FacetRules: Discovering and Describing Related Groups

Lebna V Thomas* Jiahao Deng† Eli T Brown‡

DePaul University

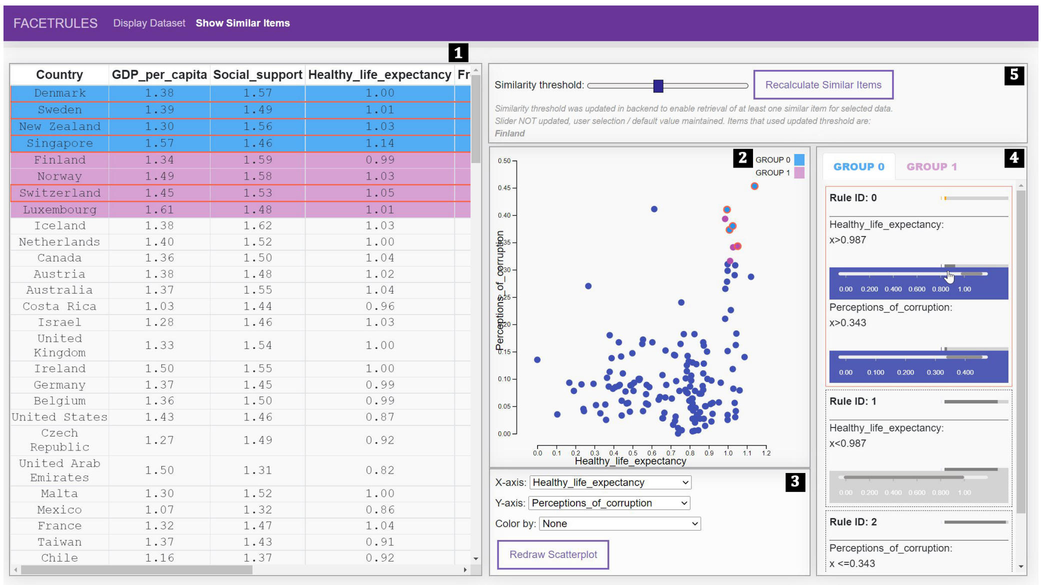


Figure 1: A picture of *FacetRules* displaying groups formed from user selected seeds and rules describing them. UI components are: (1) table with retrieved group items colored in pink and blue, (2) configurable scatterplot providing a 2D representation of the data, (3) controls for the scatterplot, (4) interactive display for rules describing the groups highlighted in the table and scatterplot, and (5) controls to adjust the similarity threshold and relearn groups.

Abstract

Domain experts, owing to their knowledge and experience, develop an intuition for patterns in the data. They may know, for example, certain points of interest. However, they may not know exactly how to characterize what makes these data special. In our work, we start from these points of interest as seeds to derive groups of similar points automatically based on surrounding cluster structure. To aid characterization, we provide descriptive rules as an interpretable model to describe the groups. We explain this technique and present a prototype to demonstrate with a usage scenario how the technique helps the user with data exploration by discovering and describing groups in the data.

Index Terms: Human-centered computing—Visualization—Visualization theory, concepts and paradigms

1 Introduction

Discovering and understanding groups in datasets is an important part of sensemaking. Domain expertise and intuitive understanding of a dataset helps in recognizing certain data points of interest, but without being able to describe the reasoning for this preference, the generalizability is limited.

*e-mail: lebnathomas@gmail.com
†e-mail: petershenxdf@gmail.com
‡e-mail: eli.t.brown@depaul.edu

The expert may know examples of a certain pattern, but not how to define it well enough to discover more. It is not possible to verify the intuition or say if the perceived pattern exists without a bigger set of examples. If we can expand the set of examples and find other similar data points, we can identify and describe the pattern better. Inspired by a real-world problem of a collaborator, we consider the problem of starting from the few seeds of interest, retrieving other similar data points to form groups and then characterizing the groups with descriptive rules. We propose using hierarchical clustering to find similar data points, and letting the user control the similarity between their points of interest and the retrieved groups. Finally, descriptive rules are automatically learned to characterize these groups by what ranges of which variables are most discriminatory.

In this paper, we explore and demonstrate this technique with a prototype. The prototype provides an interface for specifying data points of interest and iteratively relearning groups, and thus rules, with different similarity thresholds. Specifically, our contributions are: (1) the FacetRules technique of starting from seed points of interest and faceting the results of a similarity search by creating distinct, automatically extended and described groups of related points, (2) the FacetRules prototype implementation demonstrating the FacetRules technique, and (3) a usage scenario showing how this tool assists in discovering and describing groups in data. We provide additional analysis of the FacetRules technique with machine learning experiments and feedback from the domain experts who inspired the work.

2 Related Work

Existing work uses various approaches for retrieval of similar items, as in example-based search in which instead of formulating a query, the user provides examples to search with and expects similar items as a result. Lissandrini et al. [12] provide a detailed list of existing example-based methods, including techniques used for different data types and algorithms. S4 [16] and the work by Weiss et al. [27] take examples from the user and discover the queries that would have resulted in the provided examples through reverse engineering. Another tool, FlatFinder [25], uses example critiquing. It shows items based on user provided preferences and then iteratively updates the results based on user feedback. In contrast to the exiting methods, FacetRules finds similar items by leveraging the substructure in the data identified by hierarchical clustering. The substructure in hierarchical clustering gives us the ability to learn groups for different levels of similarity between items.

Another related area is interactive cluster refinement, as in ClusterSculptor [14], Cluster Sculptor [3]) and Clustrophile 2 [6]. FacetRules is not meant to craft a clustering through user feedback, but rather we use hierarchical clustering [29] to retrieve items similar to user-specified items based on the local cluster sub-structure (see Sect. 3).

Rules help characterize patterns in data and are an interpretable model [**?**]. They have been used in tools like RuleMatrix [13] for helping users understand machine learning models, and have been implemented with different machine learning backends [10, 19, 24]. Our rules generation component uses an adaptation of DRIL [4], which uses Quinlan's decision-tree-based rule algorithm, C5.0 [18]. The C5.0 algorithm includes optimizations for rule list generation when compared with other decision tree algorithms like CART [23] and ID3 [17]. This algorithm was also shown in DRIL [4] to have a significant performance advantage compared to SBRL [28] for an interactive context.

FacetRules is a Human-In-The-Loop (HIL) technique, iteratively relearning using interactions on the visual interface as parameters for the backend machine learning algorithm. The HIL system, its framework, benefits, and other concepts have been discussed in numerous works [1, 7, 11, 21, 22].

3 FacetRules

The motivation for FacetRules comes from a conversation with a collaborator, a university online-learning management team with data about class offerings at the university. From their expertise and due to their access to additional information, they know examples of classes that are doing well. But they do not know the full set of good classes, nor can they define what constitutes a class that is doing well or what the signs of trouble are. They want to expand their examples of good classes, understand what works, and eventually use that understanding to identify classes that need help in real time. From their application, we recognize the need to discover groups of similar data points to a small seed set, and to provide tools for understanding what defines these groups.

The first part of the goal is to extend a small selection with similar items. We propose using hierarchical clustering and leveraging its innate similarity structure to find similar data and group them simultaneously. For the second component, characterizing these groups, we propose using automatic descriptive rules (human-interpretable models [8]) to provide combinations of variable ranges that distinguish group members from the rest of the dataset.

3.1 Hierarchical Clustering for Groups Formation

The FacetRules technique takes a small point selection from the user and forms groups around these seed points. We use hierarchical agglomerative clustering [29] to learn groups based on a selection. The configuration details of the algorithm are mentioned in Sect. 5. Hierarchical clustering builds a tree structure of relationships between all data points. Starting from each data point as an individual cluster, it iteratively merges the closest clusters until all the data is merged into one single cluster. The resulting tree representation of clustering records the substructure of each cluster down to the individual data points.

This tree can be examined to find the relationships between data points from different levels of the tree. For example, for a given pair of data points, their closest common ancestor in the tree corresponds to the merge step of the clustering when they became part of the same cluster. The algorithm also keeps track of the distance between the clusters formed at each merge. The higher the merge distance, the less similar the clusters being merged. This feature is commonly used to make determinations like the appropriate number of clusters in the data. It serves our purposes well in this work because it can be used as a natural stopping point for expanding data from user selections. For each selected data point, x, we traverse through the tree to find the largest subtree including x such that all nodes are within a user-specified threshold of distance from each other. That subtree is returned as the group of items similar to x. These groups are simply merged when they overlap, because if two user-selected nodes are close enough relative to the threshold, they will be in the same subtree anyway.

The hierarchical clustering's recording of cluster substructure provides a natural organization of which selected items are kept separated vs. joined. If we chose related points based on similarity alone, we could miss close, relevant items. Fig. 2 provides an example. The data are shown as blue dots and the red circles are used to encode the hierarchical clustering with a containment metaphor: each circle contains a cluster, and the nested structure can be seen with nested circles. Imagine a query point and a proximity neighborhood around it, shown with the purple circle, which would be the equivalent of querying for nearest neighbors. Three points are included in the purple circle, but the immediate neighbors of its outer points are missed. Using the hierarchy to expand the selection would result in a bigger group that includes these points, which could be closely related to the query due to the way they fit in to the local structure of the data.

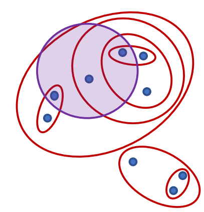
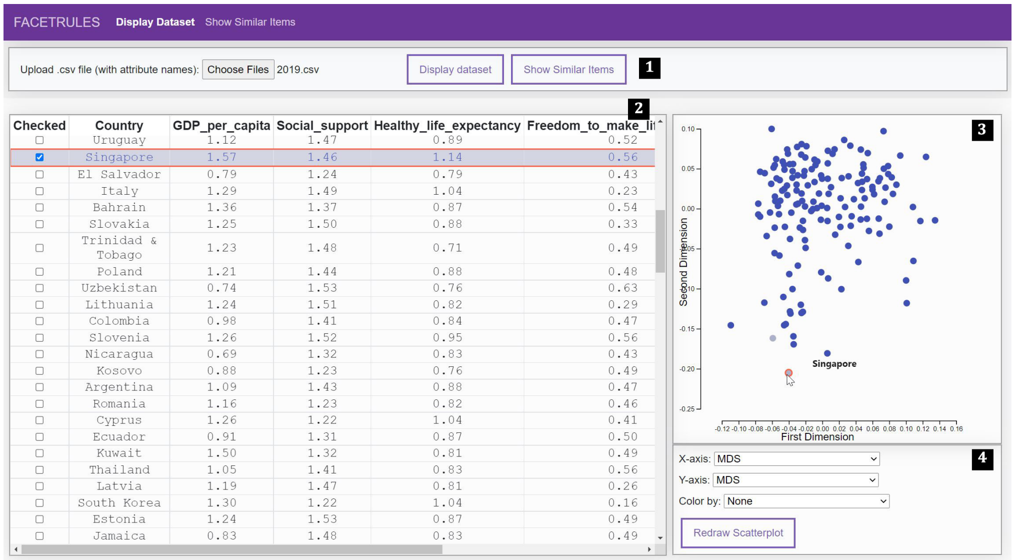


Figure 2: Illustration of hierarchical clustering and how nearest-neighbor relates to our approach of finding similar items using the HAC Tree.

One possible drawback is that if the user selects a single outlier with no neighbors except a large cluster, our method could add many points to the selection because the outlier does not join the neighboring cluster until late in the clustering process. However, a single outlier will not likely be sufficient to generate rules on its own, and providing rules may be counterproductive since there is no robust pattern. An analysis with participants to judge the effectiveness of our current group expansion compared to distance-based, or other methods, remains future work.

We use the merge distance at the $n/2$th merge, where n is the size of the data, as the default threshold for group retrieval in our tool. Since there are $n-1$ merges total, at the halfway point, most data points that have reasonably close neighbors would have merged to form a cluster. This is a desirable default so that when forming groups, most data points selected will return at least one other close neighbor, and at the same time the groups are showing only the similar items instead of expanding too far. The tool automatically increases the threshold to retrieve at least one similar item when there is none within the specified threshold, ensuring a group retrieval even for anomalies.

A slider in the UI lets the user update the threshold value. The user-system collaboration ensures that the tool always returns a group for the user, but leaves the user in control of dictating preference for group size. As the user increases the threshold, the algorithm walks further up the tree. It picks a cluster with a higher merge



(a) Initial Display Dataset screen. UI components are: (1) controls to upload, display dataset and show similar items, (2) table displaying dataset items, (3) scatterplot showing a 2D representation of the data and (4) controls for scatterplot.

(b) Scatterplot from Usage Scenario plotting two attributes and colored by a third.

Figure 3: FacetRules UI Components, as used in the Usage Scenario (Sect. 4.1)

distance, formed later in the tree formation process by the merging of sub-clusters. This results in fewer groups each with more items. Conversely, decreasing the threshold will return smaller groups and in some cases, reveal sub-groups within a retrieved group.

3.2 Rule Generation for Each Group

Each time a new set of groups is learned, or a group is extended by updating the threshold, rules are generated to describe each retrieved group compared to the rest of the dataset. The rules generation component first labels the items in the group with class 1 and the rest of the dataset with class 0. The labelled dataset is then provided to a supervised, tree-based rules generating algorithm by Quinlan [20]. The algorithm learns rules to differentiate between the in-group and outside-group classes so that the rules describe what makes in-group items different from the rest of the data (i.e., to describe the group). Each rule consists of one or more *conditions* that specify the attribute value ranges of the class being described. The rules, in the raw form as they are generated by the algorithm, are not ideal for directly presenting to a non-expert user. The preparation of the rules for the interface, like parsing the learner's output, removing duplicate clauses and sorting positive rules proceeds as in Cao et al. [4] and with the same performance characteristics. In the case of high-dimensional data, if a group requires many variables to describe it accurately, it may result in a large attribute list. However, since the technique optimizes for a parsimonious representation, unneeded attributes will be ignored and the description will use few variables when possible.

4 Prototype and Design by Usage Scenario

In this section, we discuss the interface of the FacetRules prototype. We use a usage scenario to describe the interface features in detail, while explaining design considerations[1].

4.1 Usage Scenario

Amaya is researching factors that influence citizen perception of their country. She is using the world happiness dataset from Kaggle with data about countries for the year 2019 [15]. Amaya is interested in two countries: Singapore and Finland, where citizens have a positive perception about their country. She wants to find other similar countries and understand the contributing factors.

4.1.1 Display Dataset

She uploads the dataset into FacetRules and sees the first mode, *Display Dataset*, with a screen similar to Fig. 3a. The table provides a detailed view showing attribute values for each country and the scatterplot shows the relationships in the data in 2D. The control panel for the scatterplot provides the option to pick between three dimension reduction techniques: MDS, T-SNE or UMAP. Or instead, she can use dataset attributes for the x and y axes. She can also choose to color the scatterplot by values of an attribute. Since no projection from high dimensions to 2D can be perfect, we give the user options. These are easily ignored for users who are unfamiliar with projection types, but may be appreciated by those who are. The ability to map data attributes to axes or color helps once attributes of interest are known in later steps.

Amaya keeps the default MDS view on the scatterplot. She works from the table. Hovering over the row for Singapore highlights the corresponding dot on the scatterplot. On the scatterplot, she hovers over other dots around Singapore, looking for similar countries. Mouseover causes the table to highlight countries, showing the name and making the attribute values readily available. Now she is ready to expand her countries of interest and selects Singapore and Finland on the table and hits *Show Similar Items*.

4.1.2 Show Similar Items

The prototype interface switches to its other mode, *Show Similar Items*, to display the two groups it has returned, as seen in Fig. 1. Group 0 is countries similar to Singapore, colored in blue. Group 1 is countries similar to Finland, colored in pink. There is a message under the similarity threshold slider stating that the backend had to increase the threshold in order to find similar countries for Finland. However, the slider was not updated as Singapore's group was retrieved using the default threshold of 0.5. The threshold can get adjusted in the backend from the user's preference, during the group expansion process to prevent the frustration of manually increasing the threshold to find groups for isolated data points. If only one

[1] Video of prototype walk-through is included in supplemental materials.

group is discovered, the threshold slider will be adjusted to reflect this because a lower value is not useful. However, when more than one group is formed, if the threshold had to be automatically updated and the new thresholds are different for the groups, we do not move the slider. We leave it at the user's selection and make it plain to the user by writing a message right below the slider explaining the adjustment including which of their selected points required threshold adjustment to build a suitable group.

Amaya now explores the discovered groups and the rules that describe them. The two groups are shown in the same colors, blue and pink, across the table, scatterplot and rule list. In the rules tab for each group, there is a description of what differentiates that group of points from the rest of the data. Each rule has a box containing its *conditions* on individual attributes, e.g., *Healthy life expectancy* > 0.987. The condition is listed and a range bar shows the rule's range in context of the full data. Rules that describe points that are in the group, *positive rules*, use blue in their range display. Rules that exclude points from the group, *negative rules*, use gray.

Amaya looks at the the positive rules for Group 0. The group retrieved for Singapore has *Healthy life expectancy* higher than 0.987 and *Perception of corruption*[2] higher than 0.343. Hovering over the rule *Rule ID: 0* for Group 0 highlights the data points the rule represents both in the table and on the scatterplot with a red outline. Countries in Group 0 have high *Healthy life expectancy* and *Perception of corruption* according to the rules. This makes Amaya wonder about the relation between these two variables. Amaya changes the x-axis and y-axis on the scatterplot to see *Healthy life expectancy* against *Perception of corruption*, as shown in Fig. 1 (3). The countries in this group also have higher *GDP per capita*, so Amaya wants to see how GDP is correlated with *Healthy life expectancy* and *Perception of corruption*. She uses *GDP per capita* as the attribute to *Color by* on the scatterplot. As seen in Fig. 3b, she is able to see that the countries with higher *Perception of corruption* and *Health life expectancy* also have high *GDP per capita*. She explores the group for Finland similarly. The tool has learned that for Group 1, *Perception of corruption* is between 0.31 and 0.343.

Now Amaya is curious what other countries are similar to Singapore and Finland at higher thresholds. She iteratively increases the threshold and hits *Recalculate Similar Items*. At a high threshold, the tool merges group 0 and group 1 into a single group including both the countries. The scatterplot colors the dots representing the new group and new rules are shown for this group. She recognizes that her initial hypothesis that these two countries are similar is reasonably well grounded in the data because with a group of eight countries, they are included together. She also understands much better what particular characteristics these countries have, as well as how this group breaks down to two smaller, related groups with slightly different properties.

5 Implementation

The prototype was built using the LIHCA software platform [5] which includes a Python backend running a Flask server, and a front-end Javascript API. The back and front ends communicate through Javacript APIs. The backend handles the machine learning and data storage functionalities. Group formation functionality is implemented using agglomerative hierarchical clustering from scipy [26]. The prototype supports both numerical and categorical data. We choose the distance function options in scipy for clustering based on the data type of the attributes. For non-categorical data, we use the Ward variance minimization algorithm for calculating distance between clusters and the euclidean distance function. For categorical data, we use the nearest point algorithm and Jaccard's distance. The group formation component has the same performance constraints and data limitations as hierarchical clustering. The clustering can be performed ahead of time, streamlining interaction speed. We adopt code provided by Cao et al's DRIL [4] for generating rules.

6 Evaluation With ML Experiments

We evaluated the FacetRules technique of expanding a seed selection and describing it using rules by conducting experiments on two datasets with simulated user selections. We used a subset of the Dow Jones Index dataset [2] and a preprocessed FIFA players dataset [9]. The Dow Jones Index dataset has weekly data for 30 stocks across two quarters. For our experiment, we used data from one week, with 30 instances and 14 attributes. The FIFA players dataset was preprocessed to remove missing values and duplicates, convert string values (e.g., net worth) to numerical values, and remove irrelevant attributes. We picked the players whose contract was valid at least until 2021, resulting in 1272 instances and 36 attributes. We simulated seed point selection with three different thresholds for both the datasets. We used the thresholds 0.5, 0.7 and 0.9, the tool's default threshold and two higher values to observe performance as groups grow. We ran the experiment with each data item in the dataset as an individual seed. We suggest users include more than one point, but the number of combinations of 2 is much larger, and single points are sufficient for the experiment. We recorded properties of the data points, numbers of items in the groups, numbers of rules generated, rule properties, and the performance metrics accuracy, precision and recall for the rules[3]. We use recall as the performance metric to determine the quality of rules because it measures how many of the actual positive classes the model was able to correctly predict. The labels for learning rules are predominantly negative since rules are learned for a group against the rest of the data. We are particularly interested in the ability of our technique to detect positive cases, i.e., the group items, so recall is the appropriate performance measure.

With the experiment, we wanted to evaluate if users can expect accurate groups and rules from their seed points. We assess how well the learning works based on relevant properties of the data (isolation scores) and parameters provided by the tool (threshold). For the FIFA dataset with around 1.3k items, we found that small groups (less than 5 items) generate rules less often than bigger groups. In our test with *single point* seeds, users got rules 42 percent of the time. When they did get rules, they were good quality, with high recall shown in Fig. 4 (a). Recall was positively affected by the threshold value since there were bigger groups at higher thresholds resulting in better rules and more cases where valid rules could be generated, shown in Fig. 4 (a). The contour plot shown in Fig. 4 (b) illustrates that the density of points in the neighbourhood of the selection positively affects recall. The isolation score on the x-axis measures how isolated a selected point is from other neighboring points. Isolation score is negatively associated with its neighborhood's density [30].

Figure 4: Plots from our machine learning experiments (see Sect. 6): (a) histograms faceted by threshold show the distribution of recall of the rules for threshold values, (b) contour plot of the relationship between recall (y-axis) of the rules generated and isolation score of the seed point for the threshold 0.9.

[2]This *Perception of Corruption* variable actually represents *trust*, so higher is better.

[3]The raw experimental results and graphs are included in the supplemental materials.

7 Expert Feedback

We conducted a tool evaluation session with the domain experts who inspired this work to gather feedback about the FacetRules technique and the usability of the prototype. The participants were three experts (P1-P3) from the university learning management team, each with excellent domain knowledge in different aspects of class offerings. First, we gave them an overview of how the prototype works, while answering questions[4]. Then they used the prototype to explore a preprocessed subset of their class offerings data. They shared their screens with us and were encouraged to think aloud or ask questions when necessary. After using the prototype, they answered some questions about their experience.

All the participants believed that the tool gave them interesting insights, some unexpected, about the classes they explored. All of them started by inspecting attribute values for classes and then selecting for expansion either ones that stood out or that were already familiar. The scatterplot's MDS view helped them discover items similar to their seeds before learning groups. Overall, the groups formed made sense to them. Looking at unexpected groupings, the rules and table helped them quickly understand how courses were related. In one case, the grouping and domain knowledge helped P1 explain a pattern beyond what was captured in the data. Looking at classes from different disciplines grouped together, P1 recognised that the classes had a common instructional designer. "I didn't expect to see [those classes] from such different disciplines be so similar in how they use d2l[5]. That to me was very interesting ... although they do have the same support person in our office. So maybe there is like a link there, underlying," P1 noted. For an unexpected group, P3 noted that, "There could be pedagogical or instructional elements to that. That would make it more useful to put those courses together."

After the groups were generated, all the participants used the scatterplot to check if items in the groups were close to each other and the rules helped them understand why classes were grouped together. For P3, rules helped confirm intuitions. When rules were not generated for groups with three or fewer elements, P1 and P3 expanded groups by increasing threshold. After learning how threshold value affects group size, P3 decreased the threshold and discovered sub groups within a retrieved group.

Participants demonstrated varying levels of understanding and engagement with different components, but all of them expressed their belief in the utility of the technique and the tool. They discovered groups and patterns in their data that they would not have otherwise found. They also understood the patterns discovered and how the classes grouped together were similar. Their wishlist for new features included the ability to bookmark items of interest, export generated groups, rules and scatterplots, and UI features for sorting and filtering based on attribute values. Future work includes incorporating their feedback into a tool that matches this workflow.

8 Conclusion

In conclusion, this paper presents a technique, FacetRules, which takes a user selection of data points, expands the selection to a set of related groups, and then automatically characterizes the groups with descriptive rules. We introduce a prototype to demonstrate this process, and use it to present a usage scenario. In the example scenario, we demonstrate how this technique can assist a user in augmenting points of interest and learning to characterize related data with descriptive rules. Our ML experiments show that rules learned for the generated groups were generally good quality, with high recall. Feedback from domain experts suggest that the technique helped them gather better insights.

[4] The questions asked are included in supplemental materials.

[5] d2l is the Learning Management System used at the university

References

[1] E. T. Brown, A. Endert, and R. Chang. Human-machine-learner interaction: The best of both worlds. In *Proceedings of the CHI Workshop on Human Centred Machine Learning (HCML)*, 2016.

[2] M. S. Brown, M. J. Pelosi, and H. Dirska. Dynamic-radius species-conserving genetic algorithm for the financial forecasting of dow jones index stocks. In P. Perner, ed., *Machine Learning and Data Mining in Pattern Recognition*, pp. 27–41. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.

[3] P. Bruneau, P. Pinheiro, B. Broeksema, and B. Otjacques. Cluster Sculptor, an interactive visual clustering system. *Neurocomputing*, 150:627–644, 2015. Special Issue on Information Processing and Machine Learning for Applications of Engineering Solving Complex Machine Learning Problems with Ensemble Methods Visual Analytics using Multidimensional Projections. doi: 10.1016/j.neucom.2014.09.062

[4] F. Cao and E. T. Brown. DRIL: Descriptive rules by interactive learning. In *2020 IEEE Visualization Conference (VIS)*, pp. 256–260, 2020. doi: 10.1109/VIS47514.2020.00058

[5] F. Cao, D. Scroggins, L. Thomas, and E. T. Brown. LIHCA: A human-in-the-loop software platform. In *Workshop on Machine Learning from User Interaction for Visualization and Analytics (MLUIVA), In Proceedings of IEEE VIS*, 2018.

[6] M. Cavallo and C. Demiralp. Clustrophile 2: Guided Visual Clustering Analysis. *IEEE Transactions on Visualization and Computer Graphics*, 25(1):267–276, 2019. doi: 10.1109/TVCG.2018.2864477

[7] A. Endert, W. Ribarsky, C. Turkay, B. Wong, I. Nabney, I. Díaz Blanco, and F. Rossi. The state of the art in integrating machine learning into visual analytics. *Computer Graphics Forum*, 3 2017.

[8] A. A. Freitas. Comprehensible Classification Models: A Position Paper. *SIGKDD Explor. Newsl.*, 15(1):1–10, Mar. 2014. doi: 10.1145/2594473.2594475

[9] K. Gadiya. FIFA 19 complete player dataset, 2018. Retrieved 20 FEB 2021 from https://www.kaggle.com/karangadiya/fifa19.

[10] R. C. Holte. Very simple classification rules perform well on most commonly used datasets. *Machine learning*, 11(1):63–90, 1993.

[11] D. Keim, G. Andrienko, J.-D. Fekete, C. Görg, J. Kohlhammer, and G. Melançon. Visual Analytics: Definition, Process, and Challenges. In *Proceedings of the IEEE Conference on Information Visualization*, pp. 154–175. Springer Berlin Heidelberg, 2008.

[12] M. Lissandrini, D. Mottin, T. Palpanas, and Y. Velegrakis. Example-based search: A new frontier for exploratory search. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR'19, p. 1411–1412. Association for Computing Machinery, New York, NY, USA, 2019. doi: 10.1145/3331184.3331387

[13] Y. Ming, H. Qu, and E. Bertini. RuleMatrix: Visualizing and Understanding Classifiers with Rules. *IEEE Transactions on Visualization and Computer Graphics*, 25(1):342–352, Jan 2019. doi: 10.1109/TVCG.2018.2864812

[14] E. J. Nam, Y. Han, K. Mueller, A. Zelenyuk, and D. Imre. ClusterSculptor: A Visual Analytics Tool for High-Dimensional Data. In *Proceedings of the 2007 IEEE Symposium on Visual Analytics Science and Technology*, VAST '07, p. 75–82. IEEE Computer Society, 2007. doi: 10.1109/VAST.2007.4388999

[15] S. D. S. Network. World Happiness Report, 2019. retrieved 13 DEC 2020 from https://www.kaggle.com/unsdsn/world-happiness.

[16] F. Psallidas, B. Ding, K. Chakrabarti, and S. Chaudhuri. S4: Top-k Spreadsheet-Style Search for Query Discovery. In *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*, SIGMOD '15, p. 2001–2016. Association for Computing Machinery, New York, NY, USA, 2015. doi: 10.1145/2723372.2749452

[17] J. R. Quinlan. Induction of decision trees. *Machine Learning*, 1(1):81–106, Mar 1986. doi: 10.1007/BF00116251

[18] J. R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1993.

[19] R. L. Rivest. Learning decision lists. *Machine learning*, 2(3):229–246, 1987.

[20] Rulequest Research. Information on see5/c5.0, April 2019. Retrieved

from https://www.rulequest.com/see5-info.html August 24, 2020.

[21] D. Sacha, M. Sedlmair, L. Zhang, J. Lee, D. Weiskopf, S. North, and D. A. Keim. Human-centered machine learning through interactive visualization: review and open challenges. In *In Proceedings of European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN)*, 2016.

[22] P. Y. Simard, S. Amershi, D. M. Chickering, A. E. Pelton, S. Ghorashi, C. Meek, G. Ramos, J. Suh, J. Verwey, M. Wang, et al. Machine Teaching: A New Paradigm for Building Machine Learning Systems. *ArXiv*, 2017.

[23] D. Steinberg and P. Colla. CART - Classification and Regression Trees. Salford Systems, 1997.

[24] A. K. Tung. Rule-Based Classification, 2009.

[25] P. Viappiani and B. Faltings. Design and Implementation of Preference-Based Search. vol. 4255, pp. 72–83, 10 2006. doi: 10.1007/11912873_10

[26] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. J. Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. J. Carey, İ. Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020. doi: 10.1038/s41592-019-0686-2

[27] Y. Y. Weiss and S. Cohen. Reverse Engineering SPJ-Queries from Examples. In *Proceedings of the 36th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems*, PODS '17, p. 151–166. Association for Computing Machinery, New York, NY, USA, 2017. doi: 10.1145/3034786.3056112

[28] H. Yang, C. D. Rudin, and M. Seltzer. Scalable bayesian Rule Lists. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, pp. 3921—-3930. ICML, 2017.

[29] M. L. Zepeda-Mendoza and O. Resendis-Antonio. *Hierarchical Agglomerative Clustering*, pp. 886–887. Springer New York, New York, NY, 2013. doi: 10.1007/978-1-4419-9863-7_1371

[30] Y.-L. Zhang, L. Li, J. Zhou, X. Li, and Z.-H. Zhou. Anomaly Detection with Partially Observed Anomalies. In *Companion Proceedings of the The Web Conference 2018*, WWW '18, p. 639–646. International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE, 2018. doi: 10.1145/3184558.3186580