**Project title:** Customer credit risk analysis

**Team member:** Maryam Gholamhossein

* **Introduction:**

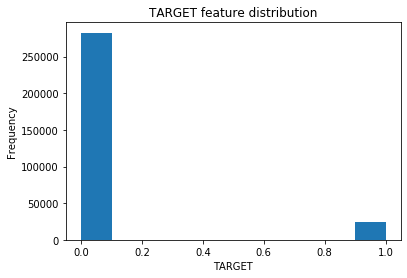
Ant finance firm focuses on giving loans to the population with insufficient or non-existence credit history. This firm is looking for better models and systems to predict their applicants’ repayment abilities. Home Credit had provided a lot of historical data including seven separate tables, as follows:

1. Main table contains information about client’s family, employment, credit info other credits on car lease, financial situations, housing status, loan type. This table also contains the TARGET feature. When 1, means the client with payment difficulties and 0, means without payment difficulties.
2. Bureau table contains information about past applications of a client in Credit Bureau including status of each credit, amount of annuity and so on.
3. Bureau Balance table contains status of credit bureau loan and month of balance relative to the application date and some behavioral data.
4. POS CASH balance table contains information if a client has passed due installments and number of installments left to pay for the previous cash loans and some behavioral data.
5. Credit card balance table contains information if a client has passed due installments and number of installments left to pay for the previous credit cards and some behavioral data.
6. Previous application table contains all the available information about previous home credit loans of an applicant.
7. Installments payments table includes repayment history for the previously disbursed credits and some behavioral data.

* **Problem description and summarize of the project and results**

Here, the problem is a supervised learning since we have TARGET variable to learn from. It is also binary classification.

The main dataset shows TARGET feature is an imbalance feature with 92% being 0 and 8% 1.



For dealing with this imbalance dataset and avoiding overfitting the model, I use K-fold cross validation and we will use *StratifiedKFold* API from sklearn library. In this project I have used 5-fold cross validation, and then I will take the average of the five models. Also, I use *auc* as evaluation metric for the model. For the model I have used lightgbm API. Lightgbm is a gradient boosting framework which uses an ensemble of decision trees and because of its accuracy, efficiency and stability is being used. [ref.1 ]

Another technique I used to avoid overfitting problem, setting “early stopping” by monitoring validation score. This can be set in the model fitting part for lightgbm model.

**AUC** (Area Under the Curve) is a probability curve that plots the True Positive Rate against False positive rate at different threshold values. It is showing how well a classifier distinguishes between classes. In other words, a model with higher AUC is doing better in distinguishing between positive and negative classes. In contrary to the F1 score in which we need to define a threshold and the results would be change depending on the selected threshold, AUC curves contain all the possible thresholds. [ref.2]

(recall) True Positive Rate = TP/(TP+FN)

False Positive Rate = FP/(FP+TN)

Missing values and some observations:

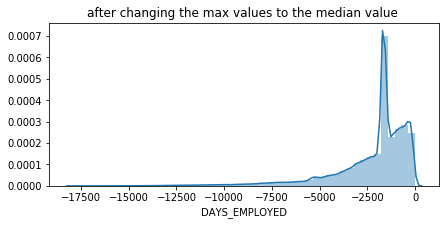
Except bureau\_balance dataset, all other datasets had missing values. After plotting

each feature using distribution plot, we decide what value should be fed to the NANs of each feature.

I used median, mean and unknown for the categorical features. Please see file “Preprocessing.py” of the project for the list of features to be filled with median, mean or unknown values for each dataset of the project.

I used label encoding for the categorical features of this project. I tried one hot encoding for some of the features, but the improvement of the model was so little that I decided to go with label encoding for all the Cat. Features.

In the main table, “DAYS\_EMPLOYED” feature has some rows with max value of more than 1000 years. First, we set these max values to nan., and then fill this nan values by median of this feature.



* **System pipeline:** 
  + Data preprocessing:

For data preprocessing, first I looked for the missing values. Also, I have checked if the missing values can be related to TARGET variable which was not the case in this project. i.e., there was no relation between the missing features and having the loan defaulted in this dataset.

I have used label encoding for the categorical data. I tried using one-hot encoding for the cat. features with more than 2 unique values. But did not see much difference in terms of model performance.

Missing values have been replaced by mean, median and unknown depending on the distribution of that specific feature.

* + Feature engineering:

For feature engineering, we started with looking at correlation between different features of the main dataset. I dropped the features with higher than 90 percent correlation. In this project between two candidate features to be removed, the feature with lower variance is removed since it contains less information.

Before creating new features, I used lightgbm model to see how is the importance features without any new features and “EXT\_SOURCE\_1”, “EXT\_SOURCE\_2”, “EXT\_SOURCE\_3”, “DAYS\_BIRTH” had the highest importance.

Next for this specific project, I used the help file provided by Yimin, youtube, Kaggle [ref. 4], [ref. 5] to get ideas in generating new features and aggregate the new features to the main dataset. These new features could be categorized into following:

1. External features:   
   “EXT\_SOURCE\_1”, “EXT\_SOURCE\_2”, “EXT\_SOURCE\_3” features are normalized so we cannot tell exactly what they are. What we can do other than aggregating their statistical information such as sum, min, max, median, we can create new features by creating polynomial features from these three important features. I created the production of these three features and weighted summation of these three features.
2. Financial status from main table and previous applications:  
   using number of family members, annual income, requested credit, age, we created many new features.
3. Behavioral features:  
   in Bureau, installments, credit card balance, we have time-series features that we can use for new features. For example, for bureau dataset, number of SK\_ID\_BUREAU for each SK\_ID\_CURR can be used.

In bureau\_balance dataset, MONTHS\_BALANCE can be sorted and then by grouping by SK\_ID\_BUREAU, we can count the number MONTHS\_BALANCE and aggregate it to the main dataset as a new feature. Important tip is that after aggregating new features to the main dataset, we will be having some NAN values, we should set the NAN values for the numerical features zero and not the median as we did in preprocessing part. Because here, maybe the applicant does not have a previous credit card or debt history so it should be set to zero if it is missing in the aggregated dataset.

* + Algorithm and Methodology:

In this project, I used lightgbm model which is a gradient **boosting** framework that uses tree-based learning algorithm.

[Ref. 6] gbdt is widely used because of its accuracy, efficiency, and stability. It is an ensemble model of decision trees and uses Gradient optimization and Boosting techniques to train the model.

Below figure [Ref. 7], shows model complexity against validation error. We want our model ideally to have a low bias (meaning model had taken all the provided information into account and had learned the relationships between the features and the target variable.) and low variance (meaning the model’s sensitivity to the fluctuations in the training dataset is low and our model hasn’t memorized the training points.) .

Machine generated alternative text:
Total Error = (Bias + Variance) + Irreducible Error 
Total Error 
Underfitting 
Overfitting 
Variance 
Bias2 
Model Complexity 

In order to achieve optimal model, we have used K-fold cross validation using sklearn library. We implemented 5-fold cross validation so that we built 5 models, the training and validation part of our dataset would have devided in 5 parts and for each model one different part would be used for validation phase. This way, all the data (except test part) are given chance to be exposed to the algorithm for training and also for validation. In the end, we used the average of these 5 models to evaluate our mode.

In order to avoid falling into local minimums of the training error, we set the early\_stopping parameter of the lightgbm hyper parameter to 300, so once the algorithm reaches a minimum error, it continues for more 300 iterations to make sure it has found the minimum training error.

* + Results:

With total of 467 features:

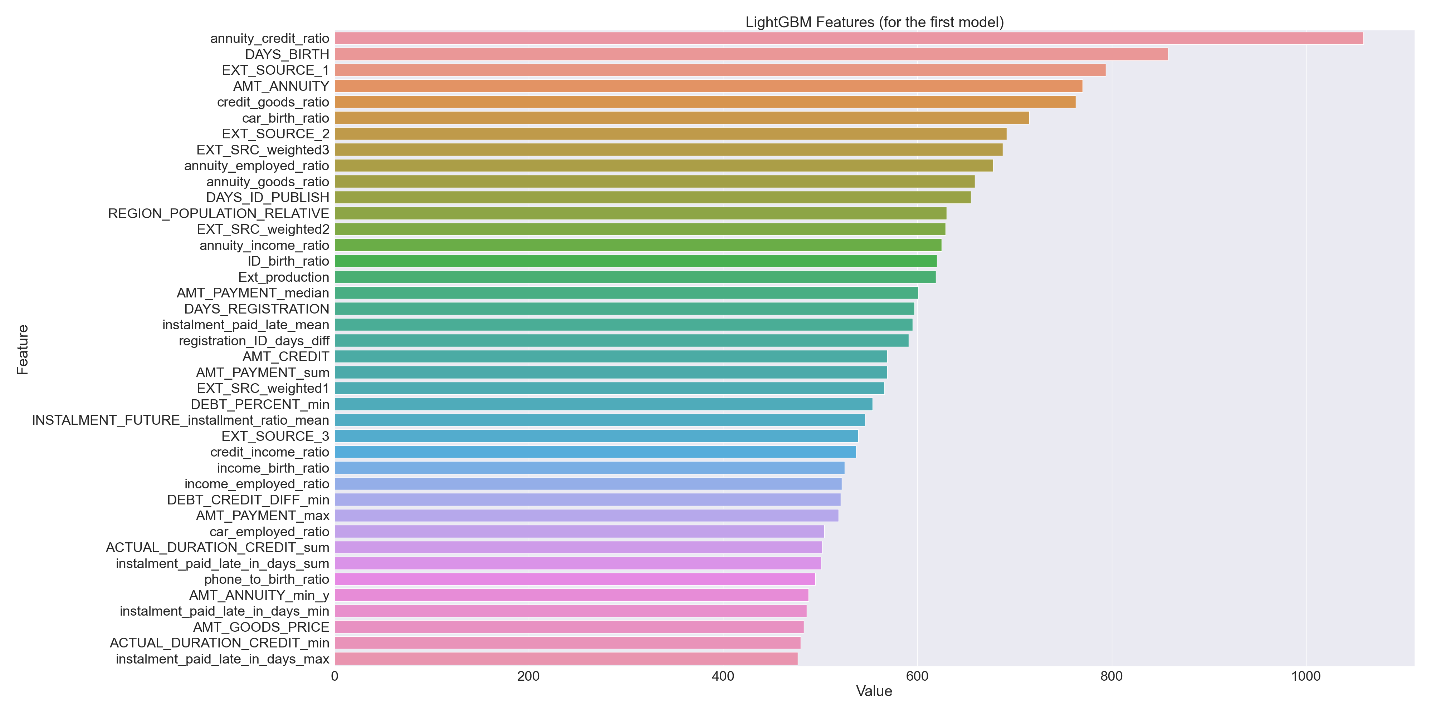
**Results** of auc:

auc list: [0.7928684026659334, 0.7897126564206951, 0.7913939053148429, 0.7863942950972072, 0.7962532350137924]

for test dataset: 0.7963688188814078

auc\_mean is: 0.7913244989024941

**Important features:** Here is the list of the 40 most important features recognized by the first model to predict the probability of an applicant to default the loan. As seen in below figure, a lot of our engineered features are among the list and the most important feature for the first model is annuity/credit ratio.



* Statement of contributions:

Steven and I did all parts of project individually and we have decided to submit our reports individually. So, all the works in this report is my work.

* References:

1- <https://neptune.ai/blog/lightgbm-parameters-guide>

2- AUC-ROC Curve in Machine Learning Clearly Explained.pdf provided as reference by the professor

3- course slides

4- https://www.youtube.com/watch?v=EVF2lqHnHq4&ab\_channel=RachelL&app=desktop

5- <https://www.kaggle.com/willkoehrsen/start-here-a-gentle-introduction/>

6- <https://neptune.ai/blog/lightgbm-parameters-guide>

7- https://www.linkedin.com/learning/applied-machine-learning-foundations/