Predicting missing values using machine learning and association rules

Eyal Michon & Arthur Malakhau

February 2023

1 Abstract

This paper compares machine learning and association rule mining for predicting missing values in large data sets. The study evaluates their effectiveness based on accuracy and provides insights into their strengths and limitations. The paper emphasizes the importance of considering the data and the problem at hand when choosing a method. The study concludes with recommendations for practitioners and future research directions, including comparison with baseline methods such as mean and mode.

2 Problem Description

The problem addressed in this paper is the difficulty in predicting missing values in large data sets. Missing values in data sets can arise from various sources such as incomplete survey responses, missing measurements, or technical failures during data collection. This lack of information can result in false conclusions and incorrect results, and can also cause significant impact when attempting to train a model on such data. This paper compares machine learning and association rule mining for predicting missing values in large data sets. Its goal is to improve the data science pipeline and provide recommendations for practitioners and future research.

Association rule mining is a widely used data mining technique for learning and analyzing correlations among items in databases. It has two associated measures, Support and Confidence, which play a crucial role in determining the accuracy of the rules generated.

Given a transaction set D and an association rule r = A $\longrightarrow$ B:
Support is defined as the percentage of transactions in D that contain both A and B. Mathematically, it can be represented as:

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

Confidence is defined as the percentage of transactions in D containing A that also contain B. It can be represented as:

$$confidence(r) := P(B|A) = \frac{P(B \wedge A)}{P(A)} = \frac{support(A \cup B)}{support(A)}$$

Lift measures how many times more often A and B occur together than expected if they were statistically independent, and is defined by:

$$lift(A \Rightarrow B) = lift(B \Rightarrow A) = \frac{conf(A \Rightarrow B)}{supp(B)} = \frac{P(B|A)}{P(B)} = \frac{P(A \cap B)}{P(A) \cdot P(B)}$$

where a lift value of 1 indicates independence between A and B.
The most frequent problem with association rules is the adjustment of threshold values for these measures. Lower values tend to produce more rules, making it difficult to determine which ones are beneficial. On the other hand, higher values can prevent important and meaningful rules from being considered. It is important to find the optimal threshold values to ensure the highest accuracy of the rules generated.
Datawig imputes missing values in data sets with varying data types using an Encoder, Featurizer, and Imputer. The Encoder identifies data types while the Featurizer converts the data into numerical representations. The MICE technique and a neural network are used to create an imputation model.

3 Solution overview

Our approach to solve the missing data problem is to use either association rules or Machine Learning techniques. Association rules are generated based on specific support and confidence levels, which are then used to impute the missing values in the training data. Machine Learning approach uses Datawig to predict the missing values. The algorithm trains a model on the available data and uses it to predict missing values. This method is different from using Association Rules, as Association Rules only identify relationships between items in a data set, while Datawig uses a deep learning model to predict missing values based on the encoded feature vectors.

3.1 Association Rules

We will begin by presenting our algorithm, which utilizes association rules to identify and impute missing values in the data set.

Algorithm 1 Pseudo-code of our association rules based algorithm

```
rules ← Generate rules with given support and confidence using Apriori
rules ← Sort rules based on lift
for row in data do
    rhs ← categories with 'NaN' value
    lhs ← categories with a value (not 'NaN') and the value itself
    relevant_rules ← []
    for rule in sorted_rules do
        if rule's rhs is a subset of rhs then
            relevant_rules ← rule
        end if
    end for
    for rule in relevant_rules do
        if rule's lhs is a subset of lhs and not filled by a disagreeing rule
        then
            missing_data ← value of rule's rhs
        end if
    end for
end for
```

Explanation of the Algorithm Flow: To help understand how the algorithm works, we will walk through an example using a specific data set. The following data set will be used as a demonstration:

Sex	Age	Occupation
Male	32	Teacher
Female	NaN	Lawyer
NaN	32	Teacher
Female	36	Lawyer

After examining the data set, we observe that there are "NaN" values in the Sex and Age columns at the second and third rows. Assuming that the algorithm has generated the following association rules:

1. Occupation, Teacher → Age, 32
2. Occupation, Lawyer → Age, 36
3. Age, 32 → Occupation, Teacher
4. Sex, Male → Age, 32
5. Age, 36 → Sex, Female

Sorting the rules by lift value prioritizes the strongest associations between items in the rule, which helps to identify and discard irrelevant rules. This method ensures that the rules used for filling missing values have the greatest impact on accuracy, improving the overall efficiency of the algorithm. We get the same list because they all have a lift of 2. We will try to complete the data that is lacking by using association rules and taking the common value. Let's start with the second row, we will split the row into two parts:
rhs = Age, lhs = Sex Female, Occupation Lawyer
Examining the Association Rules:
When evaluating the first rule, it is relevant. The reason being that the right side of the rule (Age) matches the right side of the corresponding row. As a result, this rule will be added to the list of relevant rules.
Moving on to the second rule, it can be determined that it is relevant. The right side of the rule (Age) matches the right side of the row. As a result, this rule will be added to the list of relevant rules. We do the same for the rest of the rules until there are no rules left. Now we go over all the relevant

rules and check if the left side of the rule is a subset of the left side of the row.
For the first rule we can see that the (Occupation Teacher) is not a subset of (Sex Female, Occupation Lawyer) so we move to the next rule.
Now we can see on the second rule that we have on the left side (Occupation, Lawyer) and on the left side of the row we do have (Occupation, Lawyer) so this is a subset. Additionally we can see that no other rule has filled this NaN value, so now we go over the right side of the rule and we fill in the missing value.
As we can see for the rest of the run, there are no other rules that pass the first check. If no relevant rule is found, we will not be able to make a prediction for the missing value and it will remain as "NaN."
(*) Note that we check if other rules have filled the missing value to allow a more robust rule to fill additional missing values if it is consistent with the previous rules. The previous rules hold more weight as we sort them based on lift.

3.2 Machine Learning

Explanation of the flow: To help understand how the Datawig package works, we will walk through an example using a specific data set. The same data set will be used as in 3.1. To enable Datawig to work with the data, we have to encode it, resulting in the following:
Sex = {Male: 0, Female: 1}, **Age** is a numerical feature so no encoding is required, **Occupation** = {Teacher: 0, Lawyer: 1}
We can obtain imputations by sending the data to Datawig, which performs the necessary computations and we receive the output from Datawig:

Sex	Age	Occupation
0	32	0
1	35.667	1
0.25	32	0
1	36	1

After obtaining the imputations from Datawig, we can round the resulting numbers to the nearest whole number. This enables us to extract the most appropriate value that should have been present in the respective column. After we obtain the appropriate values, we look at our encoding that we preformed at the start of the solution, and reverse our encoding.

3.3 Result

In both cases we end up with the following result: At the end we get the following data:

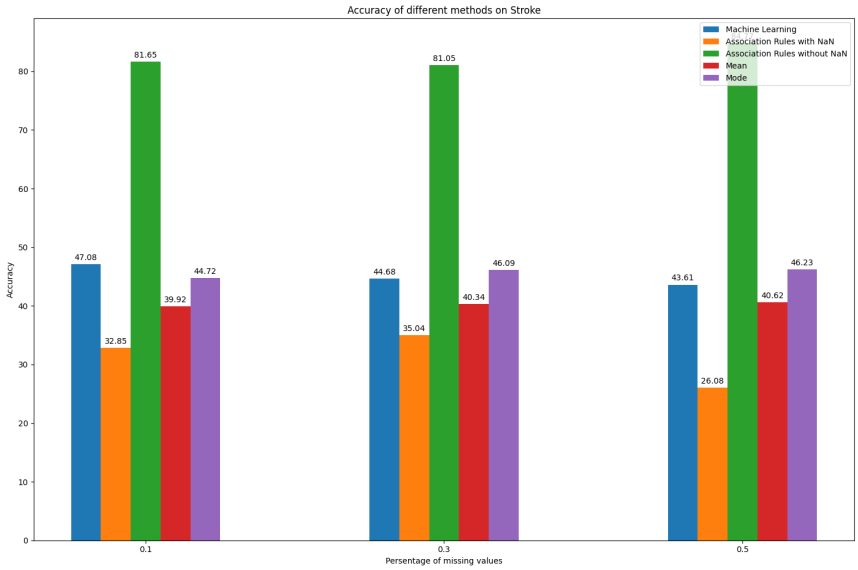
Sex	Age	Occupation
Male	32	Teacher
Female	36	Lawyer
Male	32	Teacher
Female	36	Lawyer

4 Experimental Evaluation

We tested the accuracy of our algorithm on multiple data sets by randomly removing some data and feeding the incomplete data sets to the algorithm. The algorithm predicted missing values, and we compared the results to the original data set to evaluate accuracy. We tested on four data sets with categorical and numerical data, and details can be found in the notebook.

We will now present our findings for each table by selecting a specific subset of data consisting of only a few rows and columns. Red cells represent the missing data which we will mark as NaN, orange cells represent the data that was imputed incorrectly and in green cells represent the data that was correctly imputed. Afterwards, we can see the results of the different imputation techniques when present with a data set of which X percent is missing. **The tables can be found in the Observation section in the notebook**.

4.1 Final Results



4.2 Experimental Evaluation Analysis

Firstly, our graph has two subsections to show the accuracy of the association rules based algorithm. This is because it's the only algorithm that may have missing values after imputation. This is because if the algorithm doesn't have a suitable rule for imputation, it can't impute anything. Secondly, the decrease in accuracy achieved by the association rules algorithm when accounting for missing values is caused by fewer rules being available for mining from the data. This leads to fewer predictions being made and, therefore, less accurate results. Thirdly, if we refer to other results which can be found in the notebook, we can see that mostly numerical data like the loans data set is a lot harder to impute for all types of algorithms, and also the baseline approaches produce less accurate results on this data set. Finally, the machine learning algorithm showed promising results by outperforming the mean calculation approach in all cases, although it had difficulty surpassing the mode approach.

5 Related Work

"Using association Rules for better treatment of missing values" - This paper motivated our project, which compared KNN and association rules to other machine learning algorithms for predicting missing values.
"Treatment of Missing Values for Association Rules" - This paper gave us valuable insights for our experimentation on addressing missing data values.
"Datawig: Missing Value Imputation for Tables" - This paper discusses the Datawig system's architecture, algorithms, and machine learning techniques for imputing missing values in various table formats. It offers insights into how Datawig can help address the issue of missing data values in data sets.

6 Conclusion

In conclusion, our experiments revealed strengths and weaknesses of both approaches for handling missing data. The machine learning approach is superior in completely eliminating missing values and works well on most data sets, however for the majority of cases that we've found, it doesn't perform better than the baseline mode approach. In terms of correctness, the association rules based algorithm is better at filling missing values correctly, with around 64% accuracy at its worst which far exceeds all other options. In terms of execution time, the machine learning algorithm was the slowest to produce results, but it is possible for it to achieve a faster run time when compared to the association rules based algorithm, as the former is far more affected by the amount of rows in a data set and the latter is far more affected by the amount of columns. However, both algorithms are slower than the baseline options.

One potential area for further investigation is combining methods, such as using the association rules based approach first, followed by the mode or machine learning approaches to fill in the remaining missing values, depending on the data set. The mode approach is preferred, but the machine learning approach may be more appropriate for evenly distributed data.