CREDIT SCORING

Aldo Escobedo

Konfío

Summary

- To better choose which clients to offer better loans, we created a process that use the internal payments data, credit reports, and other external data for credit scoring purposes.
- We used this information to build a random forest model.
- We carefully picked the best model by grid search method.
- With this model, we give each client a score that shows how likely they are to be risky customers.

Business Understanding

Problem statement

- Create a model to pick best clients in order to give them a better loan offer.

Definition of target

- The first step to translate a business problem into a model is the definition of a target, which in this case is defined in the "target" column of general info.
- In this case the target is binary, therefore it will be addressed as a classification problem.

Definition of features

- Because for each application we have several internal and external history records, it is necessary to define features that summarize the most important information.

Business Understanding

Internal payments

- In internal payments we have information on credits prior to the current request, granted by the same company.

Credit records

- In credit records we have the summary of each credit account associated with an application. To use this information, it is necessary to summarize these reports by calculating features such as the number of open accounts, the number of closed accounts, the total balance, the total credit limit, the total balance, etc.

Note: To simulate the real prediction scenario, we will filter these datasets up to the limit dates.

External features

- Because for each application we have several internal and external history records, it is necessary to define features that characterize each application.

Data preparation

What characteristics can be useful to separate good from bad clients?

To analyze the characteristics that can help distinguish good and bad clients, we will start defining the following features related to payment history:

- **Internal payment history**
  - *num_prev_contracts: number of fully paid contracts (maturity before limit_date)*
  - *avg_notional: average notional from previous credits (before limit_date)*
  - *pct_late_payments: percentage of payments completed after payment date*
  - *Internal_credit_payments: number of total_payments over all prev_loans*

Konfio

Data preparation

Descriptive statistics of internal features

	num_prev_contracts	avg_notional	pct_late_payments	internal_credit_payments
count	921.000000	921.000000	921.000000	921.000000
mean	8.459283	171136.110146	0.139530	14.106406
std	14.534843	165464.192803	0.172726	15.600849
min	0.000000	10000.000000	0.000000	1.000000
25%	0.000000	80000.000000	0.000000	5.000000
50%	3.000000	100000.000000	0.090909	9.000000
75%	11.000000	198217.391304	0.200000	17.000000
max	123.000000	1000000.000000	1.000000	134.000000

Data preparation

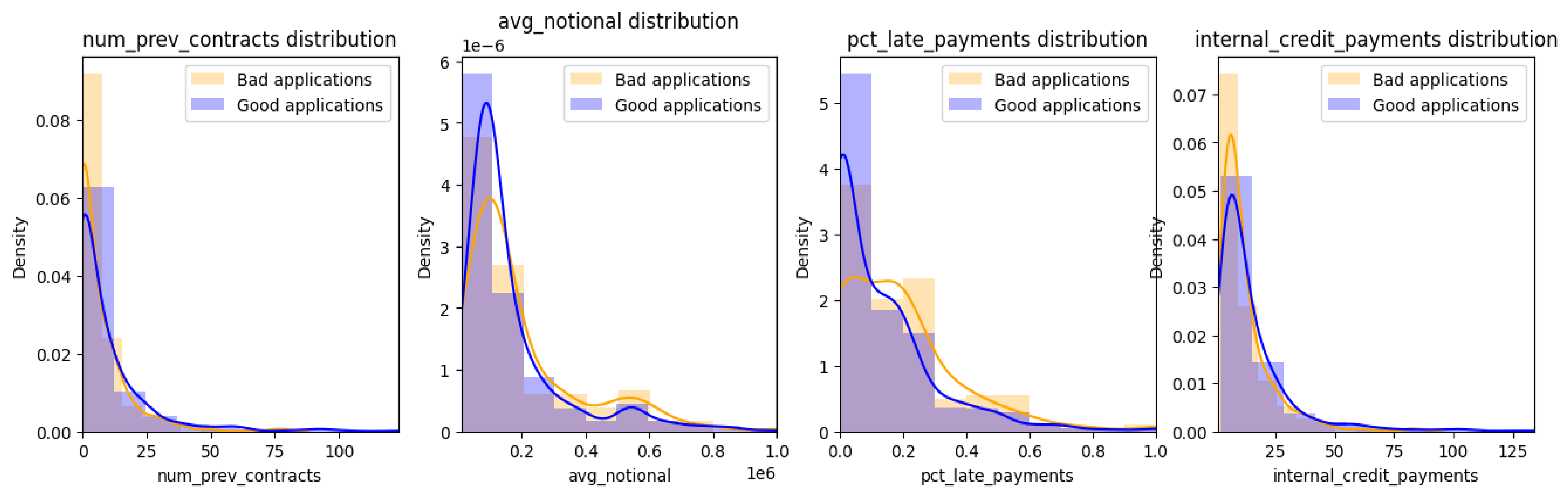


Figure: distributions of internal features grouped by target.

Insights example:

- Few prev contracts -> more risk
- lower notional amount -> less risk
- Lower percentage of late payments -> less risk
- Shorter payment history -> more risk

Konfio

Data preparation

- Credit reports
  - *open_accounts: number of open accounts at limit date*
  - *closed_accounts: number of closed accounts at limit date*
  - *maximum_credit_amount: maximum amount of credit over all accounts*
  - *current_balance: sum of current balance over all accounts*
  - *past_due_balance: sum of past due balance over all accounts*
  - *total_credit_payments: sum of length of the credit over all accounts*
  - *worst_delinquency_past_due_balance: worst accumulated delinquent over all*
  - *credit_limit: sum of credit limits*
  - *past_due_ratio: past_due_balance/credit_limit*
  - *current_balance_ratio: current_balance/credit_limit*

Konfío

Data preparation

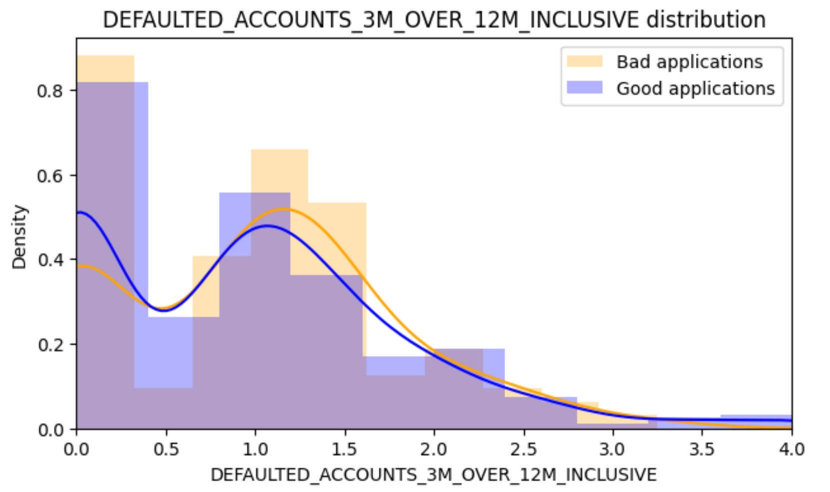
Descriptive statistics from internal features

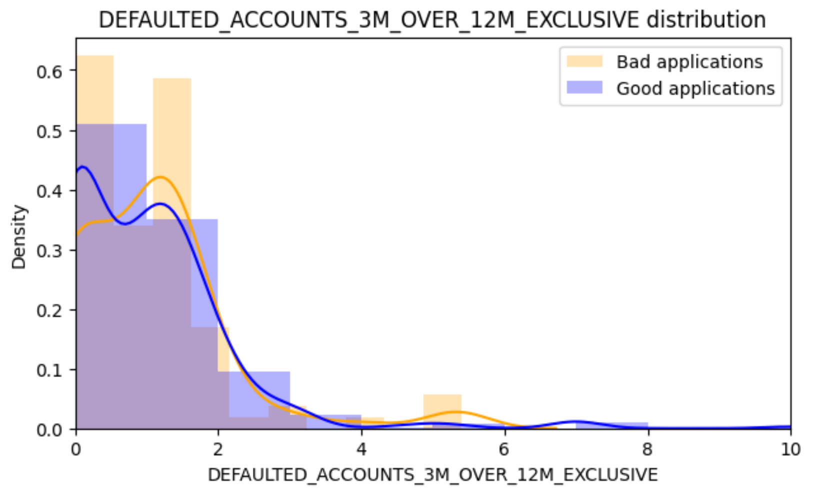
	open_accounts	closed_accounts	max_credit_amount	current_balance	past_due_balance	total_credit_payments	worst_delinquency_past_due_balance	credit_limit	past_due_ratio	current_balance_ratio
count	921.000000	921.000000	9.210000e+02	9.210000e+02	921.000000	921.000000	921.000000	9.210000e+02	921.000000	921.000000
mean	10.537459	15.137894	4.848281e+05	7.897580e+05	6143.836048	618.563518	4373.407166	1.255601e+06	0.012499	0.757210
std	5.954443	18.396227	6.407955e+05	1.102643e+06	29542.562775	790.819399	10303.248722	1.817979e+06	0.055283	0.729340
min	0.000000	0.000000	7.892000e+03	0.000000e+00	0.000000	0.000000	0.000000	3.696000e+03	0.000000	0.000000
25%	6.000000	5.000000	1.500000e+05	1.764840e+05	0.000000	143.000000	0.000000	3.286800e+05	0.000000	0.395512
50%	10.000000	10.000000	2.855000e+05	4.298700e+05	0.000000	377.000000	1046.000000	7.109640e+05	0.000000	0.607665
75%	13.000000	19.000000	5.300000e+05	8.876760e+05	1047.000000	786.000000	4178.000000	1.416254e+06	0.001389	0.846216
max	46.000000	189.000000	5.100000e+06	9.781258e+06	494494.000000	7988.000000	154402.000000	2.192811e+07	0.818363	7.619477

Data preparation

- **External features description**
  - *These features have many missing features, and no clear difference is observed in the comparison of the distributions. These features will be omitted.*

	application_id	DEFAULTED_ACCOUNTS_3M_OVER_12M_INCLUSIVE	DEFAULTED_ACCOUNTS_3M_OVER_12M_EXCLUSIVE
count	921.000000	498.000000	497.000000
mean	929.928339	0.947711	1.016511
std	555.317811	0.851456	1.160751
min	1.000000	0.000000	0.000000
25%	419.000000	0.000000	0.000000
50%	960.000000	1.000000	1.000000
75%	1436.000000	1.421429	1.500000
max	1790.000000	4.000000	10.000000





Konfío

Modeling

For this problem, classification models will be used to subsequently estimate the probabilities that each application is good or bad.

Pipeline definition

- **-Data transformation.**
  For categorical features: most frequent imputer
  For numerical features: mean imputer and standardization
- **-Undersampling**
- **-Varaince threshold**
- **-Feature Selector**
- **-Classification model (Logistic regression, Random Forest)**

Konfio

Modeling

Compare model performance

The best logistic regression and random forest model was selected through grid search using cross validation and the results were as follows.

	Logistic regression	Random Forest
Best specificity CV	0.58	0.60
Selected features	'num_prev_contracts', 'avg_notional', 'pct_late_payments', 'internal_credit_payments', 'max_credit_amount', 'current_balance', 'past_due_balance', 'past_due_ratio', 'current_balance_ratio', 'institution_FINANCIERA', 'institution_OTRAS FINANCIERA', 'institution_SERVICIO DE TELEVISION DE PAGA', 'institution_SERVICIOS', 'institution_TIENDA', 'account_type_Hipoteca', 'credit_type_Otros (Múltiples Créditos)'	'num_prev_contracts', 'avg_notional', 'pct_late_payments', 'internal_credit_payments', 'open_accounts', 'closed_accounts', 'max_credit_amount', 'current_balance', 'total_credit_payments', 'worst_delinquency_past_due_balance', 'credit_limit', 'past_due_ratio', 'current_balance_ratio', 'credit_type_Otros (Múltiples Créditos)'

For parsimony, the random forest model will be chosen.

Note: Because the context of the problem is to identify good clients, the grid search was implemented with a specificity scoring TN/(TN+FP)

Evaluation

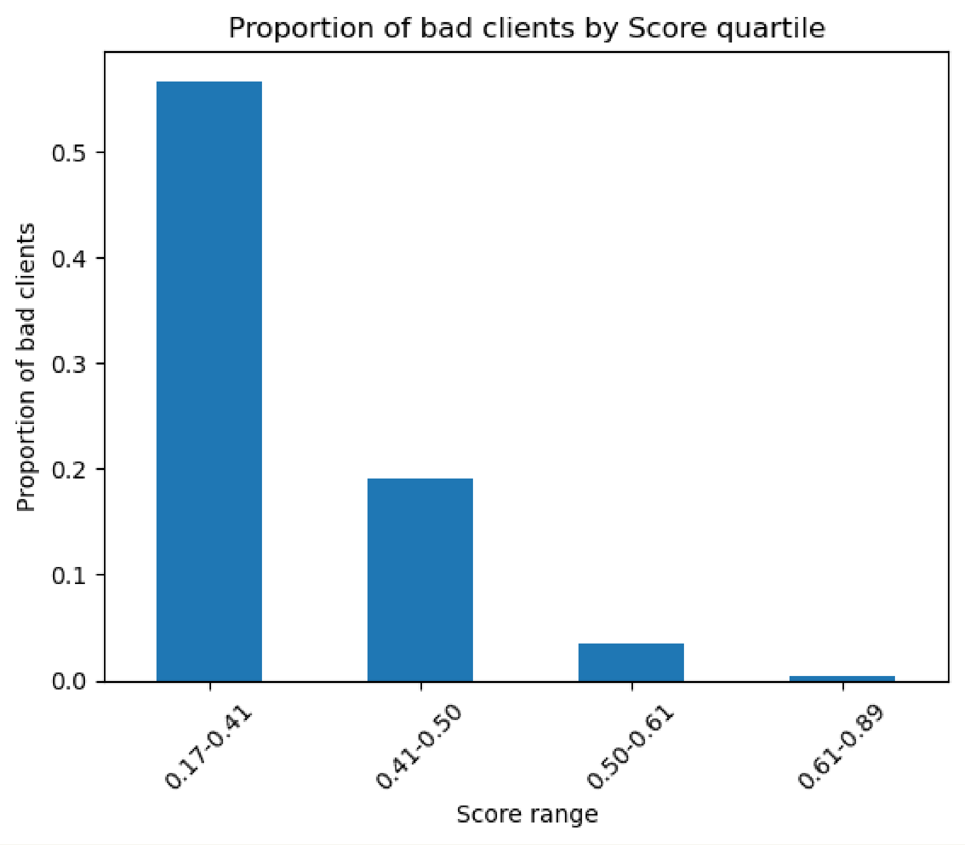
Best model parameters

classifier__max_depth: 15,

classifier__min_samples_split: 20,

classifier__n_estimators: 100

In the following figure we see that this model sorts the applications well in the sense that as the score increases, the proportion of bad clients decreases.



Deployment

the best model was saved as a pickle object so it could be loaded, and scores calculated in the Inference.py.

In inference_sample_run we can get an example on how the inference functions work.

Conclusions

With this process we can assign a score to every user_id as it's shown in inference_sample_run notebook.

According this procedure, the top 10 best clients are the following:

user_id	score
1023.0	0.956492
1145.0	0.947264
1857.0	0.946858
1242.0	0.942399
1248.0	0.931740
1000.0	0.926190
1184.0	0.906029
1120.0	0.902632
1178.0	0.901023
1017.0	0.883459