Mining Association Rules via Feature Extraction over Similar Samples

Ofek Tabak (315854695) & Mika Tal (323820654)

Final project report for the Tabular Data Science course, Bar Ilan University

1 Abstract

In the domains of Machine Learning and Data Mining, association rules illustrate relationships between variables within extensive datasets. These rules identify the connections between items in any transaction involving a diverse range of items. Association rules are mainly measured by support and confidence, and are generated by a variety of algorithms, such as Apriori. In the algorithm, there are thresholds that should be fixed prior to the run, and those affect the outcome rules – it may produce trivial association rules or irrelevant ones.

In order to get informative and meaningful rules, we suggest an association rules mining technique where we first cluster similar samples, extract the most significant features of each cluster, and then mine each cluster for association rules independently.

To overcome this problem and raise the chances of the generated association rules to be informative, we propose an association rules mining technique where we first cluster similar samples and then mine each cluster for its association rules independently. We try to show that doing so results in mining of more insightful association rules. We hope to see more insightful association rules by performing this technique.

2 Problem Description

The Data Science pipeline element that we are trying to improve is Association Rules Mining. Association rules are a data mining technique that is widely used for learning and analyzing correlations among items in databases. They have two associated measures: **Support** and **Confidence**.

- Given a transaction set D and an association rule $R := A \Rightarrow B$, the **Support** of rule R is the percentage of transactions in D that contains both A and B:

$$Support(R) := P(A \cup B) = \sum_{T \in D} \frac{[I_{A \cup B \in T}]}{|D|}$$

- Given a transaction set D and an association rule $R := A \Rightarrow B$, the **Confidence** of rule R is the percentage of transactions in D containing A that also contain B:

$$Confidence(R) := P(B|A) = \frac{P(B \wedge A)}{P(A)} = \frac{Support(A \cup B)}{Support(A)}$$

The problem of mining association rules in algorithms such as **Apriori**, is that there are thresholds to fix for the **Support** and **Confidence** measures, and they affect the results greatly; Lower values will usually produce more rules but not all are at the same level of importance and some may be irrelevant or trivial, while higher values may prevent the production of important and meaningful rules. High support and confidence thresholds usually produce rules that are not necessarily relevant - rules that are most likely formed by dominating item sets and tend to be more obvious. Our goal is to create a tool that will improve and automate the process of mining the most promising and insightful association rules from a given dataset.

3 Solution Overview

As discussed, association rule mining algorithms typically produce a large number of rules, many of which are uninteresting or already known. Discovering new insights among these generated rules presents a challenge. One potential solution is to increase the support and confidence thresholds, which would result in fewer rules being generated. However, the issue with this approach is that as the support and confidence thresholds approach 1, the algorithm may produce too few rules, if any. Worse yet, the rules mined might be trivial, misleading, and probably lacking any new or interesting information.

We will attempt to mine association rules in a more sophisticated technique. Extracting association rules from the whole dataset might overlook interesting rules that appear in different segments of the data. Therefore, a preprocessing clustering method will separate the dataset into similar sample groups, which holds insightful information to mine. Mining each cluster independently will enable each cluster to express its associations, without interference from other clusters, which usually consist of different patterns.

In order to cluster the data with the Spectral clustering algorithm, we will preprocess the datasets by converting the columns to only numerical values, so we will perform one-hot encoding on the textual categorical columns, and label encoding on textual ordinal columns. We will then extract meaningful features from each cluster and convert them into transactional form. We will mine each cluster for its association rules using Apriori algorithm and return a list of all the rules mined that way. We will evaluate the results with Lift and Conviction measurements.

4 Experimental Evaluation

To measure our experiment we will use three measures: **Lift**, **Conviction**, and **Leverage**. Those will measure the importance of a rule.

- Given a transaction set D and an association rule $R := A \Rightarrow B$, the **Lift** of rule R measures how many times more often A and B occur together than expected if they were statistically independent (a **Lift** value of 1 indicates independence between A and B, and value greater then 1 indicated a strong dependency - therefore a meaningful rule):

$$Lift(R) := \frac{Support(A \cap B)}{Support(A) \cdot Support(B)} = \frac{Confidence(R)}{Support(B)} = \frac{P(B|A)}{P(B)}$$

  Lift is a symmetrical measure ($Lift(A \Rightarrow B) = Lift(B \Rightarrow A)$).

- Given a transaction set D and an association rule $R := A \Rightarrow B$, the **Conviction** of rule R is the ratio between the probability that A appears without B if they were dependent to the actual frequency of the appearance of A without B ($\overline{B}$ is the complement event for B):

$$Conviction(R) := \frac{1 - Support(B)}{1 - confidence(R)} = \frac{P(A) \cdot P(\overline{B})}{P(A \cap \overline{B})}$$

4.1 House Prices Dataset

The House Prices dataset is the same one used in class. This dataset holds 76 features, mostly categorical, and 1460 records. We followed the preprocessing shown in class while adding one hot encoding to each categorical column for running our clustering algorithm. After that, we used K-Means with $1 < k < 15$ to find the best K (using the knee method). We also tried spectral clustering but it did not seem to end in time. we followed that by removing small clusters since we believe they represent clusters that do not hold enough data to describe the distribution of the group. Now, all we had left to do is to select the "good" columns, treat the numerical data like in class, make the data transactional, and run the Apriori algorithm with relatively high support and confidence on the partitioned and unpartitioned data and compare the result.

4.1.1 Comparing the House Price Dataset results

In front of us is the Lift, Conviction and Confidence measures.

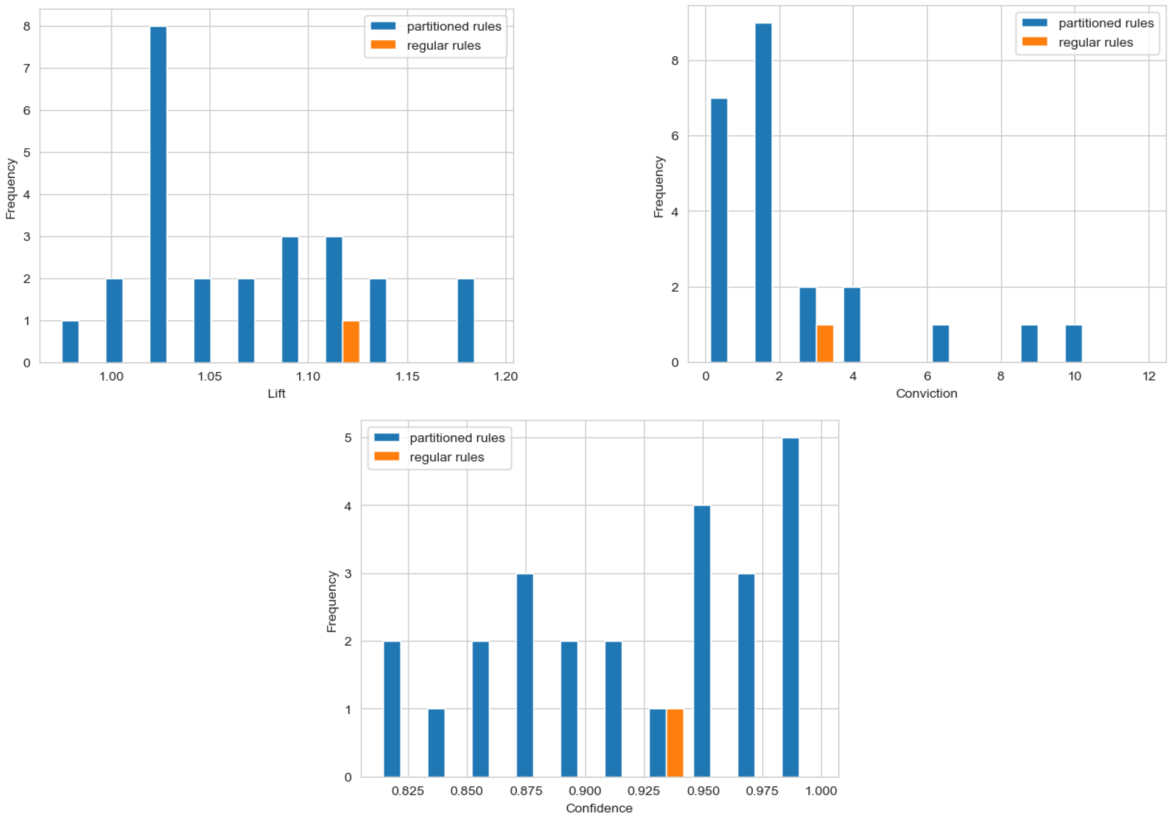

Figure 1: Lift, Conviction and Confidence graphs for (support=0.5, confidence=0.8) for regular datasets and (support=0.5, confidence=0.8) for partitioned dataset.

While examining these results, it is clear that our technique produced many more rules (25 to 1 to be exact). We could lower the confidence and support in the Apriori algorithm (as we did in other datasets where we had 0 rules), but as lowering the support might help us find more rare and interesting rules, it also introduces us to more noise, and lowering the confidence might make the results less reliable. We also must remember that using clustered data makes the support and confidence of each item in each group different so we assume our clustering was "good" and represents real groups in the data. Following that, using the Lift measure, we have found 23 interesting rules ($Lift > 1$) using our method and only 1 interesting rule using the Apriori algorithm over the unpartitioned dataset.

In terms of Conviction, we see that there are a more than a few partitioned found rules have a higher Conviction value, which means that the presence of the antecedent strongly indicates the presence of the consequent in our rules. We dropped 2 rules with an extremely high Conviction to make a plot more readable (around 70k conviction for each).

4.2 Titanic Dataset

The Titanic dataset provides information about passengers who were aboard the RMS Titanic when it sank after hitting an iceberg. it is a labeled dataset where the ’survived’ column represent the target label.

In the preprocessing, we dropped the ’Name’ and ’Ticket’ columns, treated missing data with the same approach as in the House Prices Dataset, performed one hot encoding for clustering, found the optimal K for the k-means clustering algorithm, removed small clusters, treated the numerical data like in class, and ran the Apriori algorithm on the partitioned and unpartitioned data.

4.2.1 Comparing the Titanic Dataset results

In front of us are the Lift, Conviction, and Confidence measures.

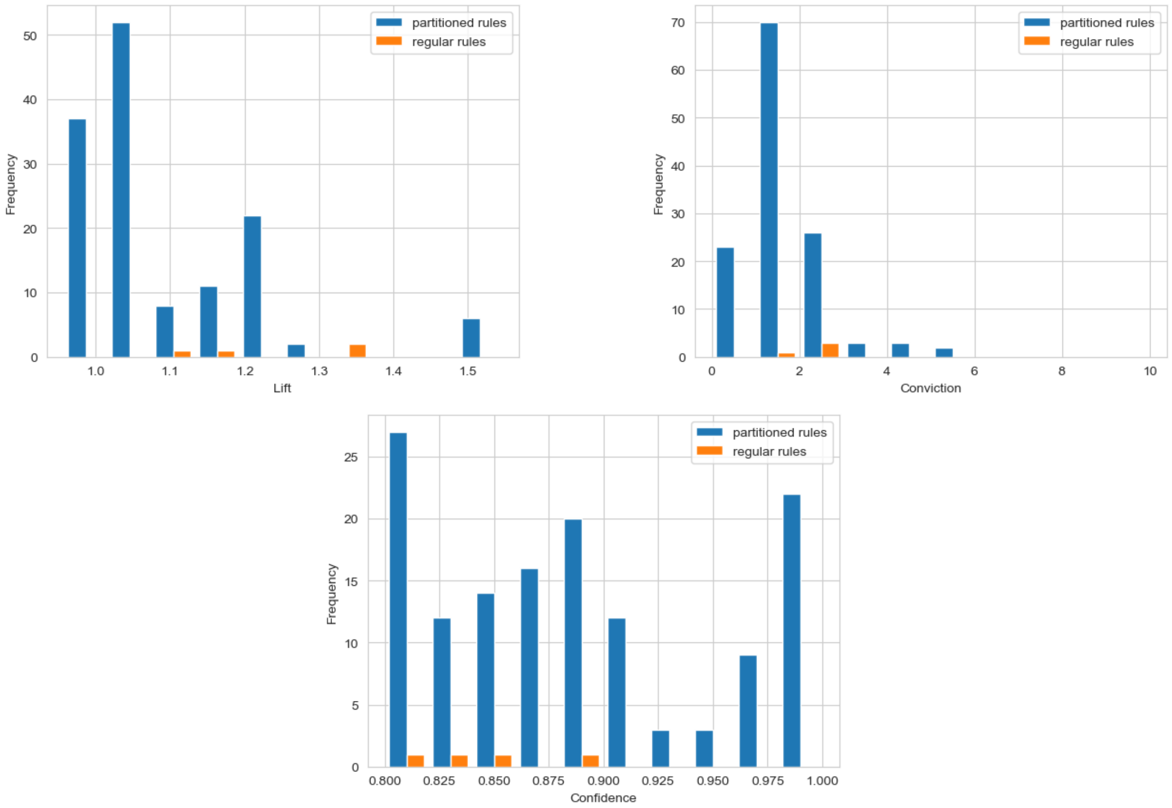

Figure 2: Lift, Conviction and Confidence graphs for (support=0.5, confidence=0.8) for regular datasets and (support=0.5, confidence=0.8) for partitioned dataset.

As displayed in the graphs, the Lift, Conviction and Confidence values of the rules resulting from the partitioned dataset are higher consistently than those resulting from the unpartitioned dataset.

4.3 Daily Transactions Dataset

The Daily Transactions dataset captures the details of transactions that occur on a daily basis.
It is a transactional dataset with the following 8 columns:’Date’, ’Mode’, ’Category’, ’Subcategory’, ’Note’, ’Amount’,’Income/Expense’, ’Currency’ preprocessing was similar to the others datasets but with one extra step, processing the Date column for clustering. We started out by doing feature extraction on the Date column and splitting it into 4 columns: day of the week (categorical), day (numerical), month (numerical), and year (numerical). Logically, the hours, minutes and seconds hold valuable data but since records with an older date were missing that data, we preferred not to use it. After that, we dropped the ’Date’ and ’Note’ columns, and followed the same procedure as in the other datasets.

4.3.1 Comparing the Daily Transactions Dataset results

In front of us are the Lift, Conviction, and Confidence measures.

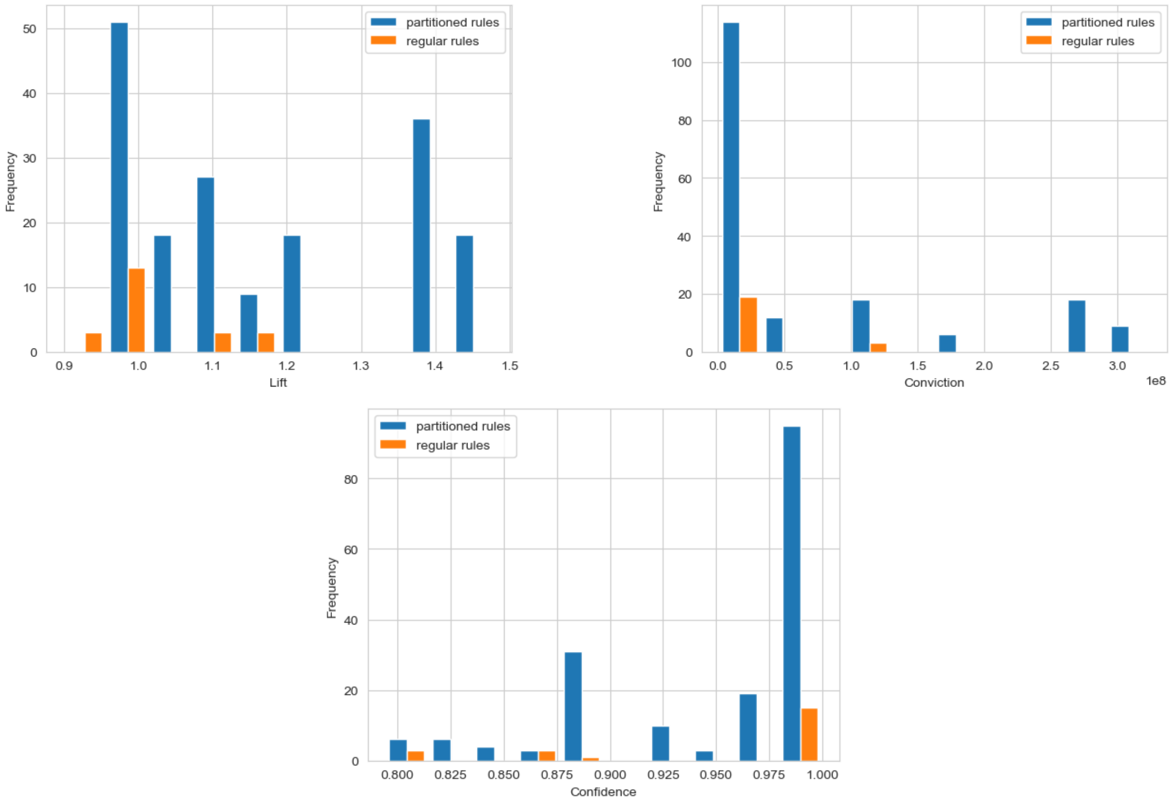

Figure 3: Lift, Conviction and Confidence graphs for (support=0.5, confidence=0.8) for regular datasets and (support=0.5, confidence=0.8) for partitioned dataset.

As displayed in the graphs, the Lift, Conviction and Confidence values of the rules resulting from the partitioned dataset are higher consistently than those resulting from the unpartitioned dataset.

4.4 Customer Dataset

The Customer dataset provides detailed information for segmenting customers based on various attributes. This dataset is useful for performing customer segmentation to identify different customer profiles and tailor marketing strategies accordingly.

In the preprocessing level, we treated missing data with the same approach as in the House Prices Dataset, performed one hot encoding for clustering, found the optimal K for the k-means clustering algorithm, removed small clusters and ran the Apriori algorithm on the partitioned and unpartitioned data.

For support=0.5 and confidence=0.8 the unpartitioned dataset yielded zero rules while the partitioned dataset yielded 30 rules. We lowered the thresholds to support=0.3 and confidence=0.6 in order to get more rules, and it resulted in 10 rules.

4.4.1 Comparing the Customers Dataset results

In front of us are the Lift, Conviction, and Confidence measures.

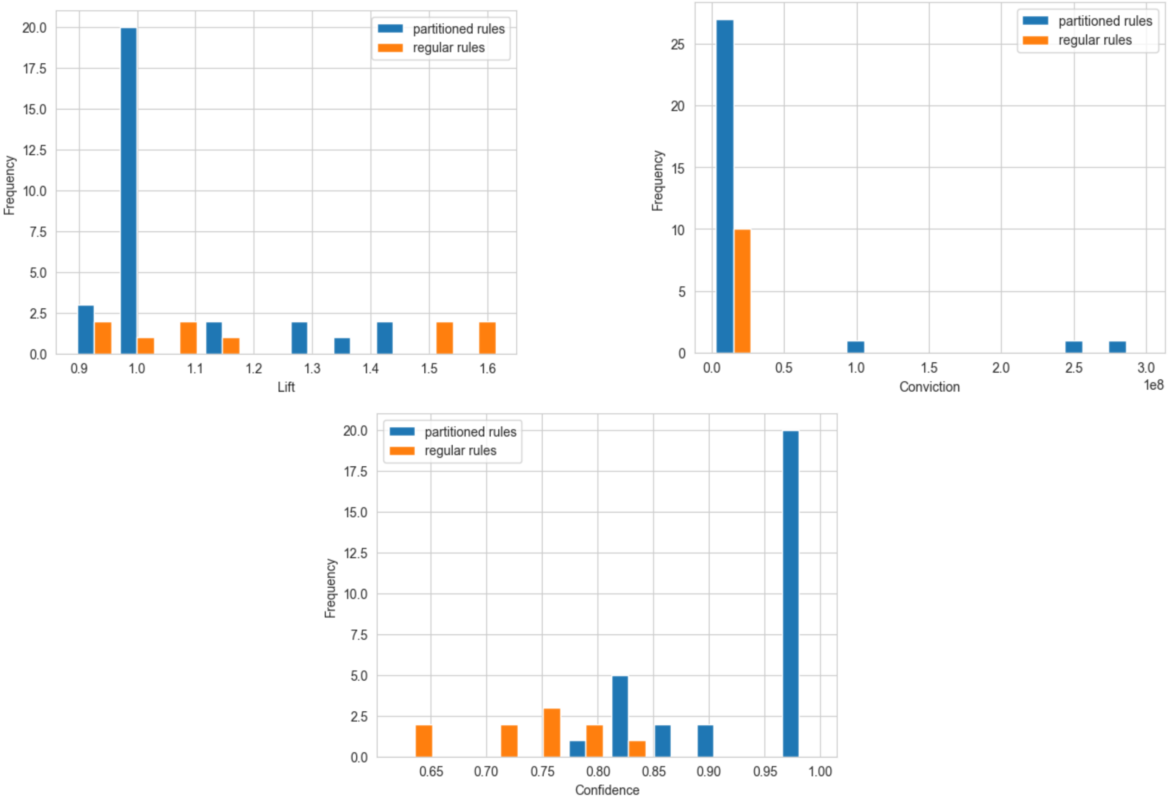

Figure 4: Lift, Conviction and Confidence graphs for (support=0.3, confidence=0.6) for regular datasets and (support=0.5, confidence=0.8) for partitioned dataset.

As displayed in the graphs, as the maximum Lift of the unpartitioned dataset is higher than the partitioned one, the Conviction and Confidence values of the rules resulting from the partitioned dataset are higher than those resulting from the unpartitioned dataset.

5 Related Work

The paper "Clustering transactions using large items" written by K., Wang, C., Xu and B., Liu (**5**), displays an algorithm for clustering transactions with large items - items contained in some minimum fraction of transactions in a cluster - that measures the similarity of a cluster of transactions.

The algorithm inspired us to cluster the dataset before the conversion into transactional form. The problem described in the paper is that clustering transactions may cause clusters of similar items rather than similar clusters of transactions, therefore the clustering at an early level has contributed to qualitative clusters based on multiple columns and features.

The k-means algorithm on the other hand has a few disadvantages, It can only cluster spherically and it uses l2 distance for measuring how close 2 points are, which fails in high dimensional data as the distance-based similarity measure converges to a constant value. We came across another paper (**6**) about the spectral clustering algorithm, which first uses whitening and PCA to lower the dimension of the data and normalize it, and perform K-means on the processed data.this should outperform K-means on many occasions and work best over datasets with many dimensions. In practice, the algorithm was too heavy for us in terms of running time(We could not utilize a GPU) so we used the K-means algorithm.

6 Conclusion

In this project, we presented a new technique that aimed to discover rare and interesting rules from a given dataset in a more subtle way (rather than changing the thresholds for the Apriory algorithm).

We investigated different clustering methods, parameters, and datasets, and were able to build the proper transactional form for each cluster. By that, we could mine each cluster and find the rules that are associated directly with the cluster's features. Therefore, the rules mined from each cluster are less trivial.

As shown in the graphs in the prior section, the Lift, Conviction and Confidence measures, the rules resulting from the partitioned dataset are consistently more interesting than those resulting from the unpartitioned dataset. That proves our thesis was right and the technique does improve the association rules mining process.

7 References

1. House Prices Dataset, Kaggle.
   Link: `https://www.kaggle.com/c/house-prices-advanced-regression-techniques/data`

2. Titanic Dataset, Kaggle.
   Link: `https://www.kaggle.com/c/titanic/data`

3. Daily Transactions Dataset, Kaggle.
   Link: `https://www.kaggle.com/datasets/prasad22/daily-transactions-dataset`

4. Customer Dataset, Kaggle.
   Link: `https://www.kaggle.com/datasets/priyanshcode/customer-segmentation`

5. Clustering transactions using large items. K., Wang, C., Xu and B., Liu. New York, NY, USA: S. Gauch, Ed. CIKM '99. ACM, Nov. 1999. In Proceedings of the Eighth international Conference on information and Knowledge Management. pp. 483-490

6. Ng, Andrew, Michael Jordan, and Yair Weiss. "On spectral clustering: Analysis and an algorithm." Advances in neural information processing systems 14 (2001).