Banking Prediction

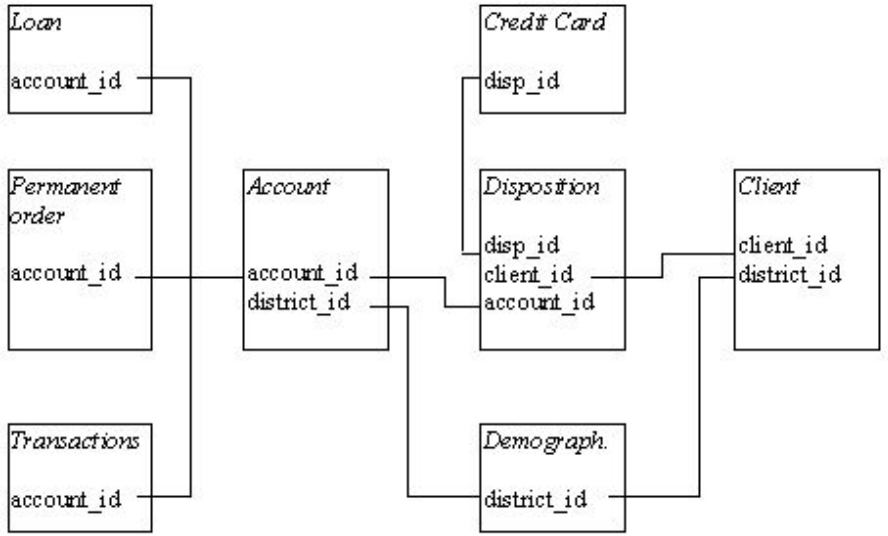
Catarina Ramos - up201406219
Tiago Almeida - up201305665

Domain Description

A bank stores information about its: **clients, accounts, credit cards, transactions** and **loans.**

Goal

Predict if a loan request should be accepted or not.



Descriptive Analysis

Problem Definition

Goal: Understand what happened on the **past**.

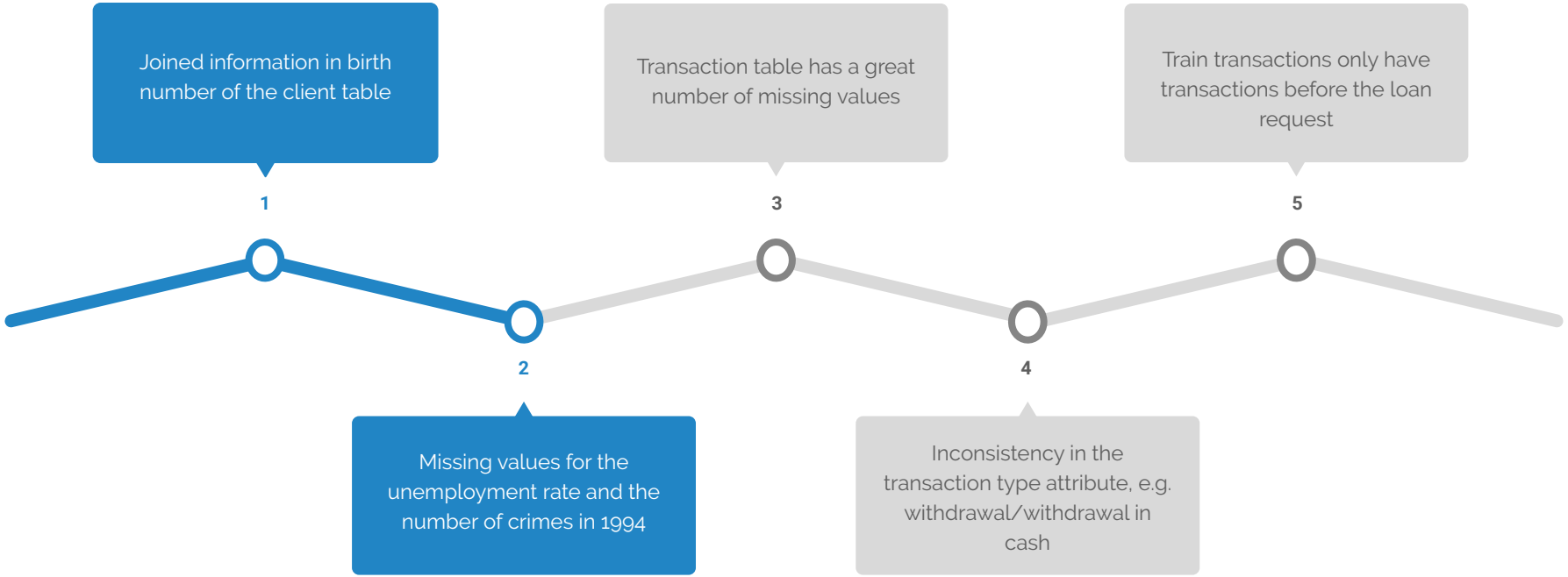
How is data organized ?

How to enhance data ?

What statistics can be obtained ?

Is there a pattern ?

Data Analysis



Data Analysis

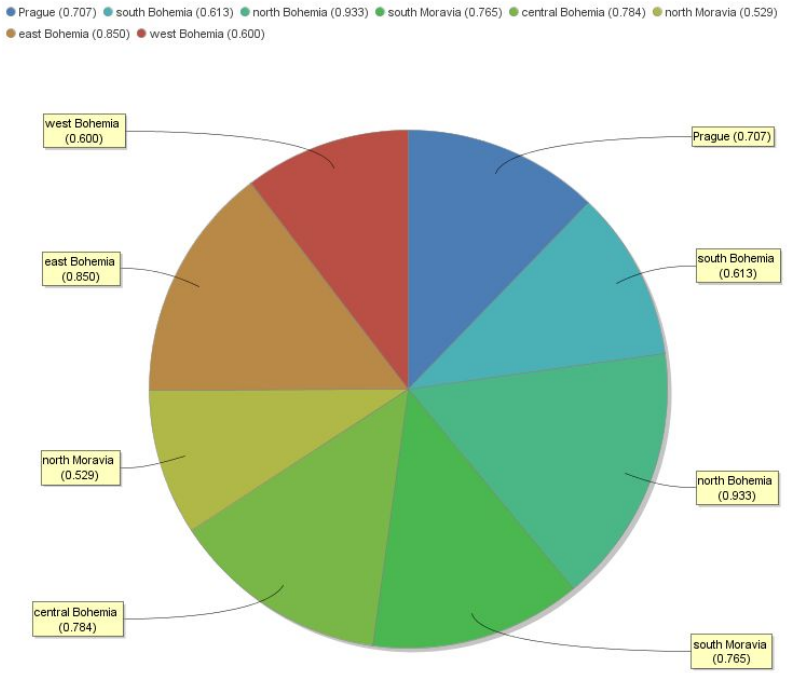
	Number of Attributes	Missing Values	Data Types
Client Table	3	0	Integer; Polynominal
Account Table	4	0	Integer; Polynominal
Card Table	4	0	Integer; Polynominal
Disponent Table	4	0	Integer; Polynominal
District Table	16	2	Integer; Polynominal; Real
Loan Table	7	0	Integer; Polynominal
Transaction Table	10	868969	Integer; Polynominal; Real

Data Preparation

1. Separate birth number information into gender and birthdate
2. Linear regression between the year of 95 and 96 for unemployment rate and crime rate
3. Converted all date attributes to Rapidminer's date type and unemployment and number of crimes to numerical types
4. Created three new attributes: average_credit_amount, avergare_withdrawal_amount, average_balance_per_month

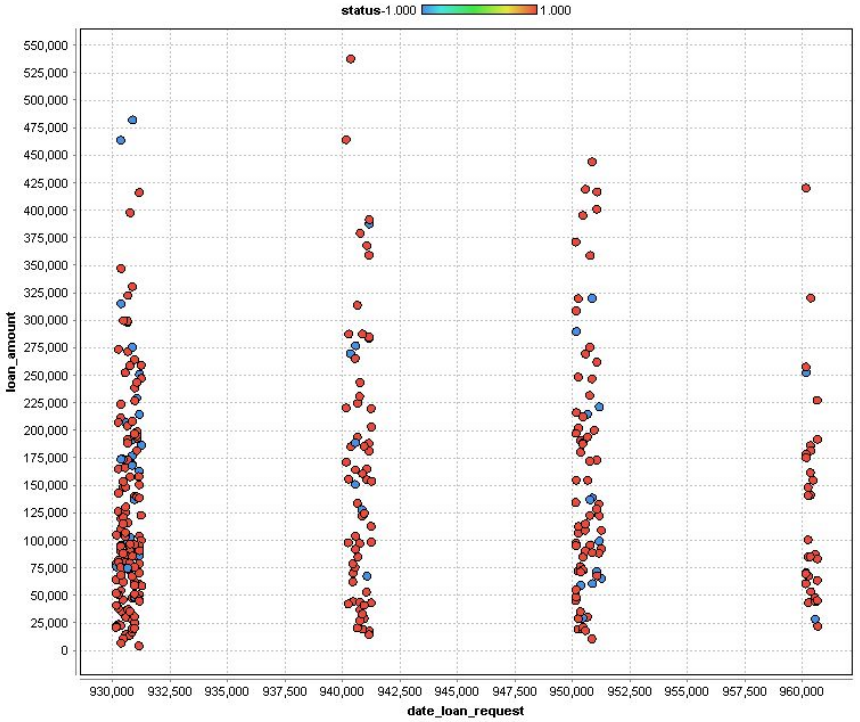
Exploratory Analysis

Data Analysis



Observations:

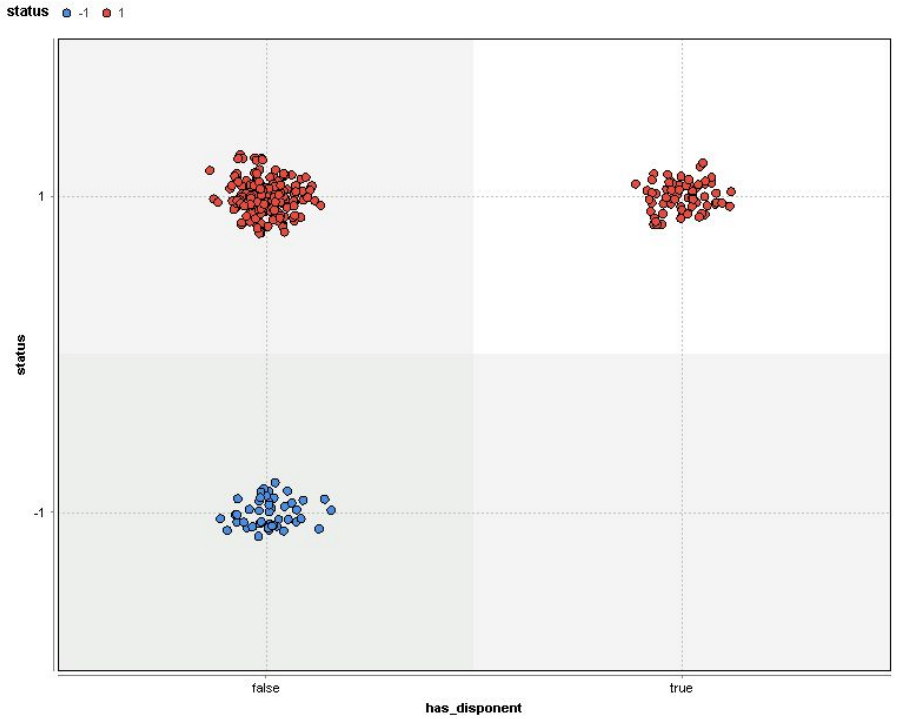
- North Bohemia has the biggest percentage of successful loans.
- The past loans starts at 1993 and end at 1996.

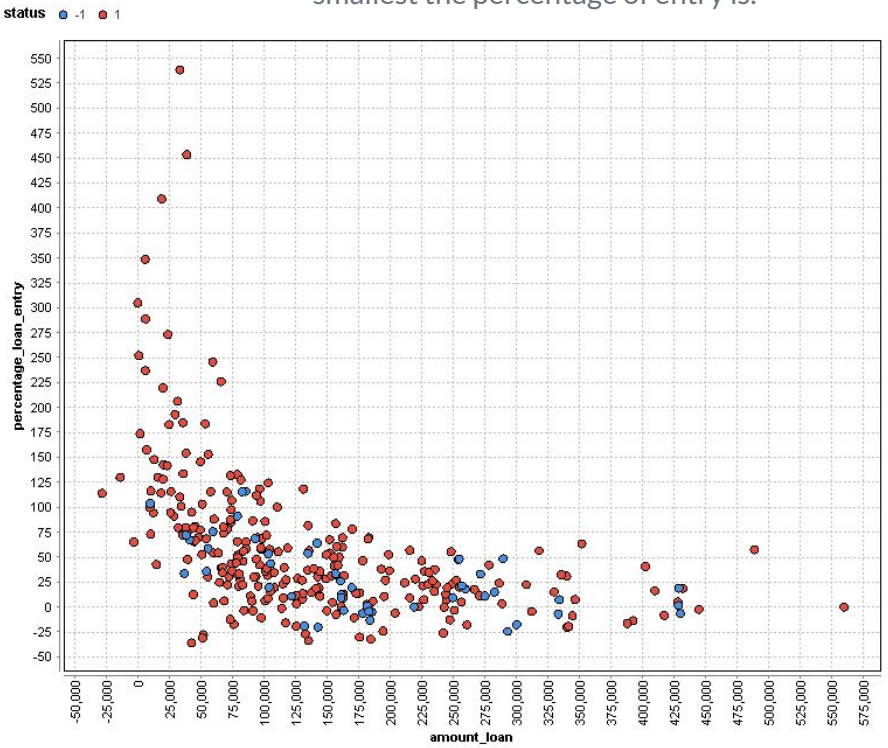


Data Analysis

Observations:

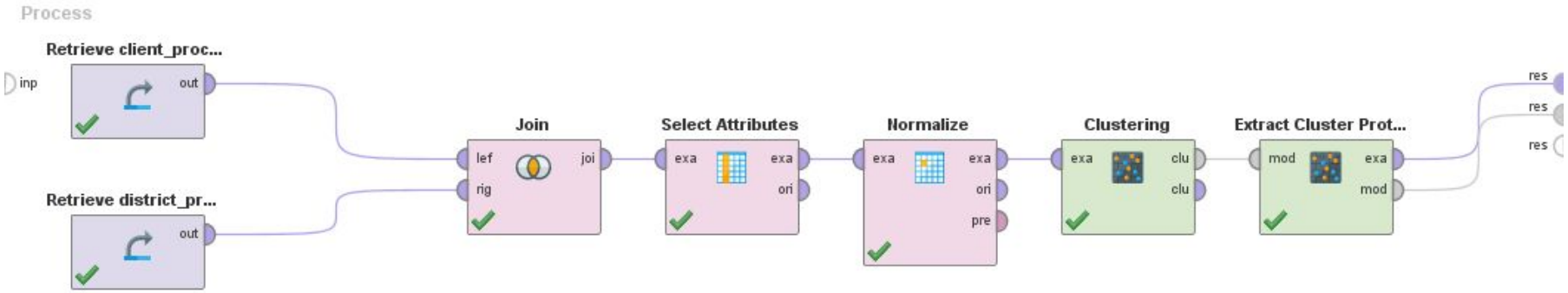
- Accounts with disponent always pay their loans.
- The bigger the loan amount the smallest the percentage of entry is.



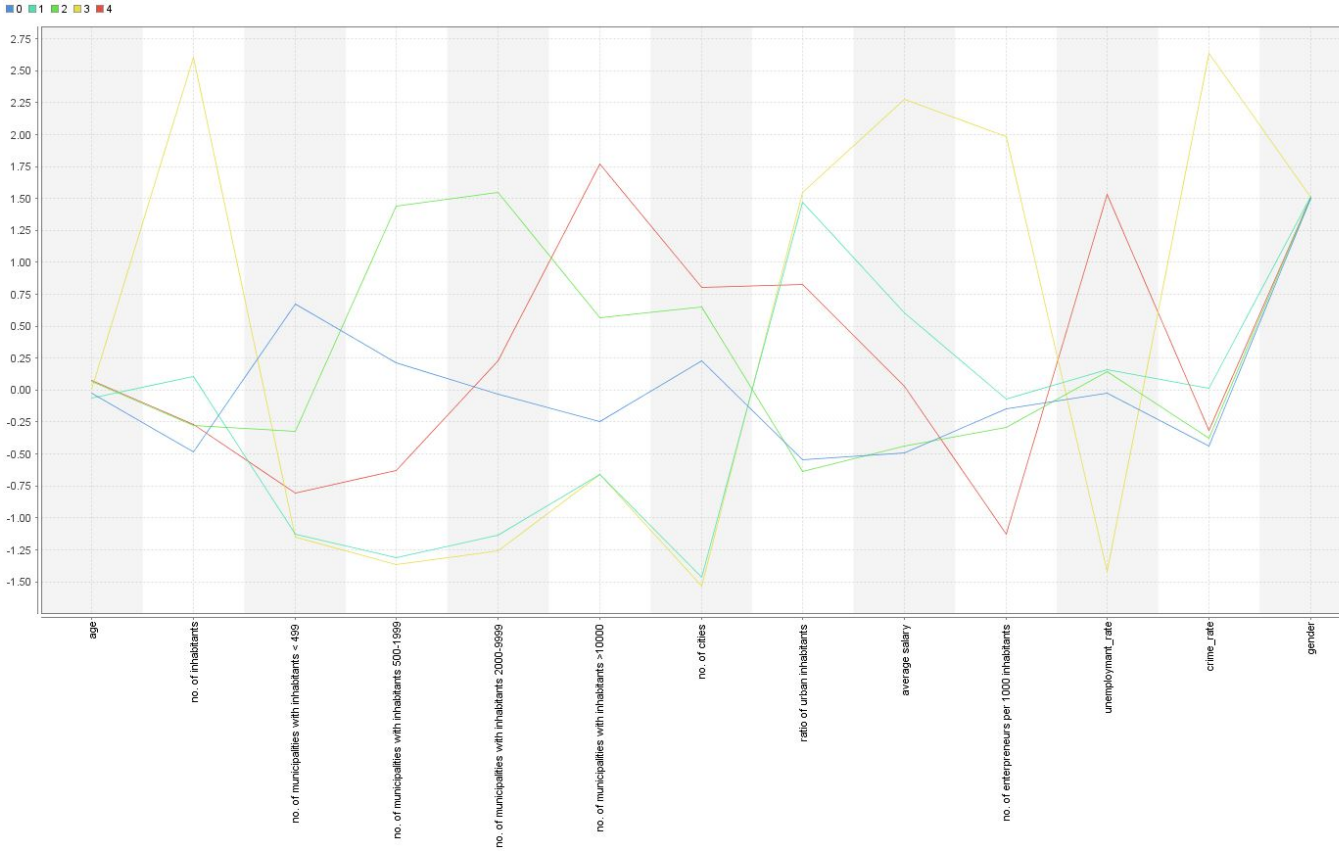


Experimental Setup

Sociodemographic Data



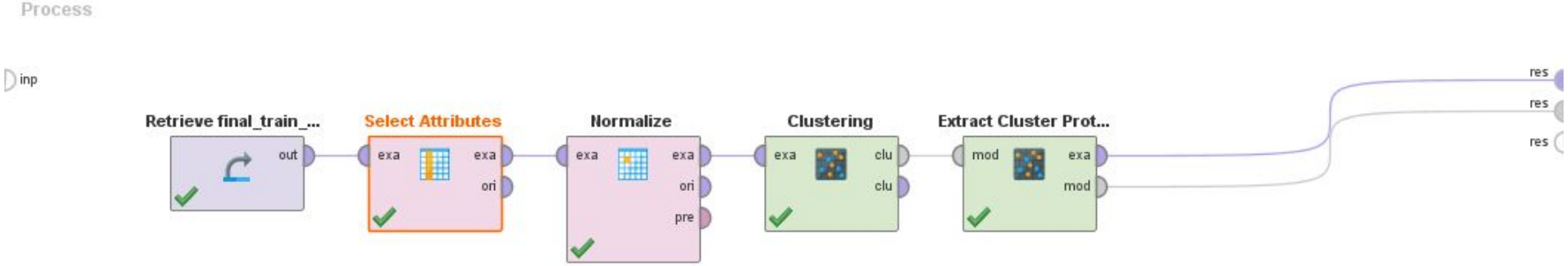
Results



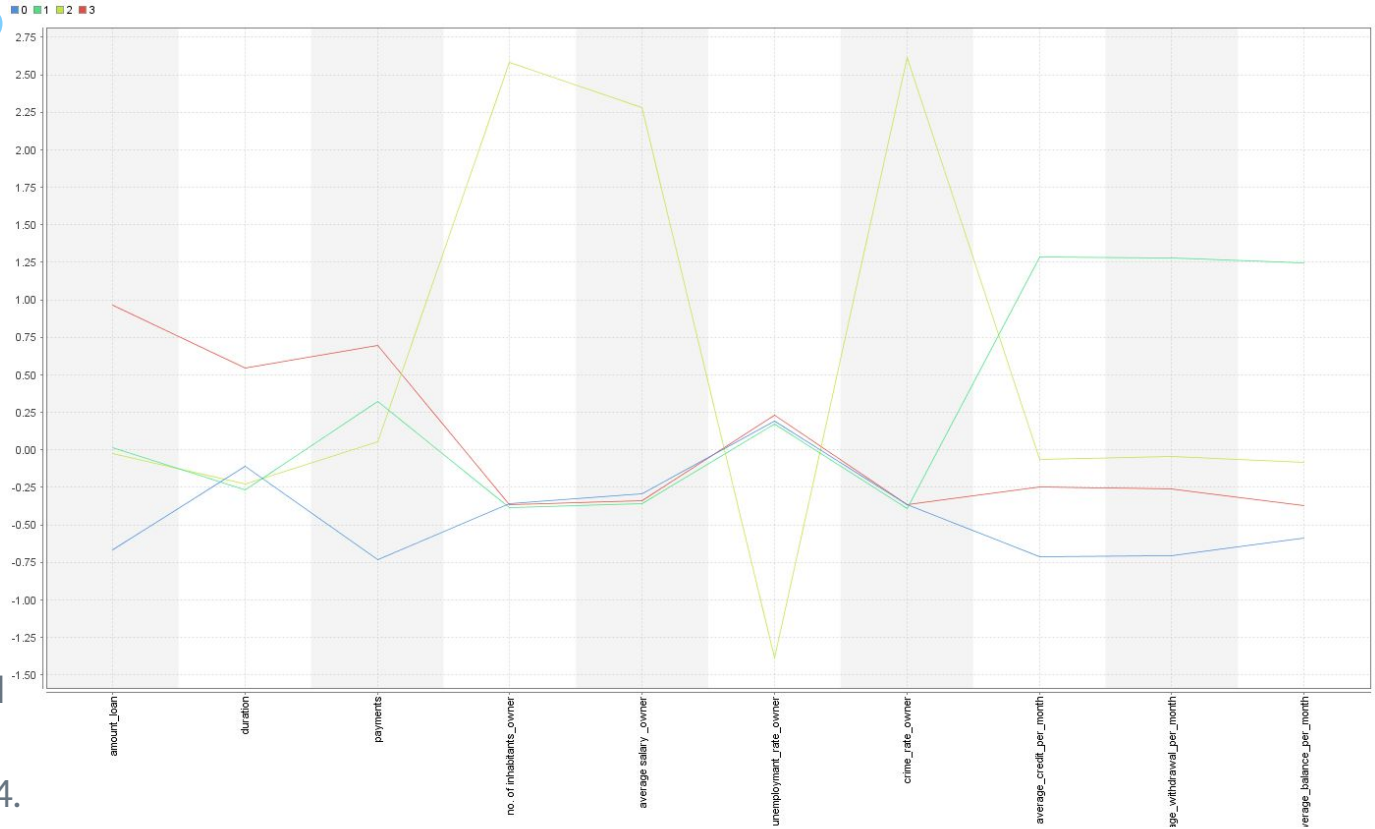
Sociodemographic Data
K-means algorithm with K = 5

Experimental Setup

Sociodemographic-Loan Data



Results



Sociodemographic Data and Loan Information

K-means algorithm with K = 4.

Predictive Analysis

Problem Definition

Goal: Understand what will happen on the future.

Should a loan be granted or not?

Will the loan be successfully paid ?

Data Preparation

Rename Attributes

So that attributes from different tables with the same name wouldn't be lost in the next process.

Join Tables

Join information across all tables into a single and final table.
Consider only transactions before loan.
This is done for the test and train data.

Pre select Attributes

Remove unnecessary information, such as, id's and attributes with a great percentage of missing values.

Data Preparation

Created attributes

More than 70 attributes were created in order to improve the prediction performance.

Removed Correlated Attributes

Rapid Miner's operator 'Remove Correlated Attributes' was used to remove redundant information.

Removed Outliers

The data was normalized for outlier detection and elimination. In the end, the data was de-normalized back to its original state.

New Attributes

Statistics about transactions:

- **minimum, maximum, average, variance, standard deviation credit/withdrawal/balance**
- **difference between credit and withdrawal** (estimated income)
- **last balance** (before loan)

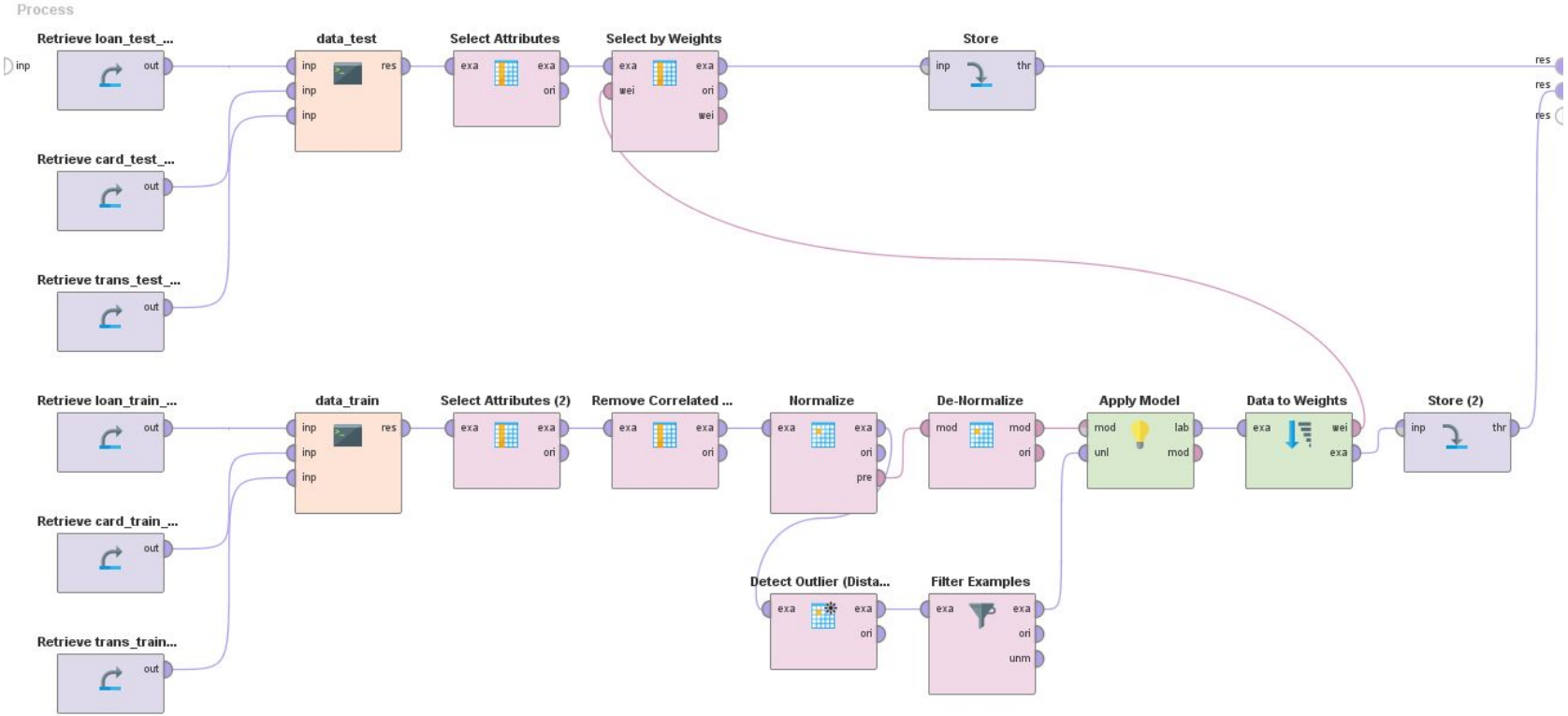
Loan:

- **age of owner when loan was requested**
- **age of disponent when loan was requested** (if no disponent then = 0)
- **has disponent** (boolean)

Other:

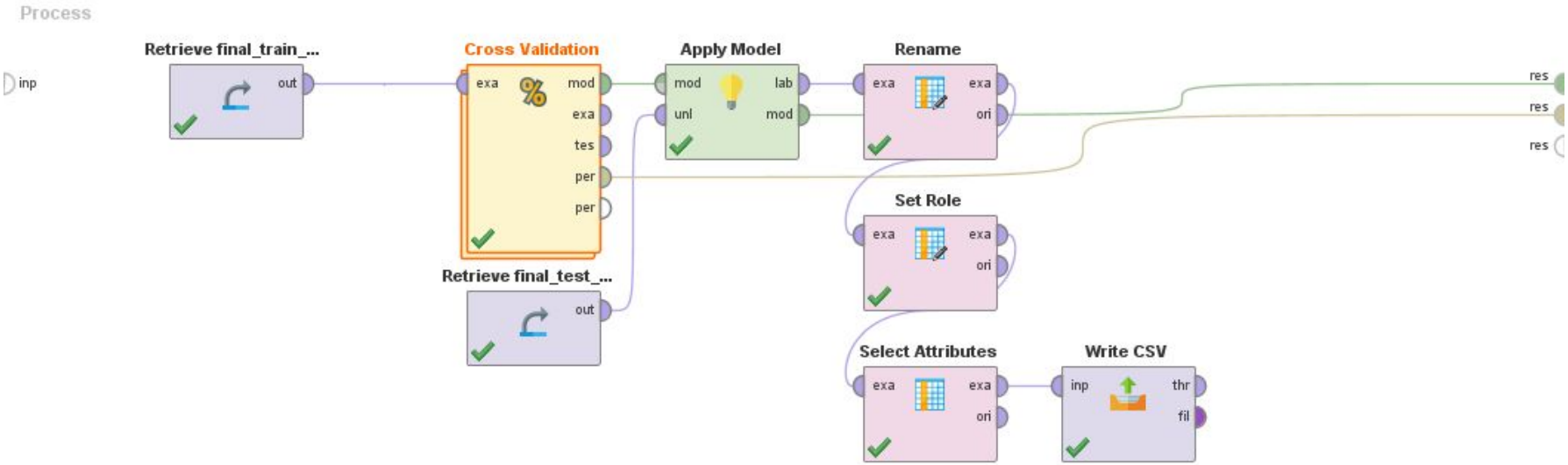
- **percentage of loan entry in bank** (last_balance/loan_amount)
- **percentage of loan payments covered based on average income per month** (payments/diff_credit_withdrawal)
- **money spent on debts on the month before** (sum of remittance to another bank withdrawals)
- **ratio between debts and income** (sum(debts)/diff_credit_withdrawal)
- **estimated balance at the end of the loan** (last_balance + duration*diff_credit_withdrawal-(total_debts_last_month + payments)*duration)

Data Preparation



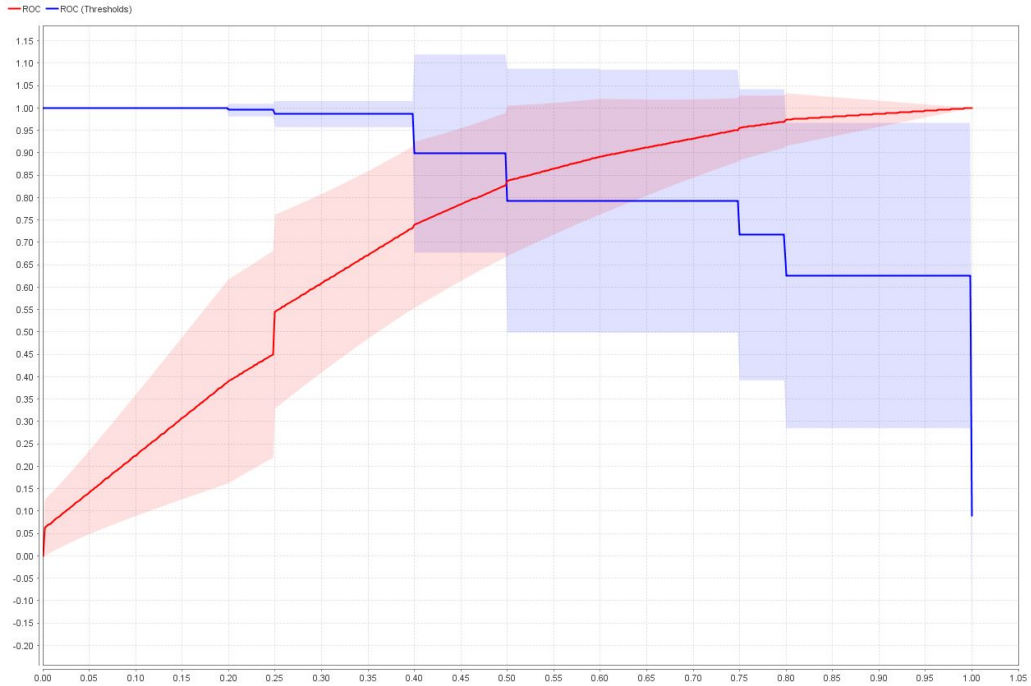
Experimental Setup

Decision Tree



Results

Decision Tree

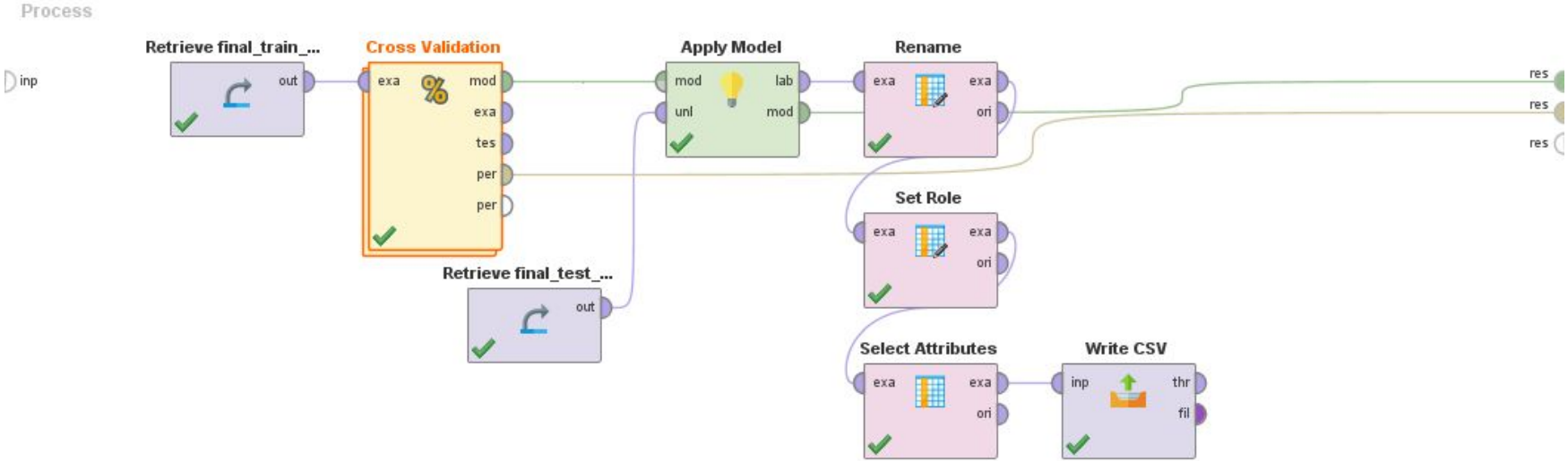


accuracy: 87.82% +/- 2.98% (micro average: 87.83%)

	true -1	true 1	class precision
pred. -1	18	9	66.67%
pred. 1	28	249	89.89%
class recall	39.13%	96.51%	

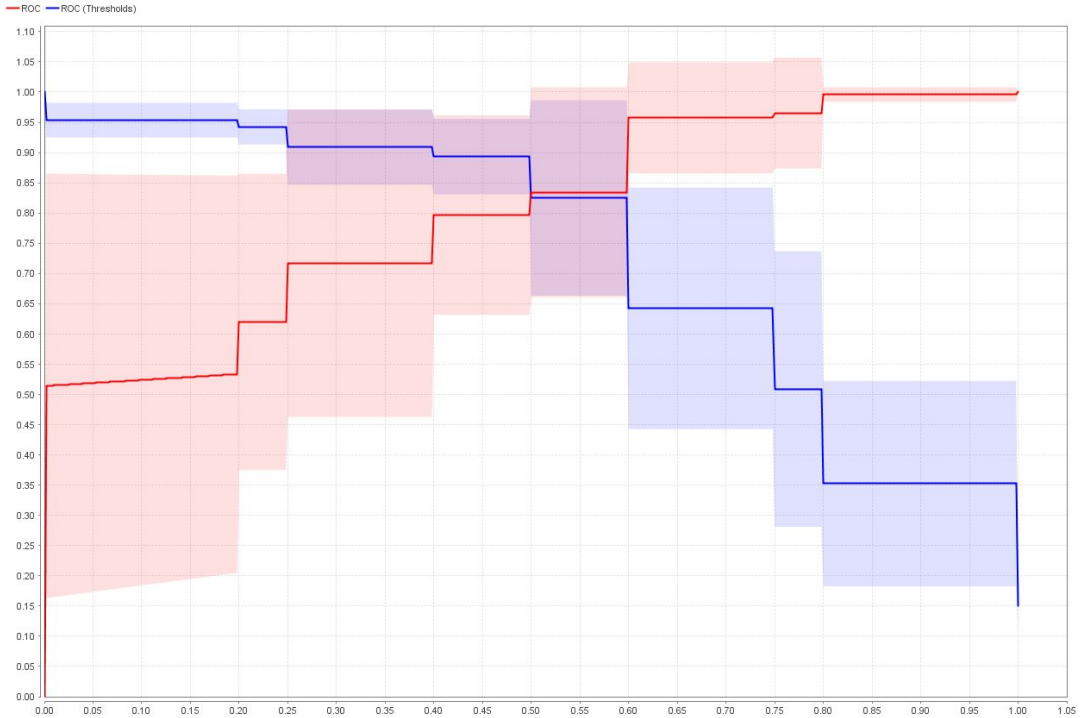
Experimental Setup

Gradient Boosted Trees



Results

Gradient Boosted Trees

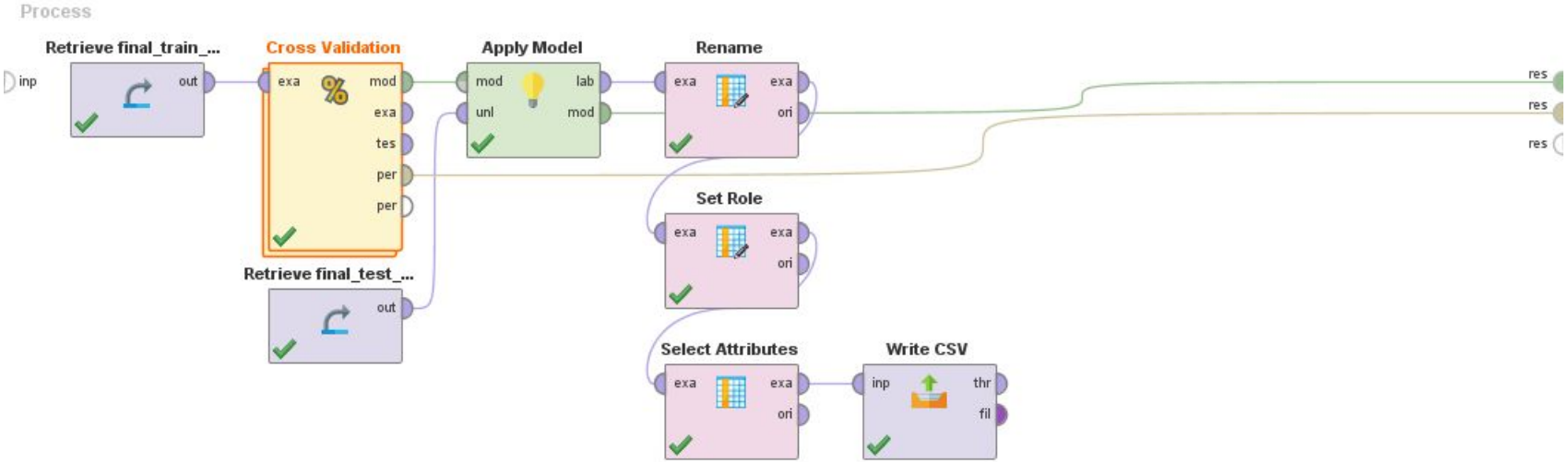


accuracy: 85.87% +/- 9.36% (micro average: 85.86%)

	true -1	true 1	class precision
pred. -1	20	17	54.05%
pred. 1	26	241	90.26%
class recall	43.48%	93.41%	

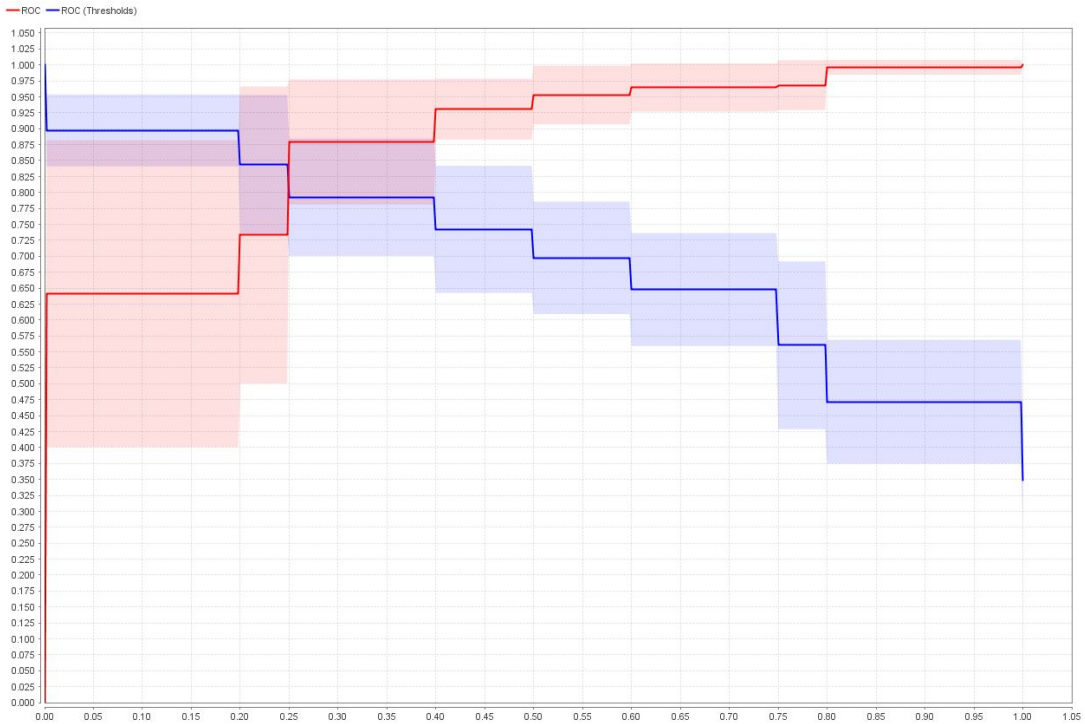
Experimental Setup

Random Forest



Results

Random Forest



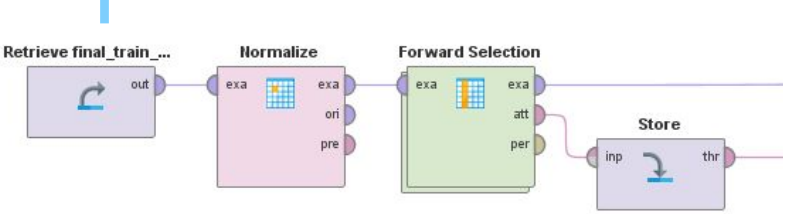
accuracy: 88.45% +/- 3.48% (micro average: 88.49%)

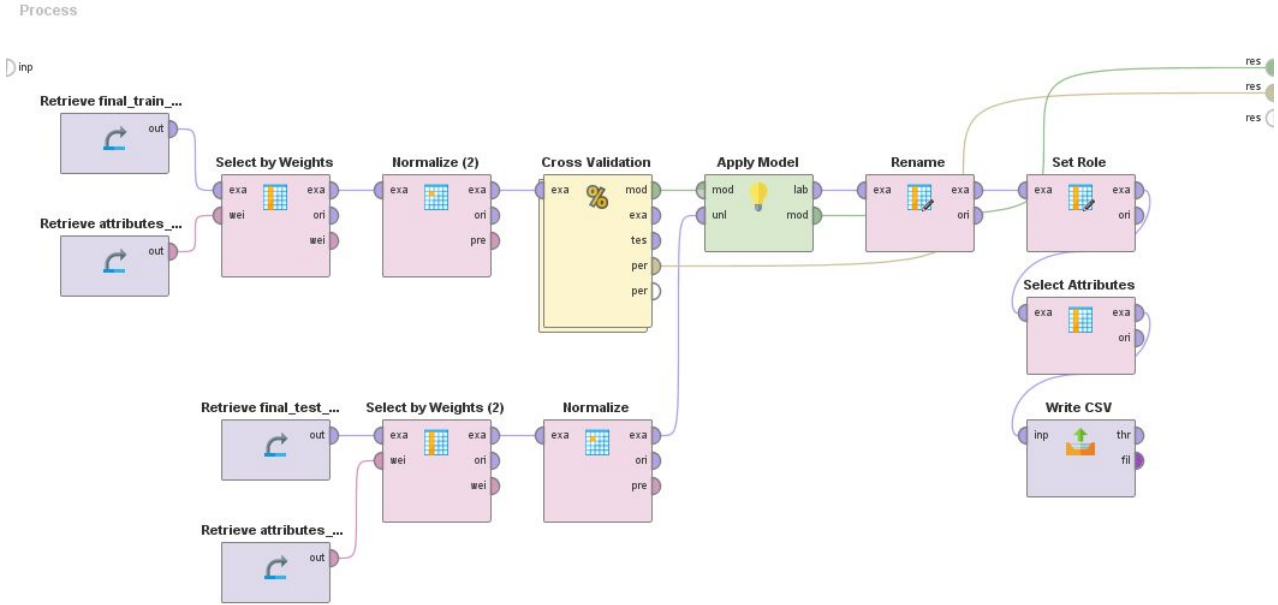
	true -1	true 1	class precision
pred. -1	15	4	78.95%
pred. 1	31	254	89.12%
class recall	32.61%	98.45%	

Experimental Setup

k-NN

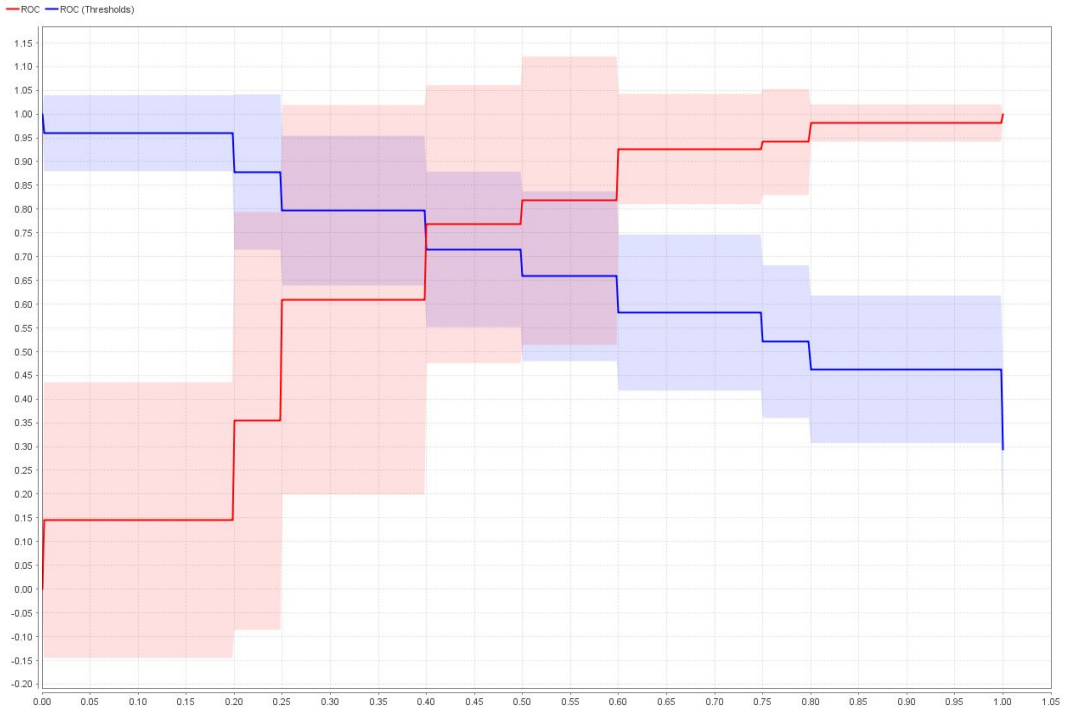
Feature selection →





Results

k-NN



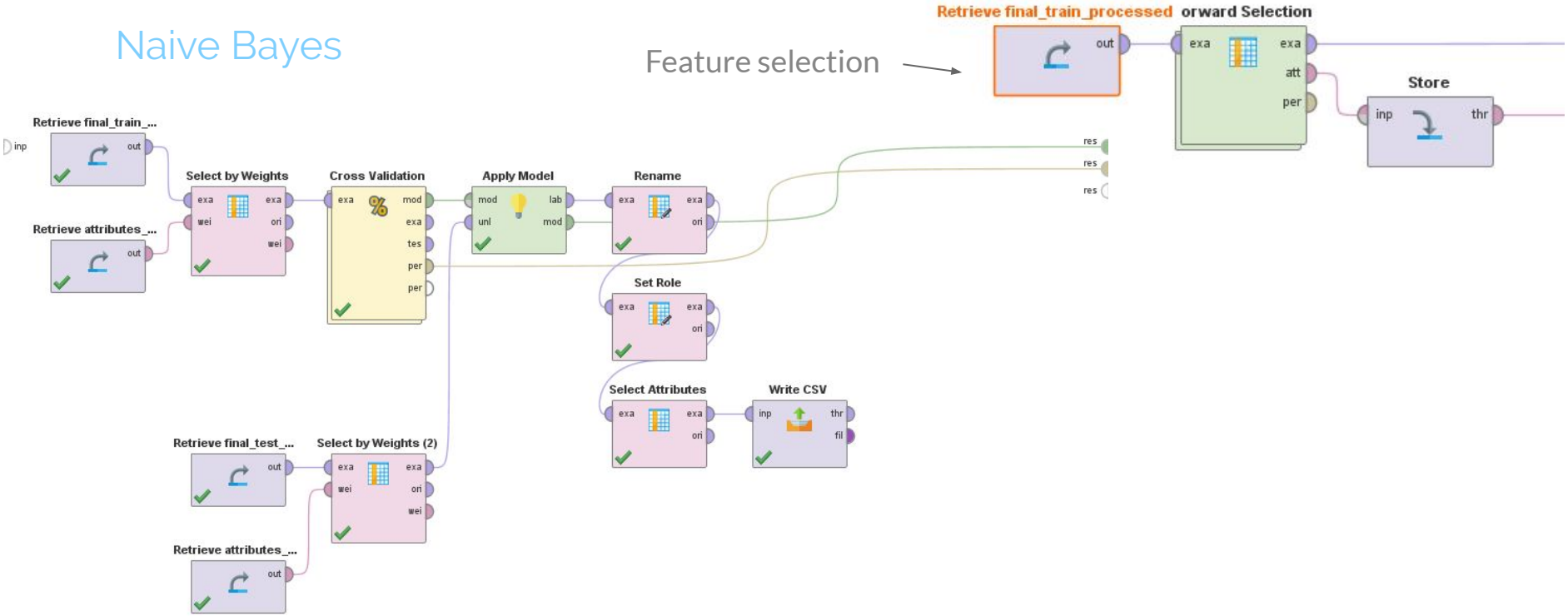
accuracy: 90.15% +/- 4.35% (micro average: 90.13%)

	true -1	true 1	class precision
pred. -1	19	3	86.36%
pred. 1	27	255	90.43%
class recall	41.30%	98.84%	

Experimental Setup

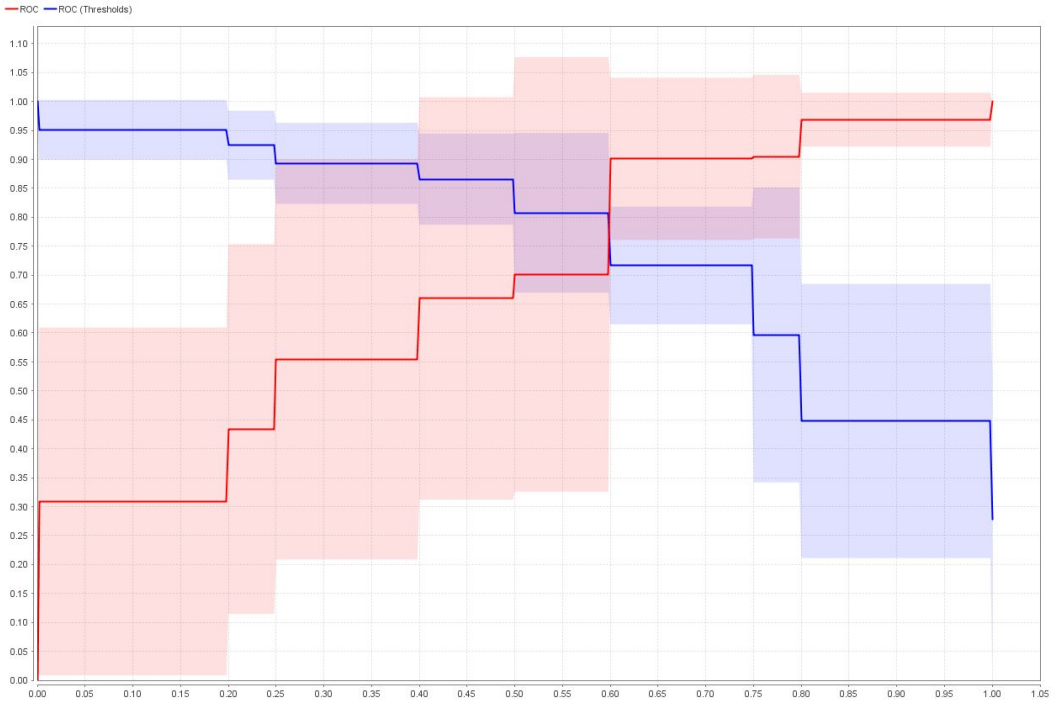
Naive Bayes

Feature selection



Results

Naive Bayes

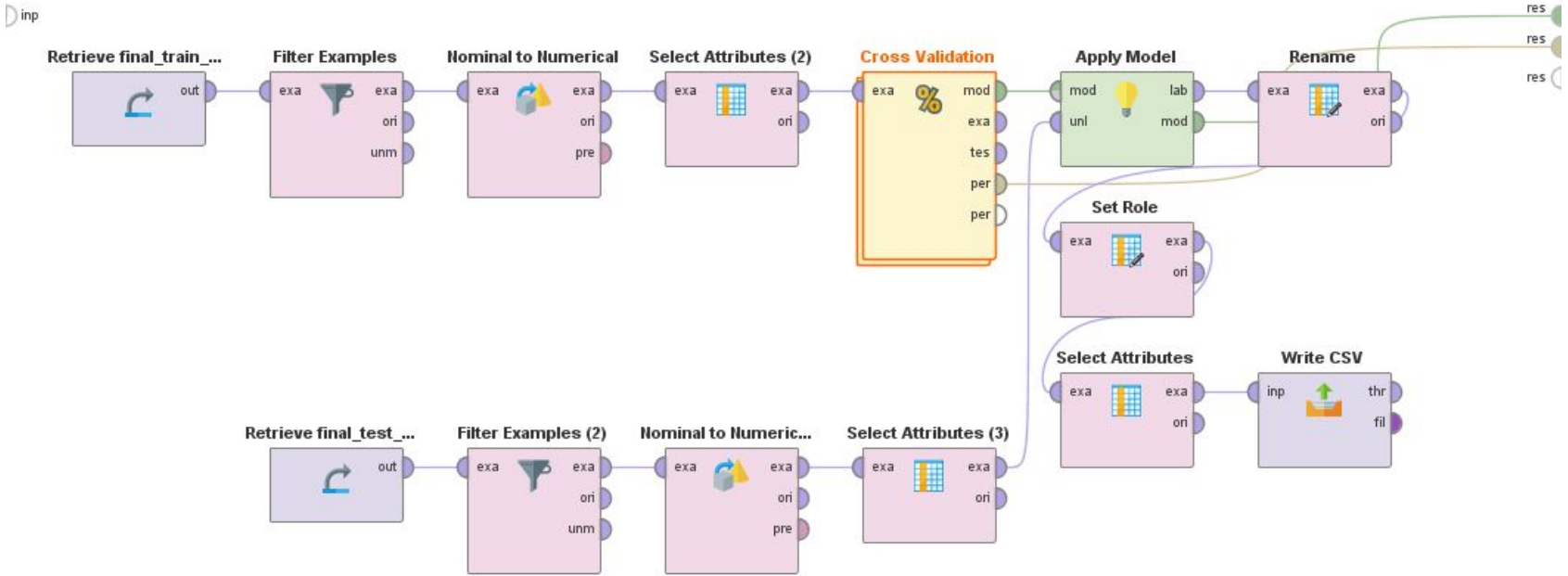


accuracy: 88.47% +/- 3.43% (micro average: 88.49%)

	true -1	true 1	class precision
pred. -1	13	2	86.67%
pred. 1	33	256	88.58%
class recall	28.26%	99.22%	

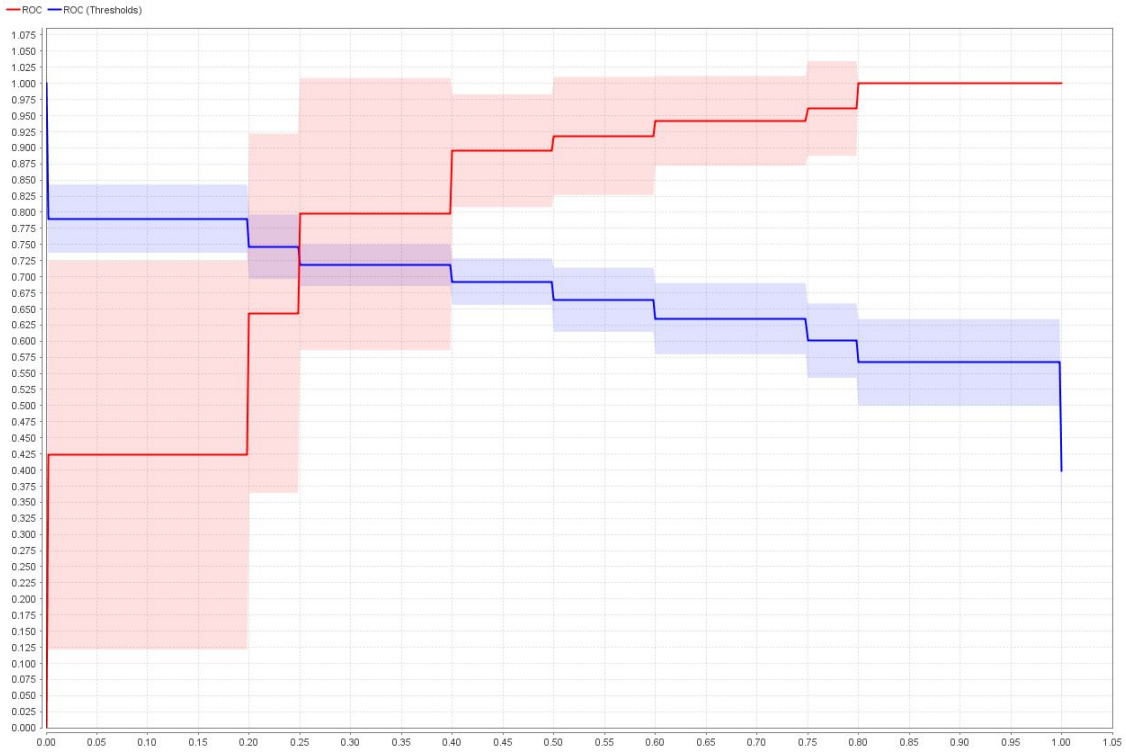
Experimental Setup

SVM



Results

SVM



accuracy: 87.81% +/- 1.99% (micro average: 87.79%)

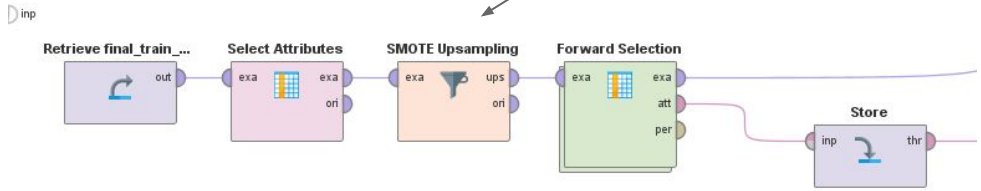
	true -1	true 1	class precision
pred. -1	8	0	100.00%
pred. 1	37	258	87.46%
class recall	17.78%	100.00%	

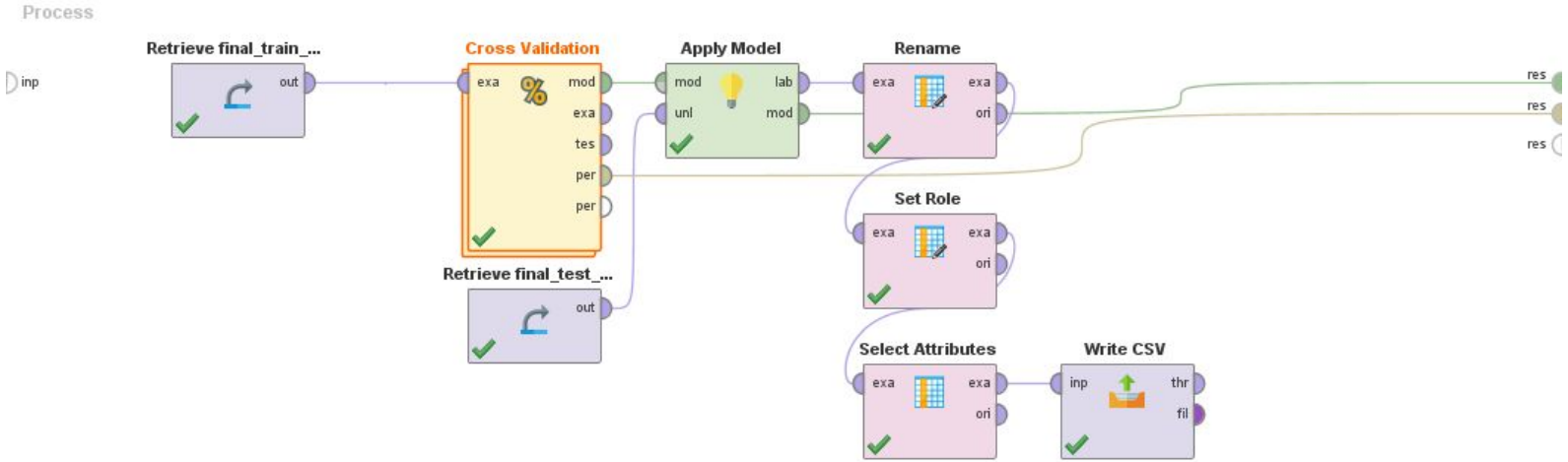
Experimental Setup

Upper Sampling

Deep Learning

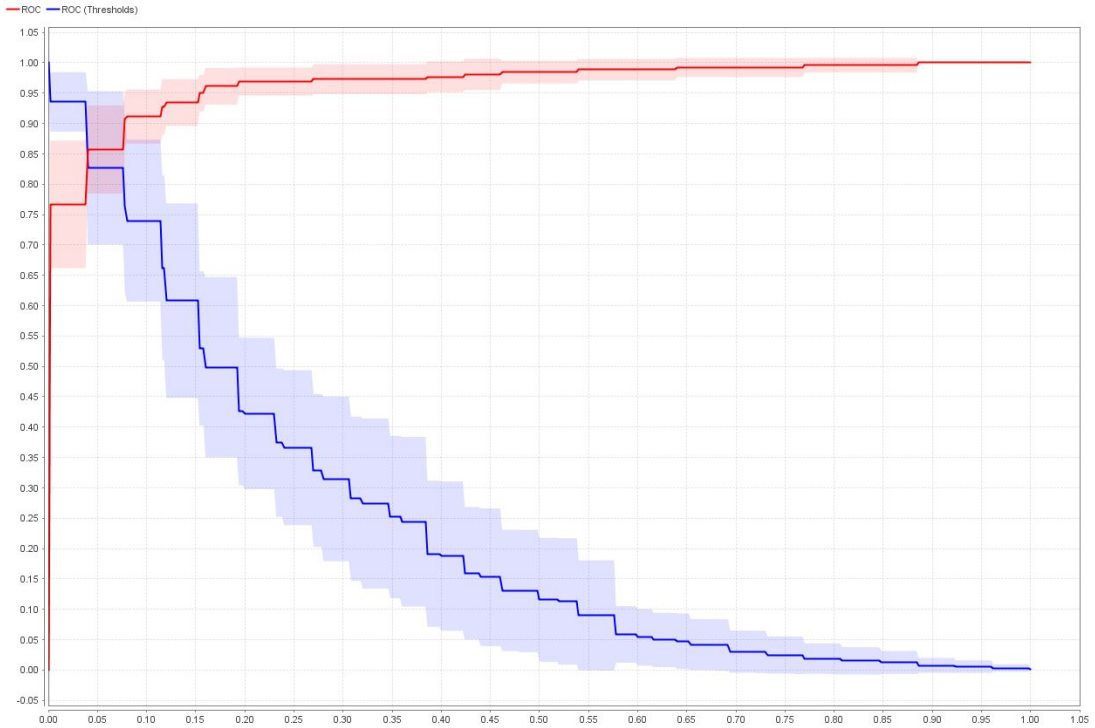
Feature selection





Results

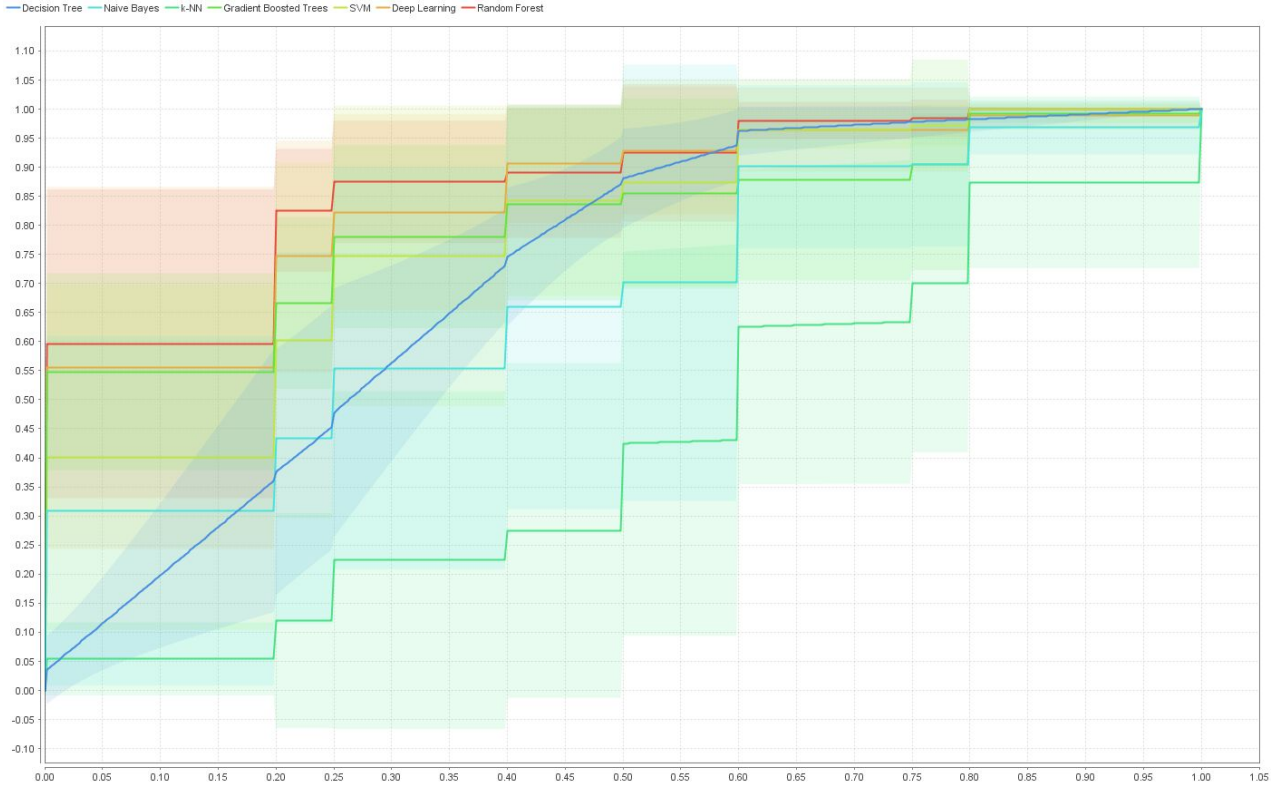
Deep Learning



accuracy: 89.72% +/- 3.26% (micro average: 89.73%)

	true -1	true 1	class precision
pred. -1	235	30	88.68%
pred. 1	23	228	90.84%
class recall	91.09%	88.37%	

ROC Curves Comparison



Conclusion

Experimented algorithms: **decision tree, deep learning, gradient boosted trees, k-nn, naive bayes, random forest, support vector machine.**

Best score on kaggle: ~86% with Deep Learning

In the future:

- test different parameters configurations
- test other algorithms
- calculate more attributes