OSDA BIG HW

Rudenkov Dmitriy

December 13, 2022

In homework, we need to select a dataset of binary classification and implement the Lazy FCA method.

Dataset

I chose «Contraceptive Method Choice Data Set» from UCI Mashing learning Repository https://archive.ics.uci.edu/ml/datasets/Contraceptive+Method+Choice.

This dataset is a subset of the 1987 National Indonesia Contraceptive Prevalence Survey. The samples are married women who were either not pregnant or do not know if they were at the time of interview.

The problem is to predict the current contraceptive method choice (no use, long-term methods, or short-term methods) of a woman based on her demographic and socio-economic characteristics. Number of Attributes: 9. Number of Instances: 1473.

Attribute Information:

1. Wife's age (numerical)

2. Wife's education (categorical) 1=low, 2, 3, 4=high

3. Husband's education (categorical) 1=low, 2, 3, 4=high

4. Number of children ever born (numerical)

5. Wife's religion (binary) 0=Non-Islam, 1=Islam

6. Wife's now working? (binary) 0=Yes, 1=No

7. Husband's occupation (categorical) 1, 2, 3, 4

8. Standard-of-living index (categorical) 1=low, 2, 3, 4=high

9. Media exposure (binary) 0=Good, 1=Not good

10. Contraceptive method used (class attribute) 1=No-use, 2=Long-term, 3=Short-term

	w_age	w_edu	h_edu	num_child	w_relig	w_work	h_occup	live_ind	med_expos	contr
0	24	2	3	3	1	1	2	3	0	1
1	45	1	3	10	1	1	3	4	0	1
2	43	2	3	7	1	1	3	4	0	1
3	42	3	2	9	1	1	3	3	0	1
4	36	3	3	8	1	1	3	2	0	1

This homework focuses on the task of binary classification. So we will be classify if women use or not contraceptive (1->False, {2,3}->True).

	contr
0	False
1	False
2	False
3	False
4	False

Using Lazy FCA baseline algorithm firstly we must binary our attributes.

We don’t change binary attribute. It would be transformed like: 1->True, 0->False

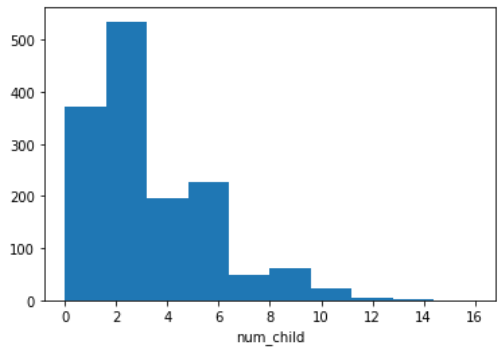
	w_relig	w_work	med_expos
0	True	True	False
1	True	True	False
2	True	True	False
3	True	True	False
4	True	True	False

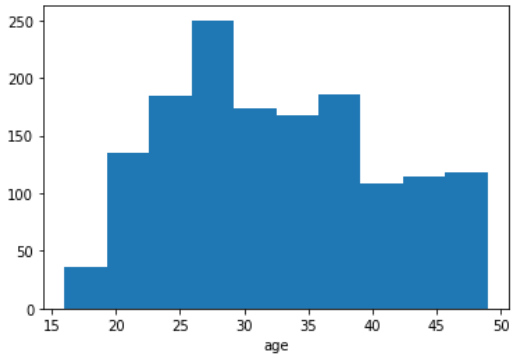
Categorical attributes were converted into a set binary features (for example, the attributes "Standard-of-living index" was transform to for binary attributes "living index 1", " living index 2", "living index 3","living index 4")

	w_edu: 1	w_edu: 2	w_edu: 3	w_edu: 4	h_edu: 1	h_edu: 2	h_edu: 3	h_edu: 4	h_occup: 1	h_occup: 2	h_occup: 3	h_occup: 4	live_ind: 1	live_ind: 2	live_ind: 3	live_ind: 4
0	False	True	False	False	False	False	True	False	False	True	False	False	False	False	True	False
1	True	False	False	False	False	False	True	False	False	False	True	False	False	False	False	True
2	False	True	False	False	False	False	True	False	False	False	True	False	False	False	False	True
3	False	False	True	False	False	True	False	False	False	False	True	False	False	False	True	False
4	False	False	True	False	False	False	True	False	False	False	True	False	False	True	False	False

We binarize numerical signs using intervals that we choose based on logic and quantitative data.

```
0 child = 97
1 child = 276
2 child = 276
3 child = 259
4 child = 197
5 child = 135
6 child = 92
7 child = 49
8 child = 47
9 child = 16
10 child = 11
```





For «Wife's age»:

1) <=21 (age of majority)

2) 22<=…<=30 (mid life)

3) >=31

For « Number of children ever born »:

1) 0 (child free temporary)

2) 1 and 2 (average European family)

3) 3 and 4

4) more and equal 5

	w_age: ..-21	w_age: 22-30	w_age: 31-..	num_child: 0	num_child: 1-2	num_child: 3-4	num_child: 5-..
0	False	True	False	False	False	True	False
1	False	False	True	False	False	False	True
2	False	False	True	False	False	False	True
3	False	False	True	False	False	False	True
4	False	False	True	False	False	False	True

After binarization of all attributes we have 26 binary attributes.

```
X = pd.concat([numerical_attr_data, binary_attr_data, category_attr_data], axis=1)
print(X.shape)
X.head()
```

```
(1473, 26)
```

	w_age: ..-21	w_age: 22-30	w_age: 31-..	num_child: 0	num_child: 1-2	num_child: 3-4	num_child: 5-..	w_relig	w_work	med_expos	...	h_edu: 3	h_edu: 4	h_occup: 1	h_occup: 2	h_occup: 3	h_occup: 4	live_ind: 1	live_ind: 2
0	False	True	False	False	False	True	False	True	True	False	...	True	False	False	True	False	False	False	False
1	False	False	True	False	False	False	True	True	True	False	...	True	False	False	False	True	False	False	False
2	False	False	True	False	False	False	True	True	True	False	...	True	False	False	False	True	False	False	False
3	False	False	True	False	False	False	True	True	True	False	...	False	False	False	False	True	False	False	False
4	False	False	True	False	False	False	True	True	True	False	...	True	False	False	False	True	False	False	True

5 rows × 26 columns

Algorithm

In the homework we use baseline algorithm for lazy FCA-based classification. And it is called "Generators framework".

Assume that we want to make a prediction for description x⊆M given the set of training examples Xtrain⊆2^M and the labels yx∈{False,True} corresponding to each x∈Xtrain.

First, we split all examples to positive and negative examples.

$$X_{pos} = \{x \in X_{train} \mid y_x \text{ is True}\}, \quad X_{neg} = X \setminus X_{pos}.$$

To classify the descriptions, we follow the procedure:

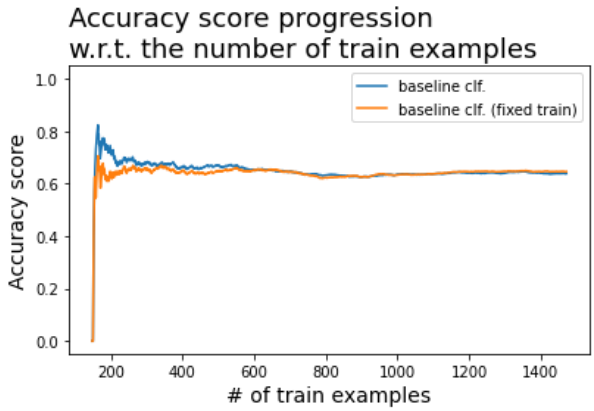
1) Count the number of counterexamples for positive examples. For each positive example we compute the intersection. Then, we count the counterexamples for this intersection, that is the number of negative examples containing intersection;
2) Dually, count the number of counterexamples for negative examples. For each negative example we compute the intersection. Then, we count the counterexamples for obtained intersection, that is the number of positive examples containing intersection.

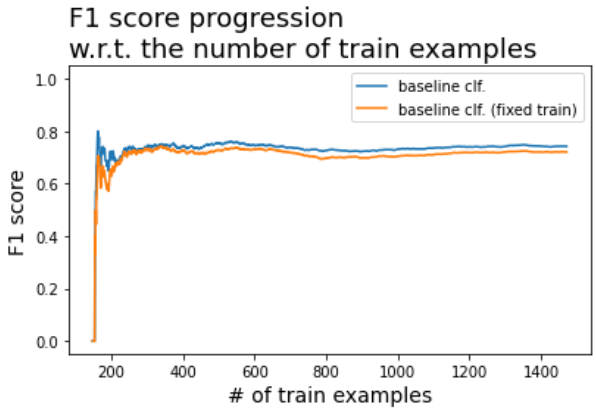
Finally, we compare the average number of counterexamples for positive and negative examples. We classify as being positive if the number of counterexamples for positive examples is smaller the one for negative examples.

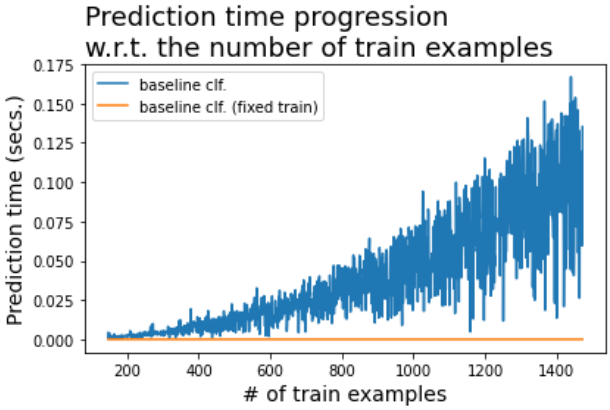
Analyze results

To reduce the running time and according to the logical solution I set minimal cardinality of intersection level is 3 (cause <3 is not enough).

There are various scores. But most often they use those that are calculated based on confusion matrix: Accuracy, Recall, Precision. But accuracy depends on the ratio of classes, unlike Precision and recall, which are applicable in conditions of unbalanced samples. Therefore, for example, there is an F-measure that combines information about the accuracy and completeness of our algorithm.







	Name	TP	FP	FN	TN	Accuracy	Precision	Recall	F1
0	baseline clf.	156	404	76	690	0.638009	0.630713	0.900783	0.741935
1	baseline clf. (fixed train)	251	309	161	605	0.645551	0.661926	0.789817	0.720238

As we can see the scores show that our classification does not perform very well and there is no big difference between unlimited and fix train. This is due to the fact that the naked classification algorithm was used without modifications and hints.

We will be able to use the fact of the power of intersections: compare the powers of positive and negative intersections, check them not through the number of occurrences in the opposite group, but rather in the initial one, do not binarize numerical attributes but use interval intersections, add cross validation, use coefficients to find a closer option, and other.

I tried to implement several options to improve performance, but unfortunately did not succeed in this.

"Generators framework" does not work well enough with low predictive performance needs further work on refinement and improvement.