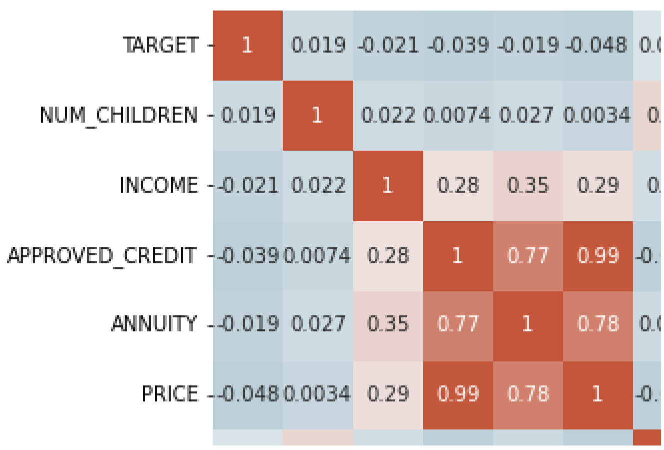
Rizky Luthfianto

1a. Describe the data pre-processing step that you did

First I summarize the data

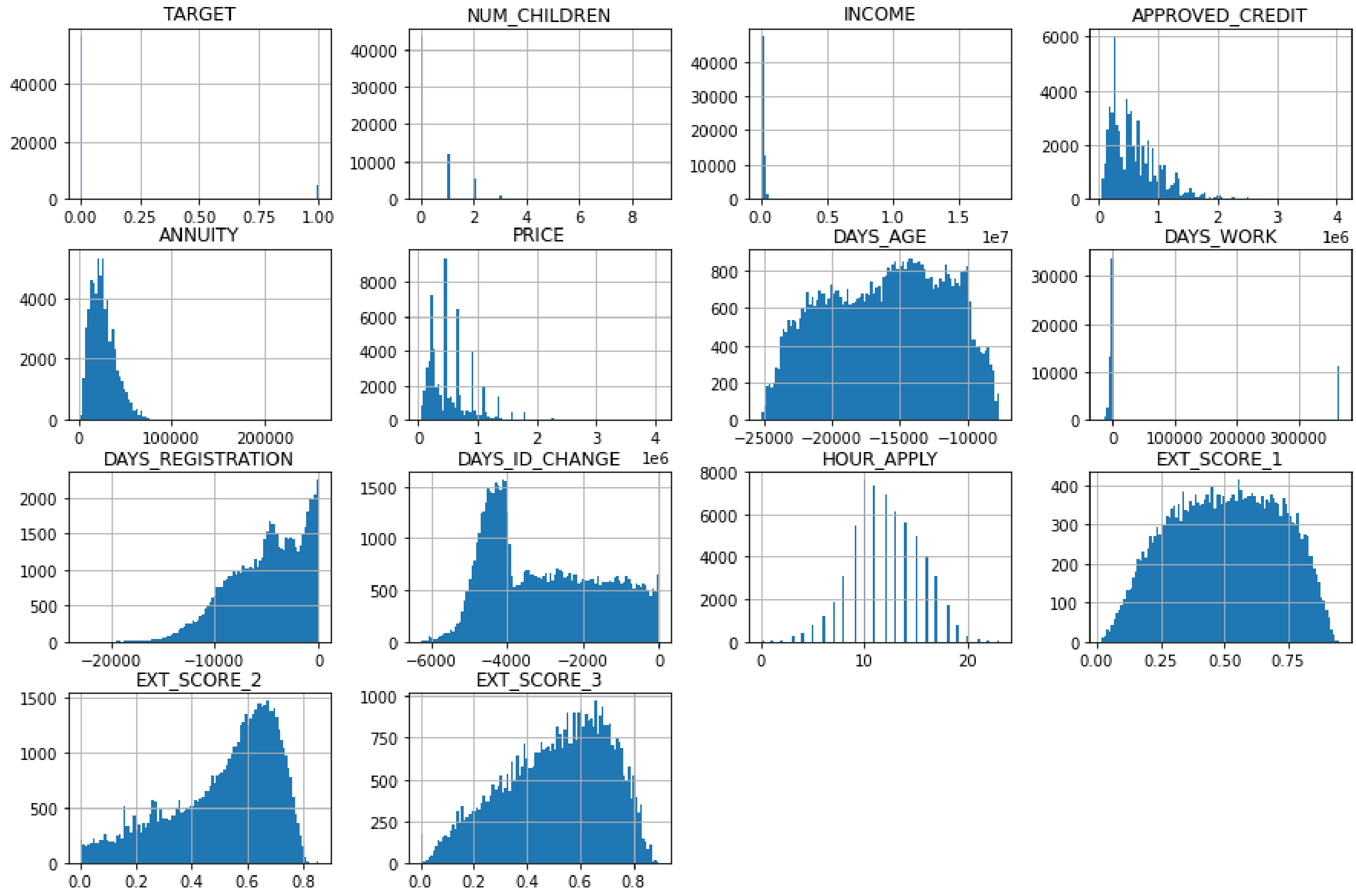
	has nan?	num of notnan	dtypes	num of unique values
Unnamed: 0	False	61503	int64	61503
LN_ID	False	61503	int64	61503
TARGET	False	61503	int64	2
CONTRACT_TYPE	False	61503	object	2
GENDER	False	61503	object	2
NUM_CHILDREN	False	61503	int64	10
INCOME	False	61503	float64	861
APPROVED_CREDIT	False	61503	float64	3562
ANNUITY	True	61502	float64	9374
PRICE	True	61441	float64	541
INCOME_TYPE	False	61503	object	7
EDUCATION	False	61503	object	5
FAMILY_STATUS	False	61503	object	5
HOUSING_TYPE	False	61503	object	6
DAYS_AGE	False	61503	int64	16257
DAYS_WORK	False	61503	int64	8524
DAYS_REGISTRATION	False	61503	float64	13153
DAYS_ID_CHANGE	False	61503	int64	5824
WEEKDAYS_APPLY	False	61503	object	7
HOUR_APPLY	False	61503	int64	24
ORGANIZATION_TYPE	False	61503	object	58
EXT_SCORE_1	True	26658	float64	25814
EXT_SCORE_2	True	61369	float64	46296
EXT_SCORE_3	True	49264	float64	744

Then do the correlation plot:



approved_credit, annuity, and price are very correlated. we will just use **approved_credit** because it has no missing_values, and drop the other two.

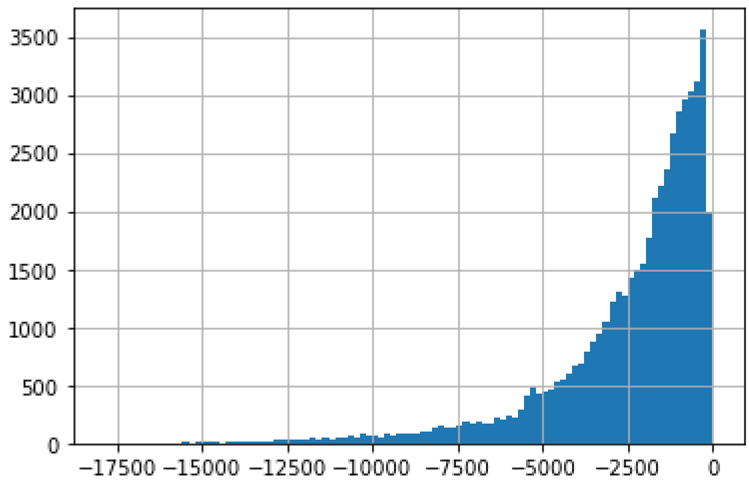
Then, I plot a histogram



Here we see an anomaly in DAYS_WORK

```
[15]: df.DAYS_WORK[df.DAYS_WORK<365243].hist(bins=100)

[15]: <AxesSubplot:>
```



I also checked the test set and see the same value 365243. Values greater than 0, especially 365243 will be marked as special values and binned separately later.

Null values are also not imputed but are binned separately.

Besides the existing feature, I also generate new features. Note that we use LN_ID granularity:

- from the train df:
  - credit_per_income: user APPROVED_CREDIT / INCOME
  - insure_rate: sum of approved_loans_that_use_insurance / count of all approved loan

- for convenience purposes:
    - year_age: absolute of DAYS_AGE / 365
    - year_work: absolute of DAYS_WORK / 365
- from previous application df:
  - contract_status_approval_rate: sum of approved loan / count of loan applied
  - contract_status_refusal_rate: sum of refused loan / count of loan applied
- from installment df:
  - overdue_days: sum of overdue days
  - has_overdue_day: sum of 1 if overdue regardless of days, 0 if not
  - overdue_amt: sum of overdue amount
  - has_overdue_amt: sum of 1 if overdue regardless of days, 0 if not

Then I split the data into train & validation sets. Here I use **stratified sampling** so the original training data, splitted training set, and validation set have the same proportion of TARGET.

Then I bin the training set using optimization to maximize IV, without monotonic constraints. The purpose is for exploratory only. Then after, exploratory, I override the binning by adding these constraints:

- That should have **ascending monotonic** trend to the event rate (more value -> more risky) :
  - NUM_CHILDREN
  - Credit_per_income
  - Overdue_days
  - Overdue_amt
  - Has_overdue_day
  - Has_overdue_amt
  - 'APPROVED_CREDIT'
- That should have **descending monotonic** trend to the event rate (more value -> less risky) :
  - 'INCOME'
  - 'Year_work'
  - 'Year_age'
  - Insure_rate
- INCOME_TYPE bins working people and unemployed together, it might be because unemployed has only 4 sample in the training data. So I make the constraint to **separate** working people and unemployed

After overriding and adding constraint to the binning. Here is the IV for each features, the predictiveness criteria is taken from Naeem Siddiqi book:

column	IV	predictiveness
EXT_SCORE_3	0.383135	Strong predictive Power
EXT_SCORE_2	0.324745	Strong predictive Power
EXT_SCORE_1	0.192043	Medium predictive Power
year_age	0.106269	Medium predictive Power

ORGANIZATION_TYPE	0.088333	Weak predictive Power
Approved	0.076050	Weak predictive Power
Refused	0.070099	Weak predictive Power
INCOME_TYPE	0.068895	Weak predictive Power
EDUCATION	0.061190	Weak predictive Power
DAYS_ID_CHANGE	0.047378	Weak predictive Power
year_work	0.047284	Weak predictive Power
APPROVED_CREDIT	0.046801	Weak predictive Power
overdue_amt	0.042602	Weak predictive Power
GENDER	0.042327	Weak predictive Power
has_overdue_amt	0.035500	Weak predictive Power
FAMILY_STATUS	0.027817	Weak predictive Power
DAYS_REGISTRATION	0.025519	Weak predictive Power
terms_payment	0.018505	Not useful for prediction
CONTRACT_TYPE	0.014794	Not useful for prediction
INCOME	0.014287	Not useful for prediction
HOUSING_TYPE	0.013729	Not useful for prediction
insure_rate	0.009523	Not useful for prediction
NUM_CHILDREN	0.008398	Not useful for prediction
Unused offer	0.007848	Not useful for prediction
overdue_days	0.006732	Not useful for prediction
has_overdue_day	0.006730	Not useful for prediction
credit_per_income	0.001910	Not useful for prediction

While some features are not useful for prediction, I include them because I see a decrease in AUC when I exclude them completely.

The data with the WOE bin is fitted to the model. I am aware of the class imbalance, to overcome the imbalance don't use SMOTE for oversampling/undersampling, rather than using the class_weight on the Logistic Regression itself.

class weight value is :1 - class_proportion = {0: 0.08079280685495016, 1: 0.9192071931450498}

1b. Choose the most appropriate metrics to measure the model performance and provide an explanation of why you choose them

- AUC: to measure the ability of a classifier to distinguish between classes
- KS: to discriminate between good" and "bad" customers, by comparing the distribution between "good" customers and "bad" customers.
- Recall: because false negatives (bad borrowers who are misclassified as good) are much more harmful than false positives (good borrowers who are misclassified as bad)

- F1: because even when false negative is more harmful, we still need to have a balanced metric between precision and recall

1c. Choose 3 of the most important features (original or derived features) and explain how and why they are important
- EXT_SCORE_1
- EXT_SCORE_2
- EXT_SCORE_3

because they have the highest Information Value, but it’s hard to deduce what they are, because we don’t own the data

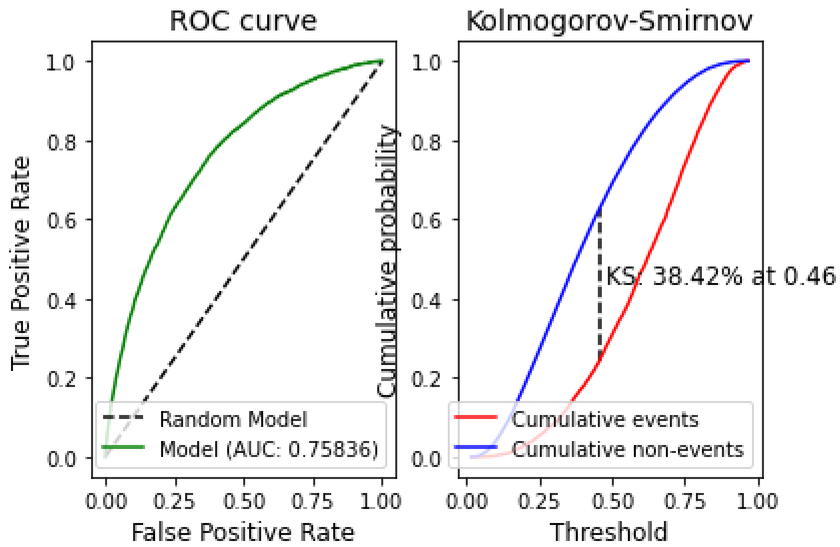
Besides external scores, these are three columns that also have high IV:
- **Year_age**: older borrowers tend to pay their debt on time more than younger borrowers
- **ORGANIZATION_TYPE**:
  - Safest org type:
    - 'Trade: type 5', 'Industry: type 12', 'Insurance', 'Bank', 'Military', 'Industry: type 6', 'Trade: type 6', 'Transport: type 1', 'Security Ministries', 'Police', 'Emergency', 'University', 'NA1', 'School', 'Industry: type 5', 'Industry: type 13'
  - Riskiest org type:
    - 'Legal Services', 'Realtor', 'Advertising', 'Industry: type 3', 'Restaurant', 'Industry: type 10', 'Construction', 'Transport: type 3', 'Industry: type 1', 'Industry: type 4', 'Industry: type 8', 'Religion', 'Cleaning'
  - **Note**:
    - NA1 got binned into the safest org type. It needs to be penalized, because if not it will incentivize the borrower to leave their org type blank so they could get a better chance at borrowing, increasing our risk.
- **Approved** (approval rate for the previous loan): user who has higher loan approval tends to pay their debt on time

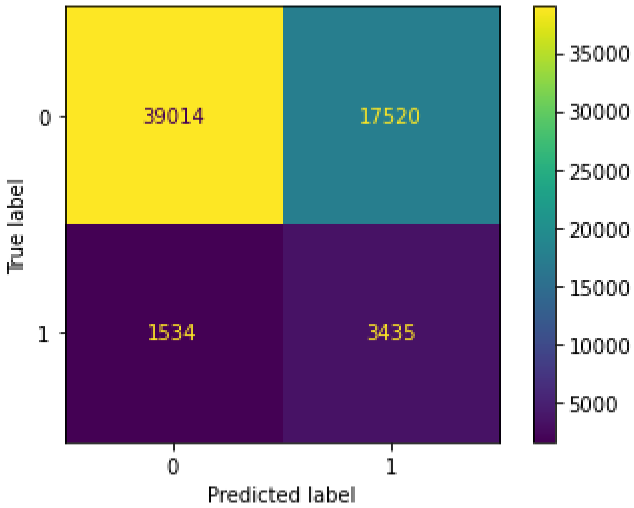
1d. Choose the most appropriate model and provide an explanation of why and how the model can solve the lenders' problem
The most appropriate model is Logistic Regression + Monotonic Binning because it is interpretable and, robust.

The model can solve the lenders' problem by predicting which borrowers pay late, which causes losses to the lenders.
The predicted potential borrowers who pay late, could be rejected or approved but with reduced approved_credits.

1e. Submit the model and all the analyses that you made complete with the test set result (Accuracy, Precision/Recall, F1, AUC, etc)

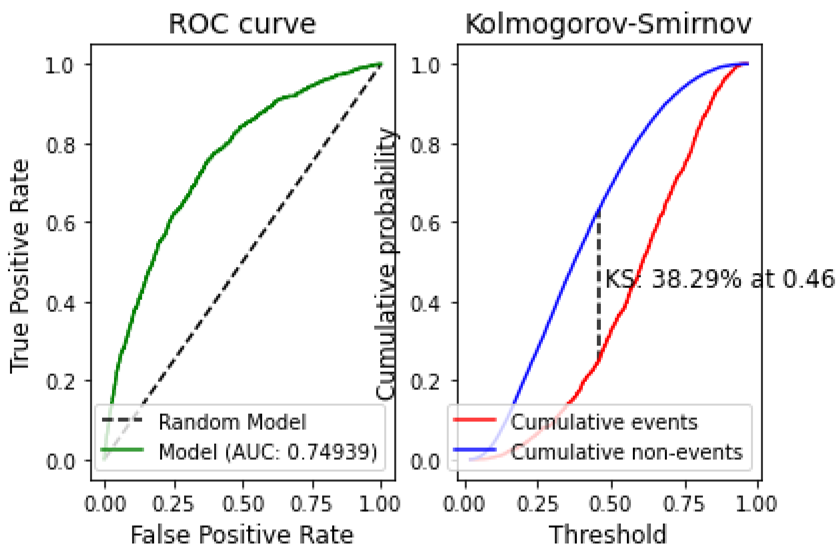
Train

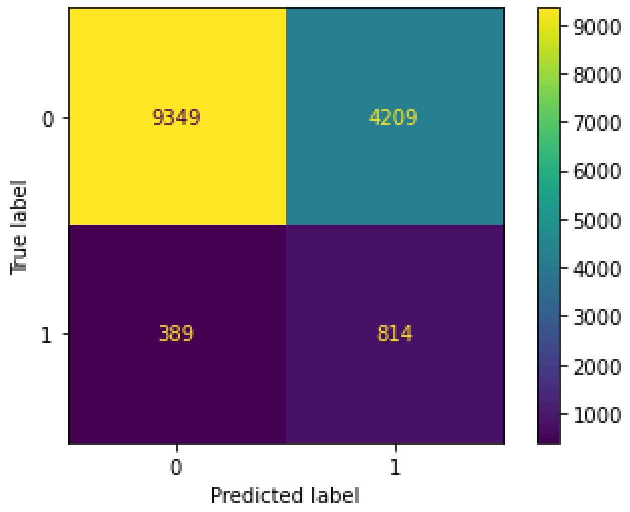




precision=0.56
recall=0.69
fscore=0.53

Test





precision=0.56
recall=0.69
fscore=0.53

Note: I attached the notebook and the model in this zip

---

2. Which Campaign did better and why?

Before we conclude which campaign is better, first we have to determine if the campaign difference is significant
1 -> business marketing
2 -> customer marketing

n_1 = 10928
n_2 = 9668
p_1 = 13.4%
p_2 = 14.5%
p = $\frac{(0.134*10928) - (0.145*9668)}{10928+9668} = 0.003$
a=0.05, Z_a= 1.645
$H_0 = p_2 >= p_1$
$H_1 = p_2 < p_1$

$$Z_{hit} = \frac{p_1 - p_2}{\sqrt{p\,(1-p)\,(\frac{1}{n_1} + \frac{1}{n_2})}}$$

$$Z_{hit} = \frac{0.134 - 0.145}{\sqrt{0.003\,(0.997)\,(\frac{1}{10928} + \frac{1}{9668})}}$$

$$Z_{hit} = \frac{-0.011}{\sqrt{0.003\,(0.997)\,0.000194942062}}$$

$$Z_{hit} = \frac{-0.011}{\sqrt{5.83071707e-7}}$$

$$z_{hit} = \frac{-0.011}{0.000763591322}$$

$$z_{hit} = -14.4056116$$

$$z_{hit} < z_a$$

H_0 is rejected when $Z_{hit} > Z_a$
But Z_{hit} is not more than Z_a, so H_0 is accepted, $p_2 >= p_1$.
Therefore the CTR difference between the campaigns is significant.

Which campaigns did better?
a = $ 0-500
b = $ 500-1000
c = $ 1000+

Assumption: The revenue for each group will be multiplied by the middle value of each interval. For the $1000+ group, it will be multiplied by 1000

Business marketing emails - Estimated Revenue
x_{a1} = 0.12 * 9105 = 1092.6
x_{b1} = 0.19 * 1491= 283.29
x_{c1} = 0.28 * 332 = 92.96
Expected Revenue: 1092.6 * 250 + 283.29* 750 + 92.96 * 1000 = 578577.5

Consumer marketing emails - Estimated Revenue
x_{a2} = 0.1 * 3087 = 308.7
x_{b2} = 0.16 * 4461 = 713.76
x_{c2} = 0.18 * 2120 = 381.6
Expected Revenue: 308.7 * 250 + 713.76 * 750 + 381.6 * 1000 = 994095

Thus we could conclude that **Consumer marketing emails** Campaign performed better

---

3. What could be some issues if the distribution of the test data is significantly different than the distribution of the training data?

The model will lose its predictive power and performs very poor. That is why we have to monitor the model performance and distribution between training data and testing data, especially when the model/scorecard is deployed.