of the number of clusters in Fig: 9.
In this case, the choice was obvious. All the metrics converge to the same result, agreeing on finding as two the optimal number of clusters.

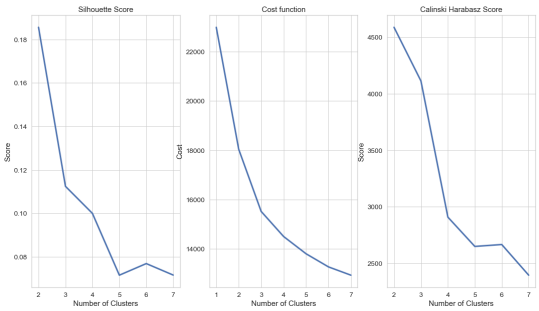

Figure 9. Metrics for K-Prototype Evaluation

5.4. Hierarchical Clustering

We also used the hierarchical clustering algorithm that works with both the dissimilarity matrix and raw data, making it applicable to both numerical and categorical variables by choosing the appropriate metrics.
Hierarchical clustering can be performed using either an agglomerative approach or a divisive approach. In this case,

we will use the agglomerative approach. Agglomerative clustering begins with individual data points and merges them. This algorithm begins by treating each data point as an individual cluster. It calculates the proximity between clusters or data points by using a distance or similarity measure. The closest clusters or data points were then merged to create a new cluster. This process continues iteratively with clusters or data points merged based on their proximity. The result was a hierarchical structure represented by a dendrogram. At each level of the dendrogram, clusters could be cut to form different numbers of clusters.
Since we had categorical data, we used three different metrics: Hamming distance, Jaccard similarity, and City Block distance. Regarding the merging method, we experimented with four different approaches: single, average, complete, and weighted linkages.
By performing all possible combinations of merging and metrics and cutting the dendrogram at different heights, we concluded that the optimal number of clusters is two.
In Table 2 and Table 3 we can observe the results of

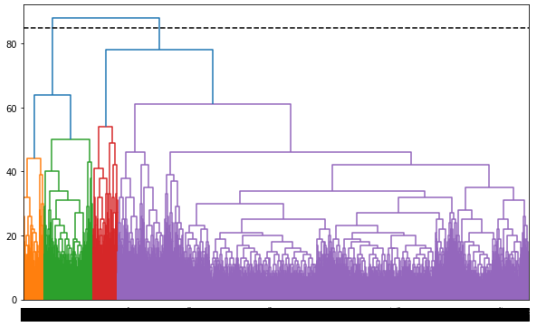

Figure 10. Dendrogram of Hierarchical clustering with method merging complete and City Block distance

different combinations with two clusters. The first observation is that the combination that maximizes both the silhouette and Calinski-Harabasz scores is the City Block metric with the complete merging method. The complete merging method is less sensitive to noise, which makes it a favorable choice. Furthermore, we noticed that the single method, which is more susceptible to noise, yields lower values compared to both metrics used. On the other hand, the average and weighted methods produced similar but worse results compared to those of the complete method.

	Single	Average	Complete	Weighted
Jaccard	0.5	0.91	338	147
City Block	3	17711	**41111**	20288
Hamming	2	40	1199	1119

TABLE 2. CALINSKI-HARABASZ OF HIERARCHICAL CLUSTERING WITH TWO CLUSTER

	Single	Average	Complete	Weighted
Jaccard	0.21	0.11	-0.11	-0.01
City Block	0.43	0.61	**0.62**	0.61
Hamming	-0.07	0.16	0.03	0.04

TABLE 3. SILHOUETTE-SCORES OF HIERARCHICAL CLUSTERING WITH TWO CLUSTER

6. Supervised Evaluation

Once for each clustering algorithms we obtained the optimal number of clusters we decided to perform Supervised Evaluation between the different algorithms to find which performs best on our dataset.
To perform this evaluation we used three different metrics and in addition, we studied the Confusion Matrix. Our attention in particular is referred to as:

- F1 Score: is an harmonic mean of precision and recall. We decided to implement this metric because is particularly adequate for dataset with high class imbalance, as in our case, where the positive labels (meaning the presence of the pathology) is less respect to the case of healthy individual
- Rand Score: the Rand Score is inserted in the framework of the Peer-to-Peer correlation, that is a set of metrics which measure the correspondence of same labels between the two clusters.
- Purity: Purity is the last measure used. Is defined as the percentage of the corrected sample in the cluster. Is based on a confusion matrix.

To perform the supervised evaluation we used the three target variables (Diabetes, Stroke and Hypertension) and we also added four new variables; in particular we focused on patients which have more than one pathology.
We created new variables for all the possible combinations between two pathologies and the case for all the three pathologies together.
In the following tables we report the results for the different algorithms using the previously defined metrics.
For notation purposes we will indicate the combos in an abbreviated form, in particular each combo is abbreviated with the first letter of each pathology (for example Diabetes and Hypertension is represented by DH)

	D	H	S	DH	SH	DS	HSD
Purity	0.60	0.58	0.94	0.63	0.95	0.95	0.96
F1	**0.56**	**0.56**	0.14	**0.51**	0.12	0.12	0.10
Rand-Score	0.52	0.51	0.52	0.53	0.52	0.52	0.52

TABLE 4. K-MODE RESULTS

We now consider these results.
First, it is worth noting that the metric that is more interesting to analyze is, as expected, the F1 Score.
For clarity, we highlighted the highest values for this metric

	D	H	S	DH	SH	DS	HSD
Purity	0.56	0.54	0.94	0.66	0.95	0.95	0.96
F1	**0.32**	**0.31**	0.22	**0.33**	0.2	0.19	0.18
Rand-Score	0.51	0.5	0.73	0.55	0.74	0.74	0.74

TABLE 5. HIERARCHICAL CLUSTERING RESULTS

	D	H	S	DH	SH	DS	HSD
Purity	0.62	0.54	0.94	0.63	0.95	0.95	0.96
F1	**0.6**	**0.54**	0.12	**0.52**	0.1	0.11	0.1
Rand-Score	0.53	0.5	0.51	0.53	0.51	0.51	0.51

TABLE 6. K-PROTOTYPE RESULTS

in all tables.
A part from the F1 Score the purity is not as interesting, this is due to the fact that it represents how well each value is guessed but suffers from class imbalance. We decided to keep this metric equal for our analysis because it can be seen as a measure, together with the confusion matrices that we will plot afterwards, of the class imbalance.
In addition, the same considerations can be made regarding the Rand Score. Except that the values are pretty similar across all the difference metrics, is noteworthy that the highest values are for the Stroke variable, which is also the variable with the highest number of negative sample (meaning that very few patients suffer from Stroke).

By analyzing the F1 Score, we can conduct some interesting statistical analyses.
The variables with the highest accuracy were Diabetes, Hypertension, and variables that represented patients with both Diabetes and Hypertension. Thus, we can confirm our initial hypothesis that there is a high correlation between the two diseases.
The highest F1 Score is for Diabetes, obtained considering with the K-Prototypes algorithm and is equal to 0.6; generally speaking a value over 0.5 is acceptable but is not generally defined as *optimal*; we assume that, due to the type of dataset we are working on rich of categorical data, is not possible to reach a much higher score.
As an additional consideration, we notice that the K-Mode and K-Prototypes have identical results. This is probably because having only one numerical variable does not significantly influence the final results. Furthermore, this confirms that our initial assumption of treating BMI as a categorical variable is valid.
The confusion matrices were calculated by counting the number of true positives, true negatives, false positives, and false negatives and normalizing them based on the predicted values. In particular, we chose to observe the confusion matrices in Fig 11 that describe the variables "Diabetes" and "Hypertension" because of their strong correlations. We can observe that the k-mode and K-prototype more or less predict the same clusters (in the figure, the values are approximately equal) and correctly predict 57% of non-DH (Diabetes and Hypertension) cases, while incorrectly predicting 38% of cases where they would be DH. In contrast, hierarchical clustering predicts non-DH cases very well, probably because it generally predicts almost all zeros.

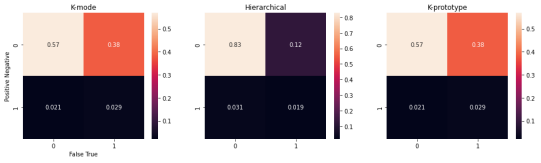

Figure 11. Confusion-Matrix

7. Classification

Finally, we added an additional task because we think it could be interesting to add an *explainable* component to our report. We refer to a new branch of Artificial Intelligence as explainable AI.
The objective was to better understand the variables that are important for our model. Until now, we have assumed that all the variables are used, but all the models are like a big black box and are very difficult to access.
We have structured this final section in two sections.
First, we built a decision tree classifier from the training set, which is the dataset without outliers, and the target values are the labels predicted by the clustering algorithm (we chose to use K-Modes).
The idea is that if the clustering is sufficiently good, the algorithm should be able to correctly classify our data along the labels found by the algorithm.
The second phase uses the Shapley Values. Once we had our classification algorithm fitted to our data, we defined an explainer and plotted which features were found to be the most important by computing the Shap value for each feature.
Because is not the objective of this report, we will not focus on the mathematical description of the Shapley Values; we cite the original paper in which they have been discussed [8]. The library used to implement the Shapley Values is called SHAP and a description of it is available at [9]. Is just worth notice that higher the value, bigger is it's impact on the model.

We used a Decision Tree Classifier with default values. The resulting Shap plot is shown in Fig: 12. Some interesting conclusions can be done:
First, only seven features among the fifteen were found to be useful for the model. This is very interesting because by performing this type of analysis beforehand, one can lighten the model by using a reduced number of features without losing information.

Moreover, the most important variables are expected to be the most relevant risk factors. Since our target variables are medical conditions, there is a vast body of literature that explains which are the most important risk factors for these diseases. In particular, taking again as a reference the CDC

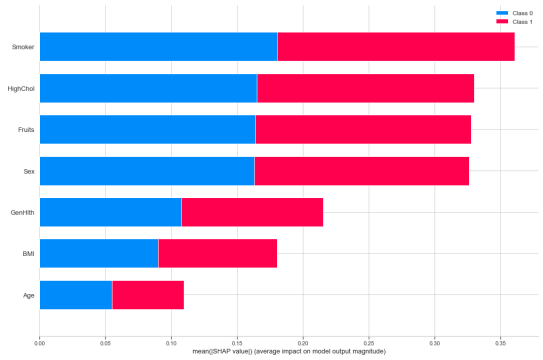

Figure 12. Plot of the Feature importance

[7], we can see that the risk factors for the Type 2 Diabetes are the same as the variables we found as important. The only exception is characterized by the variable fruit, but it has a very logical explanation; one of the risk factors for all the pathologies correlated to both diabetes and hypertension (so cardiovascular diseases) is a healthy diet, and probably the variable fruit can be seen as an indicator for this parameter.

8. Conclusion

In conclusion, the problem requires searching for possible correlations between certain variables and estimating three diseases: stroke, diabetes, and hypertension. After examining the dataset and finding mixed data, we converted all the variables into categorical variables. We then searched for and eliminated outliers using the K-nearest neighbor method with appropriate metrics.
At this point, we applied two different categories of clustering algorithms: Partitioning methods and hierarchical methods. For the first category, we applied the K-prototype and K-mode algorithms, whereas for the second category, we used hierarchical clustering with suitable metrics. We evaluated these algorithms using unsupervised metrics and, based on the results, selected the best combination of hyperparameters for each algorithm.
Finally, we evaluated the top three models using supervised metrics and arrived at two main conclusions:

- The best-predicted class was the "Diabetes-Hypertension" class. This was likely due to the strong correlation between these two variables.
- The best models for predicting this class were the K-mode and K-prototype models. This is likely because the hierarchical method tends to create globular clusters.

Finally, we trained a decision tree classifier on the training set. The target values were the labels predicted using the clustering algorithm. Using the Shapley Value, we determined the most important features of the model. We found that the most important features were Highchol, Fruit, Smoker, and Sex.
We also found another interesting result: just seven of all the variables are found useful for the model. The other variables are not used by the classifier to predict the different clusters. As a possible future work could be interesting to conduct the same study using only the variables found useful, in this way it would be possible to simplify the data collection phase for the entire process.

Disclosure Statement

The authors declare that this report is entirely original and does not contain any plagiarism. The research explained in this report was conducted by the authors themselves, and all the sources have been cited in the Reference Section. None of the content was generated using automated language models.

Take Home Message

How does Unsupervised Learning works?
Well, something like this [8]:

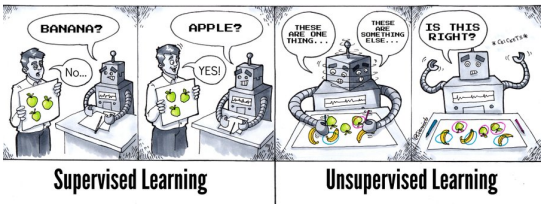

Figure 13. Unsupervised Learning

7. https://www.cdc.gov/diabetes/basics/risk-factors.html

8. Credits to: @Charaioch

References

[1] Sergios Theodoridis, Konstantinos Koutroumbas,
Chapter 16 - Cluster Validity,
Pattern Recognition (Fourth Edition),
Academic Press,
2009,
Pages 863-913,
https://doi.org/10.1016/B978-1-59749-272-0.50018-9.

[2] Leonard Kaufman, Peter J. Rousseeuw,
Finding Groups in Data,
Wiley Series in Probability and Statistics,
1990.

[3] Julio-Omar Palacio-Niño and Fernando Berzal,
Evaluation Metrics for Unsupervised Learning Algorithms,
ArXiv,
2019,
doi: 10.48550/arXiv.1905.05667.

[4] Shahla Faisal and Gerhard Tutz,
Nearest Neighbor Imputation for Categorical Data by Weighting of Attributes,
ArXiv,
2017,
doi: 10.48550/arXiv.1710.01011.

[5] B. Hopkins,
A new method of determining the type of distribution of plant individuals" Annuals of Botany (vol. 18),
1954
Pages 213-226.

[6] A. Banerjee and R. N. Dave,
"Validating clusters using the Hopkins statistic"
2004 IEEE International Conference on Fuzzy Systems (IEEE Cat. No.04CH37542),
Budapest, Hungary,
2004,
Pages 149-153
doi: 10.1109/FUZZY.2004.1375706.

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

[8] Roth, Alvin Eliot.
The Shapley Value: essays in honor of Lloyd S. Shapley.
Cambridge University Press,
1988.

[9] Scott M. Lundberg and Su-In Lee.
A Unified Approach to Interpreting Model Predictions.
https://arxiv.org/abs/1705.07874