Banking Data Analysis

Data Mining Case Study

Extração de Conhecimento e Aprendizagem Computacional

2015 / 2016

Francisco Maciel
Hugo Sousa

Domain

The domain represents a simplified banking schema, with data related to clients and accounts.

There are 4500 accounts, which can be accessed by one or more clients, although only one of them is the owner. Demographic information is linked with each client, and any client can have a credit card.

Records are kept for every transaction, but some standard fixed expenses are also represented as static payment orders.

There are 682 accounts who asked for loans from the bank, who has important information regarding the outcome of the loan contract.

Data Preparation

Data cleaning/formatting

Client:

- Split *birth_number* in *gender* and *birth_date_real* columns.
- Transformation from “yyMMdd” to “yyyyMMdd”, otherwise Rapidminer assumes future dates (20xx).

District:

- Replace one district's’ missing values of *unemployment_95* and *crimes_95* by the average of the other districts

Payment Order:

- Replace missing *k_symbol* by value “generic”

Data Preparation

Data construction

New fields were extracted from transactions and payment orders to the account, adding new data possibly relevant for further analysis, whether descriptive or predictive.

New processed fields:

- avg_mensal_expenses - from permanent order (which are **not** loans)
- avg_other_mensal_expenses - from transactions
- avg_mensal_income - from transactions
- avg_transaction_count - from transactions
- min_balance and max_balance - from transactions

Fields from the client table:

- owner_age
- owner_cc_type

Note: Only the account owner is considered as the account client (simplification assumption)

Exploratory Analysis

Correlations in supplied data

- **Districts** with more *inhabitants* and *ratio_urban* tend to have a higher *average_salary* (see Annex A).
- **Loan** amount is related heavily with the loan duration and the individual payment value (see Annex A).

Correlations in constructed data

- The mensal income is highly related to the expenses of an account. This means that the more a client has, the more he will spend (see Annex A).

Exploratory Analysis

Loans analysis

There are 682 accounts with loans (15% of the 4500), and only one loan per account. This means that, for loan predictive and descriptive analysis only few accounts can be considered. The loan contract outcome (status) also varies:

Loan status:

- A - 203/682 (30%)
- B - 31/682 (5%)
- C - 403/682 (59%)
- D - 45/682 (6%)

Loan status (binary):

- Y (A or C) - 606/682 (89%)
- N (B or D) - 76/682 (11%)

Descriptive Task

Who asks for loans?

Descriptive: Problem Definition

Clustering
Understand the profile of clients who ask for loans, regardless of the outcome of the contract (*status*). For this problem, all 4500 accounts in the dataset are considered, having 15% (682 accounts) the *status* of “LOAN” and the remainder 85% (3818 accounts) the *status* of “NONE”.

Is it possible to cluster this accounts? Or will clustering algorithms disregard loan contract outcome finding no significant differences?

What makes clients ask for a loan and what are their characteristics?

Descriptive: Data Selection & Preparation

District data was not found to be as relevant as account data regarding transactions, consistently resulting in similar cluster centroids.

Fields found relevant in the clustering:

- frequency
- owner_cc_type
- owner_age
- avg_mensal_expenses
- avg_other_mensal_expenses
- avg_transaction_count
- avg_mensal_income
- min_balance
- max_balance
- status (binarized to LOAN/NONE)

Descriptive: Experimental Setup

The used algorithm was k-means (k = *2*; measure types = *BregmanDivergences*). A centroid-based algorithm was useful because we wanted to previously specify the number of clusters to address our problem. The results confirm the main differences between these 2 kinds of clients.

Why k = 2?
Because we want to agglomerate 2 kinds of clients: the ones who have loans and the ones who don’ t. This matches the clustering done by k-means.

Other algorithms
k-medoids creates clusters with equal values of loan status. Thus, it’s not suitable for this problem (see Annex C).

Other types of algorithms (non centroid-based) generate lots of clusters and so, don’t fit this problem. (see Annex D - DBSCAN).

Descriptive: Results (Annex B)

Credit Card	Accounts with loan: most owners have a credit card, which varies in a similar way among the 3 existent types (junior, classic and gold). Accounts without loan: most of the owners don’t have a credit card, this could mean that only clients who are **compromised** with the bank ask for loans. The bank can take advantage of this to increase client commitment by **encouraging credit card ownership**.
Frequency of issuance statements	Most accounts with loan have weekly or after transaction issuances, while most accounts without loan have monthly issuances.
Owner age	Accounts with loan have younger owners (41.7) compared to accounts without loan (47.7).
Transactions	Accounts with loan have higher average mensal fixed expenses and considerably higher average mensal income, other mensal expenses and number of transactions, compared to accounts without loan.
Maximum & Minimum balance	Accounts with loan have considerably higher maximum balance and lower minimum balance (thus the need of a loan) compared to accounts without loan.

Predictive Task

Is a loan going to be paid according to contract?

Predictive: Problem Definition

Contrary to the descriptive problem that analyzes all accounts, this problem considers only the accounts that have asked for a loan (15%). The goal of this task is to create a model that can predict the outcome of a loan contract given the client data, seeking to distinguish success and failure.

This can help the bank managers to decide if they should accept or reject a loan request, based on the client information.

This problem was simplified for the loan outcome, given that the loan status is represented in the dataset as four different results but was binarized to fit a success versus failure prediction. This is also the goal of the Kaggle competition, but adding new loans to the set.

In this problem, the status can be regarded as the current status of a loan, or the future status for a specified loan. Considering the initial data and the new data supplied for the competition, a careful exploration of the transactions data suggest that the status represents the last data in the transactions, considering the amount of the loan that an account has paid in the records.

Predictive: Data Selection & Preparation

To generate the predictive model, all constructed data was joined to the loan data for an initial experimentation. The label column is the *status*, which was converted into a binomial type with the range [N,Y], using replace operations. The *id* fields were removed, as well as the fields in the *loan* table which were directly correlated with each other, such as the *amount* and *duration*, sufficing only one of them for the final decision tree.

This process was split in two methods after the initial approach revealed a problem with the dataset. Ultimately, after adding and removing fields from the subset, the fields found relevant for the predictive task were:

Method 1

- min_balance

Method 2

- avg_mensal_expenses
- avg_other_mensal_expenses

Predictive: Experimental Setup/Results - Method 1

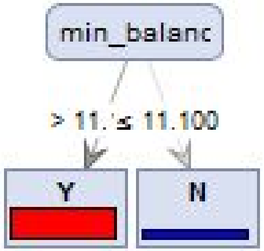
Method 1

Generating the **decision tree** below, we discovered that unsuccessful loans always had a negative *min_balance* and vice-versa. This resulted in a 100% accuracy when used on test data.

Results

100% accuracy (see Annex E).

Resulting decision tree



Predictive: Experimental Setup/Results - Method 1

Prediction vs Causality

The odd nature of this outcome lead us to think that the approach to the problem was not correct. In data mining there are no 100% prediction certainties, and in this case the result can only be explained by the causality of the *min_balance* value. Because the data represents ongoing loans, the clients with negative balance are those whose loans have already failed with the contract. This means that instead of predicting, this model is only describing the outcome. This approach will not work in a real problem, where a new client asks for a loan but has no history of negative balance because he has not failed. The Kaggle competition also had this data in the transactions, and on our first attempt to test if this had been considered we scored 100%. This means that for the competition, no ongoing loans should be on the records, and neither on the transactions. To turn this into a data mining problem we believe that this data shouldn't be accessible when a loan request is done, so we we discarded fields such as the penalties and the negative balance, and used a different approach, providing method 2.

Predictive: Experimental Setup/Results - Method 2

Method 2

Being the data sample small for the negative labels, we opted to use cross-validation with 10 validations and stratified sampling.

Other classification algorithms were also tested. In addition to **decision tree**, **rule induction** and **k-NN** were the ones performing better.

Results

Decision Tree	93.98% +/- 0.82%	see Annex F
Rule Induction	94.31% +/- 0.91%	see Annex G
k-NN	92.11% +/- 1.68%	see Annex H

Conclusions and Limitations

The fact that our descriptive task is directly related to the predictive one helped us to figure out what the most important fields could be and also the discovery of the *min_balance* influence on the loan *status* (method 1). This method is implausible in a real life context, as it uses data posterior to the loan request.

Only 76 (11%) of the loans had a contract with irregularities (*status* “B” or “D”). This means that very few examples are left for the training data and for testing. We used cross-validation and stratified sampling to address this problem, but more irregular loans would be useful for better predictions. Predicting all “Y” on the predictive task actually results in a f-measure score of 94.10%.

Future Work

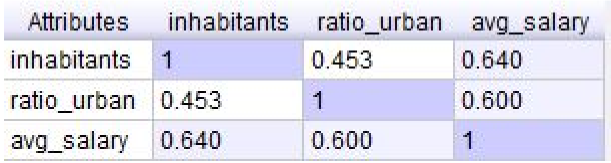
In the future, the descriptive analysis could be taken further to try to find an association rule between having a credit card and asking for a loan, or other factors that could help the bank with the marketing.

To enhance the predictions, more data could be constructed from the transactions, using other statistics such as mean deviation and transaction variance to hopefully find more determining factors.

The data could also be split between before and after a loan to try to compare the differences in behaviour, and use the first half to improve the comparison with new clients.

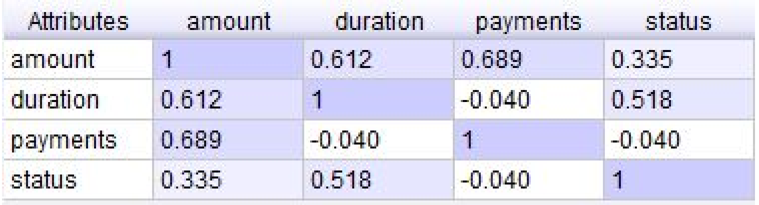
Annexes

A | Exploratory Task: Correlation Matrices



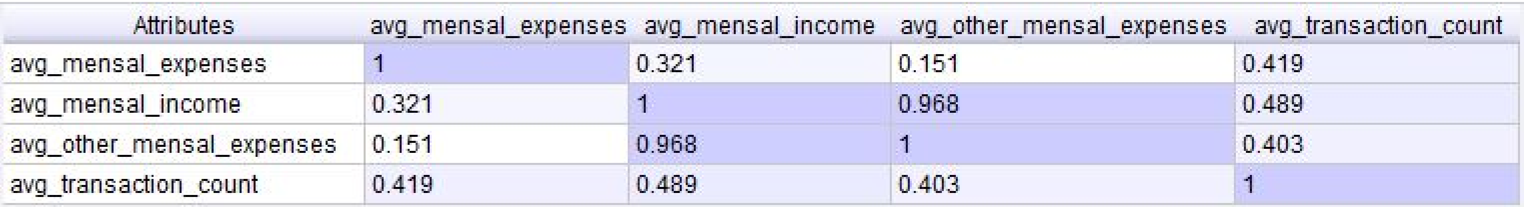
Attributes	inhabitants	ratio_urban	avg_salary
inhabitants	1	0.453	0.640
ratio_urban	0.453	1	0.600
avg_salary	0.640	0.600	1

District Correlation Matrix



Attributes	amount	duration	payments	status
amount	1	0.612	0.689	0.335
duration	0.612	1	-0.040	0.518
payments	0.689	-0.040	1	-0.040
status	0.335	0.518	-0.040	1

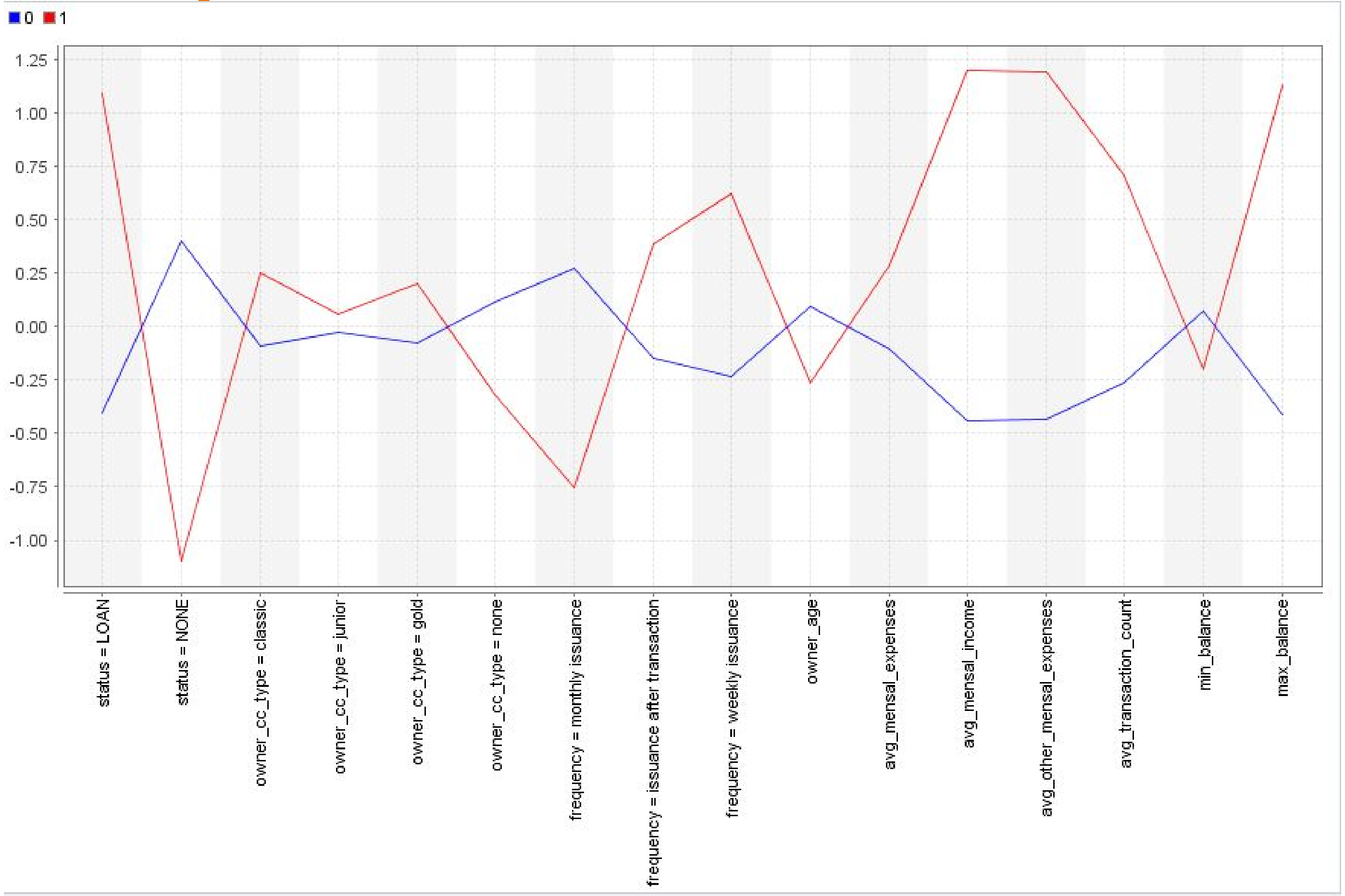
Loan Correlation Matrix



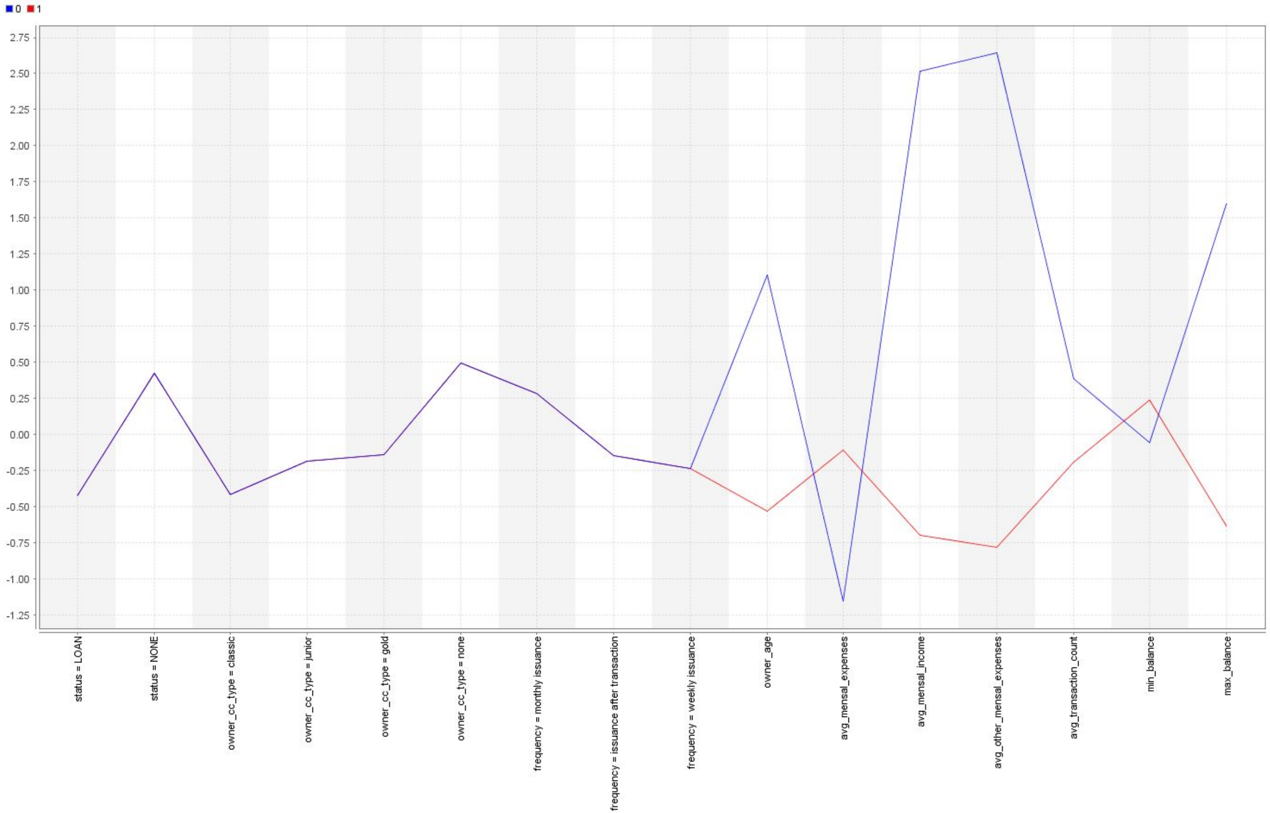
Attributes	avg_mensal_expenses	avg_mensal_income	avg_other_mensal_expenses	avg_transaction_count
avg_mensal_expenses	1	0.321	0.151	0.419
avg_mensal_income	0.321	1	0.968	0.489
avg_other_mensal_expenses	0.151	0.968	1	0.403
avg_transaction_count	0.419	0.489	0.403	1

Account constructed data Correlation Matrix

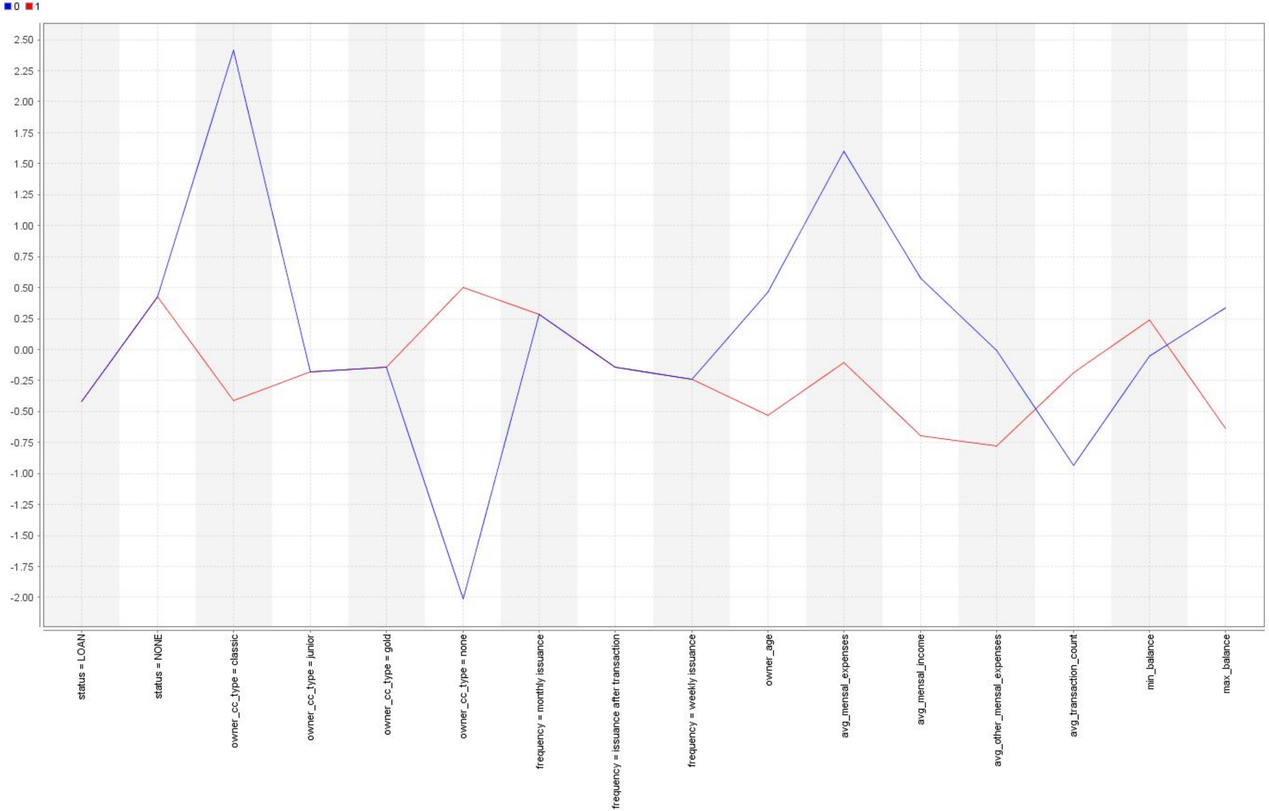
B | Descriptive Task: k-means



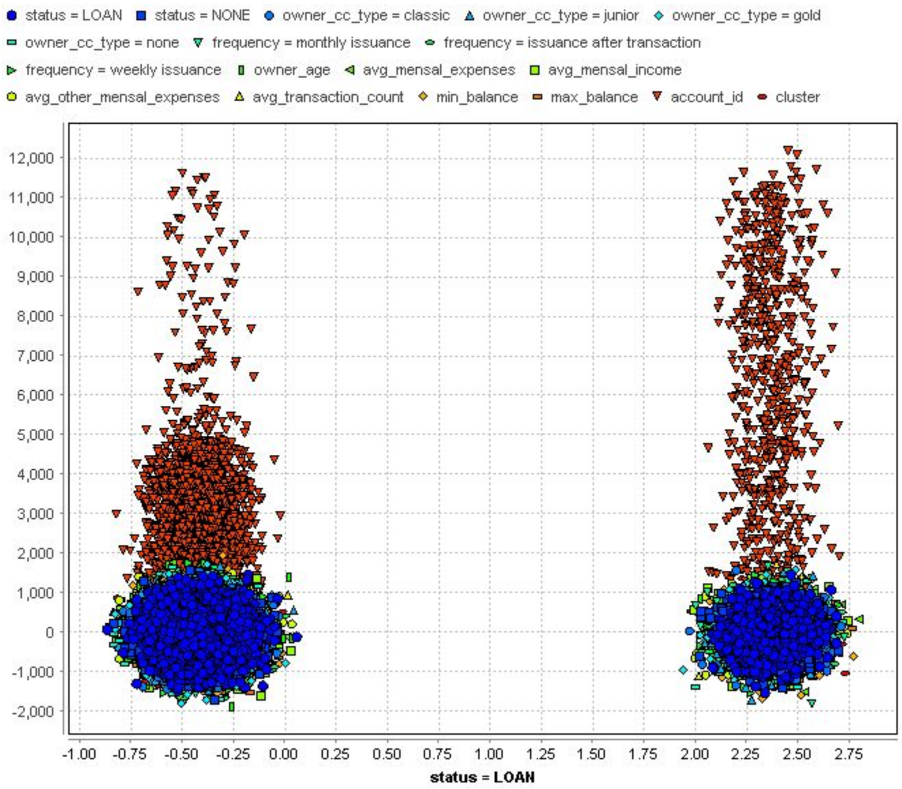
C | Descriptive Task: k-medoids (BergmanDivg.)



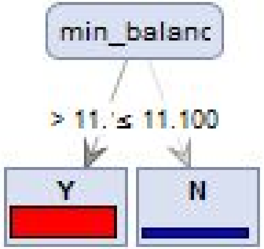
C | Descriptive Task: k-medoids (MixedMeasures)

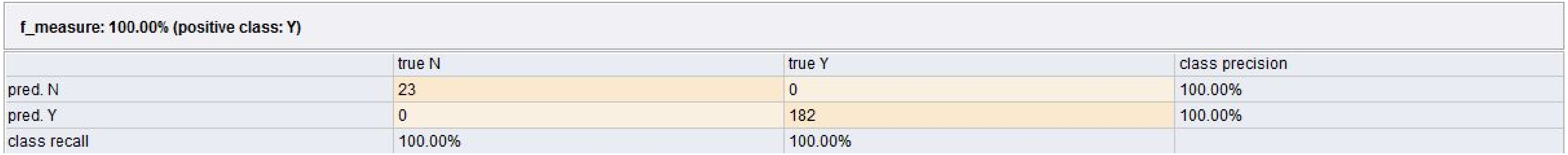


D | Descriptive Task: DBSCAN



E | Predictive Task: Method 1 - Decision Tree

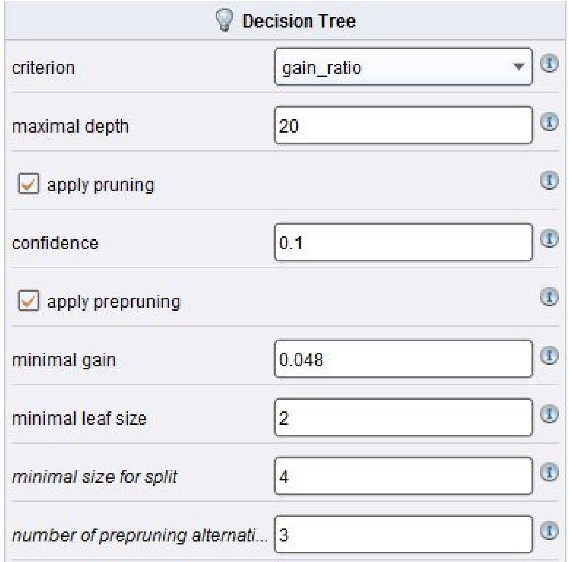




f_measure: 100.00% (positive class: Y)

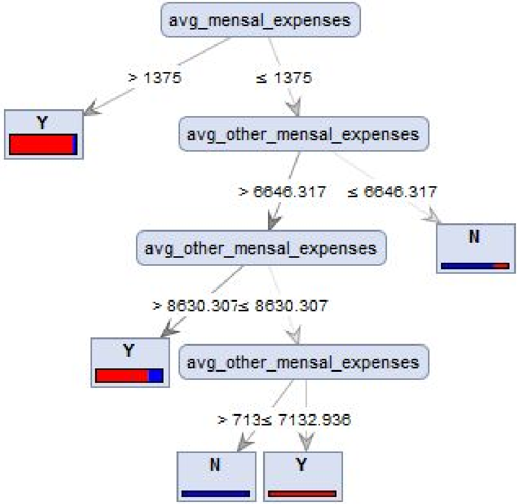
	true N	true Y	class precision
pred. N	23	0	100.00%
pred. Y	0	182	100.00%
class recall	100.00%	100.00%	

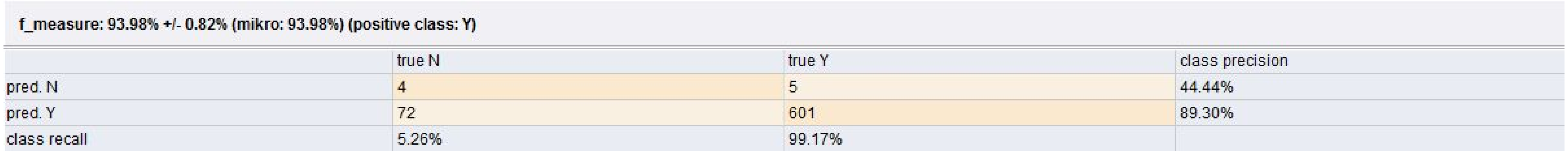
F | Predictive Task: Method 2 - Decision Tree



Decision Tree

criterion	gain_ratio
maximal depth	20
☑ apply pruning	
confidence	0.1
☑ apply prepruning	
minimal gain	0.048
minimal leaf size	2
minimal size for split	4
number of prepruning alternati...	3

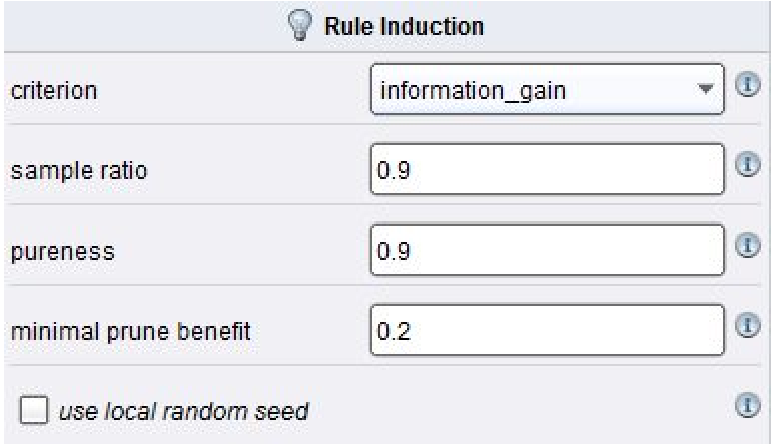




f_measure: 93.98% +/- 0.82% (mikro: 93.98%) (positive class: Y)

	true N	true Y	class precision
pred. N	4	5	44.44%
pred. Y	72	601	89.30%
class recall	5.26%	99.17%	

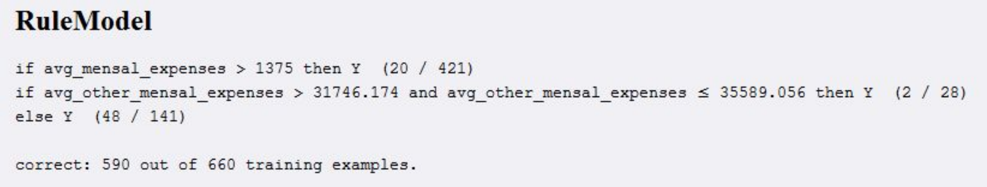
G | Predictive Task: Method 2 - Rule Induction



Rule Induction

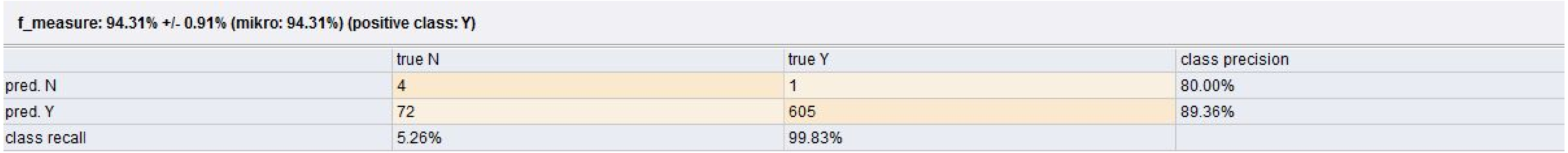
criterion	information_gain ▾
sample ratio	0.9
pureness	0.9
minimal prune benefit	0.2

□ *use local random seed*



RuleModel

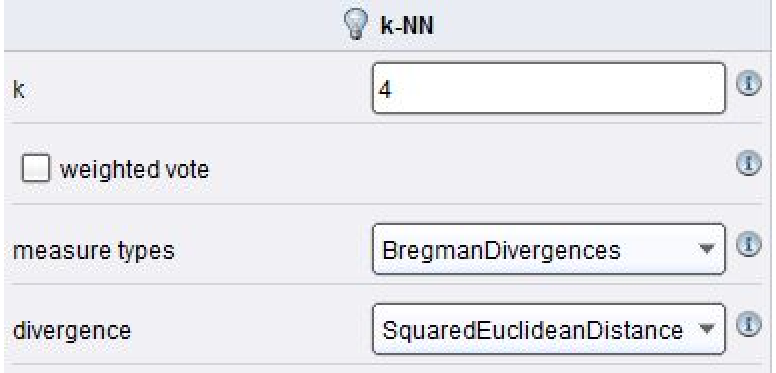
```
if avg_mensal_expenses > 1375 then Y  (20 / 421)
if avg_other_mensal_expenses > 31746.174 and avg_other_mensal_expenses ≤ 35589.056 then Y  (2 / 28)
else Y  (48 / 141)

correct: 590 out of 660 training examples.
```



f_measure: 94.31% +/- 0.91% (mikro: 94.31%) (positive class: Y)

	true N	true Y	class precision
pred. N	4	1	80.00%
pred. Y	72	605	89.36%
class recall	5.26%	99.83%	

H | Predictive Task: Method 2 - k-NN



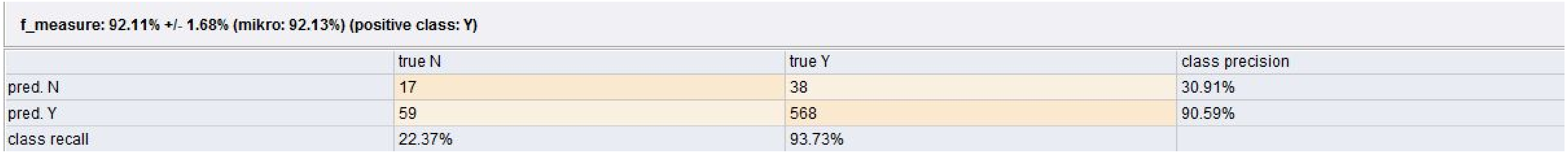
k-NN

k: 4

□ weighted vote

measure types: BregmanDivergences

divergence: SquaredEuclideanDistance



f_measure: 92.11% +/- 1.68% (mikro: 92.13%) (positive class: Y)

	true N	true Y	class precision
pred. N	17	38	30.91%
pred. Y	59	568	90.59%
class recall	22.37%	93.73%	