NATIONAL RESEARCH
UNIVERSITY

PROJECT REPORT

LAZY LEARNING WITH FCA

Simov Gleb

Faculty of Computer Science

Moscow 2021

OVERVIEW

Pattern structures allow us to approach the problem of knowledge extraction in the case of an arbitrary description of an object. They allow the application of formal concept analysis (FCA) techniques to non-binary contexts. However, in order to produce classification rules a concept lattice should be built. For non-binary contexts this procedure may take much time and resources. In order to tackle this problem, we used a modification of the lazy associative classification algorithm. The resulting quality of classification is compared to popular classification algorithms.

DATA

This dataset contains records of 700 good and 300 bad credits with 20 predictor variables, collected from 1973 to 1975.

Predictor variables

- laufkont: status of the debtor's checking account with the bank (categorical)
- laufzeit: credit duration in months (quantitative)
- moral: history of compliance with previous or concurrent credit contracts (categorical)
- verw: purpose for which the credit is needed (categorical)
- hoehe: credit amount in DM (quantitative; result of monotonic transformation; actual data and type of transformation unknown)
- sparkont: debtor's savings (categorical)
- beszeit: duration of debtor's employment with current employer (ordinal; discretized quantitative)
- rate: credit installments as a percentage of debtor's disposable income (ordinal; discretized quantitative)
- famges: combined information on sex and marital status (categorical)
- buerge: Is there another debtor or a guarantor for the credit? (categorical)
- wohnzeit: length of time (in years) the debtor lives in the present residence (ordinal; discretized quantitative)
- verm: the debtor's most valuable property, i.e. the highest possible code is used. Code 2 is used, if codes 3 or 4 are not applicable and there is a car or any other relevant property that does not fall under variable sparkont. (ordinal)
- alter: age in years (quantitative)
- weitkred: installment plans from providers other than the credit-giving bank (categorical)
- wohn: type of housing the debtor lives in (categorical)
- bishkred: number of credits including the current one the debtor has (or had) at this bank (ordinal, discretized, quantitative)
- beruf: quality of debtor's job (ordinal)
- pers: number of persons who financially depend on the debtor (i.e., are entitled to maintenance) (binary, discretized quantitative)
- telef: Is there a telephone landline registered on the debtor's name? (binary)
- gastarb: Is the debtor a foreign worker? (binary)
- kredit: Has the credit contract been complied with (good) or not (bad) ? (binary)

For FCA classification approach data should be binary, so we need somehow binarize the non-binary features. Thus, all numerical features were uniformly distributed across 4 equally spaced bins.

	kredit	laufkont_0	laufkont_1	laufkont_2	laufkont_3	laufzeit_0	laufzeit_1	laufzeit_2	laufzeit_3	moral_0	...	pers_2	pers_3	telef_0	telef_1	telef_2	telef_
0	1	0	0	0	0	0	0	0	0	1	...	1	1	0	0	0	
1	1	0	0	0	0	0	0	0	0	1	...	0	0	0	0	0	
2	1	1	0	0	0	0	0	0	0	1	...	1	1	0	0	0	
3	1	0	0	0	0	0	0	0	0	1	...	0	0	0	0	0	
4	1	0	0	0	0	0	0	0	0	1	...	1	1	0	0	0	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
995	0	0	0	0	0	1	0	0	0	1	...	0	0	0	0	0	
996	0	0	0	0	0	1	0	0	0	1	...	1	1	0	0	0	
997	0	1	1	1	1	1	0	0	0	1	...	1	1	1	1	1	
998	0	1	0	0	0	0	0	0	0	1	...	1	1	1	1	1	
999	0	0	0	0	0	1	0	0	0	1	...	1	1	0	0	0	

LAZY FCA ALGORITHM

1. We divide our dataset into positive and negative classes.
2. Our algorithm is trying to compare the classified object with each example from positive class.
3. More precisely, we compare $\sum |g' \cap g'_+|$ with $|g'| \cdot$ threshold.
4. Analogously, we compare $\sum |g' \cap g'_-|$ with $|g'| \cdot$threshold. Here g' -features of unclassified objects, g_+, g_- - features of objects from positive/negative set.
5. If the sum of intersections is greater, then this example(from positive or negative class) votes for the classified object.
6. Then we normalize the sums of votes in both classes. And the normalized ones are used for classification.

Baseline accuracy of this algorithm with default threshold (0.1) is 0.33. To maximize accuracy we searched for the best value of threshold over. We found that the $threshold = 0.8$ maximize the accuracy score for our dataset and the accuracy increased to 0.703.

METRICS

To evaluate algorithm, those metrics were used:

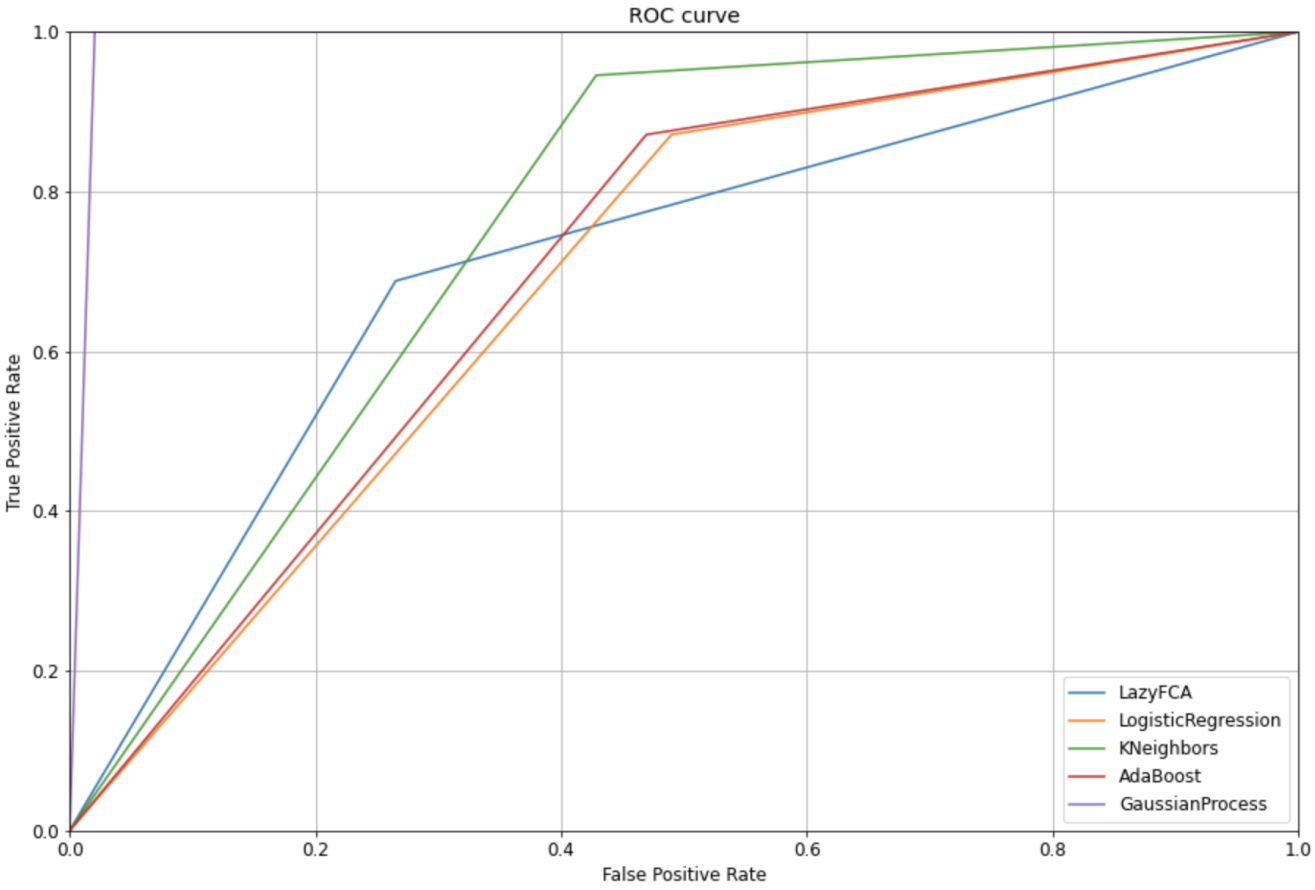
- True positive
- True Negative
- False Positive
- False Negative
- True Positive Rate
- True Negative Rate
- Negative Predictive Value
- False Positive Rate
- False Discovery Rate
- Accuracy
- Precision
- Recall
- F1 score

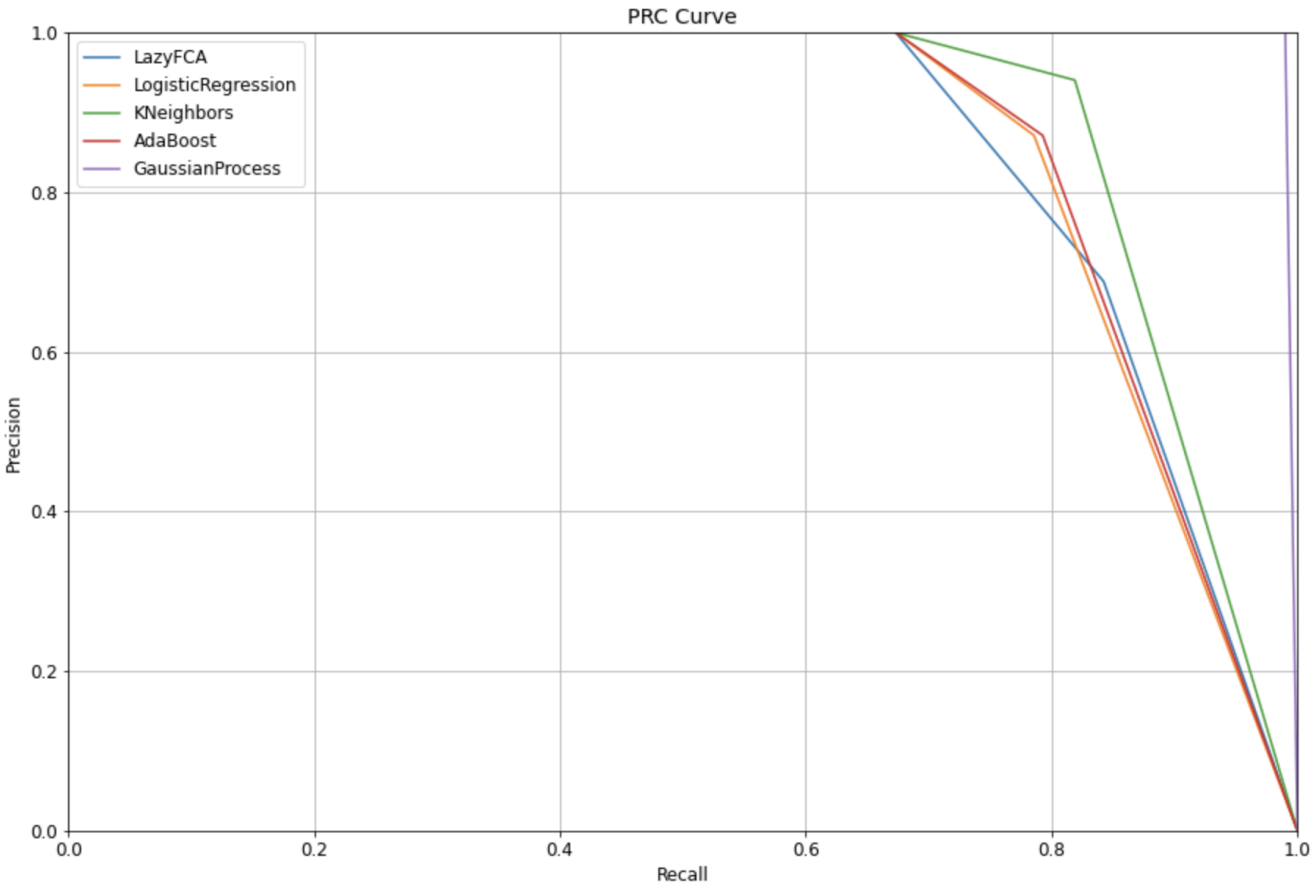
where accuracy, precision, recall and f1 measure were taken from sklearn library.

COMPARISON WITH OTHER ALGORITHMS

Those metrics are used to compare algorithm with 4 other popular algorithms, namely LogReg, KNN, AdaBoost and GaussianProcess.

Roc and PRC curves to visualise results:





CONCLUSION

As we can see from the graphs above, LazyFCA performs slightly better than LogReg and AdaBoost and at the same time slightly worse than the others. Apart from that, with attuned thershold parameter, our algorithms shows good results but still needs some improvements. The correct categorization and further binarization of data play an important role here.