SHORT-PAPER

PY-SHARQ: A Holistic Python Library for Explaining Association Rules on Relational Data

HADAR BEN-EFRAIM, Bar-Ilan University, Ramat Gan, Tel Aviv District, Israel

S. B. DAVIDSON, University of Pennsylvania, Philadelphia, PA, United States

AMIT SOMECH, Bar-Ilan University, Ramat Gan, Tel Aviv District, Israel

Open Access Support provided by:

Bar-Ilan University

University of Pennsylvania

Published: 22 June 2025

Citation in BibTeX format

SIGMOD/PODS '25: International Conference on Management of Data
June 22 - 27, 2025
Berlin, Germany

Conference Sponsors:
SIGMOD

PY-SHARQ: A Holistic Python Library for Explaining Association Rules on Relational Data

Hadar Ben-Efraim
Bar-Ilan University
Ramat Gan, Israel
hadarib@mail.biu.ac.il

Susan B. Davidson
University of Pennsylvania
Pennsylvania, USA
susan@cis.upenn.edu

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

Abstract

Association rules are an important technique for gaining insights over large relational datasets consisting of tuples of elements (i.e., attribute-value pairs). However, it is difficult to explain the relative importance of data elements with respect to the rules in which they appear. This demo showcases SHARQ (ShApley Rules Quantification), a Shapley-based measure for quantifying an element's contribution to a set of association rules. SHARQ is implemented in a convenient Python library, PY-SHARQ, which supports various explainability scenarios for association rules—including importance analysis of individual elements, rules, and attributes.

CCS Concepts

• **Information systems** → **Association rules**; • **Mathematics of computing** → **Exploratory data analysis**.

Keywords

Explainability for Unsupervised Learning & Data Mining, Shapely Values

ACM Reference Format:
Hadar Ben-Efraim, Susan B. Davidson, and Amit Somech. 2025. PY-SHARQ: A Holistic Python Library for Explaining Association Rules on Relational Data. In *Companion of the 2025 International Conference on Management of Data (SIGMOD-Companion '25), June 22–27, 2025, Berlin, Germany*. ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3722212.3725125

1 Introduction

Rule-based pattern mining over large datasets consisting of tuples of elements (i.e. attribute-value pairs) is a popular tool in a data scientist's toolbox as it does not require training data and produces clear data patterns and insights. It is heavily used in many application domains such as E-commerce, cyber security and health.

Since rule mining tools often return thousands of association rules, various techniques have been developed to manage rule sets. Solutions include ranking rules by different interestingness functions [6] and visualization techniques [3]. However, none of these techniques help users understand the relative importance of *elements* with respect to the rules in which they appear.

Measuring an element's contribution to a rule set is an *explainability* problem, yet unlike in Machine Learning (ML) where there are a plethora of techniques to explain the results of a model (see [4] for a survey), to our knowledge no such framework exists to explain association rules mined from the data. Furthermore, generic measures of an element's contribution to a set of rules, e.g. based on the score of the most interesting rule that contains the element (denoted I_{Top}) or using causality-based notions such as influence [7] (denoted $Infl$) do not adequately differentiate between elements, as illustrated below. (I_{Top} and $Infl$ are described in Section 2.1.)

Example 1.1. A data analyst, Clarice, is examining the *Adults* dataset[1] which provides demographic information on individuals. She starts by mining the rules, a toy example of which is shown in Table 1. In this table, the rules are denoted by the elements on their left and right-hand sides (LHS and RHS). For example, r_3 is the rule (*Income*, *<50K*)→ (*age*, *31-41*). The last column of the table gives the rule's interestingness (IS) score [6], a common interestingness measure for association rules. Clarice examines the rules and wonders if all elements contribute equally to the rule set.

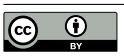
Clarice notices that element e_1, (relationship, unmarried), appears to be less important than others since if it is omitted from r_1 and r_4 then the same patterns still hold in r_2 and r_3, with roughly the same IS score. She also notices that r_3 has fewer elements than r_2, so each of its elements, e_5 and e_6, would appear to proportionally contribute more than e_2-e_4. She then applies I_{Top} and $Infl$ to see if there is a difference in element contribution to the rules. However, as shown in Table 2, all six elements have *identical* scores. □

We therefore developed in [2] a notion of *element contribution* to a set of rules called SHARQ, which captures the frequency of an element in the dataset as well as the variability in interestingness across rules of different lengths when the element is excluded. In doing so, it provides a finer measure of contribution than the generic measures mentioned earlier. As shown in the rightmost column of Table 2, SHARQ clearly differentiates between elements with respect to their contribution to the overall interestingness of the rules set: e_1 has a significantly lower score than the other elements, and e_5 and e_6 are higher than $e_2 - e_4$, confirming Clarice's intuition.

In this demo we present PY-SHARQ, a holistic python library for mining and explaining association rules on tabular data. PY-SHARQ facilitates the application of SHARQ, and augments it with numerous visual explanations based on the calculated SHARQ values. We invite the audience to choose a dataset, mine association rules, then analyze and explain them via PY-SHARQ. We will also demo three applications of SHARQ: element importance, attribute (feature) importance, and detecting redundant rules.

We briefly illustrate how PY-SHARQ is used for mining the rules and analyzing element importance, and will expand on the remaining applications in Section 3.

[1] https://archive.ics.uci.edu/ml/datasets/Adult/



SIGMOD-Companion '25, Berlin, Germany

ACM ISBN 979-8-4007-1564-8/2025/06
https://doi.org/10.1145/3722212.3725125

ID	Rule LHS	Rule RHS	IS score
r_1	(age, 44-53), (hours-per-week, 40-50), (relationship, unmarried)	(income, >=50K)	105
r_2	(age, 44-53), (hours-per-week, 40-50)	(income, >=50K)	102
r_3	(income, <50K)	(age, 31-44)	105
r_4	(income, <50K)	(age, 31-44), (relationship, unmarried)	70

Table 1: Rules and IS interestingness scores (Toy Ex.)

ID	Dataset Element	I_{Top}	$Infl.$	SHARQ
e_1	(relationship, unmarried)	1.05	0	−0.6
e_2	(hours-per-week, 40-50)	1.05	0	−0.05
e_3	(age, 44-53)	1.05	0	−0.05
e_4	(income, >=50)	1.05	0	−0.05
e_5	(age, 31-44)	1.05	0	4.6
e_6	(income, <50)	1.05	0	4.6

Table 2: Element contribution scores (Toy ex.)

Example 1.2. Clarice loads the full Adult dataset using the Pandas[2] library. She then imports PY-SHARQ, and uses it to mine and display the 71K resulting association rules (see Figure 1). Next, she uses PY-SHARQ to calculate the elements' SHARQ scores and visualize the top and bottom 5 elements (positive scores are in green and negative scores are in grey in Figure 2). The scores are provided alongside descriptive statistics for each element: its frequency within the data, and the number of rules it appears in, divided into three interestingness levels: High, Medium, and Low. Clarice immediately sees that the top five high-scoring elements are both frequent and appear mainly in highly interesting rules (*High Int.* category). In contrast, the bottom five elements are less frequent and mostly appear in less interesting rules (*Low Int.*). □

In the reminder of this demo paper we first give details of the SHARQ formula and optimizations (Section 2), then delve into additional application scenarios of PY-SHARQ for detecting redundant rules and gauging the importance of the dataset attributes based on SHARQ values.

2 The SHARQ Framework

We begin by describing our model and notions for rule-set interestingness and element contribution, and then describe the SHARQ formula and its optimized calculation.

2.1 Model and definitions

We assume that an association rules mining tool [1] has been applied to dataset D, with a set of attributes $\mathcal{A}$ and tuples T, generating the set of rules R_D. A rule $r \in R_D$ is denoted by $E_{LHS} \rightarrow E_{RHS}$, where E_{LHS}, E_{RHS} are disjoint sets of *elements*, i.e. attribute-value pairs $e = (a, v)$, where $a \in \mathcal{A}$ and value v which is a tuple $t \in T$ projected on attribute a. We use $\mathcal{E}(\cdot)$ to denote the elements of a rule, tuple or a data subset; and $attr(\cdot)$ to denote their attribute(s).

Interestingness of a rule set. A rule score function quantifies the interestingness of a rule $r \in R_D$. While PY-SHARQ allows users to specify any notion of interestingness, in this demo we use the IS score [6], which combines two well-known measures: *support* [1], which quantifies the frequency of the joint appearance of the rule's elements, and *lift* [6], which measures the independence deviation of elements on the left- and right-hand sides of the rule. The full IS score for a rule r is then defined by:

$$IS(r) \coloneqq \sqrt{support(r) \cdot lift(r)}$$

The interestingness of a *set* of rules $R \subseteq R_D$ is any aggregation of the individual scores. In our examples we use $I(R) \coloneqq max_{r \in R}$ score(r), however, PY-SHARQ provides users other aggregation operations such as summation, average, and top-k, and can take as input other measures for rule interestingness.

Measuring element contribution. Given a set of dataset elements $E \subseteq \mathcal{E}(D)$ and a set of rules $R \subseteq R(D)$, SHARQ measures the *contribution* of an element $e \in E$ to the interestingness of R, $I(R)$. There are several generic ways to quantify the contribution of an element e. For example,it could be the score of the most interesting rule that contains e, denoted as I_{TOP}, or the causality based influence [7] measure, defined in our context as the difference in the aggregative interestingness of R when removing the rules that contain e:

$$Infl(e) \coloneqq I(R) - I(R \setminus \{r | r \in R \wedge e \in r\})$$

However, these generic contribution measures do not adequately differentiate between elements as shown in Example 1.1. We therefore use a contribution measure based on the game theoretic notion of Shapley values [5] called SHApley Rules Quantification (SHARQ) [2], which captures the variability in interestingness across rules of different lengths when the element is excluded and gives a much finer measure of element contribution than other metrics.

2.2 The SHARQ Formula

The Shapley value [5] measures a *player*'s *contribution* to the *utility* of all possible *player coalitions*. In our setting, the "players" are elements in E and "coalitions" are sets of elements with disjoint attributes. That is, coalitions model rules and the contribution of an element is (roughly speaking) the difference in interestingness of rules in which they play a role and those in which they don't play a role. More precisely, given a set of elements $S \subseteq E$, let R_S denote the subset of rules of R that contain *exactly* the elements in S:

$$R_S \coloneqq \{r | r \in R \ \wedge \ \mathcal{E}(r) = S\}$$

The *utility* of an element coalition S is then defined as $I(R_S)$, i.e., the *aggregate* interestingness of R_S as defined previously. Due to this notion of utility, we consider only *valid* coalitions with respect to an element, i.e. those that can possibly form a rule when the element is included. Given an element e, and a set of elements $E \in \mathcal{E}(D)$, the set of *valid* coalitions are defined as:

$$C(e, E) \coloneqq \{S | \ S \subseteq E \ \wedge |attr(S)| = |S| \ \wedge \ attr(e) \notin attr(S)\}$$

Namely, all subsets of E that, together with element e, do not contain two elements of the same attribute (a tuple never contains two elements of the same attribute).

The SHARQ score of an element e in the context of E and R can now be defined as:

$$SHARQ_{(E,R)}(e) = \sum_{S \in C(e,E)} \frac{|S|!(|E| - |S| - 1)!}{|E|} \cdot \left(I(R_{S \cup \{e\}}) - I(R_S)\right)$$

[2] https://pandas.pydata.org/

```
#load any dataset using Pandas
adult_df = pd.read_csv('Datasets/adult.csv')

#Mine rules using PySharq
import pysharq as ps
adults_rules_set = ps.mine_rules(adult_df, binning=True, score_func=ps.funcs.is_score, support_threshold=0.05)
```

	lhs	rhs	support	lift	is_score
322	['gender_Male']	['capital-gain_(-1, 10000]', 'relationship_Husband']	0.399256	1.461180	0.763796
39	['gender_Male']	['relationship_Husband']	0.415622	1.461180	0.779293
20	['educational-num_(8, 12]']	['income_<=50K']	0.496812	1.084285	0.733952
614	['educational-num_(8, 12]', 'native-country_United-States']	['income_<=50K']	0.459192	1.080685	0.704445
...	...	...	...	...	...

71390 rows × 5 columns

Figure 1: Rules mining using PY-SHARQ

```
sharq = ps.Sharq(rules_set=adults_rules_set)
sharq.plt.element_bars(top_bottom_elements_num=5, verbose=True)
```

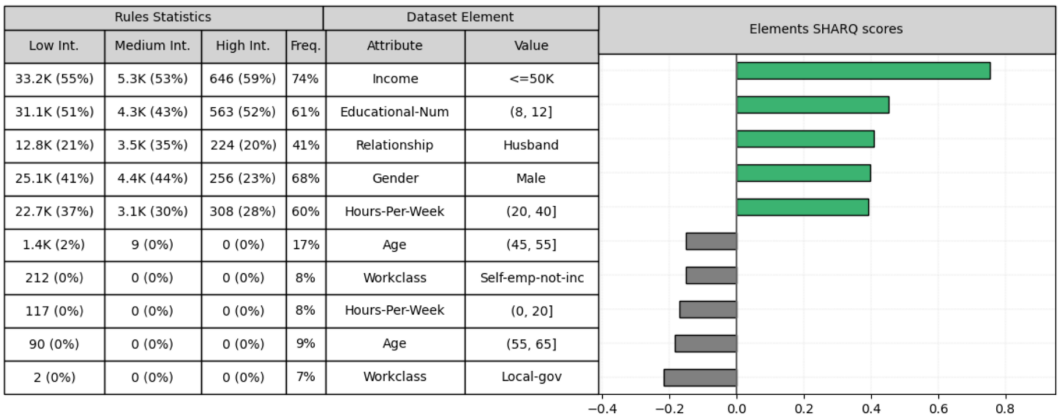


Figure 2: PY-SHARQ visual explanation of SHARQ element contribution scores alongside useful statistics

Optimizing SHARQ calculations. The cost of any application of Shapley values is determined by the number of *coalitions*, which is exponential in the number of *players*. In our context, the players are the dataset elements, which is often considerably larger than the number of attributes, and in many datasets can also surpass the number of tuples. It is therefore infeasible to calculate SHARQ scores even for small rule sets. In [2] we therefore devise an improved formula in which a substantial number of irrelevant element coalitions that do not affect the SHARQ score are pruned, and use this to develop efficient algorithms for calculating *exact* SHARQ scores for both a single element as well as for multiple elements together (while amortizing the cost of single calculations).

3 PY-SHARQ Demonstration

PY-SHARQ is implemented in Python 3.10. It uses Apriori[3] to generate the rules, and Pandas to store and manipulate the underlying data. We now describe the usage, and demonstration scenarios for PY-SHARQ.

3.1 Usage

PY-SHARQ contains several features that assist users in mining rules (such as built-in binning for numerical values and removal of infrequent categories in categorical data, rows and columns sampling) and provides numerous interestingness functions (such as Confidence, Lift, IS score, Conviction, Chi Squared, and more).

As depicted in Figure 1, to mine the rules the user uploads a dataset (using Pandas) and applies the PY-SHARQ `mine_rules` function. The resulting rules are displayed in a table, allowing the user to sort them according to score or filter by specific elements. To use SHARQ explanations, the user initializes a SHARQ object with the mined rules or a subset thereof (see Figure 2). Optionally, the user can specify which subset of elements to use (default is *all*), and how to aggregate the rule scores (default is *max*).PY-SHARQ then calculates the SHARQ score for the input rules. Once initialized, users can analyze the rules through the visual explanations generated by PY-SHARQ (detailed in Section 3.3).

[3]https://efficient-apriori.readthedocs.io/en/latest/

```
sharq.plt.attributes_heaetmap()
```

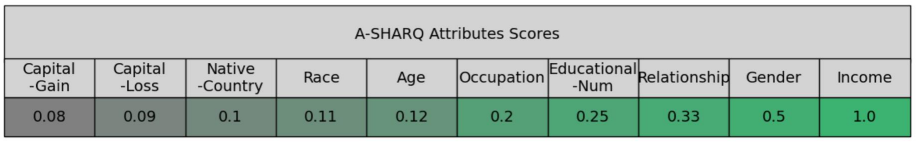


Figure 3: A-SHARQ attribute importance scores

3.2 Interactive Demonstration

We will demonstrate several different rule-mining scenarios and demonstrate the insights gained from the visual explanations produced by PY-SHARQ.

We will also provide a look under the hood and demonstrate the impact of using different rule interestingness measures, compare importance rankings with contribution measures and showcase the efficiency of our optimization methods in reducing coalition size and computation time.

3.3 Sample Applications Demo

We next describe three use cases for PY-SHARQ on rules mined from the Adult dataset. In the live demo, we will provide several other examples on different datasets from our collection.

Element Importance. As shown in Example 1.2, users can examine the importance of individual elements in the data. Figure 2 shows the top and bottom 5 elements ordered w.r.t. their SHARQ score. Using the `verbose` flag, PY-SHARQ also provides statistics for each element: the frequency of the element in the data, and the number of rules in which the element appears within three different rule groups: Low int (bottom third of the interestingness score range), Medium int (middle third) and High int (top third). We also specify in parentheses the proportion of rules that fall into each category.

Analyzing the SHARQ elements scores, we observe that the top five high-scoring elements are highly frequent in the data, each with a frequency exceeding 41%. Furthermore, these elements appear prominently in high scoring rules (more than 20%) and medium scoring rules (over 35%). In contrast, the bottom five elements are less common, appearing in less than 17% of the rules, and are predominantly found in low scoring rules.

Rule Importance. PY-SHARQ can also calculate the SHARQ scores for rules, which can be used to identify *redundant* rules that, despite their high interestingness scores, are similar to *shorter* rules with equivalent scores. Removing redundant rules helps narrow the user's focus to a more concise set of significant rules.

Rule scores are based on a normalized SHARQ element score, $\widehat{SHARQ}_{(E,R)}(e)$, where the SHARQ score is divided by the element's frequency rank (see [2] for details).

$$R\text{-}SHARQ_{(E,R)}(r) = \min_{e \in \mathcal{E}(r)} \widehat{SHARQ}_{(E,R)}(e)$$

namely, the normalized score of the rule's lowest-scoring element.

For example, the scores for rules 322, 39, 20 and 614 in Figure 1 are 0.008, 0.397, 0.452, and 0.031, respectively. Examining rule 322: (*Gender*,*Male*)⟶ (*Capital-Gain*, *-1-10K*), (*Relationship*, *Husband*) we see it that it obtains a very low R-SHARQ score of 0.008, while its IS score is high (0.76). This rule, intuitively, is redundant, as rule 39 is highly similar and obtains a higher IS score of 0.77, but does not contain the element (*Capital-Gain*, *-1-10K*). Correspondingly, rule 39 obtains a SHARQ score almost 50X higher. Rules 20 and 614 demonstrate a similar phenomenon.

Attribute importance. We also show how SHARQ can be used to gauge the importance of attributes w.r.t. the examined set of rules using an *attribute*-level SHARQ score:

$$A\text{-}SHARQ_{(E,R)}(a) = \frac{\sum_{e \in E_a \cap \mathcal{E}(R)} \widehat{SHARQ}_{(E,R)}(e)}{E_a}$$

where E_a denotes the elements subset that belong to attribute a. Intuitively, this is the mean *normalized* SHARQ score for the elements of attribute a that appear in at least one rule.

Figure 3 depicts the importance scores of sample attributes in the Adult dataset, sorted from lowest (left) to highest (right). Notice that the *Income* and *Gender* columns obtain the highest attribute scores, as both contain only high-importance elements.

Note that the *Income* and *Gender* columns obtain the highest attribute scores, as both contain only high-importance individual elements (see Figure 1). The column *Educational-num* obtains a lower score even though it contains the top-2 scoring element (*Educational-num*,*[8-12]*)) because the rest of the elements in this attribute obtain lower SHARQ scores.

The attributes importance scores can be used, e.g., to reduce the dimensionality of the data (by disregarding low-scoring attributes) when performing analytical tasks.

Additional use-cases. In the live demonstration, we will preview additional rule explainability use cases not covered here, such as feature interaction—examining how the importance of one feature changes with the values of another; the calculation of element *subset* importance; and SHARQ correlations.

Acknowledgments. This research was partially supported by the Israel Science Foundation grant 2121/22.

References

[1] Rakesh Agrawal, Ramakrishnan Srikant, et al. 1994. Fast algorithms for mining association rules. In *PVLDB*, Vol. 1215. 487–499.
[2] Hadar Ben-Efraim, Susan B Davidson, and Amit Somech. 2025. SHARQ: Explainability Framework for Association Rules on Relational Data. *PACMMOD* 3, 1 (2025), 1–25.
[3] Michael Hahsler and Radoslaw Karpienko. 2017. Visualizing association rules in hierarchical groups. *Journal of Business Economics* 87, 3 (2017), 317–335.
[4] Pantelis Linardatos, Vasilis Papastefanopoulos, and Sotiris Kotsiantis. 2021. Explainable ai: A review of machine learning interpretability methods. *Entropy* 23, 1 (2021), 18.
[5] L. S. Shapley. 1953. *17. A Value for n-Person Games.* Princeton University Press, Princeton, 307–318. https://doi.org/doi:10.1515/9781400881970-018
[6] Pang-Ning Tan and Vipin Kumar. 2000. Interestingness measures for association patterns: A perspective. (2000).
[7] Eugene Wu and Samuel Madden. 2013. Scorpion: explaining away outliers in aggregate queries. *PVLDB* 6, 8 (2013), 553–564.