Would people without credit history be trusted?

Introduction

Nowadays every organisation faces the challenge of detecting any abnormality correctly and promptly – such as disease detection in medical institutes, fraud detection in banking business and depression detection in mental health society. Catching any symptom of risky behaviours or situations is very important to save their business resources as well as prevent public loss. Fortunately, with advanced technologies in computing, there are enormous amounts of data collected and stored everywhere, and it helps us to provide evidence-based insights to make better decisions in the business environment.

Problem Definition

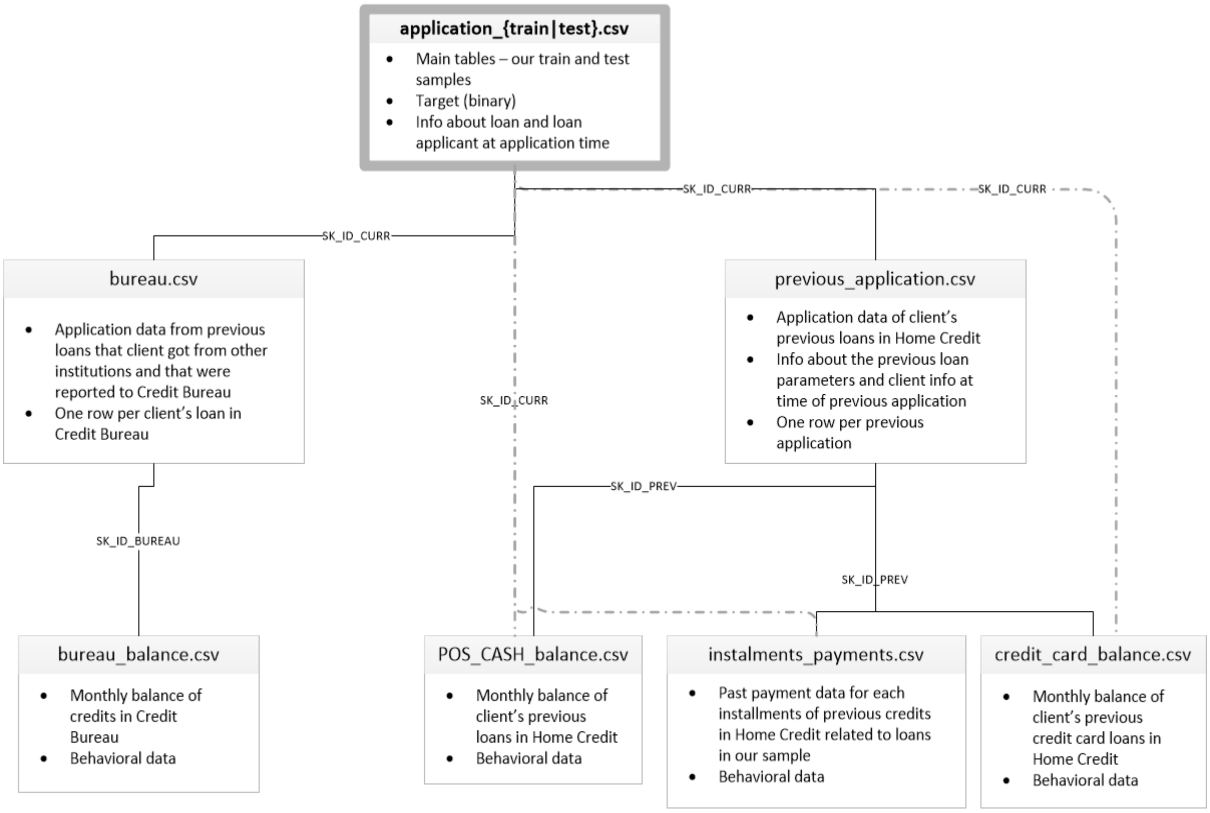
A loan service agency collected big data for people who applied for a loan with them and wants to build a precious model to accurately predict that an applicant would fail to repay the loan or not if they borrowed money from them. The data consist of several files including current application, credit bureau info, bureau balance, the point of sale balance, credit card balance history, previous application and payment history for over 300000 applicants – there are over 200 variables in total as well. We want the simplest prediction model, which includes only a few relevant variables among these 200 as long as it fits well in the metric of the area under the ROC (AUC). This project would focus on:

- Which variables are most relevant to predict the default status?
- Which aggregation method from the historical data is better – mean, median or sum?
- Would the weighted aggregation with time work better than equal weight?

Data Source

The data came from the kaggle website (https://www.kaggle.com/c/home-credit-default-risk/data); the list of files provides an abbreviated description of the data and followed by the diagram showing the relationship of these data tables:

- Application: the main table in which a row represents an application, dim (307511, 122)
- Bureau: client's previous credits by other financial institutions, dim (1716428, 17)
- Bureau balance: monthly balance of previous credit, dim (27299925, 3)
- POS cash balance: monthly balance of previous POS and cash loan, dim (10001358, 8)
- Credit card balance: monthly balance of previous credit card, dim (3840312, 23)
- Previous application: previous application history with hosting agency, dim (1670214, 37)
- Instalment payment: payment history for the previous disbursed credit, dim (13605402, 8)


application_{train|test}.csv
Main tables – our train and test samples
Target (binary)
Info about loan and loan applicant at application time
SK_ID_CURR
SK_ID_CURR
SK_ID_CURR
SK_ID_CURR
bureau.csv
Application data from previous loans that client got from other institutions and that were reported to Credit Bureau
One row per client's loan in Credit Bureau
previous_application.csv
Application data of client's previous loans in Home Credit
Info about the previous loan parameters and client info at time of previous application
One row per previous application
SK_ID_BUREAU
SK_ID_PREV
SK_ID_PREV
bureau_balance.csv
Monthly balance of credits in Credit Bureau
Behavioral data
POS_CASH_balance.csv
Monthly balance of client's previous loans in Home Credit
Behavioral data
instalments_payments.csv
Past payment data for each installments of previous credits in Home Credit related to loans in our sample
Behavioral data
credit_card_balance.csv
Monthly balance of client's previous credit card loans in Home Credit
Behavioral data

Methods

First I would store the data in the PostgreSQL database and clean and aggregate the historical datasets by applicants to derive simple statistics such as mean, standard deviation, median and sum using SQL.

I will also join aggregated tables to the main table, or Application, using SQL codes.

For the explanatory analysis, I would summarise the proportion of missing values, statistics of numerical variables, frequencies of categorical variables and correlations with the target variable for main variables in Application table with appropriate visualisation -using histogram, boxplot, density plot, and so on. I will implement missing values with average (mean or median) if its proportion is less than 0.8, otherwise drop the variable in both analysis and modelling.

For the modelling, I would apply Light GBM algorithm, which is most commonly used for this kind of imbalanced big-size data modelling, with the random search optimization. To validate the model selected form the process, cross-validation with five folds would be sufficient. At the prior of this process, the whole data will be split into two – namely training and test - at random, with the ratio of 7:3. Training set uses for the modelling process, and the model derived from the process is applied to testing set, then the outcome in the metric of AUC would be reported.

Reports

The full report consists of descriptive analysis, modelling process and outcomes. Python codes and presentation slides are also presented separately.