Cluster-Explorer: Explaining Black-Box Clustering Results

Sariel Tutay
Bar-Ilan University
Ramat Gan, Israel
sarielt@cs.biu.ac.il

Amit Somech
Bar-Ilan University
Ramat Gan, Israel
somecha@cs.biu.ac.il

ABSTRACT

Interpreting clustering results is a challenging, manual task, that often requires the user to perform additional analytical queries and visualizations. To this end, we demonstrate Cluster-Explorer, an interactive, easy-to-use framework that provides explanations for black-box clustering results. Cluster-Explorer takes as input the raw dataset alongside cluster labels, and automatically generates multiple coherent explanations that characterize each cluster.

We first propose a threefold quality measure that considers the conciseness, cluster coverage, and separation error of an explanation. We tackle the challenge of efficiently computing high-quality explanations using a modified version of a *generalized frequent-itemsets mining* (gFIM) algorithm. The gFIM algorithm is employed over multiple filter predicates which are extracted by applying various binning methods of different granularities. We implemented Cluster-Explorer as a Python library that can be easily used by data scientists in their ongoing workflows. After employing the clustering pipeline of their choice, Cluster-Explorer opens an integrated, interactive interface for the user to explore the various different explanations for each cluster.

In our demonstration, the audience is invited to use Cluster-Explorer on numerous real-life datasets and different clustering pipelines and examine the usefulness of the cluster explanations provided by the system, as well as its efficiency of computation.

CCS CONCEPTS

• **Mathematics of computing** → **Cluster analysis**; *Exploratory data analysis*; • **Information systems** → *Data analytics*.

KEYWORDS

Explainability for unsupervised learning; Data Exploration

ACM Reference Format:
Sariel Tutay and Amit Somech. 2023. Cluster-Explorer: Explaining Black-Box Clustering Results. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management (CIKM '23), October 21–25, 2023, Birmingham, United Kingdom.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3583780.3614734

1 INTRODUCTION

Clustering is an important, effective tool in the data scientist's arsenal, used for discovering patterns and structures in data. Dozens of different clustering algorithms have been devised to segment the data points into groups based on their feature distributions. While the results of these algorithms can be easily visualized on a two dimensional plain (See Figure 1), interpreting them can be challenging. To unravel the unique characteristics of each cluster and find out what differentiates them, users often need to manually perform additional analytical queries and data visualizations.

To this end, we present Cluster-Explorer, a system that provides explanations for black-box clustering results. Given the dataset and the cluster *labels* resulted from a clustering pipeline, Cluster-Explorer automatically generates coherent explanations that characterize each cluster. For illustration, consider the following example.

Example 1.1. Consider Clarice, a data analyst examining the well-known "Adult" dataset[2], containing demographic information on individuals alongside their income (See Table 1 for a sample). Clarice uses an Agglomerative Clustering algorithm [11] then visualizes the results (after projecting the data into a 2D plane using a method such as PCA [6]. While the clusters (see Figure 1) seem fairly separated in the projected space, Clarice still needs to figure out what attributes in the original data characterize each cluster. After extensive manual analysis, including several analytical queries and visualizations, Clarice is able to characterize the clusters, understanding that, for example, Cluster 0 mainly contains *individuals up to 35 years of age, with 6-14 years of education*, whereas Cluster 1 comprises *married adults, over 35, with 0-5 years of education.* □

While in this example the clusters' characteristics were manually extracted by the user, the goal of Cluster-Explorer is to generate such explanations automatically. To do so, we tackle three important challenges: (1) How to gauge the quality of a cluster explanation? (2) How to efficiently compute good explanations? (3) How to design an interactive interface for exploring the different explanations?

(1) Quality of explanation. Naturally, the most accurate explanation is one that describes all data-points in the cluster. This can be done, naively, by creating a disjunction of all feature values of a cluster's members. While accurate, such an explanation is long and incoherent. On the other hand, shorter explanations may be less accurate as they can also be valid for some of the points in different clusters, as well as invalid for points in the explained cluster.

To tackle this challenge, we devise a threefold quality measure that considers the explanation's (1) *conciseness* – measures how short is an explanation, (2) *cluster coverage* – evaluates the proportion of the cluster's points that can be described by the explanation, and (3) *separation error* – which counts the number of points in different clusters that the explanation describes.

Ideally, a good explanation is concise (short), has high cluster coverage, and a low separation error.

(2) Efficient Computation of Explanations. As we formally define in Section 2, explanations are comprised of filter conditions on the

CIKM '23, October 21–25, 2023, Birmingham, United Kingdom

ACM ISBN 979-8-4007-0124-5/23/10.
https://doi.org/10.1145/3583780.3614734

data (equivalent to, e.g., SQL Where clause). Since, naturally, many such explanation candidates exist, another challenge is finding high-quality explanations in a timely fashion.

To overcome this challenge, Cluster-Explorer combines multiple binning methods of different granularity levels, and feeds their results into a *generalized* frequent itemset mining [14] algorithm. This step yields high-coverage explanation candidates, which are then sorted by their conciseness and separation error.

(3) Interactive UI. Cluster-Explorer often outputs several high-quality cluster explanations. To allow users to interactively browse through them, we first calculate the skyline [3] w.r.t. the explanations' conciseness, coverage and separation error. Then, we select only the dominating explanations, and let the user sift through them by controlling the Explanation Quality Sliders (see Figure 2).

Related Work. A multitude of works exists in the domain of Explainable AI (XAI), mainly focused on supervised learning tasks [1, 9, 12, 13]. Unfortunately, such systems (which, e.g., reveal important features w.r.t. a certain prediction), cannot be trivially adapted for the case of explaining unsupervised clustering algorithms. Another line of work proposes cluster algorithms that are interpretable by design [5, 8]. Differently, Cluster-Explorer is not tied to a particular clustering algorithm. This is useful, as there are dozens of clustering methods, each suitable for different data types and distributions [18]. Closer to our work, [7] suggests an algorithm-agnostic explainability tool for clustering. The tool provides feature importance scores for each cluster. Differently, Cluster-Explorer provides cluster explanations comprised of tight filter conditions (as opposed to only feature names) that characterize each cluster in a concise and coherent manner.

2 SOLUTION ARCHITECTURE

We begin by describing our model and definitions, then present the quality assessment and efficient algorithm for cluster explanations.

2.1 Model & Definitions

Cluster-Explorer takes as input a dataset and a black-box clustering algorithm: A dataset $D = \langle X, A \rangle$ with n data points $X = x_1, \ldots, x_n$, projected over m attributes $A = a_1, \ldots, a_m$. We denote by x_i the i-th data point, and by x_{ij} the projection of x_i over attribute a_j. A clustering function $F : X \longrightarrow C$ maps each point x_i to a cluster $c \in C$ (C is a set of cluster labels).

In Cluster-Explorer, a *cluster explanation* is defined as a conjunction of predicates $E(x) = \{P_1 \wedge P_2 \ldots P_l\}$. The predicates are of the form $P := \langle a, op, V \rangle$, where $a \in A$, op is an operator (e.g., <, >, 'contains', 'between', ...), and V is a set of literal values.

Given a data point $x \in X$, the explanation $E(x)$ is said to be *true* if x satisfies all predicates in E, and *false* otherwise.

Example 2.1. Table 1 contains a sample of the Adult dataset, alongside cluster labels. Rows 1-3 in the table are labeled as Cluster 0. Table 2 depicts three candidate explanations for Cluster 0 (ignore, for now, the three right-most columns). Explanation E_0^2, for example, comprises of the predicates $P_1 := \langle 'age', between, (16, 35) \rangle$ and $P_2 := \langle 'education-num', between, (4, 13) \rangle$.

Out of the rows in Table 1 that indeed belong to Cluster 0, see that Explanation E_0^2 holds for Rows 124 and 53, yet is not true for Row 32 (having 'age' of 41). By contrast, Explanation E_0^1 holds for all three rows (IDs 124, 32, and 53) yet unfortunately – it also holds for rows 5631 and 39 of Cluster 1 and Cluster 2.

Row ID	Age	Edu.num	Relationship	Gender	...	Hrs-per-week	Income	Cluster
124	25	7	Unmarried	Male	...	40	$\leq 50K$	0
32	41	10	Unmarried	Female	...	50	$\geq 50K$	0
53	34	12	Unmarried	Male	...	50	$\geq 50K$	0
...	...	...	...	...	...	...	...	...
342	36	3	Husband	Male	...	50	$\geq 50K$	1
521	40	3	Husband	Male	...	50	$\leq 50K$	1
5631	45	5	Wife	Female	...	60	$\geq 50K$	1
...	...	...	...	...	...	...	...	...
39	46	12	Wife	Female	...	30	$\leq 50K$	2
938	33	15	Husband	Male	...	60	$\geq 50K$	2
693	36	14	Husband	Male	...	50	$\geq 50K$	2
...	...	...	...	...	...	...	...	...

Table 1: Adult Dataset Sample, with Cluster Labels

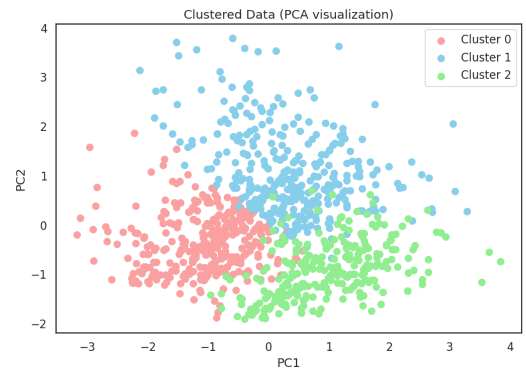

Figure 1: Clustering Results Visualization (Adult dataset)

2.2 Quality Measures for Cluster Explanations

The question of what constitutes a 'good' explanation has been investigated in various different domains such as cognitive science, philosophy and psychology. Relying on this vast body of research, [10] suggests that in the context of XAI, a *good* explanation is primarily *contrastive* (i.e., why event P happened *instead* of an event Q), but also *simple*, *coherent*, and *truthful*.

In Cluster-Explorer, we adapt these criteria to the use case of explaining clustering results and develop corresponding quality metrics for an explanation. Given a cluster $c \in C$, a good explanation E_c has (1) high *coverage* of the points in c while retaining a (2) low *separation* error. Namely, the explanation is valid for the majority of the points in c, and invalid for the points associated with any other cluster $c' \in C, c' \neq c$. The higher the score w.r.t. (1) and (2), as defined below, the more contrastive and truthful the explanation is. However, as mentioned above, a naive explanation with a perfect score could be the union of the description of each point in c. Naturally, such an explanation is long and incoherent. We therefore introduce a (3) *conciseness* measure which corresponds to the number of predicates in the explanation E_c. We next define the three explanation quality metrics.

1. Cluster Coverage. Given a dataset D, clusters C and mapping function CL, the *coverage* of an explanation E_c (for cluster c) is defined as the ratio of the points associated with cluster c that the explanation E_c describes:

$$Coverage(E_c) := \frac{|\{x \in D \mid E_c(x) = True \wedge CL(x) = c\}|}{|\{x \in D \mid CL(x) = c\}|}$$

Exp.num	Explanation	Cluster label	coverage	Separation Error	Conciseness
E_0^1	⟨'age',between,(16,48)⟩ ∧ ⟨'education-num',between,(4,13)⟩ ∧⟨'relationship',!=,husband⟩	0	0.99	0.05	0.33
E_0^2	⟨'age',between,(16,35)⟩∧ ⟨'education-num',between,(4,13)⟩	0	0.95	0.04	0.5
E_0^3	⟨'age',between,(16,53)⟩ ∧ ⟨'hours-per-week',between,(10,72)⟩ ∧⟨'education-num',between,(4,14)⟩	0	0.88	0.04	0.33

Table 2: Example Candidate Explanations

2. Separation Error. This is the ratio of points that the explanation E_c is valid for, yet do not belong to cluster c:

$$SeparationErr(E_c) := \frac{|\{x \in D \mid E_c(x) = True \wedge CL(x) \in C \setminus \{c\}\}|}{|\{x \in D \mid E(x) = True|}$$

3. Conciseness. Following [10], the length and simplicity of the explanations are important for its comprehension by the users. We therefore define the conciseness of an explanation to be the inverse number of predicates it contains:

$$Conciseness(E_c) := \frac{1}{|\{P \mid P \text{ is a predicate in } E_c\}|}$$

The following example shows how the metrics are calculated for the explanations in Table 2.

Example 2.2. Consider Explanation E_0^1, as depicted in Table 2. Assume that the total number of data points belonging to Cluster 0 is 373, out of which 370 satisfy E_0^1. In addition, 20 other data points that belong to Clusters 1 and 2 also satisfy E_0^1. Calculating the scores for E_0^1 we obtain: $Coverage(E_0^1)) = \frac{370}{373} = 0.99$, $SeparationErr(E_0^1)) = \frac{20}{390} = 0.05$, and $Conciseness(E_0^1)) = \frac{1}{3} = 0.33$.

Explanations Quality Skyline. There's a natural trade-off between the quality measures. For instance, an explanation obtaining a very high *coverage*, may have a lower *conciseness* score and higher *separation error*, whereas a highly *concise* explanation may fall short on *coverage*. Cluster-Explorer allows the users to provide an initial threshold for each measure (denoted $\theta_{cov}, \theta_{sep}$, and θ_{con}, for coverage, separation, and conciseness, resp.) Then, to balance the measures, we use the *skyline operator* [3] (also known as Pareto Frontier) calculation. Namely, we are interested only in the *dominating* explanations, having no other explanation surpassing it w.r.t. all three measures. Formally, explanation E_c is dominating *iff*:

$$\begin{aligned} \nexists E'_c \mid\ & Coverage(E'_c) \geq Coverage(E_c) \\ & \wedge SeparationErr(E'_c) \leq SeparationErr(E_c) \\ & \wedge Conciseness(E'_c) \geq Conciseness(E_c) \end{aligned}$$

Example 2.3. Consider again the candidate explanations depicted in Table 2. Explanation E_0^1 is better than E_0^2 w.r.t. coverage (0.99 to 0.95) yet is inferior w.r.t. the separation error (0.05 to 0.04) and conciseness (0.33 to 0.5). While E_0^1 and E_0^2 are both incomparable and will appear on the skyline, see that Explanation E_0^3 is dominated by E_0^2 as it has the same separation error, yet inferior coverage and conciseness (0.88 and 0.33, resp.).

2.3 Generating Cluster Explanations

Naively, generating the best explanations w.r.t. the quality measures, as defined above, requires instantiating all possible explanations (i.e., with all possible predicates) and assessing their quality. Since this is infeasible in practice, we suggest an efficient algorithm based on *generalized frequent itemsets mining* (gFIM), first introduced in [14]. While originally gFIM is used to mine frequent item categories in a given transactional dataset, we use it in our context to find conjunctions of predicates with a high coverage. We then further refine the explanation candidates and calculate the skyline w.r.t. the coverage, separation and conciseness.

In more detail, given a dataset D and cluster labels CL, the user is interested in explaining cluster $c \in C$ with quality thresholds $\theta_{cov}, \theta_{sep}$, and θ_{con}. Our algorithm for generating the skyline of cluster explanations works as follows:

1. Creating a taxonomy of binned data. For numeric attributes, binning is essential for finding promising values for *range* predicates (e.g., *'age' between 16 and 48*). Cluster-Explorer supports any binning method that splits a value domain of an attribute to non-overlapping intervals: $I = \{[l_1, m_1], [l_2, m_2], \dots\}$ s.t. $l_i < m_i \wedge m_i < l_{i+1}$. In particular, our implementation contains tree-based and 1d clustering [16] as well as equal-width binning.

Next, we unify all intervals obtained by the different binning methods into a *taxonomy*, which is passed to the gFIM algorithm, as explained below, to find a candidate set of cluster explanations. The taxonomy depicts a hierarchy of the intervals according to the partial order $[l_i, m_i] < [l_j, m_j] \iff l_i > m_j \wedge l_i < m_j$.

2. Applying gFIM to get a promising set of explanation candidates. A gFIM algorithm, as described in, e.g., [14] mines frequent *generalized* itemsets. Given a taxonomy of items and their containing categories, a transactional dataset, and a support threshold θ, the gFIM algorithm mines sets of items or categories which appear with a frequency of above θ. As described below, applying the gFIM algorithgm on X^c, the subset of the data points that are labeled by $CL(x) = c$, results in a candidate set of explanations with coverage and conciseness thresholds above θ_{cov} and θcon.

First, we convert X^c to a transactional format, s.t. each row x_i is a multiset of *key-value* items (a_j, x_{ij}). Each such item is connected to all corresponding categories $[l, m]$ in the taxonomy of attribute a_j, s.t $l \leq x_{ij} \leq m$.

Next, we run the gFIM algorithm on this transactional encoding of X^c, denoted T^c, using θ_{cov} as the input *support threshold* while restricting the maximal itemset size to be $\frac{1}{\theta_{con}}$. The gFIM execution yields a set of all maximal frequent *generalized* itemsets, each containing a combination of items (a_j, x_{ij}) or intervals from the taxonomy of the form $(a_j, [l, m])$. We denote the output itemsets of the gFIM algorithm by $\mathcal{IS}^c$.

```
from  ClusterExplorer import Explainer
explainer = Explainer(adults_df, labels)
explainer.explain(cluster=0)
```

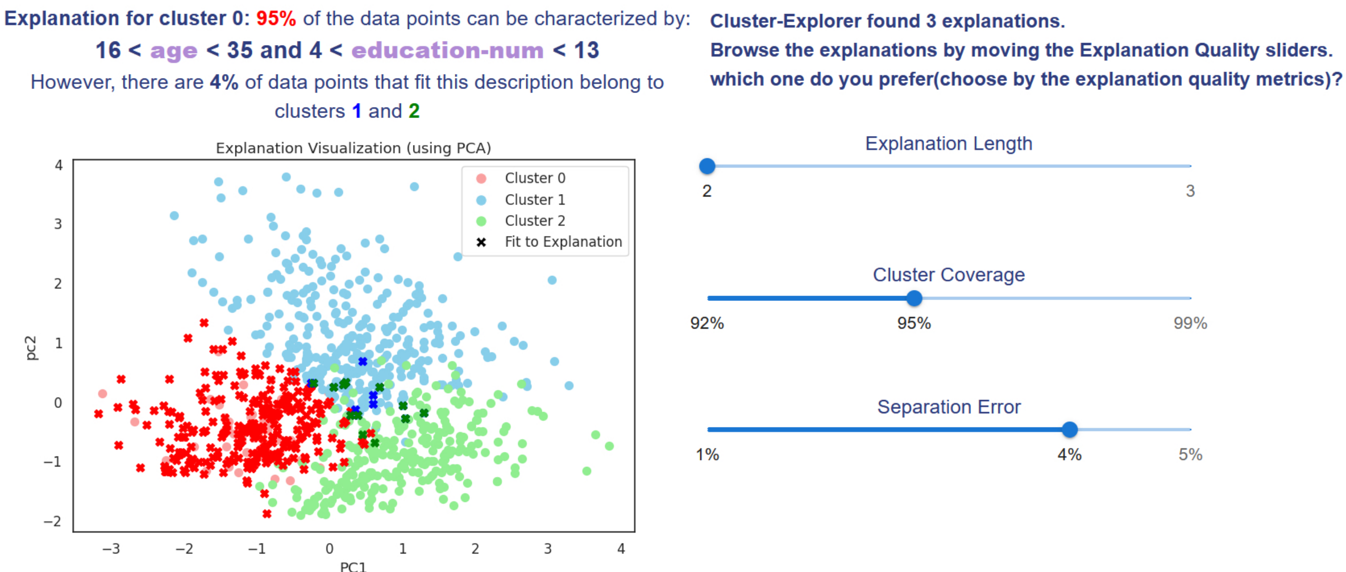

Figure 2: Example usage of Cluster-Explorer

We now make three important observations. First, see that each generalized itemset $IS^c \in \mathcal{IS}^c$ is equivalent to a cluster explanation E_c: Items correspond to *equality predicates* of the form $\langle a_j, \text{equals}, x_{ij}\rangle$ and intervals are *range* predicates of the form $\langle a_j, \text{between}, [l, m]\rangle$. Second, the coverage of E_c is higher than θ_{cov}, since the frequency of IS^c is higher than θ_{cov}. Third, E_c has a conciseness score higher than θ_{con}, because IS^c contains a maximum of $\frac{1}{\theta_{con}}$ items or categories.

3. Calculating explanations skyline. Taking the results $\mathcal{IS}^c$ of the gFIM algorithm, we first transform each of them to an explanation candidate E_c, as described above. As the gFIM algorithm ensures that the explanations have sufficient coverage and conciseness, we now need to filter out all explanation candidates with separation error *higher* than θ_{sep}, in order to obtain the set of explanation candidates that comply with all input criteria. We denote the set compliant explanation candidates by EX_θ. Last, to obtain only the *dominating* explanations for cluster c, denoted EX_c, we employ the skyline operator [3] on the candidate explanations set EX_θ. Namely, $EX_c = SKYLINE_{E \in EX_\theta}(Coverage, SepError, Conciseness)$.

As described below, we create a coherent, captioned visualization for each explanation $E_c \in EX_c$, and allow the user to browse through them in an interactive UI (See Figure 2).

3 USER INTERFACE & DEMONSTRATION

Implementation & UI. We implemented Cluster-Explorer as a Python library (See [15] for code), allowing users to seamlessly obtain cluster explanations in their current analytical workflows, without using an external interface or software.

To use Cluster-Explorer, the user first initializes an Explainer Object which takes as input the dataset as a Pandas [17] Dataframe alongside corresponding cluster labels. The latter can be the results of any clustering pipeline built by the user (e.g., one-hot encoding and k-means clustering; dimensionality reduction with PCA and hierarchical clustering).

Then, the user specifies a cluster of interest. Cluster-Explorer then computes the dominating explanations (as detailed in Section 2.3) and presents them in an integrated, interactive web interface, as depicted in Figure 2.

The user can browse through the generated explanations via three sliders, one for each quality score described in Section 2.2: Explanation Length (Conciseness), Cluster Coverage, and Separation Error. When dragging one of the sliders to the right or to the left, Cluster-Explorer dynamically changes the explanation view (see the left-hand-side of Figure 2), and shows an explanation fitting to the desired score. The rest of the sliders are automatically set according to the scores of the presented explanation.

For each explanation, Cluster-Explorer generates a natural language description that comprises its predicates – but also indicates the explanation's misses and wrong hits from other clusters.

To further analyze the explanation's quality, Cluster-Explorer also generates a corresponding visualization, which highlights, on top of the two-dimensional illustration of the clusters, all data points that are covered by the explanation.

Interactive Demonstration. We invite the conference participants to interact with Cluster-Explorer, and examine its usefulness. First, the audience is invited to upload their own dataset or select one from our collection, which contains numerous Kaggle [4] datasets. The participant can then select a ready-made clustering pipeline, which include several stages of data pre-processing, feature engineering, and the application of a clustering algorithm. Once cluster labels are obtained, users can explore cluster explanations generated by Cluster-Explorer.

REFERENCES

[1] Amina Adadi and Mohammed Berrada. 2018. Peeking inside the black-box: a survey on explainable artificial intelligence (XAI). *IEEE access* 6 (2018), 52138–52160.

[2] Adult Income Dataset (UCI). 2022. https://archive.ics.uci.edu/ml/datasets/Adult/.

[3] Stephan Borzsony, Donald Kossmann, and Konrad Stocker. 2001. The skyline operator. In *Proceedings 17th international conference on data engineering*. IEEE, 421–430.

[4] Kaggle Community. 2023. https://www.kaggle.com.

[5] Sanjoy Dasgupta, Nave Frost, Michal Moshkovitz, and Cyrus Rashtchian. 2020. Explainable k-means and k-medians clustering. *arXiv preprint arXiv:2002.12538* (2020).

[6] George H Dunteman. 1989. *Principal components analysis*. Number 69. Sage.

[7] Charles A Ellis, Mohammad SE Sendi, Eloy Geenjaar, Sergey M Plis, Robyn L Miller, and Vince D Calhoun. 2021. Algorithm-Agnostic Explainability for Unsupervised Clustering. *arXiv preprint arXiv:2105.08053* (2021).

[8] Nave Frost, Michal Moshkovitz, and Cyrus Rashtchian. 2020. ExKMC: Expanding Explainable k-Means Clustering. *arXiv preprint arXiv:2006.02399* (2020).

[9] Scott M Lundberg and Su-In Lee. 2017. A unified approach to interpreting model predictions. *Advances in neural information processing systems* 30 (2017).

[10] Tim Miller. 2019. Explanation in artificial intelligence: Insights from the social sciences. *Artificial intelligence* 267 (2019), 1–38.

[11] Daniel Müllner. 2011. Modern hierarchical, agglomerative clustering algorithms. *arXiv preprint arXiv:1109.2378* (2011).

[12] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 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*. 1135–1144.

[13] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2018. Anchors: High-precision model-agnostic explanations. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 32.

[14] Ramakrishnan Srikant and Rakesh Agrawal. 1995. Mining generalized association rules.

[15] Sariel Tutay and Amit Somech. 2023. https://github.com/analysis-bots/cluster-explorer.

[16] Haizhou Wang and Mingzhou Song. 2011. Ckmeans. 1d. dp: optimal k-means clustering in one dimension by dynamic programming. *The R journal* 3, 2 (2011), 29.

[17] Wes McKinney. 2010. Data Structures for Statistical Computing in Python. In *Proceedings of the 9th Python in Science Conference*, Stéfan van der Walt and Jarrod Millman (Eds.). 56 – 61. https://doi.org/10.25080/Majora-92bf1922-00a

[18] Rui Xu and Donald Wunsch. 2005. Survey of clustering algorithms. *IEEE Transactions on neural networks* 16, 3 (2005), 645–678.