NATIONAL RESEARCH UNIVERSITY
«HIGHER SCHOOL OF ECONOMICS»
Faculty of computer science

Ordered Sets in Data Analysis
PROJECT REPORT

LAZY FCA TOOLBOX

Student:
Shenker Anastasia

Moscow 2020

Task Description

Lazy FCA is a method of solving binary classification problem. Binary classification problem is a kind of problem when target feature takes one of two values – positive or negative, which can also be denoted as '+' and '−', or 1 and 0. The value of target feature is known for train set, the purpose is to estimate the values of target feature for the test set based on the values of the remaining variables.

This algorithm is called 'lazy' because the step of building the classification model is eliminated. The class label is assigned not according to the parameters of some trained model, but according to the existing data from the training set for each test object separately.

Algorithm

The general algorithm is to compare with which of the contexts the test object has the most matches in other variables.

The first step is to split the train set into two parts: plus-context C_+, which contains objects with positive value of target feature, and minus-context C_-, which contains objects with negative value of target feature.

The next step is to calculate intersection of i-th test object with plus-context C_+. Then the same operation for minus-context C_-: calculating intersection of i-th test object with minus-context C_-.

$$h_+(c) = \sum_{c_+ \in C_+} c \cup C_+$$
$$h_-(c) = \sum_{c_- \in C_-} c \cup C_-$$

C is a set of all objects, C_+ is a set of plus-context and C_- is a set of minus-context.

The next step is to compare, with which of context the intersection is greater.

In my project, I tried three modifications of that algorithm. I estimated their metrics of goodness on the test set, chose the best one and estimated its metrics on the KFold cross-validated data.

Algorithm 1

The modification of this algorithm is a threshold value of the number of intersections, on the basis of which the classification is carried out. After calculating of intersection with plus-context, I check, weather this intersection is presented in any examples in minus-context. If the amount of such examples is greater, than the threshold, then the example is classified as minus-context. And analogically for intersection with minus-context.

According to this algorithm, an example is classified positively if each of its intersections with objects from the plus-context is nested in no more than 3 descriptions from the minus-context (and vice versa).

If in the end the amount of positive and negative classification labels is equal, the classification is a random choice with equal probabilities.

The resulting metrics for this algorithm on the one spited train-test set are as follows.

Name of metric	Value of metric
True Positive	26
True Negative	16
False Positive	16
False Negative	35
Accuracy score	0.45
Precision Score	0.61
Recall Score	0.43
Roc AUC Score	0.46
Time of algorithm work	2.19

It can be seen from the table, that the accuracy score is less than 50%, which is very low. This algorithm does not classify well, it tends to give a negative class label too often. Moreover, its results are always different because it uses random choice too often.

Algorithm 2

In this algorithm, each plus-context object 'votes' for a positive classification if its intersection with the example does not fit into the minus-context descriptions (and vice versa). An example is classified positively if the number of 'votes' for the positive classification prevails (and vice versa).

The resulting metrics for this algorithm on the one spited train-test set are as follows.

Name of metric	Value of metric
True Positive	61
True Negative	32
False Positive	0
False Negative	0
Accuracy score	1.00
Precision Score	1.00
Recall Score	1.00
Roc AUC Score	1.00
Time of algorithm work	272.77

As can be seen from the table, the accuracy score of this algorithm is 100%, that is strange. This algorithm identifies all objects correctly, which cannot be true and most likely happens due to some error.

Also, this algorithm has the longest time of work.

Algorithm 3

The simplest modification of Lazy FCA. An example is classified positively if its intersection with plus-context is greater than with minus-context and vice versa.

Name of metric	Value of metric
True Positive	61
True Negative	0
False Positive	32
False Negative	0
Accuracy score	0.66
Precision Score	0.66
Recall Score	1
Roc AUC Score	0.5
Time of algorithm work	0.36

As can be seen from the table, the accuracy score is 66%, which is not great but enough to admit that the algorithm is good. But also, we can see, that the amounts of true and false negatives are zeros, that means that the algorithm identified all objects as positive.

According to the accuracy score the best and the most reliable algorithm is the third one.

Cross-validation

I took ‘tic-tac-toe’ algorithm for cross-validation. It consists of 958 observations. There are 10 categorical variables, one of which is the target feature with value ‘positive’ and ‘negative’.

For the third algorithm, the data must be presented in quantitative form, so I transformed all categorical features, except the target one, to dummy variables, enveloped into three binary variable each. The target feature was just presented as binary, where ‘positive’ = 1, ‘negative’ = 0.

For cross-validation I took KFold cross-validation with 10 splits. The realization of it was taken from *sklearn.model_selection* specification.

After launching the third algorithm on each of 10 test-splits, I took an average of 10 for each metric. The results are presented in a table below.

Name of metric	Value of metric
True Positive	31.9
True Negative	17.8
False Positive	15.4
False Negative	30.7
Accuracy score	0.52
Precision Score	0.67
Recall Score	0.51
Roc AUC Score	0.52
Time of algorithm work	0.303

As can be seen from the table, the problem with the absence of negative values is no longer actual, which means that was the singularity of data. But the accuracy score after cross-validation fell by more than 10 points to 52%, which is low.