Lazy FCA

Sifei Meng

https://github.com/mengsifei/osda_lazy_fca

Dataset

- 45000 records
- 5 categorical features
- 8 numeric features
- 1 target feature (loan_status)

Loan_status:

- 0 - rejected
- 1 - approved

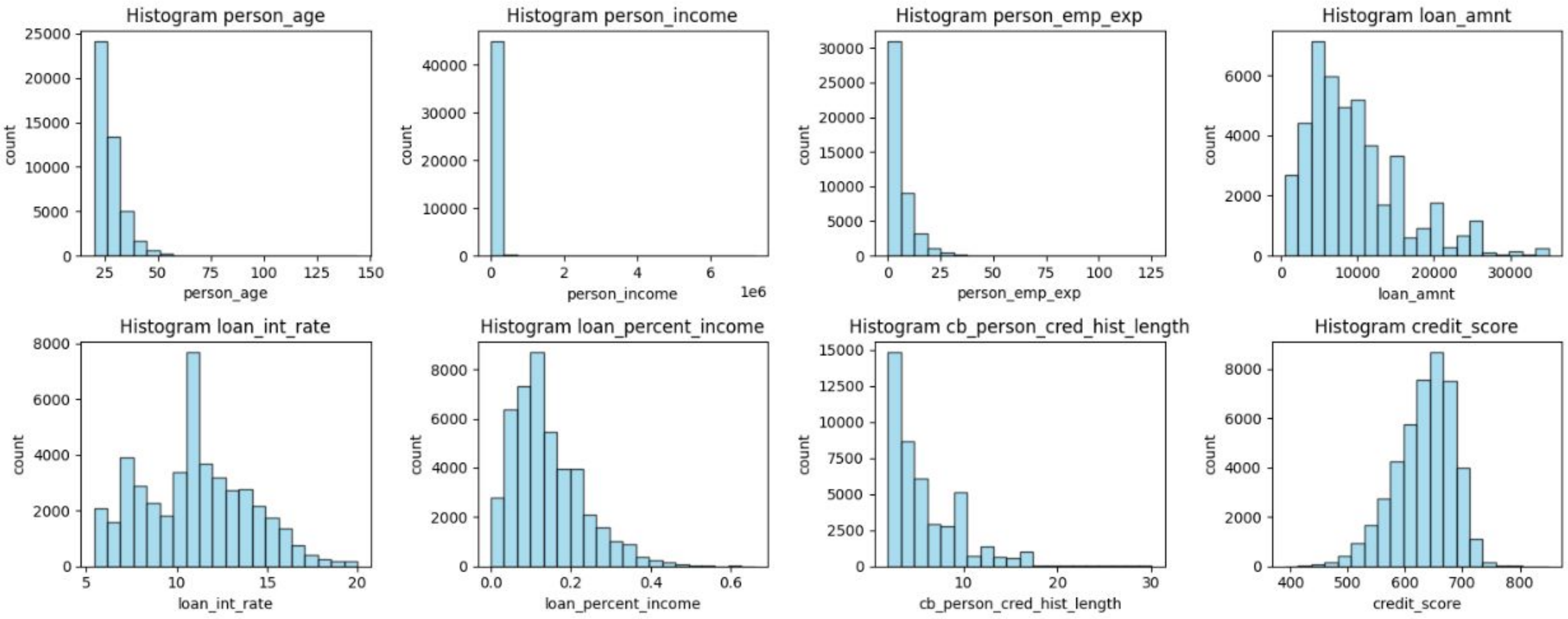
Feature Name	Description	Data Type
person_age	Age of the person	Float
person_gender	Gender of the person	Categorical
person_education	Highest education level	Categorical
person_income	Annual income	Float
person_emp_exp	Years of employment experience	Integer
person_home_ownership	Home ownership status (e.g., rent, own, mortgage)	Categorical
loan_amnt	Loan amount requested	Float
loan_intent	Purpose of the loan	Categorical
loan_int_rate	Loan interest rate	Float
loan_percent_income	Loan amount as a percentage of annual income	Float
cb_person_cred_hist_length	Length of credit history in years	Float
credit_score	Credit score of the person	Integer
previous_loan_defaults_on_file	Indicator of previous loan defaults	Categorical

Table 1: Feature details of the dataset.

Preprocessing

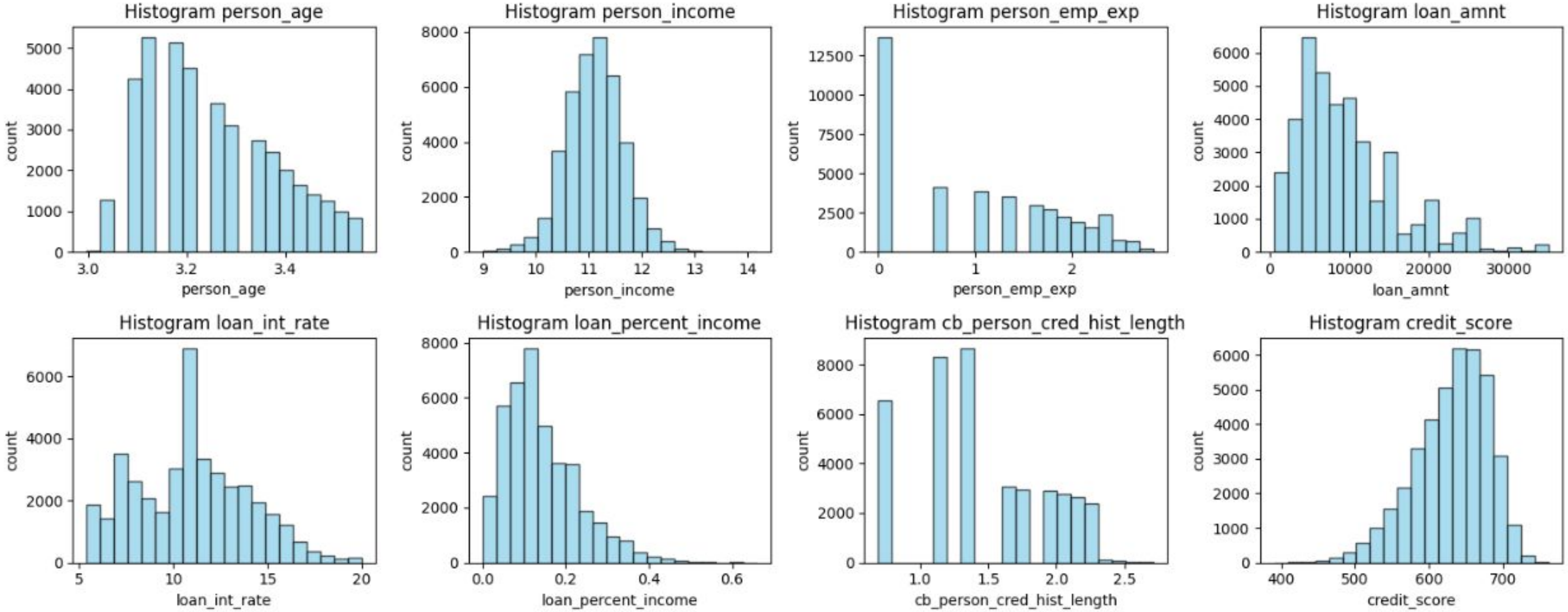
Problem 1: Outliers

- Detected applicants over 100 years old.
- Identified long tails in distributions.



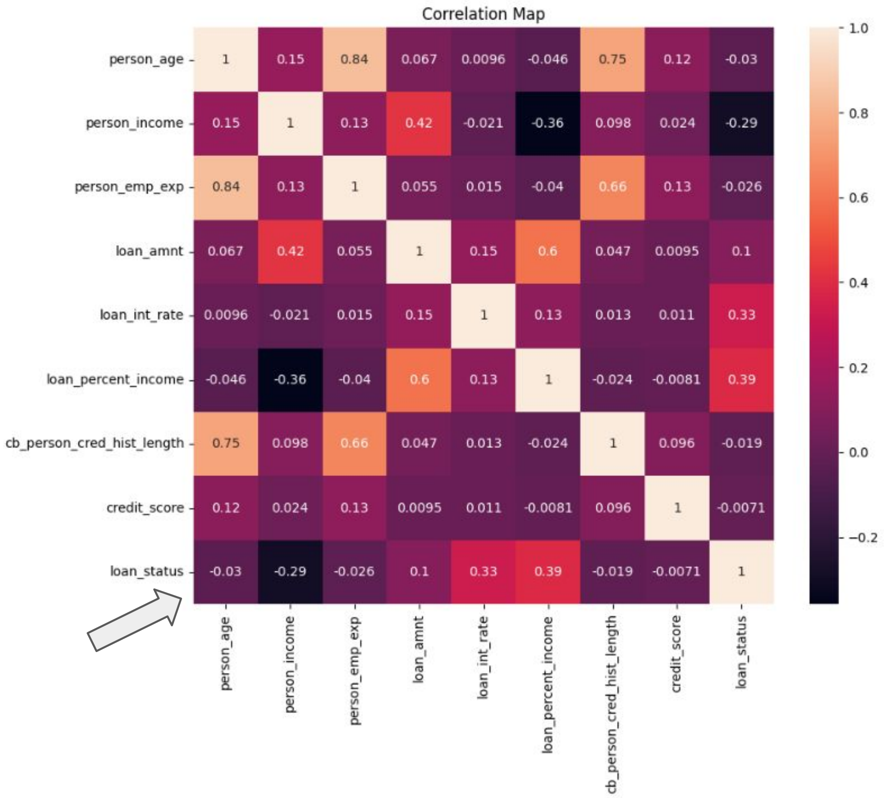
Solution 1: Delete outliers and perform logarithm

- Removed applicants above the 90th percentile for age.
- Applied a natural logarithm transformation to long-tailed features.

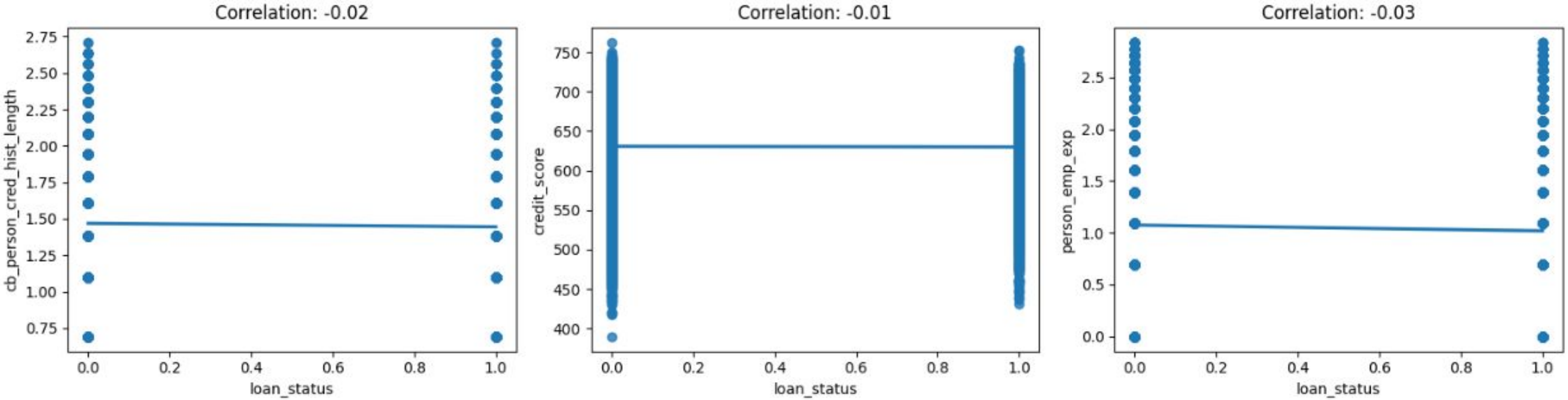


Correlation map of numeric features

- Person_income: Strongly negatively correlated with the target.
- Loan_int_rate & Loan_percent_income: Strongly positively correlated.
- Some features show no correlation with the target



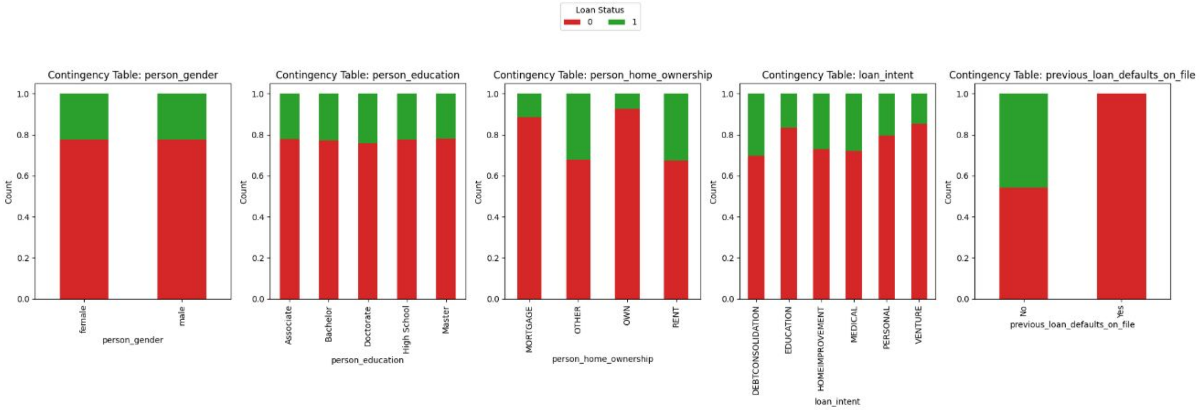
Analysis of “uncorrelated” features



Indeed not related.

Analysis of categorical features

- Contingency tables are calculated.
- Gender and education do not affect the target.



Numeric features

- Binarized using an inter-ordinal scale with balanced thresholds from X_train.

```
AgeCategory
20-23    8740
>28      8425
23-25    7639
25-28    7607
Name: count, dtype: int64
```

Categorical features

- Applied nominal scaling for convenience, ensuring no overlap between categories.

previous_loan_defaults_on_file_No	previous_loan_defaults_on_file_Yes
1	0
0	1
1	0
1	0
1	0
...	...

Classification

Metrics used

- True Positive.
- True Negative.
- False Positive.
- False Negative.
- True Negative Rate (Specificity).
- Negative Predictive Value.
- False Positive Rate.
- False Discovery Rate.
- Accuracy.
- Precision.
- Recall (True Positive Rate).
- F1 Score

Prioritize **F1 Score** due to dataset imbalance, while also calculating other metrics.

Vanilla Lazy FCA

- Why "Lazy"?
  - Classification occurs only when needed, avoiding pre-built global models.
- How?
  - Split training data by target labels.
  - Match patterns by computing intersections with training data:
    - Matches with positive patterns & mismatches with negative patterns count as positive.
    - Matches with negative patterns & mismatches with positive patterns count as negative.
    - If no classifiers match, assign a default class (1).
    - Otherwise, predict the class with the most valid classifiers.

Metric	Value
True Positives	360
True Negatives	12
False Positives	2
False Negatives	240
Accuracy	0.6059
Precision	0.9945
Recall (Sensitivity)	0.6000
Specificity	0.8571
Negative Predictive Value	0.0476
False Positive Rate	0.1429
False Discovery Rate	0.0055
F1 Score	0.7484

Table 2: Performance Metrics of LazyClassifierFCA

Results of state-of-the-art methods

- Unbinarized dataset performs better than binarized version.

Metric	KNN	NB	LR	SVM	DT	RF	XGB
True Positives	333	309	338	343	334	342	338
True Negatives	176	199	189	184	182	183	192
False Positives	29	53	24	19	28	20	24
False Negatives	76	53	63	68	70	69	60
Accuracy	0.8290	0.8274	0.8583	0.8583	0.8404	0.8550	0.8632
Precision	0.9199	0.8536	0.9337	0.9475	0.9227	0.9448	0.9337
Recall (Sensitivity)	0.8142	0.8536	0.8429	0.8345	0.8267	0.8321	0.8492
Specificity	0.8585	0.7897	0.8873	0.9064	0.8667	0.9015	0.8889
Negative Predictive Value	0.6984	0.7897	0.7500	0.7302	0.7222	0.7262	0.7619
False Positive Rate	0.1415	0.2103	0.1127	0.0936	0.1333	0.0985	0.1111
False Discovery Rate	0.0801	0.1464	0.0663	0.0525	0.0773	0.0552	0.0663
F1 Score	0.8638	0.8536	0.8860	0.8875	0.8721	0.8849	0.8895

Table 3: Performance Metrics for Binarized dataset

Metric	KNN	NB	LR	SVM	DT	RF	XGB
TP	343	346	342	346	351	345	337
TN	188	177	182	190	176	186	206
FP	19	16	20	16	11	17	25
FN	64	75	70	62	76	66	46
Accuracy	0.8648	0.8518	0.8534	0.8730	0.8583	0.8648	0.8844
Precision	0.9475	0.9558	0.9448	0.9558	0.9696	0.9530	0.9309
Recall	0.8428	0.8219	0.8301	0.8480	0.8220	0.8394	0.8799
Specificity	0.9082	0.9171	0.9010	0.9223	0.9412	0.9163	0.8918
NPV	0.7460	0.7024	0.7222	0.7540	0.6984	0.7381	0.8175
FPR	0.0918	0.0829	0.0990	0.0777	0.0588	0.0837	0.1082
FDR	0.0525	0.0442	0.0552	0.0442	0.0304	0.0470	0.0691
F1 Score	0.8921	0.8838	0.8837	0.8987	0.8897	0.8926	0.9047

Table 4: Performance Metrics for Unbinarized dataset

Lazy FCA Updated

Improvement Directions

- Allow **approximate matches** using similarity thresholds.
- Manually set **minimum cardinality** and **maximum counter-examples**.
- Apply **class weights** to address imbalances.

Hyperparameter introduction

- Maximum Counter-Examples
- Minimum Cardinality
- Threshold for positive class
- Threshold for negative class
- Class weight

Vanilla Lazy FCA vs. Optimized Lazy FCA

Metric	Previous Value	Updated Value
True Positives	360	**344**
True Negatives	12	**188**
False Positives	2	**18**
False Negatives	240	**64**
Accuracy	0.6059	**0.8664**
Precision	0.9945	**0.9503**
Recall (Sensitivity)	0.6000	**0.8431**
Specificity	0.8571	**0.9126**
Negative Predictive Value	0.0476	**0.7460**
False Positive Rate	0.1429	**0.0874**
False Discovery Rate	0.0055	**0.0497**
F1 Score	0.7484	**0.8935**

Table 5: Comparison of Previous and Updated Performance Metrics of LazyClassifierFCA

Comparison with other models (Binarized dataset)

Metric	KNN	NB	LR	SVM	DT	RF	XGB	LazyClassifierFCA
True Positives	333	309	338	343	334	342	338	**344**
True Negatives	176	199	189	184	182	183	192	**188**
False Positives	29	53	24	19	28	20	24	**18**
False Negatives	76	53	63	68	70	69	60	**64**
Accuracy	0.8290	0.8274	0.8583	0.8583	0.8404	0.8550	0.8632	**0.8664**
Precision	0.9199	0.8536	0.9337	0.9475	0.9227	0.9448	0.9337	**0.9503**
Recall (Sensitivity)	0.8142	0.8536	0.8429	0.8345	0.8267	0.8321	0.8492	**0.8431**
Specificity	0.8585	0.7897	0.8873	0.9064	0.8667	0.9015	0.8889	**0.9126**
Negative Predictive Value	0.6984	0.7897	0.7500	0.7302	0.7222	0.7262	0.7619	**0.7460**
False Positive Rate	0.1415	0.2103	0.1127	0.0936	0.1333	0.0985	0.1111	**0.0874**
False Discovery Rate	0.0801	0.1464	0.0663	0.0525	0.0773	0.0552	0.0663	**0.0497**
F1 Score	0.8638	0.8536	0.8860	0.8875	0.8721	0.8849	0.8895	**0.8935**

Table 6: Performance Metrics for Binarized Dataset Including LazyClassifierFCA

Comparison with other models (Unbinarized dataset)

Metric	KNN	NB	LR	SVM	DT	RF	XGB	LazyClassifierFCA
TP	343	346	342	346	351	345	337	**344**
TN	188	177	182	190	176	186	206	**188**
FP	19	16	20	16	11	17	25	**18**
FN	64	75	70	62	76	66	46	**64**
Accuracy	0.8648	0.8518	0.8534	0.8730	0.8583	0.8648	0.8844	**0.8664**
Precision	0.9475	0.9558	0.9448	0.9558	0.9696	0.9530	0.9309	**0.9503**
Recall	0.8428	0.8219	0.8301	0.8480	0.8220	0.8394	0.8799	**0.8431**
Specificity	0.9082	0.9171	0.9010	0.9223	0.9412	0.9163	0.8918	**0.9126**
NPV	0.7460	0.7024	0.7222	0.7540	0.6984	0.7381	0.8175	**0.7460**
FPR	0.0918	0.0829	0.0990	0.0777	0.0588	0.0837	0.1082	**0.0874**
FDR	0.0525	0.0442	0.0552	0.0442	0.0304	0.0470	0.0691	**0.0497**
F1 Score	0.8921	0.8838	0.8837	0.8987	0.8897	0.8926	0.9047	**0.8935**

Table 7: Performance Metrics for Unbinarized Dataset Including LazyClassifierFCA

Conclusion

- Preprocessed dataset:
  - Deleted outliers
  - Performed logarithm transformation for distributions with long tails.
- Conducted EDA:
  - Analyzed correlation between numeric features.
  - Analyzed contingency table of categorical features.
  - Deleted unrelated features.
- Performed binarization:
  - Numeric features: inter-ordinal scaling.
  - Categorical features: nominal scaling.
- Resplitted the binarized dataset:
  - A smaller dataset of about 3000 records is sampled from the original dataset.
  - The class distribution is almost balanced.
- Simple Lazy FCA algorithm is implemented
- Other state-of-the-art methods are performed
- Lazy FCA algorithm is improved
- The result is comparable with state-of-the-art methods