Hypothyroidism Data Set: Anomaly Detection

Benedetta Bensi
867143
MSc Artificial Intelligence for Science and Technology
b.bensi@campus.unimib.it

Stefano Carotti
911255
MSc Artificial Intelligence for Science and Technology
s.carotti1@campus.unimib.it

***Abstract*— Anomaly detection is a critical task in various domains. In unsupervised settings, where labeled anomaly data is nonexistent, the challenge intensifies. This report presents a comparative analysis of different algorithms for anomaly detection within unsupervised framework.**

I. Introduction

In anomaly detection the goal is to find objects that are different from most other objects. Although unusual objects or events are, by definition, rare, this does not mean that they do not occur frequently in absolute terms. For example, an event that is "one in a thousand" can occur millions of times when billions of events are considered. Anomalies are of considerable interest in various field, as in fraud and intrusion detection, system disturbances and public health and medicine, in which it can occur that for a particular patient unusual symptoms or test results indicate potential health problems. Historically, anomaly detection has been viewed as a technique for improving analysis of data objects. Therefore, anomaly detection and removal is often a part data preprocessing.
The data set studied in this report deals with hypothyroidism, a medical condition characterized by an underactive thyroid gland, which fails to produce sufficient thyroid hormones. Thyroid hormones play a crucial role in regulating various body functions. Untreated hypothyroidism can lead to complications. Therefore, early detection and management are essential to prevent long-term health consequences and improve quality of life for individuals affected by this condition.
Since no ground truth was available, different anomaly detection methods were employed in order to have a better understanding of the setting. The objective was not only detecting the anomalies, but also estimating the percentage of such data points.

II. Preprocessing

A. Data set description

The data set consisted of twenty-one variables and 7200 observations; the first thing that was noticed is that fifteen of the variables were binary while six of them were numerical.

One-hot encoding has already been applied to all the categorical variables. Furthermore all the numerical ones were normalized with min-max scaling between zero and one.

	Dim_15=0	Dim_16	Dim_17
count	7200.000000	7200.000000	7200.000000
mean	0.951111	0.009172	0.108506
std	0.215651	0.043357	0.042001
min	0.000000	0.000000	0.000000
25%	1.000000	0.001340	0.091922
50%	1.000000	0.003208	0.109192
75%	1.000000	0.005094	0.119777
max	1.000000	1.000000	1.000000

Listing 1: Statistical description of a categorical (dim_15=0) and two numerical variables (Dim_16 and Dim_17).

Before visualizing the data, correlation was studied. Correlation analysis helps in understanding the relationships between different variables in the data set. When anomalies occur, they can disrupt these relationships.
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.

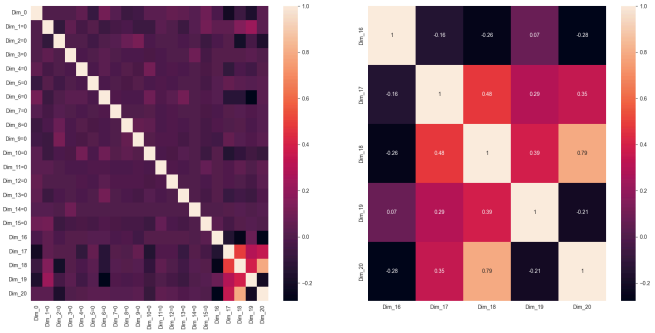

Figure 1: Correlation on the whole data set (left), zoom on the last five variables (right).

It was also interesting to visualize the six numerical variables employing box-plot:

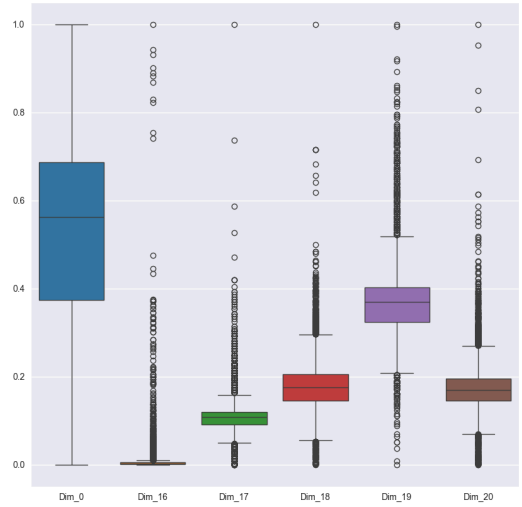

Figure 2: Numerical variables box-plot.

As it can be noticed from this figure, a lot of outliers have already been recognized, but only considering a small fraction of the total variables.

B. *Distance Measures*

When all the variables are continuous, the most commonly used distance measure is the Euclidean distance. If the data is a mix of both continuous and categorical type, one way to proceed could be to ignore the categorical variables (as in the box-plot implementation) or to transform the numerical variables into categorical ones. However these approaches involve loss of information, therefore it was decided to implement a distance measure that could manage both variable types.
Gower (1971) defined a general coefficient which measures the similarity between two units [1]. The Gower's distance can be defined as the complement to one of the Gower's similarity coefficient:

$$d_{G,i,j} = 1 - s_{G,i,j} = \frac{\sum_{t=1}^{p}(\delta_{ijt}d_{ijt})}{\sum_{t=1}^{p}(\delta_{ijt})}$$

It is a dissimilarity or distance measure between object i and j where $d_{ijt} = 1 - s_{ijt}$ is the distance calculated on the t-th variable. s_{ijt} is the similarity between the two objects with respect to the t-th variable and depends on the type of variable itself: for numerical data $1 - s$ is the Manhattan distance scaled by the range $R_t = \max(x_t) - \min(x_t)$, while for categorical data distance is zero if $x_{jt} = x_{it}$ and one otherwise [2]. A distance matrix was computed with Gower's metric which was employed in the algorithms that will follow in this report.

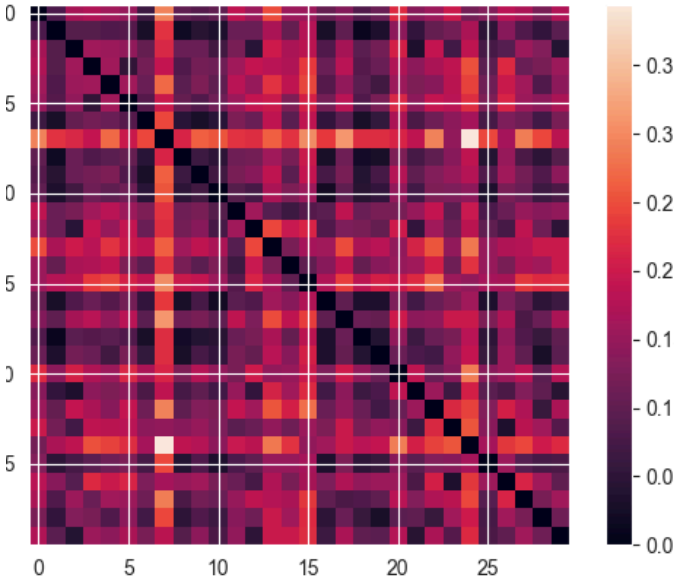

Figure 3: Distance matrix using Gower, on a sampled reduced data set to have a better visualization.

C. *Data Visualization*

In order to better envision the data set and understand also in a graphical sense the future results of the employed algorithms, dimensionality reduction techniques have been considered. In particular, t-Distributed Stochastic Neighbor Embedding (T-SNE) and Multi-Dimensional Scaling (MDS) are suited to this framework.
T-SNE is designed to preserve local structure, meaning that nearby data points in the original high-dimensional space are expected to remain close to each other in the lower-dimensional embedding. However, it does not preserve global structure as strictly, therefore, interpreting the exact meaning of distances in its embedding can be challenging due to the emphasis on local relationships.

On the contrary, MDS aims to preserve the overall structure of the data, including both local and global relationships. It focuses on maintaining the pairwise distances and dissimilarities between data points. As a consequence, we considered MDS a good option in the anomaly detection task, since the results could be better interpreted.

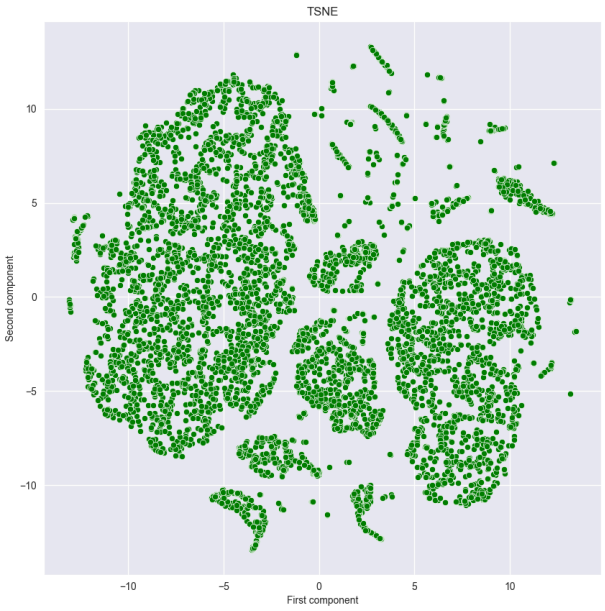

Figure 4: Data set visualization using T-SNE.

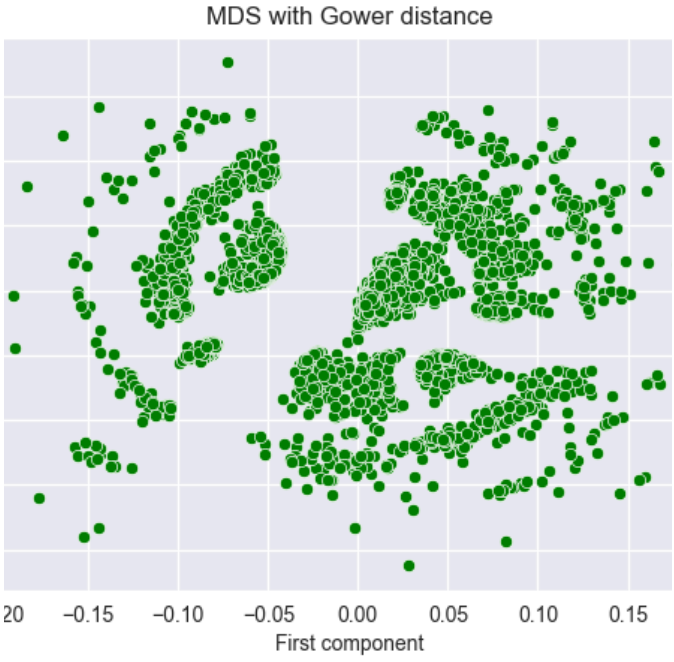

Figure 5: 2-D data set visualization using MDS.

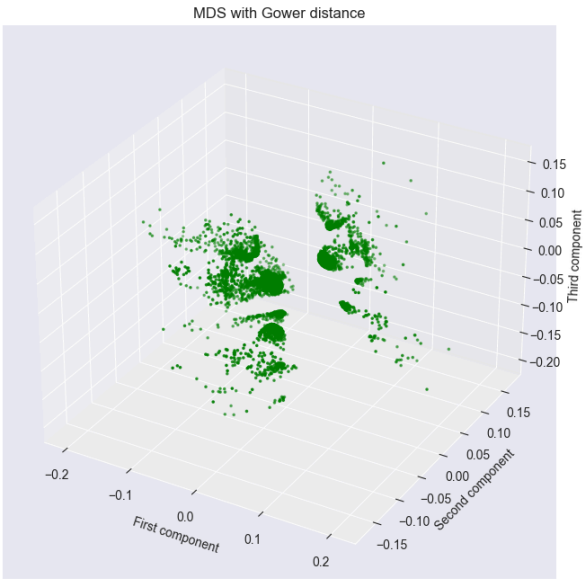

Figure 6: 3-D data set visualization using MDS.

III. Anomaly detection

Anomaly detection techniques focus on finding objects that differ substantially from most other objects, and the techniques themselves should not be affected by these data points. In order to validate the results, various algorithms were employed. The algorithms in this report had to perform properly with mixed data. They can be distinguished between three macro-categories: ensemble, proximity-based and reconstruction-based methods.

A. Ensemble method - Isolation Forest

Isolation Forest builds an ensemble of isolation trees (iTrees) for a given data set, then anomalies are those instances which have the shortest average path lengths on the iTrees. Isolation trees separate an instance from the rest of the population. Since anomalies are few and different with respect to the normal class, they are more susceptible to isolation. Each tree is trained on a subset of the training data, therefore introducing diversity in the predictions, which are then aggregated in the final prediction through averaging the path length to compute an anomaly score. This method performs well with high dimensional data, showing more promising results compared to proximity based models [3]. Considering how promising this algorithm is in anomaly detection and in handling categorical data, we decided to utilize it as a guideline in order to assess an estimation of the total number of outliers in this data set. For this reason, we set the *contamination* parameter (percentage of outliers) equal

to *'auto'* value. In this way the offset used in the algorithm is not chosen as the percentage of outliers desired, but it is a known value that permits to discriminate *'definitely anomalies'* and *'quite safe to be regarded as normal instances'*[4]. With *'auto'* setting, 3.8% outliers were obtained.

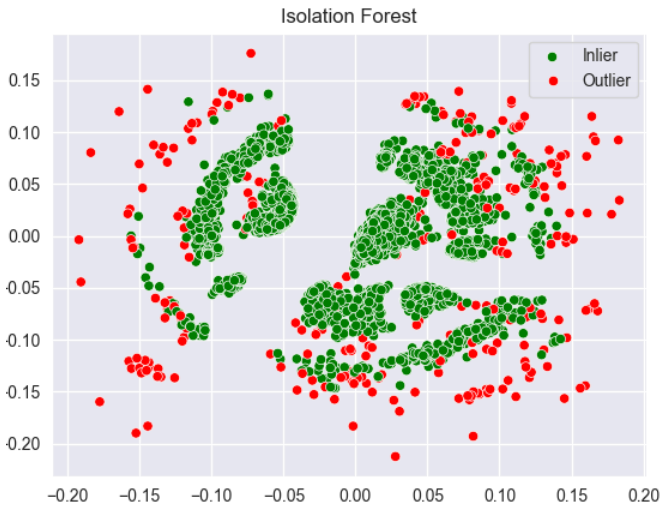

Figure 7: Isolation Forest classification results visualized on 2-D MDS.

For what has been said, a percentage not too far from $\sim 4\%$ is what we would expect from the following methods.

B. Proximity-based method: Local Outlier Factor

The goal is to validate the results by comparing different algorithms. A popular unsupervised outlier method is Local Outlier Factor (LOF), which flags as outliers all the points whose distance from a set value of nearest neighbors exceeds a certain threshold.
A score for each object in the data set is computed, indicating its degree of outlier-ness. In this way it is quantified how outlying an object is. The outlier factor is local in the sense that only a restricted neighborhood of each object is taken into account [5]. The score is based on the local reachability density, the inverse of the average reachability distance, which is the maximum between the distance from each point to its k-th nearest neighbor and the distance to the query point.
LOF from sklearn was implemented with *'algorithm'* equal to *'auto'*. This option will attempt to decide the most appropriate algorithm based on the values passed to fit method [6]. The *contamination* parameter was initially set to *'auto'* which, similarly to the previous case, sets an offset critical value. With these settings the results were

```
Number of outliers:  820
Number of inliers:  6380
Proportion of outliers:  0.11
```

which are substantially higher than the ones obtained with Isolation Forest algorithm. What we noticed by plotting the sorted distances is a second knee that separates more extreme samples, corresponding to 3.8% of outliers, similarly to Isolation Forest result.

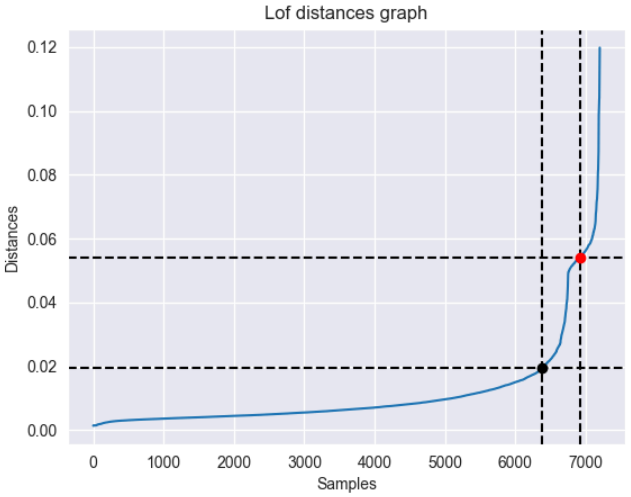

Figure 8: LOF distances: 11% and 3.8% of outliers marked in black and red respectively.

We used the latter as a threshold by setting *contamination* equal to 0.038, leading to the following classification:

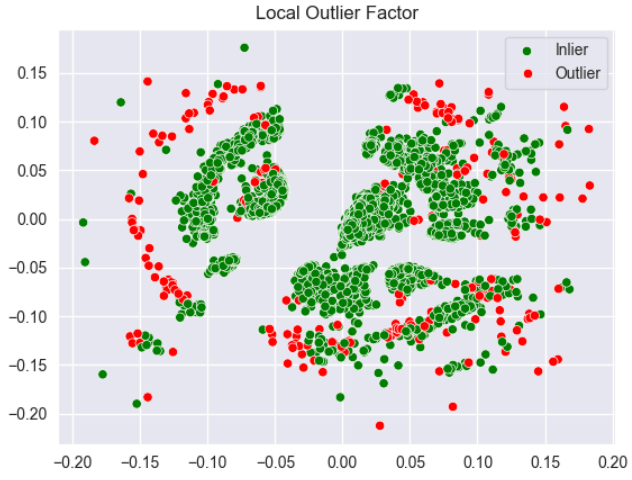

Figure 9: LOF classification results visualized on 2-D MDS.

C. Density-based method: DBSCAN

Density-Based Spatial Clustering of Applications with Noise (DBSCAN) is a popular unsupervised algorithm used for clustering data. Its strength relies in finding clusters with arbitrary shape, without setting the number of clusters and shows good efficiency on large databases [7].
The idea behind this method is to divide all the points in

three categories: core, border and noise, where the latter stands for anomaly points.

- Core point: if it has at least *min_samples* points within a radius *eps*, including itself.
- Border point: if it has fewer than *min_samples* within *eps*, but is within *eps* of a core point.
- Noise point: if it isn't a core point nor a border point.

In particular *eps* and *min_samples* are user-specified parameters. Although DBSCAN is a clustering technique, we chose to implement it since as its first step it discriminates between normal points and noise. In order to set both *eps* and *min_samples*, we decided to utilize the knee method on the sorted k-distance graph, where k was set as *min_samples* -1 and validating the results by maximizing the *silhouette* score.

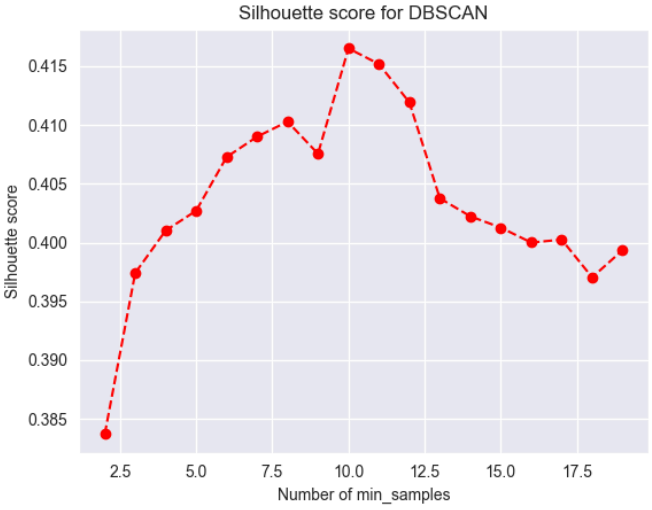

Figure 10: Silhouette score varying min_samples

After setting $eps = 0.0138$ and $min\_samples = 10$, which maximize the *silhouette*, DBSCAN algorithm was applied, with the following outcomes:

```
Number of outliers:  607
Number of inliers:  6593
Proportion of outliers:  0.084
```

In the following figure the obtained results are displayed.Indeed, the number of outliers is substantially higher, however all the farther points are labeled as outliers, unlike in the LOF visualization.

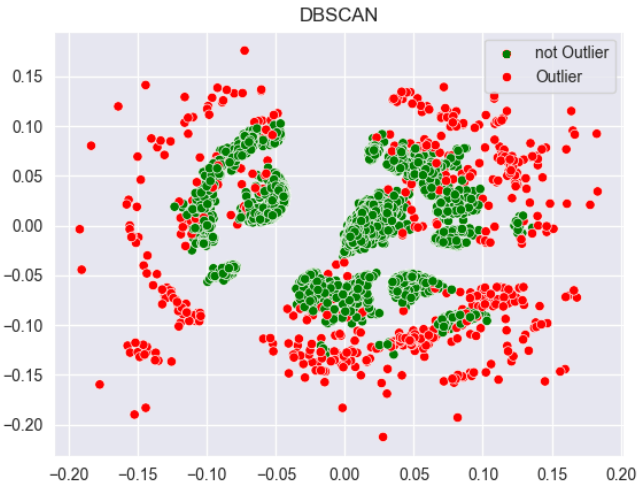

Figure 11: DBSCAN classification results visualized on 2-D MDS.

Notice that, even though we followed the described above silhouette criterion, the NN sorted distances once again presented a second knee, that corresponded to an *eps* that, if fed in the DBSCAN algorithm, resulted to approximately 1.5% of outliers.

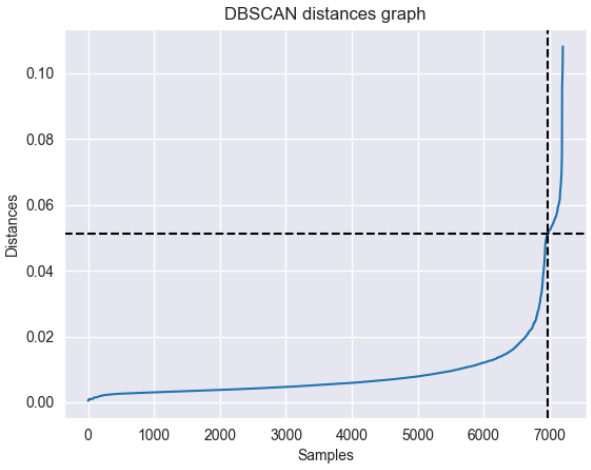

Figure 12: NN sorted distances.

For the final comparison, *eps* was set to 0.0282 to ensure that 3.8% of the data points were outliers, allowing the results to be evaluated with respect to those of the other algorithms.

D. Reconstruction-based method: PCA

Principal Component Analysis (PCA) is a powerful technique for dimensionality reduction of data while preserving most of its variance. It finds its application in anomaly detection by finding the representation of the data object in a lower-dimensional space and then projecting it back to the original, higher-dimensional, space. During this process

some information is lost, the objects with large reconstruction error are flagged as outliers. For the purpose of adapting this technique to our data set, we fitted PCA on the Gower's distance matrix. This is essentially Principal Coordinate analysis or MDS: the idea is to have a lower-dimensional embedding of the pairwise distances, so that the large reconstruction error objects are samples whose pairwise distance from all the other points is anomalous with respect to the majority of the population [8]. As a consequence, the more PCs are used the less objects are wrongly reconstructed. We think that applying a more specific algorithm to distance matrices, like MDS, could lead to better results. Nevertheless, we implemented the *sklearn* PCA method since we were already familiar with it.

We arbitrary chose the number of principal components equal to eight, leading to an explained variance $= 0.98$, just to avoid an exact reconstruction. Then chose to apply the chi-squared test to the reconstruction errors. This way we had a statically-founded method to assess whether some points have unusually high reconstruction error, flagging those points as anomalies. We chose a significance level $\alpha = 0.05$ and considered an observation anomalous if the sum of squared values of its projection normalized on the first eight principal components is larger than $\chi^2_8(0.05)$.

```
Number of outliers:  584
Number of inliers:  6616
Proportion of outliers:  0.081
```

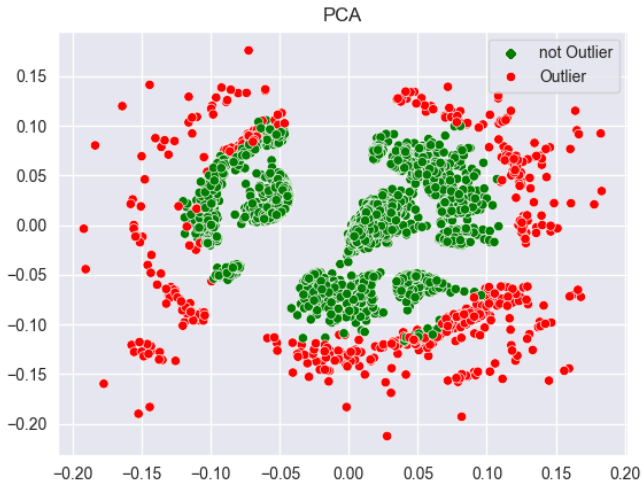

Figure 13: PCA classification results visualized on 2-D MDS.

These results rely on the parameters values α and the number of PCs, that we arbitrary chose. More specifically, as we raised the number of PCs or lowered α, the percentage of outliers diminished. This is because the algorithm becomes more accurate in the reconstruction in the first case, while in the second one only more extreme observations are flagged as anomalies. Graphically it can be noticed that all the outer points, the ones that lie outside the main cluster data, are evaluated as outliers, compared to the other algorithms. However, the perfect agreement with the visualization is also due to the fact that we are implementing a sort of MDS for both the classification and MDS itself for the visualization.

At last we set the number of principal components that corresponded to 3.8% of outliers, in order to compare which points are chosen with respect to the other algorithms.

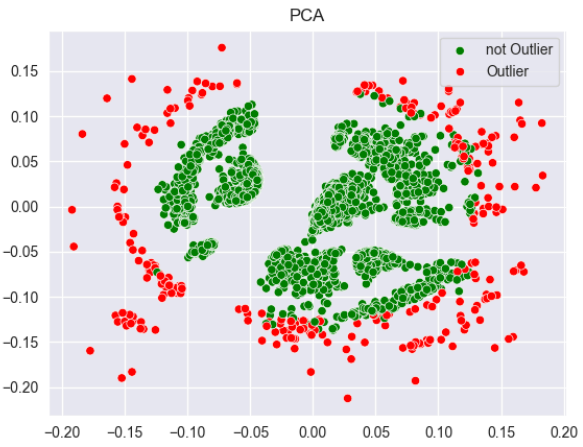

Figure 14: PCA classification 3.8% of outliers visualized on 2-D MDS.

IV. Comparison

The aim of this paragraph is to distinguish, on the equal condition of 3.8% of contamination, which observations are inliers or, if flagged as outliers, by how many algorithms.

```
139 points are flagged as inliers by 2 algorithms and
as outliers by 2
120 points are flagged as inliers by 1 algorithm and
as outliers by 3
198 points are flagged as inliers by 3 algorithms and
as outliers by 1
67 points are flagged as outliers by 4 algorithms
6676 points are flagged as inliers by 4 algorithms
```

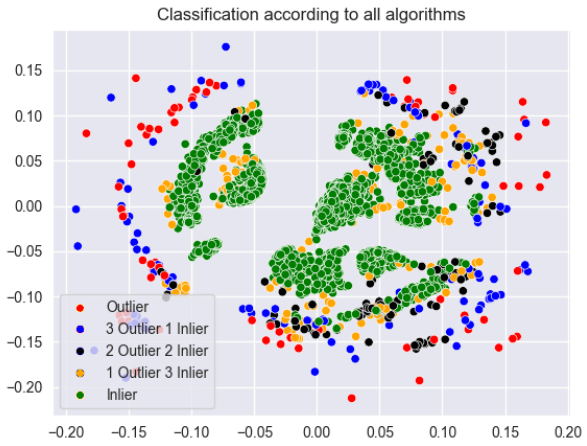

Figure 15: Combined classification results visualized on 2-D MDS.

From this comparison a good agreement is achieved: the vast majority of the points are considered inliers by all the algorithms, there are only 139 observations for which we cannot make any assumption, but keep in mind that all the methods utilized are of different nature.
Finally we assess the consistency by computing the *Rand score* that consists in taking all pairs of samples and counting the pairs that are assigned in the same or different classification.

```
Rand score between ISOF and LOF  0.898
Rand score between DBSCAN and LOF  0.954
Rand score between PCA and LOF  0.919
Rand score between ISOF and DBSCAN  0.932
Rand score between ISOF and PCA  0.944
Rand score between PCA and DBSCAN  0.95
```

As expected from the previous results, these scores are a strong indicator of consistence between the algorithms performances.

V. Conclusions

In a complete unsupervised framework, without any context about the variables meaning, we found it reasonable to compare multiple models' predictions. Since this data set belongs to an healthcare domain, we are interested in finding very precise results. The approach we took aimed to validate both the number of total outliers and more accurate predictions of anomalies. For further researches, we think it would be useful to have more knowledge about the context of the data set. If we had such information, we could exclude less meaningful variables in order to simplify the process. The cooperation with experts in the field could be helpful to interpret the data set, integrating other experimental evidence and domain information without any influence of any anomaly detection algorithms.

Strong evidence of redundant variables is the capability of PCA to have a remarkably low reconstruction error even with a very low number of principal components. Nevertheless, we find our results satisfying both with coherence and graphical visualization. Considering how Isolation Forest adapts well on this kind of data set we thought it was the most appropriate among the four models utilized, therefore made it our guideline during this whole project. Finally on the provided CSV file the last column contains the probability of the data object being anomalous according to Isolation Forest.

VI. Disclosure statement

Benedetta Bensi and Stefano Carotti assure that this project is entirely original without any plagiarism. The research detailed in this report was independently conducted by the authors, with all sources cited in the References Section. No content was AI generated.

References

[1] S. Bishnoi and B. Hooda, "A survey of distance measures for mixed variables," *International Journal of Chemical Studies*, 2020.

[2] M. D'Orazio, "Distances with mixed type variables some modified Gower's coefficients." 2021.

[3] K. M. T. Fei Tony Liu and Z.-H. Zhou, "Isolation Forest," *2008 Eighth IEEE International Conference on Data Mining*, 2008.

[4] K. M. T. Fei Tony Liu and Z.-H. Zhou, "Isolation-based anomaly detection," *ACM Transactions on Knowledge Discovery from Data*, 2012.

[5] R. T. N. J. S. Markus M. Breunig Hans-Peter Kriegel, "Identifying Density-Based Local Outliers," *Int. Conf. On Management of Data, Dalles, TX, 2000*, 2000.

[6] P. et al., "Scikit-learn: Machine Learning in Python," *JMLR 12*, 2011.

[7] H. P. K. J. S. Ester M. and X. Xu, "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise," *Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining, Portland, OR, AAAI Press*, 1996.

[8] A. Agovic, A. Banerjee, A. Ganguly, and V. Protopopescu, "Anomaly detection using manifold embedding and its applications in transportation corridors," *Intell. Data Anal.*, vol. 13, pp. 435–455, 2009, doi: 10.3233/IDA-2009-0375.