# Tabular Data Science Project

Shani Sperling and Shaked Arel

March 2022

## 1 Abstract:

In the Data Science field, the goal is to analyze large amounts of data by using scientific methods and extract relevant conclusions and knowledge from it. A common problem with datasets is the lacking of information which causes the data to be less quality. In order to try and solve this problem, we would like to find a sophisticated way that would help us fill in those missing values. Hopefully, those predictions would be as accurate as the real values. In this paper, we proposed a way to predict missing values in a dataset by using association rules. We used the value, which was implied from the rule, in order to estimate what its true value was supposed to be. To check the effectiveness of our proposed solution we performed an experiment on real-world datasets. The results show that in some cases, using association rules could help fill missing data in datasets but it is not always the best option. In some cases, we will not be able to predict the right value and sometimes we will not have a prediction at all.

## 2 Problem description:

The data science field helps us solve machine-learning problems. The knowledge we gain by analyzing a lot of data helps us to generalize the vast amount of information and reach conclusions that can be applied to new information that we have not yet seen. In order to do so, we use different algorithms called "learning algorithms". The maximum accuracy of those algorithms is usually determined by two important factors: the first one is the quality of the training data and the second one is the values of hyper-parameters set for them.

What makes a dataset be considered a quality one? The quality of a dataset depends upon many factors such: as the amount of information, its diversity,

and so on. The problem of filling in missing values in datasets has become an important research issue in the field of data mining and machine learning in particular. This paper came to solve this problem.

While collecting the information there is a possibility that the collected data would not be complete. There are several ways to collect data. One way is collecting information from people, for example by surveys. In this case, perhaps there was not enough time to complete the questionnaire. Or, the person taking the quiz did not understand the question. Or, respondents may refuse to answer questions due to privacy issues. Another way to collect data is by performing research. In this case, a researcher might forget to take specific measurements. Or, the software in which he conducted the study fell. All of the above are reasons why the information may be missing.

Lack of information can lead learning algorithms and in general other statistical algorithms not to work or return incorrect results (because partial information can lead to false conclusions). E.g. if we would like to run an algorithm for classification on a specific observation, but in the training dataset there would be no similar samples to the given observation, the algorithm could not classify the observation to its class. Another problem that missing information can cause is that it can provide a wrong idea about the data itself, for example: calculating an average for a column with half of the information unavailable gives the wrong metric. Sometimes when data is unavailable, some algorithms might not work at all. Some machine learning algorithms with datasets containing Nan values throw an error.

# 3 Solution overview:

Our solution to the problem described above is using association rules in order to fill the missing data. i.e. wherever there is lacking information, we will try to find a suitable association rule that will help us complete it.
The general framework of the algorithm is: Scan the data-set ⇒
Create Association Rules on given support and confidence thresholds ⇒
For each sample of Training data with a Missing value ⇒
Impute missing values using Association Rules

**Algorithm 1** Pseudo-code of our algorithm

```
¡Go over the dataset and find the rows that have at least one 'Nan' value¿
¡Generate Association rules on given support and confidence¿
For Each row ∈ HasNanValue do
    ¡Separate the row categories to¿
    right ← categories with 'Nan' value
    left ← categories with a value (not 'Nan') and the value itself
    For Each rule do
        relevant rules ← []
        if rule's rhs contains right AND left contains rule's lhs then
            relevant rules ← rule
    values ← []
    For Each 'Nan' value in a row do
        For Each relevant rule do
            if rule's rhs contains the category with the 'Nan' value then
                values ← value of rhs
        ¡Fill the missing value with the Mode of values¿
=0
```

**Explanation about the course of the algorithm:** We will demonstrate a run for a specific example that will help to understand the way the algorithm works. we will view the following dataset:

| age | gender | income range in k(1,000) |
|---|---|---|
| 30s | Male | 20-30 |
| 20s | Female | **Nan** |
| 30s | Female | **Nan** |
| 20s | Female | 10-20 |

Going over this dataset, we notice 'Nan' values at the second and third rows in the income column.

Suppose that the algorithm found the following association rules:

1. age 30's → income 20-30
2. age 20's, gender female → income 10-20
3. gender male → age 30's

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:
right = income
left = age 20's, gender female.
We will go over the association rules and find the relevant ones:
Looking at the first rule we see that: the right side of the rule (income) contains the right side of the row and that the left side of the rule (age 30's) is not contained in the left side of the row therefore this rule is irrelevant.
Looking at the second rule we see that: the right side of the rule (income) contains the right side of the row and that the left side of the rule (age 20's, gender female) is contained on the left side of the row, therefore, this rule is relevant so we will add it to the list of relevant rules.
Looking at the third rule we see that: the right side of the rule does not contain the right side of the row therefore this rule is irrelevant.
That is, there was found only one suitable association rule.
In the values list, we will save the value deriving from the rules we previously found, which is 'income 10-20'.

Finally, to fill the missing data we will take the frequent value in the values list (in our case there is only one thus it will be chosen) and update the row to be: 20's female 10-20.

Similarly, for the third row, the algorithm will work the same and will choose the first rule (age 30's $\rightarrow$ income 20-30) and eventually, the updated row that we will get is: 30's female 20-30.

In case we will get more than one suitable rule, we will decide to take the common value, i.e. the one that appears in most of the rules. In case we have not found any relevant rule, we will not be able to reach a prediction about the missing value and therefore will keep it 'Nan'.

**A short explanation about our intuition:**
A rule is found if there are a number of instances that uphold the rule. In order to handle the lack of information, we would like to find rules that their values correspond to the values of the row in which the data is missing. If such a rule is found then the value derived from the rule will be, in high probability, pretty close to the real value that is missing.
In the example shown above, we saw that in many cases women in their 20s earn 10,000-20,000$ therefore, we found the rule:
age 20's, gender female $\rightarrow$ income 10-20.

Now when given a piece of information about a woman in her 20s without knowing how much she earns, according to the rule we will estimate that she earns between 10,000 to 20,000$ since that is the value that was expected.

# 4 Experimental evaluation:

To evaluate the correctness of the suggested algorithm, we performed several experiments on different datasets. In each experiment, we took a complete dataset and randomly deleted parts of its data. Then sent those datasets with missing values to the algorithm, which returned a new dataset with predictions about the missing values. At last, we compared the original dataset with the returned dataset.
We performed our experiment on four different datasets, where their columns contain categorical information. The details about those datasets are presented in the following table:

| dataset | number of instances | number of attributes |
|---|---|---|
| car_data | 1728 | 7 |
| balance | 625 | 5 |
| shopping_data | 200 | 5 |
| ta | 151 | 6 |

For two of those four datasets, we will see three tables. The first table will be the dataset with 'Nan' values, the second table will be the original dataset and the third table will be the dataset that returned from the algorithm.
In addition, for each dataset, we will provide its own success rate (how many values the algorithm predicted correctly among all the missing values)
car_data:
car_data_nan (rows 11-17):

| buying price | maintenance cost | number of doors | number of persons | lug_boot | safety | decision |
|---|---|---|---|---|---|---|
| vhigh | vhigh | 2 | 4 | small | nan | unacc |
| vhigh | nan | 2 | 4 | small | med | unacc |
| vhigh | vhigh | 2 | nan | small | high | nan |
| vhigh | vhigh | 2 | 4 | med | low | unacc |
| vhigh | vhigh | 2 | 4 | med | med | unacc |
| vhigh | vhigh | 2 | 4 | med | high | nan |
| vhigh | vhigh | 2 | 4 | big | nan | unacc |

original:

| buying price | maintenance cost | number of doors | number of persons | lug_boot | safety | decision |
|---|---|---|---|---|---|---|
| vhigh | vhigh | 2 | 4 | small | low | unacc |
| vhigh | vhigh | 2 | 4 | small | med | unacc |
| vhigh | vhigh | 2 | 4 | small | high | unacc |
| vhigh | vhigh | 2 | 4 | med | low | unacc |
| vhigh | vhigh | 2 | 4 | med | med | unacc |
| vhigh | vhigh | 2 | 4 | med | high | unacc |
| vhigh | vhigh | 2 | 4 | big | low | unacc |

Prediction:

| buying price | maintenance cost | number of doors | number of persons | lug_boot | safety | decision |
|---|---|---|---|---|---|---|
| vhigh | vhigh | 2 | 4 | small | low | unacc |
| vhigh | nan | 2 | 4 | small | med | unacc |
| vhigh | vhigh | 2 | 2 | small | high | unacc |
| vhigh | vhigh | 2 | 4 | med | low | unacc |
| vhigh | vhigh | 2 | 4 | med | med | unacc |
| vhigh | vhigh | 2 | 4 | med | high | unacc |
| vhigh | vhigh | 2 | 4 | big | low | unacc |

Results: we can see that in the yellow cells, the algorithm predicted correctly according to the association rules. In the blue cell, the algorithm has found a prediction yet a wrong value. And in the red cell, the algorithm couldn't predict at all and left it with 'Nan'.

Overall, the success rate for this dataset is 43/50 = 86%

ta:

ta_data (rows 61-64)

| language | course instructor | course | summer or regular | size | attribute |
|---|---|---|---|---|---|
| nan | 25 | 7 | regular | 27 | m |
| Hebrew | 25 | nan | regular | 23 | m |
| Hebrew | 2 | 9 | regular | 31 | nan |
| Hebrew | 1 | 15 | nan | 22 | m |

original:

| language | course instructor | course | summer or regular | size | attribute |
|---|---|---|---|---|---|
| Hebrew | 25 | 7 | regular | 27 | m |
| Hebrew | 25 | 7 | regular | 23 | m |
| Hebrew | 2 | 9 | regular | 31 | m |
| Hebrew | 1 | 15 | summer | 22 | m |

Prediction:

| language | course instructor | course | summer or regular | size | attribute |
|---|---|---|---|---|---|
| Hebrew | 25 | 7 | regular | 27 | m |
| Hebrew | 25 | 7 | regular | 23 | m |
| Hebrew | 2 | 9 | regular | 31 | m |
| Hebrew | 1 | 15 | summer | 22 | m |

The success rate for this dataset is 10/17 = 58.8%.

For the rest of the datasets, we will only note the success rate:

Balance: 18/25 = 72%

A note about this dataset: we found out that all the rules that have been found imply only on a certain column thus this column will be the only one we can predict. In the rest of the columns, the algorithm wouldn't be able to predict and will leave 'Nan' values.

Shopping_data: 7/17 = 41.2%

A note about this dataset: we noticed that since it is a small dataset, by deleting data we reduced it even more. Which causes finding only a small

amount of rules. To find more rules we lowered the confidence threshold but this made the rules less quality, which led to filling the information incorrectly.

# 5 Related work:

While choosing a topic for this project we reviewed many articles. From which we came across the article "Treatment of Missing Values for Association Rules: A Recent Survey" from which we took inspiration for the project theme. We have seen that dealing with missing information is a common problem that we should approach and try to bring a solution to.
In addition, we have seen that there are a number of different ways to deal with this problem and different solutions that use association rules, which has given us direction and inspiration while thinking about the algorithm.
While creating the algorithm we were assisted by the article "Using Association Rules for Better Treatment of Missing Values" which introduced an algorithm similar to our idea and expands it with improvements such as references to non-categorical information and use of the 'knn' algorithm. However, the way our algorithm runs is a bit different. For example, when checking whether a rule is relevant for a particular missing value, in the article they checked only if the dragging side of the rule is contained on the left side of the row while we also checked the dragged side to reduce the number of rules and dismiss rules that do not contribute to us.

# 6 Conclusion:

The conclusion from the study we conducted, is quite complex. First, we have seen that there are cases in which the association rules can actually be used to fill in missing information. In cases where there is a relatively large number of good rules (with high confidence) and when the parts drawn from the rules are spread over as many categories in the dataset (i.e. different rules drag different categories so each category will have rules that can predict it) the algorithm significantly improves the accuracy of the dataset content.
In addition, we have seen that when there are not enough good rules, we will not be able to fill in the information correctly and when there are not enough rules, we will not be able to fill in at all. Overall, the use of association rules can be a good idea for improving a dataset with missing information.