# Understand your clusters: a link between the clustering data and explanatory metadata

Maciej Mozolewski[1], Samaneh Jamshidi[2], Szymon Bobek[1] and Grzegorz J. Nalepa[1]

[1]*Jagiellonian Human-Centered Artificial Intelligence Laboratory (JAHCAI) and Institute of Applied Computer Science, Jagiellonian University, Cracow, Poland*

[2]*Center for Applied Intelligent Systems Research (CAISR), Halmstad University, Halmstad, Sweden*

**Abstract**

In this preliminary work, we present an approach for augmentation of clustering with Natural Language Explanations. In clustering there are 2 main challenges: a) choice of a proper, reasonable number of clusters and b) cluster analysis and profiling. There is a plethora of technics for a) but not so much for b), which is in the general a laborious task of explaining obtained clusters. In this work we propose a method that aids experts in cluster analysis by providing iterative, human-in-the-loop methodology of generating cluster explanations. In a convincing example, we show how the process of clustering on a set of *objective variables* could be facilitated with textual *metadata*. In our case images of products from online fashion store are used for clustering. Then product descriptions are used for profiling clusters.

**Keywords**

XAI, clustering, metadata, Natural Language Processing, explanations, narratives

## 1. Introduction

Categorization is one of the ways how humans describe the world. To classify means to notice that some phenomena differ from each other. And more importantly how they differ. Finally, one is giving names to those different classes of entities. Clustering in machine learning in essence is a no different process.

Clustering methods are standard in many fields of human prosperity. In the advent of an ever-increasing amount of data, we use tools to automate the process, as manual clustering is too laborious. Indeed, there are many statistical, machine learning, and deep learning algorithms. The difficulty arises to convince end-users that derived clusters make sense. Will it be clustering sensor data in Industry 4.0 or behaviors of consumers, results need to be actionable. We believe that this is impossible if we do not explain what the algorithm has learned. It also follows that the process is iterative. We continually hypothesize about the significant differences between classes, then test results by looking at classified objects.

From our expertise in Industry 4.0 and e-commerce, we often see distinctions between 2 types of data. There are *objective data* and the "subjective" data or *metadata*. For instance in e-commerce popular approach for recommendations is Collaborative Filtering. It is based on finding users similar to each other in terms of interactions with products. Thus, *objective data* are shoppers' behaviors. Categories, titles, and descriptions of products are *metadata*, which are usually the result of the joint work of many e-store employees. For rolling steel factories, predictive maintenance models are derived mainly from *objective sensory data*, like temperature, force, etc. Factory accounting data are *metadata*.

The more *objective data* are, the more they are suited for modeling the phenomena, be it physical, business, sociological or psychological in nature. *Metadata* are more suited for explaining the model to the user, convincing her or him, and prompt to make decisions and actions based on this knowledge. They are more prone to error, because of their conventionality and subjectivism, but they speak to humans.

In whis work we propose a method that allows for clustering dataset with *objective data*, and explain differences between clusters with *metadata*. We use XAI methods to explain differences between clusters using *metadata* which are perfectly understandable by humans, but may not be of enough

*IJCAI-ECAI 2022, the 31st International Joint Conference on Artificial Intelligence and the 25th European Conference on Artificial Intelligence, July 23–29, 2022 Messe Wien, Vienna, Austria*

EMAIL: m.mozolewski@doctoral.uj.edu.pl (M. Mozolewski); samaneh.jamshidi@hh.se (S. Jamshidi); szymon.bobek@uj.edu.pl (S. Bobek); grzegorz.j.nalepa@uj.edu.pl (G. J. Nalepa)
URL: https://github.com/mozo64 (M. Mozolewski)
ORCID: 0000-0003-4227-3894 (M. Mozolewski); 0000-0001-7055-2706 (S. Jamshidi); 0000-0002-6350-8405 (S. Bobek); 0000-0002-8182-4225 (G. J. Nalepa)





CEUR Workshop Proceedings (CEUR-WS.org)

quality to perform valid clustering. The selection of most interpretable *metadata* is iterative and human-guided. In our example we show how image-based clustering can be enhanced with textual description of clusters. We argue that such an approach can lead to better utilisation of *metadata* for cluster analysis purposes, which results in better understanding of clusters which is the final goal of every clustering task. Furthremore, it allows for checking the consistency between two or more possible instance reporesentations (image and text) which might be crucial in domains that rely on both (e.g. e-commerce).

The rest of the paper is organised as follows. In Section 2 we present current research in the area of interactive clsutering and human-guided clsuter analysis. The description of our method along with use-case studies is given in Section 3. Finally, we conclude our work and show perspectives of its further development is presented in Section 4.

## 2. Related works

Explainable AI approaches have become particularly important, and although most work is generally focused on supervised learning, some works have been done to explain clusters. One of the most common methods for understanding clustering methods is visualization. By using low-dimensionality embedding and displaying them in two or three dimensions, one can get an overview of the clusters and their data. However, these visualizations are not always understandable and explainable.

The decision tree is one of the inherently interpretable algorithms. So one common way to explain models is to use decision trees. Nevertheless, the critical point for explaining the decision tree is its depth because decision trees with high depth no longer are interpreted, so we must pay attention to the depth of the tree produced. Using a small decision tree to divide a dataset into k clusters provides explainable clusters, but this approach has a trade-off between being explainable and accuracy. IMM algorithm [1] approximates k-means and k-median clustering by a threshold tree with k leaves. While ExKMC [2] uses a threshold tree to provide an explainable k-mean clustering in which the number of tree leaves is more than the number of clusters.

Besides visualization or providing some conditions on features, using text data is reasonable to explain to users. [3] uses the captions of the images along with images to create a more discriminative classification. In addition, they use this *metadata* to provide language explanation and generate a text description for each class.

Another approach is given in [4], where authors present a toolkit for conformance checking between expert knowledge with automatic clastering. The differences in expert-based clustering and automated clustering is justified with XAI methods and the process is iterative. However, the explanations are not human-guided, and the expert has no impact on the way they are generated. In particular it is not possible to provide additional *metadata* for explanations, nor modify the set of conccepts that are used for explanations.

## 3. Cluster analysis with *metadata*

In this section, we will show how our method could be applied to real case scenarios. We choose an example from the e-commerce field because the authors have experience working in this industry. Specifically, we work with online stores to provide them, among others, with recommendations of products to their end-users (clients).

In real-life scenarios, data about products are stored in product catalogs in shop databases, and most often exchanged with so-called product feeds (XML documents). We used a public dataset from Kaggle[1]. This dataset in terms of content resembles a product feed for an online store of a medium size product catalog. It consisted of 44000 products with category labels, titles, and images. For the code accompanying this example see GitHub repository[2].

As been said before, we treat images as *objective data*. We used embeddings of images obtained via MobileNetV2 [5]. The fully-connected layer at the top of the network was disregarded because we were not interested in the classification done by the model. The output of the final layer of the model was of length 20480. We used Singular Value Decomposition (SVD) with normalization to reduce the dimensionality of embeddings, leaving at least 90 percent of the variance.

In this section, we will present tools dedicated to data scientists who would like to perform clustering. We propose a 2-step clustering loop, which consists of k-means clustering and textual explanations of clusters. Data preparation also could be performed more than once, if needed. For the sake of simplicity, we call it "step 0" in this work.

[1] See: https://www.kaggle.com/datasets/paramaggarwal/fashion-product-images-small

[2] See: https://github.com/mozo64/xai-survey/blob/sklearn-text-clustering-example/src/example1-clustering-products-fashion.ipynb

### 3.1. Data preparation

The method requires 2 types of data: *objective* and *metadata* as defined in the previous section. In "step 0" method provides users with helper functions to prepare both types of data. For *objective data* there is a function that performs a reduction of dimensionality via SVD followed by normalization. It works on any numerical data, which could be as well as one-hot variables and continuous real values (floats). User sets percent of explained variance left after SVD reduction. The optimal count of new dimensions could be determined automatically by our algorithm. This is done by probing different dimension counts with scipy.optimize package, so the user does not need to do this manually. As for *metadata* which are textual, there are wrappers built on top of SpaCy[3] and NLTK[4] libraries. Users can contact text columns, lemmatize, remove stopwords and perform Tf-Idf vectorization. For numerical *metadata*, we found a way to incorporate them into textual explanations. For instance, the year could be recoded as the label "year2022", which will be easily interpreted along the pipeline. Other numerical variables could be recoded to low/medium/high bins, based on quartiles. Finally, the user constructs the "Pipeline" object and initializes it with 2 datasets: *objective* and *metadata*.

### 3.2. Assistance in clustering

The first step corresponds to running the unsupervised algorithm. Typically, the person who performs the analysis starts with the dilemma of choice of the number of clusters. It can be resolved with her/his background knowledge, intuition, practicality prerequisites, or just a trial and error approach. To give our users a hint in this regard, we use the T-SNE 2-dimensional projection of the data. At the moment, this is a solely visual clue. It is depicted on Figure 1. If data have an underlying structure, points representing observations will cluster, which would be observed on the chart. As T-SNE on massive data could be resource intensive, the default is to run this process on random subsample and cache results. Additionally, users can apply textual labels to the T-SNE chart, plotted on a subsample of data, to avoid cluttering the chart. Labels could represent the most important pieces of *metadata*, like the label, observation id, and summary of description. The next clue is derived from the silhouette score on a plot in the Figure 2. The range of the number of clusters to be tested is provided in accordance

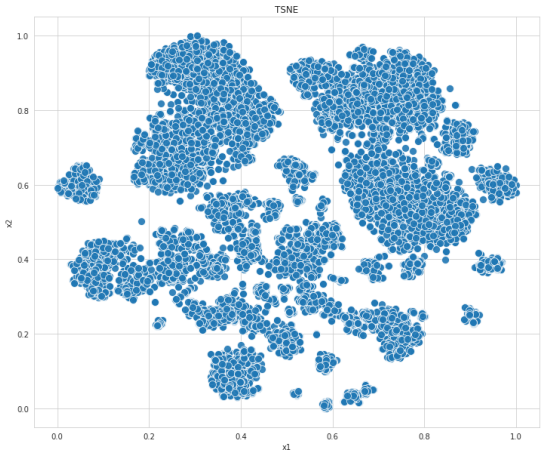


**Figure 1:** Preliminary visualisation of *objective data* in 2-D projection

with the previous clue. To speed-up computations, this plot could be obtained on a random subset, and results are cached for further reference. For now, the user interprets the plot on her/his own. Finally, clustering with k-means is performed on all observations. Visualization with T-SNE is presented, this time with clusters colored different colors, which is depicted on Figure 3.

### 3.3. Interactive explanations

The second step is to explain clusters, so the person who performs data analysis can assess the result. We would like to give users agency in refining explanations. Thus, we provide her or him with the possibility to influence explanations by extending stopwords with his own terms. On the other hand, we initialize the whitelist with keywords like "year2022", defined in "step 0". Then we use the Tf-Idf vectorizer, taking into account the aforementioned lists. Vectors are used for training decision tree classifiers. The size of the list of additional terms is under the control of a user. She or he can change it and interactively observe the changes in a Figure 6. Moreover, there are presented example observations on Figure 4, word clouds describing clusters on Figure 5 and LIME explanation for one instance of *metadata* Figure 7.

The last stage is a plot of the word cloud of each cluster, using the same Tf-Idf vectorizer. Plots are accompanied by examples of observations if data scientists should define visualization function and pass into Pipeline class API. Moreover, there is random observation with the LIME explainer for a

[3]See: https://spacy.io/
[4]See: https://www.nltk.org/

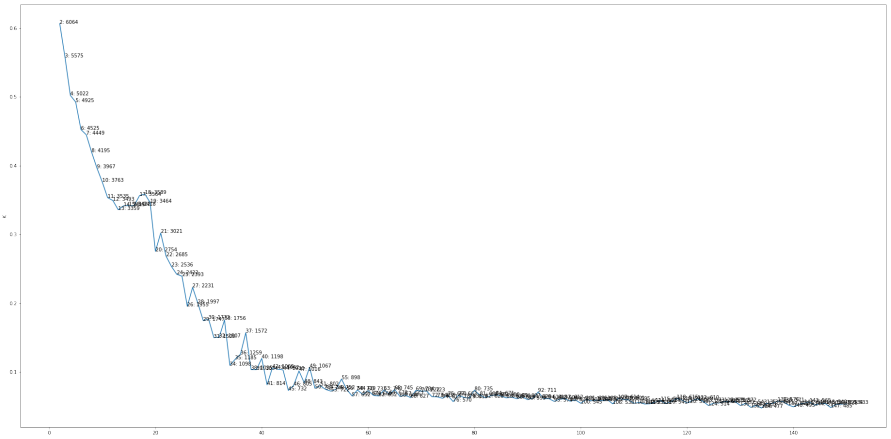

**Figure 2:** Silhouette score with cluster counts and values


**Figure 3:** Preliminary visualisation of *objective data* in 2-D projection

**Figure 4:** Example products

given class.

## 4. Summary

In this work we presented the method that allows for explaining clusters with concepts that could be more human-readable that data which was used


**Figure 5:** Word cloud for a given category

as sn input to clustering algorithm. We based our method on the observation that different types of data are suitable in different degrees to clustering and explaining tasks. We demonstrated the feasibility of our approach on the e-commerce example, where images were treated as input for clustering and textial descriptions of images as basis for cluster descriptions.

In future work we would like to improve our method with several extensions. We will focus on automatically proposing number of clusters based both on embedding features with technics similar to T-SNE and metrics like silhouette score. We want to test other techniques clustering than k-means. For instance hierarchical clustering could more suited in e-commerce, where taxonomies of products are multilayer. Word clouds could be replaced with topic analysis with Latent Dirichlet Allocation or techniques derived from Natural Language Generation. Another interesting direction is to construct explanations with other modalities, like visual, by

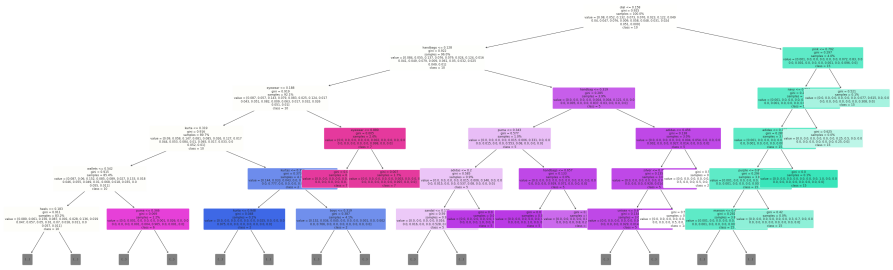

**Figure 6:** Decision tree classifier which explains how clusters differ in terms of *metadata*


**Figure 7:** LIME explanation for observation in category "handbags"

something more sophisticated than presenting example images. It could be done for instance with image captioning.

## Acknowledgments

The work of Szymon Bobek has been additionally supported by a HuLCKA grant from the Priority Research Area (Digiworld) under the Strategic Programme Excellence Initiative at the Jagiellonian University (U1U/P06/NO/02.16).

The work of Samaneh Jamshidi was supported by CHIST-ERA grant CHIST-ERA-19-XAI-012 funded by Swedish Research Council.

## References

[1] S. Dasgupta, N. Frost, M. Moshkovitz, C. Rashtchian, Explainable $k$-means and $k$-medians clustering, arXiv preprint arXiv:2002.12538.

[2] N. Frost, M. Moshkovitz, C. Rashtchian, Exkmc: Expanding explainable $k$-means clustering, arXiv preprint arXiv:2006.02399.

[3] L. A. Hendricks, Z. Akata, M. Rohrbach, J. Donahue, B. Schiele, T. Darrell, Generating visual explanations, in: European conference on computer vision, Springer, 2016, pp. 3–19.

[4] S. Bobek, M. Kuk, J. Brzegowski, E. Brzychczy, G. J. Nalepa, Knac: an approach for enhancing cluster analysis with background knowledge and explanations, CoRR abs/2112.08759. arXiv:2112.08759.
URL https://arxiv.org/abs/2112.08759

[5] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, L.-C. Chen, Mobilenetv2: Inverted residuals and linear bottlenecks (2019). arXiv:1801.04381.