Lazy FCA (Formal Concept Analysis) Classifcation on Loan Approval Dataset.

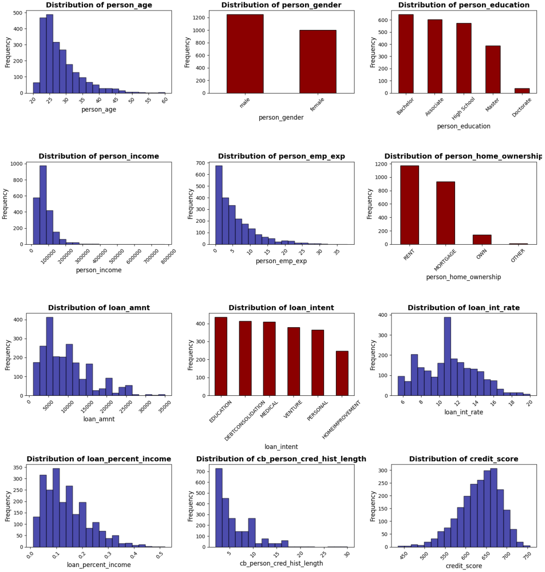
Slovyagina Anna

December 8, 2024

Outline

- Dataset overview
- Binarization
- Comparison of Standard and Modified Algorithm
- Standard Classification Algorithms
- Results

Dataset overview



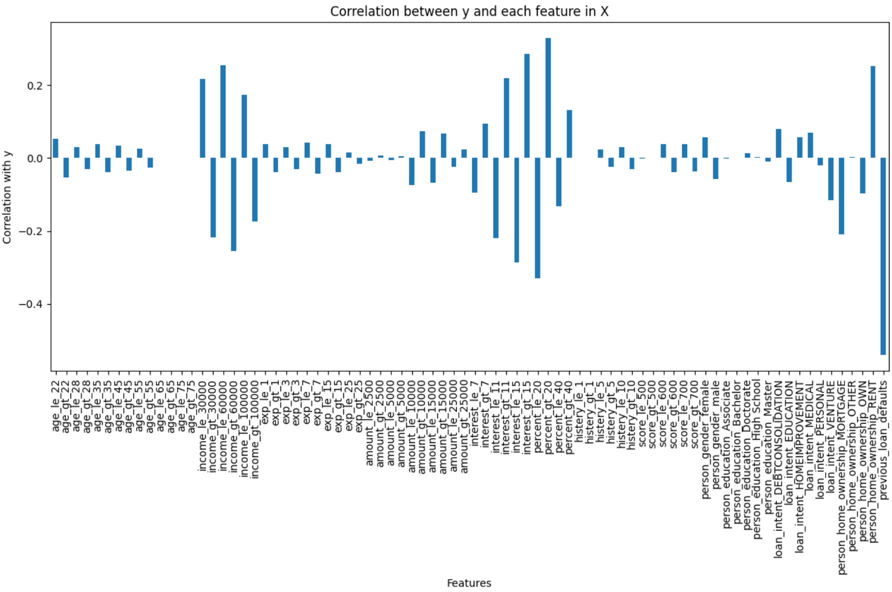
Binarization

One-Hot Encoding: gender, education, home ownership, loan intent.
Inter-Ordinal Scaling: age, income, experience in years, loan amount, interest rate, loan amount as a percentage of annual income, length of credit history in years, credit score.

Example of inter-ordinal scaling: loan amount encoding

- ≤ 2500: Lower end loans
- ≥ 2500: bove minimum
- ≤ 5000: Moderate credit needs
- ≥ 5000: Substantial commitments
- ≤ 10000: Common ceiling
- ≥ 10000: Higher-value loans
- ≤ 15000: High-value assessment
- ≥ 15000: Above average debt
- ≤ 25000: Upper lending limit
- ≥ 25000: Extreme cases

Binarization



Comparison of Standard and Modified Algorithm

The problem with the standard classifier lies in the uneven distribution of classes: the number of positive classes is significantly lower than that of negative classes, which is why a modification to the algorithm was applied. Various experiments were conducted on the validation set, during which different thresholds for positive classifiers were considered. The most optimal solution turned out to be selecting the positive class on the condition that positive classifiers make up at least 10% of the identified negative classifiers.

	TP	**TN**	**FP**	**FN**	**TNR**	**NPV**	**FPR**
Lazy_FCA	31	166	9	19	0.95	0.89	0.051
Not modified FCA	5	175	0	45	1.00	0.79	0.00

	Accuracy	**Precision**	**Recall**	**F1**
Lazy_FCA	0.88	0.78	0.62	0.69
Not modified FCA	0.80	1.00	0.10	0.18

Standard Classification Algorithms

For all classical classifiers, hyperparameter tuning was performed using GridSearchCV, accompanied by cross-validation. The best hyperparameters found for each classifier are as follows:

KNeighborsClassifier	`n_neighbors: 7` `weights: distance`
GaussianNB	`default`

LogisticRegression	`C: 1`
	`max_iter: 100`
	`multi_class: multinomial`
	`penalty: l2`

SVC	`C: 1`
	`degree: 2`
	`gamma: scale`
	`kernel: rbf`

DecisionTreeClassifier	`max_depth: 10`
	`min_samples_leaf: 2`
	`min_samples_split: 10`

RandomForestClassifier	`max_depth: 10`
	`min_samples_leaf: 1`
	`min_samples_split: 2`
	`n_estimators: 50`

XGBClassifier	`learning_rate: 0.1`
	`max_depth: 5`
	`min_child_weight: 2`
	`n_estimators: 50`
	`subsample: 1.0`

Results

The Lazy FCA classification approach demonstrated competitive performance compared to traditional machine learning algorithms, achieving an accuracy of 88% and maintaining a good balance between precision (0.76) and recall (0.62). While it may not achieve the highest absolute performance metrics, its strong interpretability features and competitive accuracy make it a viable alternative to traditional machine learning approaches, especially in scenarios where understanding the model's reasoning is as important as its predictive accuracy.

	Lazy_FCA	**FCA**	**KNN**	**NB**	**LR**	**SVC**	**DT**	**RF**	**XGB**
TP	31	5	31	50	36	40	38	39	41
TN	166	175	164	110	167	169	167	172	170
FP	9	0	11	65	8	6	8	3	5
FN	19	45	19	0	14	10	12	11	9
TNR	0.95	1.00	0.94	0.63	0.95	0.97	0.95	0.98	0.97
NPV	0.89	0.79	0.89	1.00	0.92	0.94	0.931	0.94	0.95
FPR	0.051	0.00	0.06	0.37	0.046	0.034	0.046	0.02	0.03
FDR	0.23	0.00	0.26	0.57	0.18	0.13	0.17	0.07	0.10
Accuracy	0.88	0.80	0.87	0.71	0.902	0.93	0.91	0.94	0.94
Precision	0.76	1.00	0.74	0.43	0.81	0.87	0.83	0.93	0.89
Recall	0.62	0.10	0.62	1.00	0.72	0.80	0.76	0.78	0.82
F1	0.69	0.18	0.67	0.60	0.76	0.83	0.79	0.84	0.85

Thank you for your attention!