*Journal of Information and Data Management*doi: 10.5753/jidm.2025.4663

Explainable Clustering: A solution to interpret and describe clusters

Guilherme Sérgio de Oliveira [**Universidade Federal de Viçosa** | ***guilherme.sergio@ufv.br***]
Fabrício A. Silva [**Universidade Federal de Viçosa, Manna Team** | ***fabricio.asilva@ufv.br***]
Ricardo V. Ferreira [**Cinnecta - part of Matera** | ***ricardo.ferreira@cinnecta.com***]

Laboratório de Inteligência em Sistemas Pervasivos e Distribuídos (NESPeD-Lab)
IEF, Universidade Federal de Viçosa - Campus Florestal, Rodovia LMG 818, km 6, Florestal - MG, 35690-000, Brazil

Received: 3 July 2024 • **Published:** 20 June 2025

Abstract Unsupervised learning algorithms represent a set of techniques for finding hidden patterns or characteristics in data without a previously defined label. An unsupervised learning technique is clustering, which consists of grouping data with similar characteristics into the same group, while data with different characteristics belong to other groups. Despite being a technique with many applications, understanding the output of clustering models is a complex task, requiring extensive manual analysis to understand the characteristics of each group, since the output doesn't contain much information about the cluster's characteristics. Therefore, this article proposes MAACLI: ***M****odel and* ***A****lgorithm* ***A****gnostic* ***CL****ustering* ***I****nterpretability*, a technique for generating user-friendly descriptions to help interpret groups generated by unsupervised clustering algorithms. The solution consists of two components that generate friendly descriptions of the groups and was tested on two types of datasets, one of which was provided by a partner company. The solution was able to generate simple, user-friendly descriptions of the groups, extracting only the important attributes.

Keywords: clustering explainability, unsupervised learning, Explainable Artificial Intelligence, XAI

1 Introduction

Unsupervised learning aims to extract information from data without a previously established label. One of the unsupervised strategies is clustering, which consists of grouping data with similar characteristics so that similar data belongs to the same group [Ghahramani, 2003]. The similarity of the data can be measured based on distance or density, for example. The clustering technique has many applications, such as in the health sector [Yu *et al*., 2022] and marketing [Bartels, 2022]. In general, in recent years, companies in different segments have invested in grouping their customers in order to get to know them better and thus offer personalized products and services [Li *et al*., 2021].

There are various metrics for assessing the quality of the groups created, such as silhouette score and the Calinski-Harabasz Index [Liu *et al*., 2010]. However, they do not provide information about the reasons why an item belongs to a particular group, nor about the main characteristics of each group created. Therefore, an important task after creating the groups is to interpret them [Xu and Wunsch, 2005], with the aim of understanding the characteristics that stand out most in each group. Some works in the literature try to make clustering techniques explainable, by intrinsically explainable methods [Fraiman *et al*., 2013; Dasgupta *et al*., 2020; Bertsimas *et al*., 2020] and also with concepts from *Explainable Artificial Intelligence* (XAI) [Ellis *et al*., 2021; Morichetta *et al*., 2019]. However, they are dependent on the clustering algorithm or model. In addition, the works do not provide a user-friendly description of the interpretation of the groups, which can be understood by people without technical knowledge, such as business and marketing people, who are often the main stakeholders interested in the clusters.

In this work, we create and evaluate a solution called MAACLI: ***M****odel and* ***A****lgorithm* ***A****gnostic* ***CL****ustering* ***I****nterpretability* , which is model and algorithm agnostic, generates a user-friendly output of the interpretation of clusters, and is understood by lay users in the field of technology. MAACLI is made up of two independent components. The first consists of mining the rules of a decision tree, trained with the groups created by any non-supervised algorithm. The second component consists of generating a description of the groups, using the importance of the attributes for each group, found using *XAI* techniques.

The proposed solution was evaluated on different datasets: initially, it was validated on simpler data, with few dimensions and well-separated groups, and then tested on a complex, real dataset provided by a private partner company, containing the segmentation of 263,684 customers into 14 distinct groups. The results showed that MAACLI can provide relevant descriptions even for complex data. Compared with a solution from the literature, it was possible to demonstrate that the outputs are more explanatory and that the proposal to be agnostic in terms of algorithm and model is advantageous.

This work is an extension of the paper published at the Symposium on Knowledge Discovery, Mining and Learning (KDMiLe 2023) [Oliveira *et al*., 2023]. In this new version, we have added a discussion on new relevant works in section 2, enhanced the details of the methodology in section 3, and included new experiments and results in section 4. We have also included the limitations of our proposal in section 4.3.

The paper is organized as follows: Section 2 presents related work. The proposed solution is discussed in Section 3. The results are presented and discussed in Section 4. Finally, the concluding remarks are presented in Section 5.

2 Related Work

Despite being a topic that is still little explored, work related to the explainability and interpretability of unsupervised machine learning has gained ground in recent years. In particular, it is possible to find works using Binary Trees to make the clustering technique explainable, such as in [Fraiman *et al*., 2013; Loyola-González *et al*., 2020; Dasgupta *et al*., 2020].

In [Fraiman *et al*., 2013], for example, the authors propose a 3-step clustering approach based on Classification and Regression Trees, where the original supervised strategy is adapted to unsupervised problems. In [Loyola-González *et al*., 2020], the authors use an ensemble method to generate groups through a set of unsupervised trees. The trees are created with a proposed evaluation function for splitting the nodes of the tree, which takes into account the separation between groups and the compactness of items within the same group. The work of [Dasgupta *et al*., 2020] follows the same path, proposing the creation of a tree using centroids previously found by center-based clustering methods, such as *k-means* and *k-medians*, to calculate the objective function for creating the tree.

The last mentioned work is later improved, where [Frost *et al*., 2020] propose a new method called Explainable k-means clustering algorithm, ExKMC, that allows the creation of trees with more leaves than the number of groups generated by the basic algorithms. In [Laber *et al*., 2023], the authors use metrics related to tree complexity to generate smaller and more explainable trees than ExKMC [Frost *et al*., 2020].

This proposal manages to turn *k-means* into an explainable algorithm with the addition of a small cost. The approach works for groups that have already been created, but it is dependent on the model, which must be based on centers. As a result, the proposed method generates a decision tree capable of grouping data while maintaining the natural explainability of decision trees. The main disadvantage of these solutions is that they require the clustering to be done by the proposed algorithm or model. MAACLI, on the other hand, accepts previously created groups, regardless of the algorithm and model used.

Although trees are popular for this type of problem, there are also other approaches. The work of [Ellis *et al*., 2021] seeks to adapt the *Permutation Feature Importance* [Breiman, 2001] technique to deal with unsupervised learning.

By regrouping the data with shuffled variables, the method manages to find the most important features for clustering with an algorithm-independent approach by checking for changes in clustering after breaking feature associations with the initial clustering.

On the other hand, [Morichetta *et al*., 2019] propose a methodology for performing interpretable clustering, which manages to find the most important characteristics for clustering. The methodology uses the *Support Vector Machine* (SVM) and the *Local Interpretable Model-Agnostic Explanations* (LIME) technique [Ribeiro *et al*., 2016] to understand the groups. Finally, [Corral *et al*., 2009] present a description of each group with common values for most of the samples.

However, it generates a description of all the attributes, making it unfeasible in situations with many dimensions, as well as being unable to deal with continuous numerical attributes. The work by [Moura *et al*., 2022] proposes a framework for describing groups using a method of discretizing numerical values by observing the group's unique identifier to find better bins with cutoffs at infrequent values. The work generates a unique description for each group, but does not provide information on how to deal with categorical attributes.

Table 1 shows a comparison of the related works based on the following criteria:

1. The work is agnostic to the clustering algorithm.
2. The work can handle numerical attributes.
3. The work can handle categorical attributes.
4. The work generates a descriptive output that is easy to understand and interpret.

Table 1. Comparison of Works on Cluster Explainability

Work	1	2	3	4
[Fraiman *et al*., 2013]		✓		✓
[Loyola-González *et al*., 2020]		✓		✓
[Dasgupta *et al*., 2020; Frost *et al*., 2020]		✓		✓
[Ellis *et al*., 2021]	✓	✓	✓	
[Morichetta *et al*., 2019]	✓	✓	✓	
[Corral *et al*., 2009]	✓		✓	
[Moura *et al*., 2022]	✓	✓		✓
MAACLI (This Work)	✓	✓	✓	✓

Although all of these studies have made progress in explaining groups, making it possible to understand why a particular element has been allocated to a particular group, they do not focus on a user-friendly description of groups that helps end-users interpret them. And the works that offer this user-friendly description intrinsically through trees, are model-dependent or present new clustering methods. The proposal MAACLI aims to advance the state of the art in the explainability of clusters by solving these problems.

In this sense, this work aims to provide a user-friendly and model-independent description of pre-existing groups automatically, regardless of the model and algorithm used. Last, but not least, it is worth highlighting that this work seeks to meet a market demand, which often struggles to explain the created groups to clients in areas such as *marketing*, sales, customer service, and others.

3 MAACLI: *Model and Algorithm Agnostic CLustering Interpretability*

The MAACLI solution consists of two main independent components: a first component that creates, extracts, and pro-

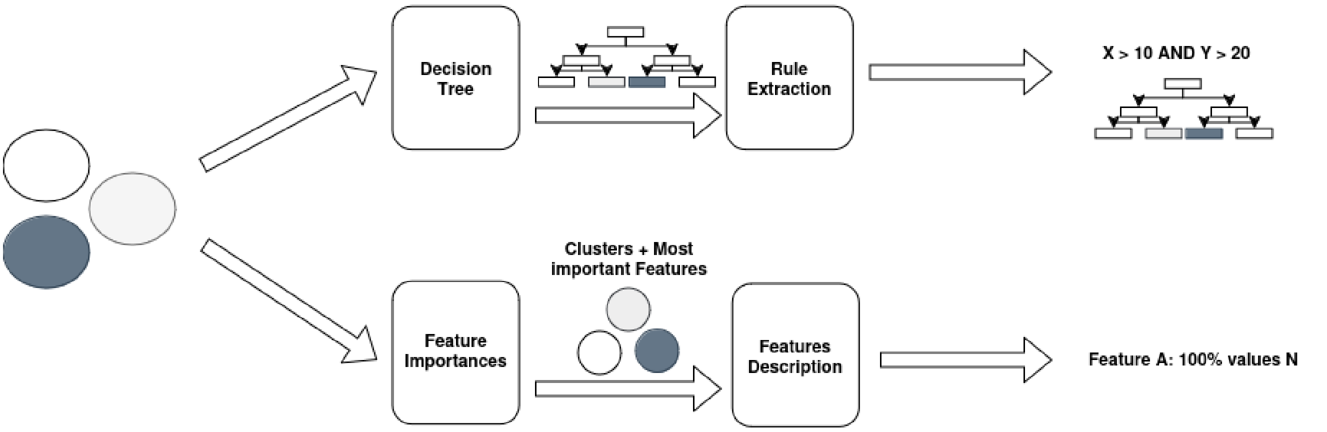


Figure 1. Overview of the proposed solution MAACLI

cesses rules from a decision tree trained with data grouped by some clustering algorithm (see Section 3.1); and a second component, which consists of generating descriptions for the most important attributes that define and characterize the groups created (see Section 3.2).

Figure 1 shows an overview of the proposed solution. The solution manages to provide complete descriptions so that end users can understand the groups created in a simple way, with a global view of the characteristics from the first component, and a complete description of the attributes with the second component. The two components together provide a detailed and user-friendly interpretation of the groups, which can be understood by end users. In summary, MAACLI is a solution for group interpretability independent of the model and clustering algorithm used, which generates three outputs (visuals, rules and descriptions) that are easily understood by end users of the cluster.

3.1 Interpretability using decision trees

A decision tree is a classification strategy that manages to separate data by creating a tree structure. To do this, they are recursively split binarily at each node of the tree, creating an axis-parallel hyperplane to split the data space into two resulting half-spaces or regions, which also induces a partition of the data points, until a leaf is reached. The aim of each split is to reduce the impurity of the data so that the classes are separated correctly [Breiman *et al.*, 1983]. Decision trees are a powerful machine learning tool, with an intrinsic characteristic: ease of interpretation [Molnar, 2022].

The tree generated by decision tree algorithms can be understood as rules in the format *If-Then* connected by the logical operator *AND*. Each path to a leaf in the tree represents a rule, where, if all the conditions are accepted for a sample, then it belongs to the class with the highest number of individuals at that node. With this, a tree can be created with the groups, so its rules can help interpret them. In this case, the group identifiers are utilized as categorical labels, and a tree is trained based on the predefined groups.

From the complete tree, two possible outputs can be obtained: visual and textual. The visual output shows the structure of the tree with its nodes and branches, while the textual output contains a representation of the rules in text form. The solution proposed in this work receives the dataset and the data labels that represent the group unique identifier assigned to each sample, indicating the cluster to which it belongs, as determined by an arbitrary clustering model, and returns a list with the rules for each group and the tree created.

Algorithm 1 outlines this component. A decision tree model is first created with the given parameters and the clustered data (line 1). Then, for each sample in a group, the rules are extracted and stored in a list (line 4). After finding the rules for all the data in the group, the unique rules are filtered out, removing redundancies, for example, the rule $X < 10$ *AND* $X < 9$ could be reduced to $X < 9$ only (line 5). After that, the number of occurrences of each rule is used to filter out those that represent a minimum number of data points (lines 6 - 9). This minimum number N is parameterizable and can be specified by the user according to their interests.

Algorithm 1 Decision tree rule extraction

Require: Dataset $D = \{x_1, x_2, \ldots, x_n\}$, Data Labels $L = \{l_1, l_2, \ldots, l_n\}$, minimum rules sample size N
Ensure: Set with rules for each of the groups and the visual tree output
1: $T \leftarrow Tree(D, L)$ ▷ Creates a decision tree with Data labels being the target
2: $Rules \leftarrow \{\}$
3: **for** each unique label $l \in L$ **do**
4: $R_l \leftarrow ExtractRules(T, l)$ ▷ Extract all rules that describe the data group with label l
5: $R_l \leftarrow RemoveRedundancies(R_l)$ ▷ Remove redundancies from the rules
6: **for** each unique rule $r \in R$ **do**
7: **if** Percentage of group l in $r < N$ **then**
8: $R_l \leftarrow R_l \setminus \{r\}$ ▷ Remove rules that are not representative
9: **end if**
10: **end for**
11: $Rules \leftarrow Rules \cup R_l$
12: **end for**
13: **return** $Rules,\ VisualOutput$

At the end of the execution of this component, visual and textual outputs are generated.

3.2 Interpretability by describing the most important attributes

The visual and textual outputs in the form of rules are important for indicating which criteria were taken into account when separating the groups. However, they still require manual and often technical analysis, making interpretation difficult. Therefore, to complement these outputs, the second component of MAACLI generates a user-friendly textual description, which is the main contribution of this work.

The second component of the solution consists of generating descriptions of the most important attributes for each group. To extract the most important attributes from each group, the proposed solution initially creates a classification model that can separate the groups, using the algorithm *XGBoost*[1]. This model is trained using the clustering data with the group identifiers of each individual as the labels. Once the model has been trained, this work proposes an adaptation of the *Permutation Feature Importance* [Breiman, 2001] technique to select the most important attributes of each group, as described below.

The *Permutation Feature Importance* technique consists of applying permutations to attributes to see how much this affects the performance of supervised models. Initially, the error of the original model is computed ($e = L(y, f(X))$), where y is the target vector, X is the data matrix, f is a trained model, and $L(y, f)$ is an error measure, such as accuracy or recall. After that, each attribute of $X = \{x_1, x_2, x_3, ..., x_n\}$ is permuted and a new error is computed ($e_{perm} = L(y, f(\hat{X}_j))$, where $\hat{X}_j$ represents the original data matrix with the attribute j permuted. The importance I_j of attribute x_j is calculated by the difference in errors ($I_j = e - e_{perm}$), and the more the permutation of an attribute negatively affects the classification results, the more important it is.

This strategy manages to extract the importance of features globally. However, to generate the description of the groups, it is necessary to find out what attributes are most important for each group individually. An attribute can be important for a specific group, characterizing the group in question and differentiating it from the others, while it is irrelevant for characterizing the other groups. The importance of attributes for each group, proposed in this work, deals with these situations.

To find the importance of each group, the *Permutation Feature Importance* was changed to analyze each class independently. To this end, the importance was calculated by observing how much the permutation of an attribute affects the recall of each class. Recall was used because it represents the sensitivity of the classification, indicating the relevant elements that were classified in each class. The adapted technique was then applied to the model trained with the grouped data and the most important characteristics for each group were extracted. Attributes with importance values greater than zero are considered important for a given group, while the rest are considered unimportant. The next step is to describe the most important attributes for each group.

Each group was described using an adaptation of the *Detailed Anti-Unified Algorithm* (DAU) [Corral *et al.*, 2009]. The idea behind the original algorithm is to create descriptions by adding a new element to each group. This new element contains as attributes the values present in most of the individuals already present in the groups and the new elements created are used as a description of each group. For MAACLI, the algorithm is adapted to also work with continuous numerical values. As DAU uses the concept of values present in the majority of elements, the original approach was unable to deal with continuous values.

Algorithm 2 Generating group descriptions

Require: Dataset $D = \{x_1, x_2, \ldots, x_n\}$, Data Labels $Y = \{y_1, y_2, \ldots, y_n\}$, Number of permutations N
Ensure: Set with descriptions for each of the groups

$M \leftarrow Model(D, Y)$ ▷ Creates a supervised model with Data labels being the target
for each label $l \in Y$ **do**
  $D_l \leftarrow \{x \in D \mid label(x) = l\}$
  $y_l \leftarrow \{y \in Y \mid y = l\}$
  $P_{baseline_l} \leftarrow L(y_l, M(D_l))$ ▷ Evaluate performance of the model for group l
  $F \leftarrow \{\}$
  for each feature $f \in D$ **do**
    $I_{f_l} \leftarrow 0$
    for $i = 1$ to N **do**
      $D' \leftarrow D$ ▷ Create a copy of the dataset
      $D' \leftarrow$ Permute the values of feature f in D'
      $P_{perm} \leftarrow L(y_l, M(D'_l))$ ▷ Evaluate the performance with permuted feature
      $I_{f_l} \leftarrow I_{f_l} + (P_{baseline_l} - P_{perm})$ ▷ Compute the importance of feature f
    end for
    $I_{f_l} \leftarrow I_{f_l} / N$ ▷ Average importance over all permutations
    if $I_{f_l} > 0$ **then**
      $F \leftarrow F \cup f$ ▷ include only important features to create descriptions
    end if
  end for
  $E \leftarrow createElement(D, l)$
  $I \leftarrow I \cup generateDescriptions(E, F)$
end for
return I

To meet this requirement, an adaptation using quartiles of the data was used. When a continuous numerical attribute is found, the values of the first and third quartiles, Q_1 and Q_3 respectively, are calculated and inserted into the description of that group. Thus, the continuous numerical value is converted into the format *75% greater than* Q_1 and *75% less than* Q_3. Thus, a categorical attribute with the first and third quartiles ($[Q_1, Q_3]$) of the continuous variable is inserted into the new group element, analyzing the distribution of group values. This adaptation respects the concept of attributes present in most group data, describing continuous attributes as well. Furthermore, using quartiles makes the description resistant to extreme values.

With this, the new elements created are used together with the importance of the attributes to describe each group. At-

[1] Another classification algorithm can be used

tributes with importance equal to zero of the new elements created are discarded, while attributes of greater importance are given preference in the description output.

Algorithm 2 details the proposed solution. Briefly, the description solution works as follows: initially, the groups are modeled with a classification algorithm (line 1), where a supervised algorithm is trained using the unsupervised dataset D and the set of data labels Y (i.e., clusters unique identifiers); after that, the most important attributes of each group are found with the adaptation of *Permutation Feature Importance*, where the model is first evaluated on the original data, and then the original data is permuted with respect to each feature, with the model being evaluated again on the permuted data (lines 2 - 19); then, a new description element is created for each group (line 20); and finally, the descriptions are generated by analyzing the most important attributes (line 21).

4 Results and Discussion

To evaluate the proposed solution, validation was first carried out on two datasets with few dimensions and few groups. This allows the correct results to be verified. Next, tests were carried out on real, complex data, with 27 dimensions and 14 groups, containing the segmentation of a company's customers. Finally, the solution was tested on synthetic datasets with a more complex distribution, demonstrating a limitation of the solution.

4.1 Validation Data Sets

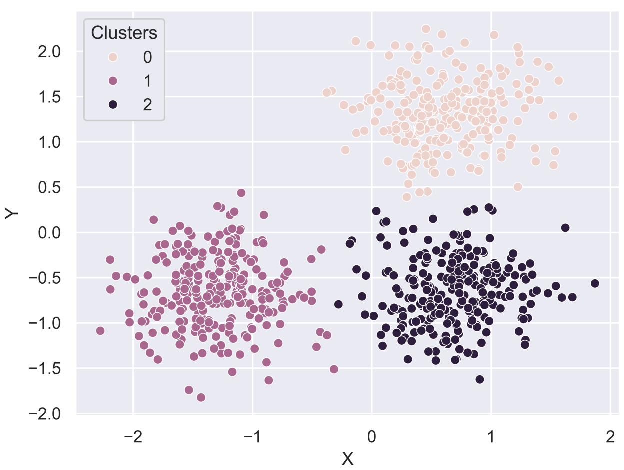


Figure 2. Synthetic - 1 Dataset Distribution

To carry out the first tests, two datasets were initially used. A synthetic set of clusters, illustrated in Figure 2, with two dimensions and 750 values separated into 3 groups with centers at $(1, 1), (-1, -1)$ and $(1, -1)$, generated manually using *Isotropic Gaussian Blobs*[2] and clustered using KMeans. And a second synthetic dataset, illustrated in Figure 3, also manually generated using *Isotropic Gaussian Blobs*[2], but with denser data and more complex clusters than the previous one, generated by the BIRCH clustering algorithm [Zhang *et al.*, 1996].

[2] `https://scikit-learn.org/stable/modules/generated/sklearn.datasets.make_blobs.html`

Figure 3. Synthetic - 2 Data Set Distribution

4.1.1 Decision Tree Rules

Table 2. Generated Rules for the Dataset Synthetic - 1

Data	Group	Rules
Synthetic - 1	0	Y > 0.11
Synthetic - 1	1	Y ≤ 0.11 AND X ≤ -0.23
Synthetic - 1	2	Y ≤ -0.15 AND X > 0.09

Table 2 shows the rules generated with the first component of the proposed solution. The tree went through the rule extraction and pre-processing stage, where only rules with at least 10% of the instances in each group were returned. This value is configurable, being used to filter out very specific rules that may bring complexity to the interpretation of the groups. It can be observed that the rules provide a simple explanation for the created groups. For the synthetic datasets, the data were delimited on the X and Y axes, according to each group. For the second synthetic dataset, the decision tree failed to model the data satisfactorily, generating too many rules with too few samples. However, the other component of the solution was able to describe the dataset well.

The visual output for the first synthetic data, using the decision tree, was also generated, containing 6 levels and 21 nodes. Appendix A presents an example of the visual output of the first component of MAACLI for synthetic data. You can see the complete tree with the detailed rules. It is worth noting that there are several rules that were not included in the rule generation, i.e., rules that contained less than 10% of the data in each group represented by them. The visual output can be useful for a better understanding of the groups, but it is more complex to analyze since it requires visual observation from the root to the leaf of the tree to understand a given rule.

4.1.2 Attribute descriptions

Table 3 presents the importance of the attributes returned for each group of the datasets used. The column *importance* contains tuples with the attribute name and its respective importance, ranging in the interval $[0, 1]$. Attributes with an importance of 0 were ignored. It is possible to observe that for the dataset synthetic - 1, attributes X and Y vary in importance across different groups. While for Group 0, attribute

Y is more important, for Group 1, attribute X shows greater importance. For Group 2, the attributes have similar importance in group creation. By the centers of the groups created at $(1, 1)$, $(-1, -1)$, and $(1, -1)$ respectively for Groups 0, 1, and 2, it is possible to observe that these importance are coherent. From Figure 2, it is possible to observe that these attribute importance are consistent with the data.

For the second synthetic dataset, it can be seen that the importance also managed to represent the data well. For groups 0 and 2, feature Y was the most important, while for group 1 feature X was the most important. Looking at Figure 3, you can see that the importance is coherent. Groups 0 and 1 are almost completely separated by the Y axis, while group 1 is separated from the others on the X axis.

Table 3. Generated importance for the Datasets

Data	Groups	importance
Synthetic - 1	0	(X, 0.17), (Y, 0.67)
Synthetic - 1	1	(X, 0.61), (Y, 0.16)
Synthetic - 1	2	(X, 0.31), (Y, 0.35)
Synthetic - 2	0	(X, 0.08), (Y, 0.48)
Synthetic - 2	1	(X, 0.80), (Y, 0.18)
Synthetic - 2	2	(X, 0.08), (Y, 0.51)

Figure 4 shows the description of the second synthetic dataset. It manages to extract the characteristics of the cluster in a user-friendly way and is a simple description. However, looking at Figure 3, you can see that the solution is unable to extract some more complex shapes, limiting itself to describing the data through a rectangular or cubic region (for three or more dimensions) in the plane with the most relevant data for the description, which can hide important information about the cluster. A more complete analysis of the limitations of the solution is conducted in Section 4.3.

Cluster 0 description:
Y:
75% greater than -1.385
75% less than -0.351
X:
75% greater than -0.375
75% less than 1.396

Cluster 1 description:
X:
75% greater than -2.128
75% less than -1.363
Y:
75% greater than -0.258
75% less than 1.368

Cluster 2 description:
Y:
75% greater than 0.387
75% less than 1.388
X:
75% greater than -1.025
75% less than 0.555

Figure 4. Synthetic - 2 Description

4.2 Real Dataset: Customer Segmentation

After validation using datasets with few dimensions and instances, MAACLI was applied to a real dataset from a company.

The dataset used consists of customer segmentation data from a private company. It contains information on 263,684 customers segmented into 14 groups using a proprietary solution from the company, where one group was used as noise to include customers who did not fit into any other group. The data comprises 27 attributes representing characteristics such as age, income, contracted products, and profession of the customers. The dataset was applied in the proposed solution to interpret the groups and thus enable better decision-making by end-users for the created groups. For privacy reasons, customer identifiers were not used, nor were any other attributes that could be used to trace them. Product names were also changed.

Tree and Rules: Figure 5(a) shows the rules generated by the first component of the solution for the first 3 groups of the dataset. The data has some *dummy* attributes and the pre-processing stage deals with this by providing simpler outputs through affirmations (i.e., *Is*) or negations (i.e., *Is not*), as can be seen in the figure. According to the rules, Group 0 is characterized by containing customers who do not use certain types of products (i.e., Products 10, 9, 6, and 7) and have Profession 3. For Group 1, the result showed that it contains customers who use Product 9 and do not belong to Profession 2. Finally, Group 2 is made up of customers who use Product 10. With these rules, it is possible to target strategic campaigns to groups of customers with a certain profession and product, for example.

Description of Important Attributes: Moving on to the second component, you can see the output of the description of the attributes in Figure 5(b). The previous rules can provide a characterization of the groups created, aiding interpretation. With the description of the attributes, it is possible to add even more value to the interpretation of the groups. Initially, it can be seen that for Group 0, 2 important attributes are identified that influence its creation. Group 0 can be characterized by customers with Profession 3 who use Product 4, while Group 1 is characterized by customers that use Product 9. However, Group 1 also has customers with a similar length of time as customers and a well-defined age, between 40 and 51. Group 1's income is in a wide range (i.e., 1,500 to 10,000), but was still considered important by the solution. As for Group 2, it is possible to observe the presence of customers with Product 10, a lower age (i.e., 21 to 27 years), a lower income (i.e., 400 to 1,200), and only one active product on average.

To visually illustrate the coherence of the descriptions, Figure 6 contains a radar chart with the distribution of the important attributes for the 3 groups analyzed (Groups 0, 1, and 2), comparing them with the other groups. In line with the output of the descriptions, Group 0 (blue) stood out from the other groups with regard to Product 4 and Profession 3 of the customers present. For Group 1 (red), Product 9 and customer time stood out from the other groups, which was described in the output of the proposed solution (see Figure 5(b)). Finally, for Group 2 (green), it is possible to observe the strong pres-

Group 0 Rules:
Not Product 10 AND
Not Product 9 AND
Not Product 6 AND
Profession 3 AND
Not Product 7
Quantity: 2713 - 100.0%

Group 1 Rules:
Product 9 AND
Not Profession 2
Quantity: 6329 - 100.0%

Group 2 Rules:
Product 10
Quantity: 7331 - 100.0%

(a) Rules of Groups 0, 1, and 2

Group 0 Description:
Profession 3: 100% of values 1
Product 4: 100% of values 1

Group 1 Description:
Product 9: 100% of values 1
Customer time:
  75% greater than 1744.0
  75% less than 1805.0
Customer age:
  75% greater than 40.0
  75% less than 51.0
Average income:
  75% greater than 1500.0
  75% less than 10000.0

Group 2 Description:
Average active products: 100% of values 1
Product 10: 100% of values 1
Customer age:
  75% greater than 21.0
  75% less than 27.0
Average income:
  75% greater than 400
  75% less than 1200

(b) Description of Groups 0, 1, and 2

Figure 5. Application of MAACLI on real data

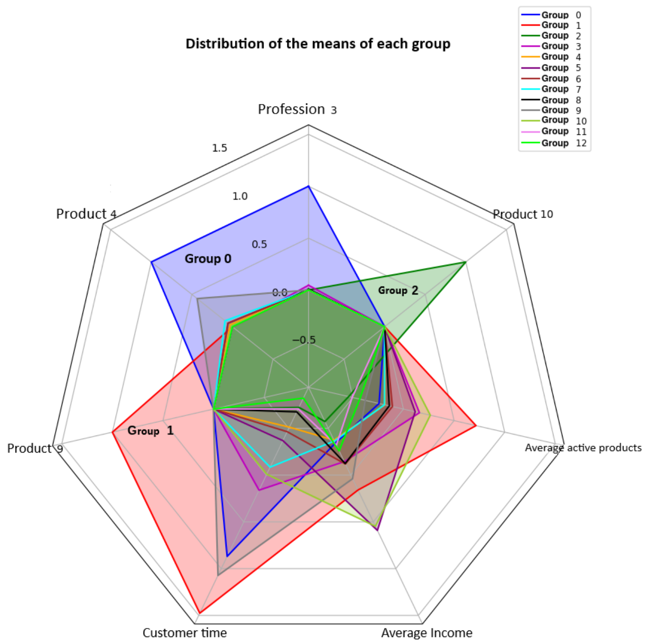


Figure 6. Distribution of some attributes for each group

ence of Product 10, as well as average income and average active products below the other groups, as also generated in the solution output in Figure 5(b). In other words, the groups analyzed were well described by the solution compared to the other groups. This analysis was also carried out for the other groups, but as each analysis generates a different figure due to the differences in the important attributes, it was decided to keep the analysis of groups 0, 1, and 2 as an example.

To complement the analysis, the Mann-Whitney [Mann and Whitney, 1947] hypothesis test was applied for continuous variables and the chi-squared (χ^2) test for categorical variables, considering the two most important attributes found by the algorithm for each group. The Chi-squared (χ^2) test is a statistical test used to determine if there is a significant association between categorical variables, where the null hypothesis is that there is no difference in the distribution of responses to the outcome across comparison groups. The Mann–Whitney test is a nonparametric test that compares two independent groups to assess whether their distributions differ and the null hypothesis is that there is no difference between the two groups in the population.

This way, we can see if there is a significant difference between one group and the others for the most important attributes found by MAACLI. For this, we conducted the experiment with the cluster for which the important feature was identified, while the remaining clusters comprised the other group we aimed to compare and apply the tests. Our null hypothesis is that there is no significant difference between these two groups.

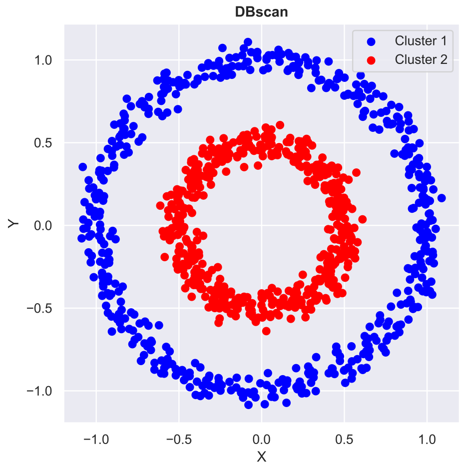
You can see the results of the tests in Table 4. The low p-values show that the important features found are significantly different for the rest of the groups. Note that some groups have only one important attribute, showing that just one attribute is enough to describe some groups.

Table 4. Hypothesis tests of the two most important attributes for each group.

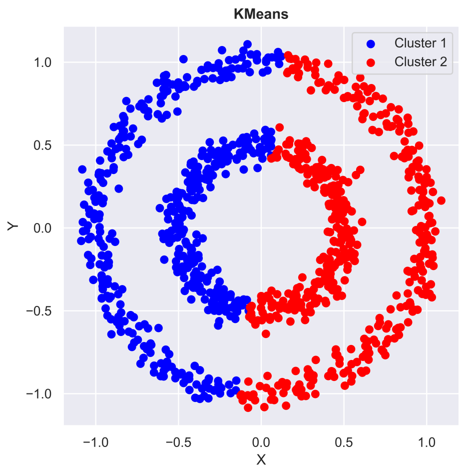
Group	p-value Attribute 1	p-value Attribute 2
Group 0	$\ll 0.0001$	$\ll 0.0001$
Group 1	$\ll 0.0001$	$\ll 0.0001$
Group 2	$\ll 0.0001$	$\ll 0.0001$
Group 3	$\ll 0.0001$	-
Group 4	$\ll 0.0001$	-
Group 5	$\ll 0.0001$	-
Group 6	$\ll 0.0001$	-
Group 7	$\ll 0.0001$	-
Group 8	$\ll 0.0001$	0.041
Group 9	$\ll 0.0001$	-
Group 10	$\ll 0.0001$	-
Group 11	$\ll 0.0001$	$\ll 0.0001$
Group 12	$\ll 0.0001$	-

Comparison with base solution: Finally, a comparison was made with the ExKMC [Frost *et al.*, 2020] solution, which creates an expanded version of *K-Means*. The result was a tree with 14 levels and 51 leaves, which led to the explanation of all groups, with very complex rules. The reason for such a complex tree is that, unlike MAACLI , ExKMC does not allow pruning according to the number of items covered by the rules. More importantly, it is worth highlighting that to run ExKMC, it was necessary to use the original raw data, as it is a specific clustering algorithm. In other words, unlike MAACLI which is agnostic in terms of algorithm and model, ExKMC does not allow the use of another algorithm. Finally, in addition to the output in tree format, MAACLI also generates textual rules and descriptions.

4.3 Limitations



(a) DBscan



(b) KMeans

Figure 7. Synthetic dataset distribution

Finally, MAACLI was tested on a synthetic dataset presented in Figure 7 representing two concentric circles, clustered with two different clustering algorithms.

Table 5. Importance generated for the circles

Group	X-axis	Y-axis
Group 1 - DBscan	0.33	0.34
Group 2 - DBscan	0.31	0.30
Group 1 - KMeans	0.48	0
Group 2 - KMeans	0.49	0

It is possible to see the importance generated for the groups in Table 5. Both features have similar importance for DBscan clustering, showing that both are important for the groups. When KMeans is used, only the X axis is considered important. This makes sense given the distribution of the data. In Figure 8 we can see the description of group 0 generated by MAACLI for DBScan clustering. Note that the description cannot capture the circular and concentric shape of the data, which may lead the user to infer that the data has a distribution that is not consistent with the actual distribution of the data. Given the simplicity of the KMeans clustering, its description has been omitted.

Cluster 1 Description:
Y:
  75$\%$ greater than -0.9
  75$\%$ less than 0.892
X:
  75$\%$ greater than -0.881
  75$\%$ less than 0.88

Cluster 2 Description:
X:
  75$\%$ greater than -0.431
  75$\%$ less than 0.445

Y:
  75$\%$ greater than -0.443
  75$\%$ less than 0.433

Figure 8. Descriptions for concentric circles

In these cases where the data presents more complex distributions, the solution may not be suitable for the interpretability of the groups. Similarly, algorithms that can generate clusters with more complex shapes, such as density-based clustering algorithms, may not have the groups fully explained by the solution descriptions. In other words, the same data clustered in different ways can influence the shape of the clusters, making it difficult to interpret them using the proposed solution.

Likewise, the rules generated by the tree with at least 10% of the samples, shown in Figure 9, despite being more interesting and complete, may not show in a simple way the distribution presented in the data. As previously mentioned, decision trees divide data into half-spaces, making it difficult, for example, for data containing concentric circles separated into two clusters.

To further analyze the solution, a benchmark [Gagolewski, 2022] called g2mg[3] was used with several datasets of synthetic data. The benchmark consists of data generated using Gaussian components, with a standard deviation equal to $variance \times \sqrt{dimension/n}$ where n is the number of clusters in the dataset. Data was generated containing two and ten well-defined centers, with the $dimension$ and $variance$ varying between smaller and larger values, to assess the impact on the results.

With this data, our solution was applied to generate each cluster's description. For a dataset with k clusters, our solution generates a description $D = \{d_1, d_2, ..., d_k\}$, with one set of conditions for each cluster i. To assess the quality of a

[3]https://clustering-benchmarks.gagolewski.com/weave/data-v1.htmlg2mg

Cluster 1 rules:
X greater than 0.58
Quantity: 153 - 30.6%

X less than or equal to 0.58 AND
X greater than -0.61 AND
Y greater than 0.63
Quantity: 104 - 20.8%

X less than or equal to 0.58 AND
X greater than -0.61 AND
Y less than -0.68
Quantity: 98 - 19.6%

X less than or equal to -0.61
Quantity: 145 - 29.0%

Cluster 2 rules:
X less than or equal to 0.58 AND
X greater than -0.61 AND
Y less than or equal to 0.63 AND
Y greater than -0.68
Quantity: 500 - 100.0%

Figure 9. Rules for concentric circles

description generated, the average coverage is calculated for each data set used. The average coverage consists of:

$$avg\_cov = \sum_{i=1}^{k} \frac{|V|}{|C_i|} \tag{1}$$

Where $V = \{\forall x_j \in C_i \mid d_i(x_j) = \text{True}\}$ and $C_i \subset X$, $C_i = \{x_1, x_2, ..., x_n\}$ the set of elements that belong to cluster i. An average coverage of 1 means that all elements of all clusters were correctly covered by the description generated by our solution.

Figures 10 and 11 show the average coverage of the descriptions generated for the benchmark for datasets with 2 and 10 clusters, respectively. Figure 10 shows that, as the dimension increases, the coverage tends to decrease slightly, but not significantly for most cases. From 16 dimensions onwards, some cases with zero coverage were identified, due to the algorithm's difficulty in finding the important features.

On the other hand, when we have 10 clusters (see Figure 11), increasing the dimension reduces the chance of not finding important features, but the solution becomes more sensitive to the number of dimensions, making it difficult to describe the clusters. In this scenario, variance had no negative effect for the same dimension value.

Based on these results, we can see that MAACLI is able to describe clusters with up to 8 dimensions well, with acceptable coverage of over 80%, even for 10 clusters. However, when the number of dimensions is 16 or higher, the coverage of cluster explanation decreases for the scenario with 10 clusters and becomes unstable for the scenario with 2 clusters, being more impacted by the variance of the data. It is therefore important to check the coherence of the description for high dimensions, regardless of the number of clusters found and the variance of the data.

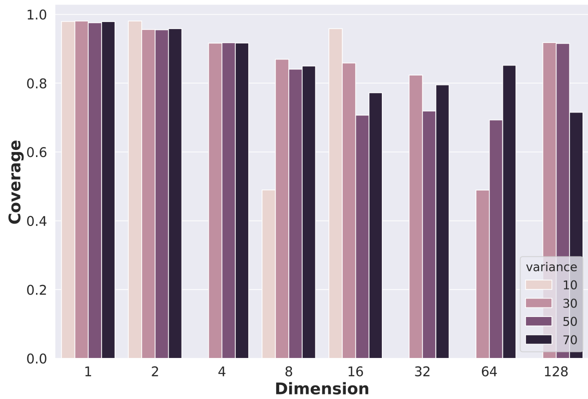


Figure 10. Data with 2 centers

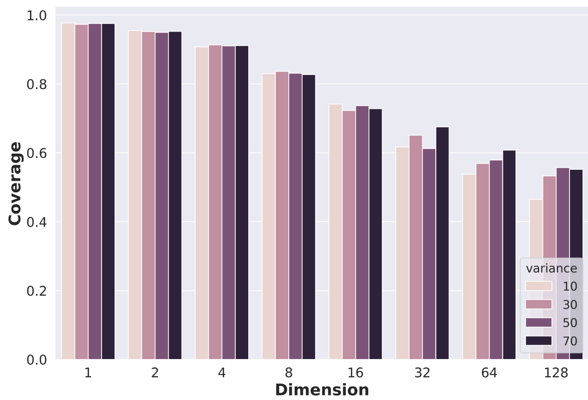


Figure 11. Data with 10 centers

5 Concluding Remark

This paper presents a solution for the interpretability of groups generated by unsupervised algorithms. The solution automatically returns user-friendly descriptions of the groups, regardless of the model and algorithm used. This makes it possible for end users, without technical knowledge, to interpret the groups created without having to carry out extensive and repetitive manual analysis. The results show that two types of simple descriptions are generated which are capable of characterizing the groups, even for real data with many attributes and groups.

For both well-separated and complex data, the solution is able to generate simple and user-friendly descriptions. However, for certain types of data, clustering, and dimensions, MAACLI should be used cautiously, as certain characteristics of the data may not be captured, since our solution describes the data as rectangles or cubes in space, which may not be ideal for certain datasets. As future work, it is possible to propose better ways to describe continuous attributes, exploring, for example, data variance or data discretization strategies. Furthermore, improvements in methods can be proposed to simplify the description of data with more complex and less clear distributions, keeping the output friendly. Finally, the way of finding important features could be improved to avoid using an auxiliary supervised model.

Acknowledgements

The authors thank Fapemig, Manna Team, Softex, and CNPq (Number 421548/2022-3).

Authors' Contributions

Oliveira, G. S.: Conceptualization, Methodology, Implementation, Validation, Writing. **Silva, F. A.**: Supervision, Conceptualization, Methodology, Validation, Writing. **Ferreira, R. V.**: Supervision, Conceptualization, Data Curation, Review.

Competing interests

The authors declare that they do not have competing interests.

Availability of data and materials

The synthetic datasets are public. The real dataset is not available.

References

Bartels, C. (2022). Cluster analysis for customer segmentation with open banking data. In *2022 3rd Asia Service Sciences and Software Engineering Conference*, ASSE' 22, page 87–94, New York, NY, USA. Association for Computing Machinery. DOI: 10.1145/3523181.3523194.

Bertsimas, D., Orfanoudaki, A., and Wiberg, H. M. (2020). Interpretable clustering: an optimization approach. *Machine Learning*, 110:89–138.

Breiman, L. (2001). Random forests. *Machine learning*, 45(1):5–32.

Breiman, L., Friedman, J. H., Olshen, R. A., and Stone, C. J. (1983). Classification and regression trees.

Corral, G., Armengol, E., Fornells, A., and Golobardes, E. (2009). Explanations of unsupervised learning clustering applied to data security analysis. *Neurocomputing*, 72(13):2754–2762.

Dasgupta, S., Frost, N., Moshkovitz, M., and Rashtchian, C. (2020). Explainable k-means and k-medians clustering. ICML'20. JMLR.org.

Ellis, C. A., Sendi, M. S. E., Plis, S., Miller, R. L., and Calhoun, V. D. (2021). Algorithm-agnostic explainability for unsupervised clustering. *ArXiv*, abs/2105.08053.

Fraiman, R., Ghattas, B., and Svarc, M. (2013). Interpretable clustering using unsupervised binary trees. *Advances in Data Analysis and Classification*, 7:125–145.

Frost, N., Moshkovitz, M., and Rashtchian, C. (2020). Exkmc: Expanding explainable k-means clustering. *ArXiv*, abs/2006.02399.

Gagolewski, M. (2022). A framework for benchmarking clustering algorithms. *SoftwareX*, 20:101270. DOI: https://doi.org/10.1016/j.softx.2022.101270.

Ghahramani, Z. (2003). Unsupervised learning. In *Summer school on machine learning*, pages 72–112. Springer.

Laber, E., Murtinho, L., and Oliveira, F. (2023). Shallow decision trees for explainable k-means clustering. *Pattern Recognition*, 137:109239. DOI: https://doi.org/10.1016/j.patcog.2022.109239.

Li, Y., Chu, X., Tian, D., Feng, J., and Mu, W. (2021). Customer segmentation using k-means clustering and the adaptive particle swarm optimization algorithm. *Applied Soft Computing*, 113:107924.

Liu, Y., Li, Z., Xiong, H., Gao, X., and Wu, J. (2010). Understanding of internal clustering validation measures. In *2010 IEEE International Conference on Data Mining*, pages 911–916. DOI: 10.1109/ICDM.2010.35.

Loyola-González, O., Gutierrez-Rodríguez, A. E., Medina-Pérez, M. A., Monroy, R., Martínez-Trinidad, J. F., Carrasco-Ochoa, J. A., and García-Borroto, M. (2020). An explainable artificial intelligence model for clustering numerical databases. *IEEE Access*, 8:52370–52384. DOI: 10.1109/ACCESS.2020.2980581.

Mann, H. B. and Whitney, D. R. (1947). On a Test of Whether one of Two Random Variables is Stochastically Larger than the Other. *The Annals of Mathematical Statistics*, 18(1):50 – 60. DOI: 10.1214/aoms/1177730491.

Molnar, C. (2022). *Interpretable Machine Learning*. 2 edition.

Morichetta, A., Casas, P., and Mellia, M. (2019). Explain-it: Towards explainable ai for unsupervised network traffic analysis. Big-DAMA '19, page 22–28, New York, NY, USA. Association for Computing Machinery. DOI: 10.1145/3359992.3366639.

Moura, M., Veras, R., and Machado, V. (2022). Caibal: cluster-attribute interdependency based automatic labeler. In *Proceedings of the 37th ACM/SIGAPP Symposium on Applied Computing*, SAC '22, page 1109–1116, New York, NY, USA. Association for Computing Machinery. DOI: 10.1145/3477314.3507140.

Oliveira, G., Silva, F., and Ferreira, R. (2023). Model and algorithm-agnostic clustering interpretability. In *Anais do XI Symposium on Knowledge Discovery, Mining and Learning*, pages 33–40, Porto Alegre, RS, Brasil. SBC. DOI: 10.5753/kdmile.2023.232618.

Ribeiro, M. T., Singh, S., and Guestrin, C. (2016). "why should i trust you?": Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, page 1135–1144, New York, NY, USA. Association for Computing Machinery. DOI: 10.1145/2939672.2939778.

Xu, R. and Wunsch, D. (2005). Survey of clustering algorithms. *IEEE Transactions on neural networks*, 16(3):645–678.

Yu, Z., Sohail, A., Nofal, T. A., and Tavares, J. M. R. S. (2022). Explainability Of Neural Network Clustering In Interpreting The Covid-19 Emergency Data. *FRACTALS (fractals)*, 30(05):1–12. DOI: 10.1142/S0218348X22401223.

Zhang, T., Ramakrishnan, R., and Livny, M. (1996). Birch: an efficient data clustering method for very large databases. In *Proceedings of the 1996 ACM SIGMOD International Conference on Management of Data*, SIGMOD '96, page 103–114, New York, NY, USA. Association for Computing Machinery. DOI: 10.1145/233269.233324.

Appendix A Visual Tree Output