Home Credit Default Risk

Kaggle Competition - Home Credit Default

Website: https://www.kaggle.com/c/home-credit-default-risk

Home credit group:

Founded in 1997, Home Credit Group is an international consumer finance provider with operations in 10 countries. We focus on responsible lending primarily to people with little or no credit history. Our services are simple, easy and fast.

Objective:

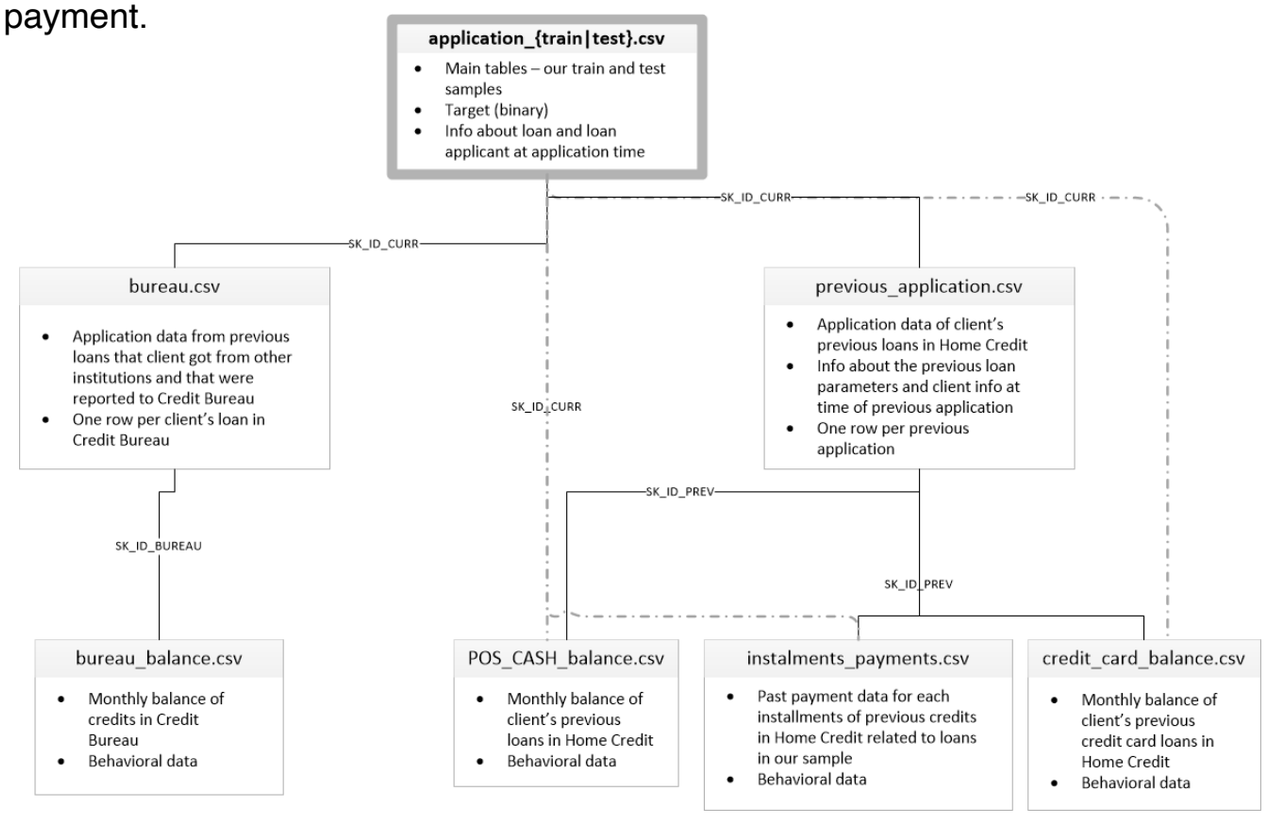
The objective of this competition is to use historical loan application data to predict whether or not an applicant will be able to repay a loan. This is a standard supervised classification task:

- Supervised: The labels are included in the training data and the goal is to train a model to learn to predict the labels from the features
- Classification: The label is a binary variable, 0 (will repay loan on time), 1 (will have difficulty repaying loan)

Data Information

There are 7 different sources of data:

- application_train/application_test: the main training and testing data with information about each loan application at Home Credit. Every loan has its own row and is identified by the feature `SK_ID_CURR`. The training application data comes with the `TARGET` indicating 0: the loan was repaid or 1: the loan was not repaid.
- bureau: data concerning client's previous credits from other financial institutions. Each previous credit has its own row in bureau, but one loan in the application data can have multiple previous credits.
- bureau_balance: monthly data about the previous credits in bureau. Each row is one month of a previous credit, and a single previous credit can have multiple rows, one for each month of the credit length.
- previous_application: previous applications for loans at Home Credit of clients who have loans in the application data. Each current loan in the application data can have multiple previous loans. Each previous application has one row and is identified by the feature `SK_ID_PREV`.
- POS_CASH_BALANCE: monthly data about previous point of sale or cash loans clients have had with Home Credit. Each row is one month of a previous point of sale or cash loan, and a single previous loan can have many rows.
- credit_card_balance: monthly data about previous credit cards clients have had with Home Credit. Each row is one month of a credit card balance, and a single credit card can have many rows.
- installments_payment: payment history for previous loans at Home Credit. There is one row for every made payment and one row for every missed payment.



Procedure

- EDA – data analysis and visualization
- Handle missingness and outliers
- Feature Engineering – add new features
- Machine Learning Modeling – xgboost, lightgbm, catboost
- Advance Strategies – Ensemble and Stacking

EDA

Application training data: 307511 clients, 122 feature

Unbalanced data：

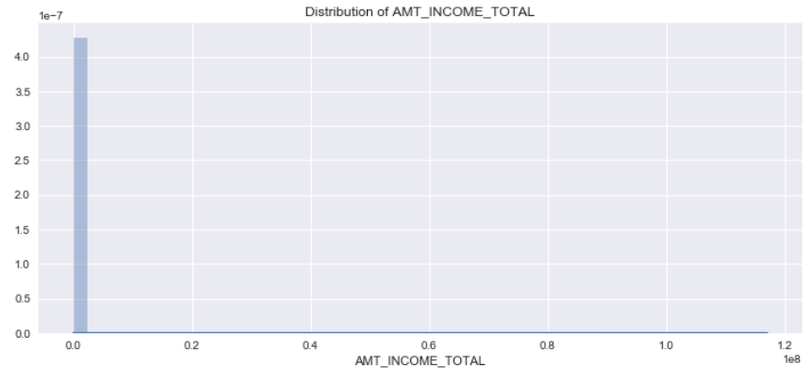
NAME_CONTRACT_TYPE	Cash loans	Revolving loans
TARGET		
0	0.829274	0.089997
1	0.075513	0.005216

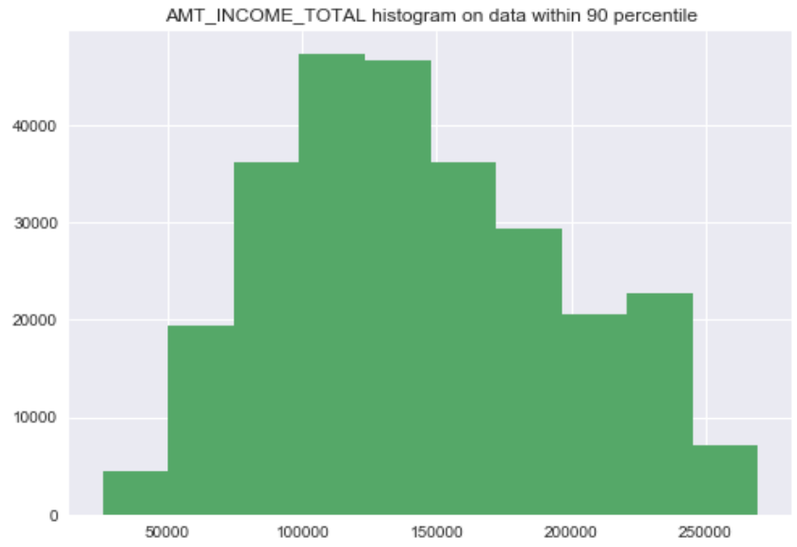
Gender, family status, occupation, income, education level, credit, Car, house, delinquency, fico, loan, credit, etc

NAME_FAMILY_STATUS	Civil marriage	Married	Separated	Single / not married	Unknown	Widow
CNT_CHILDREN						
0	20947	126575	14132	38810	2	14905
1	6588	43696	4389	5578	0	868
2	1936	22496	1111	958	0	248
3	262	3190	122	85	0	58
4	33	370	12	9	0	5
5	5	74	2	1	0	2
6	2	17	1	0	0	1
7	1	5	0	0	0	1
8	0	2	0	0	0	0
9	1	0	0	1	0	0
10	0	2	0	0	0	0
11	0	1	0	0	0	0
12	0	2	0	0	0	0
14	0	2	1	0	0	0
19	0	0	0	2	0	0

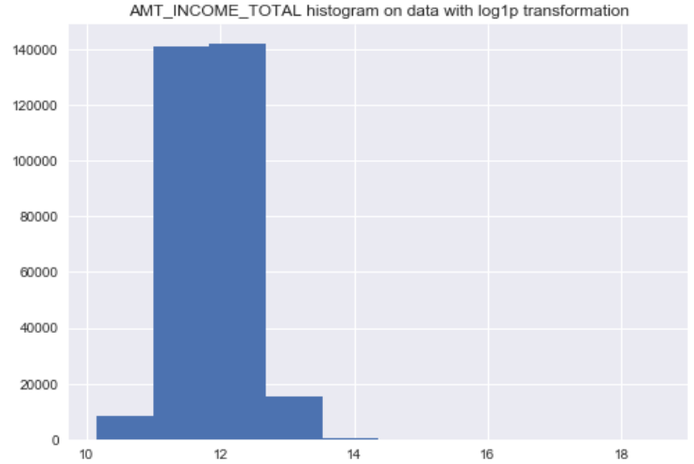
Outliers and Transforamtion

Outliers: log-transformation





Income feature has some huge outliers, if we leave only data within 90 percentile, it is almost normally distributed;
log transformation also helps.



Outliers: replacement

prev['DAYS_FIRST_DUE'].replace(365243, np.nan, inplace= True)

Missingness and Imputation

Application training data

Fill NA with 0 if applicants have no such feature

	Total	Percent
COMMONAREA_MEDI	214865	69.872297
COMMONAREA_AVG	214865	69.872297
COMMONAREA_MODE	214865	69.872297
NONLIVINGAPARTMENTS_MODE	213514	69.432963
NONLIVINGAPARTMENTS_MEDI	213514	69.432963
NONLIVINGAPARTMENTS_AVG	213514	69.432963
FONDKAPREMONT_MODE	210295	68.386172
LIVINGAPARTMENTS_MEDI	210199	68.354953
LIVINGAPARTMENTS_MODE	210199	68.354953
LIVINGAPARTMENTS_AVG	210199	68.354953
FLOORSMIN_MEDI	208642	67.848630
FLOORSMIN_MODE	208642	67.848630
FLOORSMIN_AVG	208642	67.848630
YEARS_BUILD_MEDI	204488	66.497784
YEARS_BUILD_AVG	204488	66.497784
YEARS_BUILD_MODE	204488	66.497784
OWN_CAR_AGE	202929	65.990810
LANDAREA_MODE	182590	59.376738
LANDAREA_AVG	182590	59.376738
LANDAREA_MEDI	182590	59.376738
BASEMENTAREA_MEDI	179943	58.515956
BASEMENTAREA_AVG	179943	58.515956
BASEMENTAREA_MODE	179943	58.515956

Feature Engineering

Add new features based on business domain knowledge

Create 3000 features from 200 features

Important added features:

- Application data :
    - 3 fico sources (max, min, median, mean)
    - Credit to annual loan amount ratio
    - Credit to goods price ratio
    - Credit to income ratio
- Bureau: active and closed client
- Credit card: different stage of payment trend

Machine Learning Models

Objective: max roc-auc score

Models:

Lightgbm: (https://lightgbm.readthedocs.io/en/latest/)
LightGBM is a gradient boosting framework that uses tree based learning algorithms. It is designed to be distributed and efficient with the following advantages:

- Faster training speed and higher efficiency.
- Lower memory usage.
- Better accuracy.
- Support of parallel and GPU learning.
- Capable of handling large-scale data.

Xgboost: (https://tech.yandex.com/catboost/)

XGBoost is an optimized distributed gradient boosting library designed to be highly efficient, flexible and portable. It implements machine learning algorithms under the Gradient Boosting framework. XGBoost provides a parallel tree boosting (also known as GBDT, GBM) that solve many data science problems in a fast and accurate way.

Catboost: (https://tech.yandex.com/catboost/)

CatBoost is a state-of-the-art open-source gradient boosting on decision trees library.

- Accurate: leads or ties competition on standard benchmarks
- Robust: reduces the need for extensive hyper-parameter tuning
- Practical: uses categorical features directly and scalably
- Extensible: allows specifying custom loss functions

Advanced Strategies

Ensemble

Stacking