Home Credit Default Risk report

Team Codex

Surya Teja
IMT2018080

Venkata Raghava
IMT2018023

Abdul Hadi
IMT2018041

Abstract

The objective of this project is to create a model that will predict whether a person will be able to repay loans or not when the model is given the person's financial history.

Problem Statement

Many people struggle to get loans due to insufficient or non-existent credit histories. And, unfortunately, this population is often taken advantage of by untrustworthy lenders. Then home Credit strives to broaden financial inclusion for the unbanked population by providing a positive and safe borrowing experience. In order to make sure this underserved population has a positive loan experience, Home Credit makes use of a variety of alternative data including telco and transactional information to predict their clients' repayment abilities. While Home Credit is currently using various statistical and machine learning methods to make these predictions, they're challenging Kagglers to help them unlock the full potential of their data. Doing so will ensure that clients capable of repayment are not rejected and that loans are given with a principal, maturity, and repayment calendar that will empower their clients to be successful.

Dataset Description

1) Application_test,train.csv:

In this it is divided into two parts such as train and test, where train has a target and test has no target.Static data for all applications. One row represents one loan in our data sample.

2) Bureau.csv:

All clients credit provided by other financial institutions that were reported to the credit bureau. For every loan in our sample, there are many rows as the number of credits the clients had in the credit bureau before the application date.

3) Bureau Balance.csv:

The balances of the credit bureau will be here. This table has one row for each month of history of every previous credit reported to the Credit Bureau.

4) POS_CASH_balance.csv:

Monthly balance snapshots of point of sales and cash loans of applications are present here. This table has one row for each month of history of every previous credit in Home Credit (consumer credit and cash loans) related to loans in our sample such as loans in the sample.

5) Credit_card_balance.csv:

The Monthly balance snapshots of previous credit cards that the applicant has with Home Credit are present in it. This table has one row for each month of history of every previous credit in Home Credit (consumer credit and cash loans) related to loans in the sample.

6) Previous_application.csv:

All previous applications for Home Credit loans of clients who have loans in our sample are present in it. There is one row for each previous application related to loans in our data sample.

7) Installment_payments.csv:

The history of repayments for the previous distributed credits in the home credit related to the loans are in our sample are here.There is one row for every payment that was made and one for missed payments and there is row for one payment of one installment which is corresponding to the payments of the previous home credit related to loans in our sample are given here

8) Home credit_columns_description.csv:

This file contains all descriptions of files in columns in various data files.

Data Processing Tasks

A. Visualisations and Inferences

- The graph given below (Fig 1) shows the plot between count and Target. This is plotted for showing a rough estimate of the number of data points with Target value 0 and Target value 1.

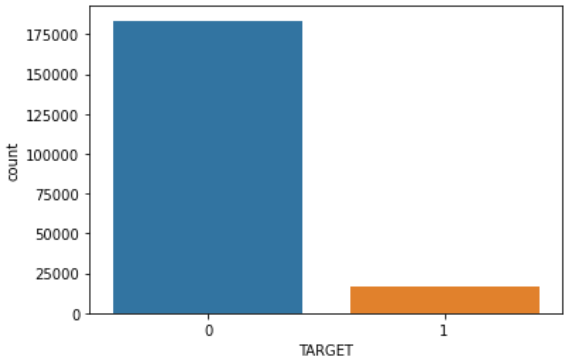

Fig 1

- In the graph given below (Fig 2), we are going to plot the ages of the client in years against the count. As all the values are meaningful (i.e all the age values of the clients are meaningful), there are no outliers in this.

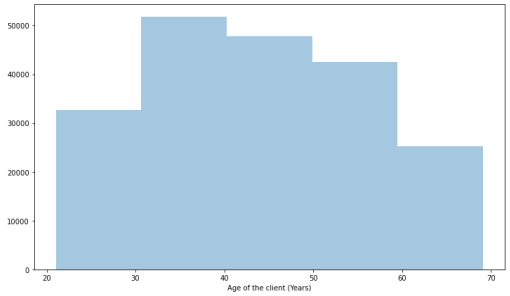

Fig 2

- The following graph (Fig 3) is plotted between Time before the loan application the persons started current employment in years and the frequency or count of the number of persons with the same number of years of employment. As clearly shown in the graph below we detect an outlier (at 1000 years the graph is showing some count).

.

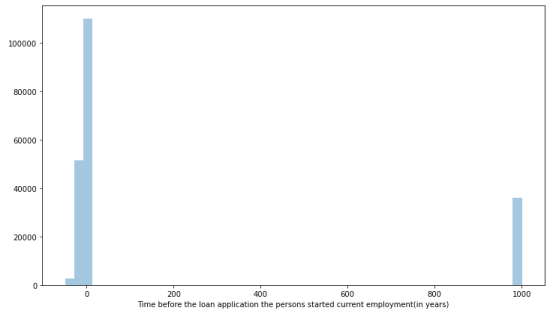

Fig 3

- The following graph (Fig 4) is plotted between the count of the outliers or anomalies and Target. This just gives us a rough idea of the distribution of Target among the outliers.

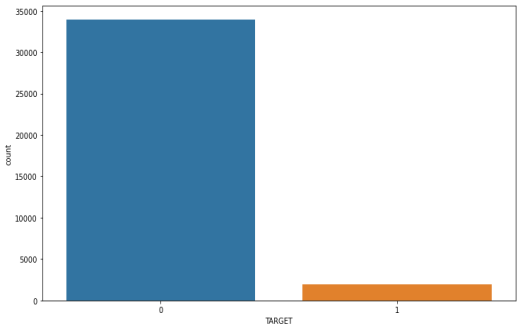

Fig 4

- The below graph (Fig 5) is plotted between years worked before application and the frequency. We can observe from the graph that there are no outliers as all the values are meaningful (the graph is distributed over a meaningful range of age value

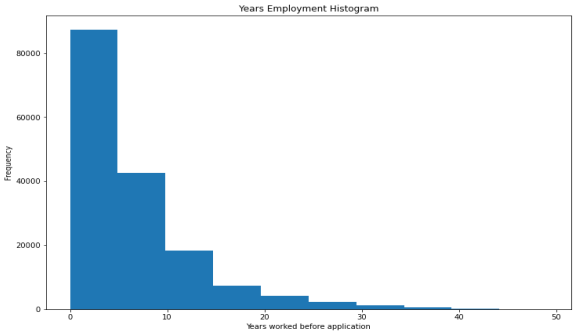

Fig 5

B. Data Cleaning and Preprocessing

I. Missing Values

For this purpose a function called, *missing_columns* is written. This function takes in a whole dataframe and then gives a dataframe which consists of the number of missing values and the missing value percentage of each column in the dataframe given to the above mentioned function

II. Outliers

The first outlier that we encounter in the data is in the 'DAYS_EMPLOYED' column of the data where we find that the maximum value is 1000 years approximately. We find all the values of DAYS_EMPLOYED which lie in between 990 to 1000 years and replace it with NaN. We do this for both test and train datasets.

III. Imputations

All the imputations are done by the SimpleImputer which is imported from the sklearn.impute library. This imputer fills up all the missing values or NaN values (which includes outliers, unavailable data).

IV. Combining/ Merging Data Sources

After using a polynomial of degree 4 for the features with the most positive correlation to fit for the data set, we merge the polynomial features into the original dataframes using their indices for further use of these polynomial

features in the various models which we will be using.

C. Feature Engineering

I. Dimensionality Reduction

On fitting the model with a polynomial of degree 4, we get various polynomial features (like *EXT_SOURCE_3^2 DAYS_BIRTH*, and so on). From these various polynomial features, we select the features on which the *TARGET* depends the most on. This can be done by finding the correlation of all these columns with the target. We select the features which highly .

II. New Features

Some new features such as *debt-to-income ratio (DIR), annuity-to-income ratio (AIR), annuity-to-credit-ratio (ACR)* and *days-employed-to-age-ratio (DAR)* are introduced. They are defined as:

- *DIR = Credit amount of the loan / Total Income* which is nothing but *AMT_CREDIT/AMT_INCOME_TOTAL.*
- *AIR = Loan annuity / Total Income* which in terms of the columns of the data is *AMT_ANNUITY/AMT_INCOME_TOTAL*.
- *ACR = Loan annuity/ Credit amount of the loan = AMT_ANNUITY/AMT_CREDIT*
- *DAR = Number of days employed/ Age of applicant* which, in terms of the columns is *DAYS_EMPLOYED/DAYS_BIRTH*

Model Training

The models we used for training are Logistic Regression, Naive Bayes, Random Forest, Lightgbm and Xgboost. First a polynomial of degree 4 was used to fit the model and then over the features that are generated by the degree 4 polynomial, different models were used. The logistic regression was implementedby importing the *LogisticRegression* package from sklearn.linear_model. An object of this which is named *log_regressor* is then created. This object is then used to get the predictions for the test data. The final predictions are then stored in *log_regression_pred_test*. After training it using Logistic Regression, we also used the *RandomForestTreeClassifier* package to train using a Random Forest tree. After this, we used the *Naive Bayes* package, *Lightgbm* package and the *Xgboost* package to train them. Further, we used a stacked ensemble of the 2 best models (the 2 models which gave us the highest scores are Xgboost and Logistic Regression).

influence the *TARGET* column (in other words they have highly positive correlation or highly negative correlation). These columns which highly influence the *TARGET* column are: *EXT_SOURCE_3*EXT_SOURCE_2, EXT_SOURCE_3*EXT_SOURCE_2*EXT_SOURCE_1.* The other columns which are not useful or which don't influence the data get dropped off (like *EXT_SOURCE_3*). These columns have less negative correlation

Results

A. Evaluation Metrics

Evaluation metrics like F1-score or ROC curve (receiver operating characteristic curve) can be used for evaluating the models mentioned above.

B. Comparison of different machine learning techniques.

As said before, we used various models like Logistic Regression, Naive Bayes, Random Forest trees, Lightgbm and Xgboost. The bestresults were seen from Logistic Regression and Xgboost (these two models gave us the 2 best scores among all the models that we tried). To improve this further we used a stacked ensemble where we stacked our two best models i.e the Logistic Regression and Xgboost.

Conclusion

Hence, with the help of data from the past, we were able to create a model (logistic regression model) which predicts whether a person can repay loans or not.

References

[1] Geeksforgeeks post like

- https://www.geeksforgeeks.org/understanding-logistic-regression/
- https://www.geeksforgeeks.org/random-forest-classifier-using-scikit-learn/
- https://www.geeksforgeeks.org/xgboost-for-regression/

[2] Towardsdatascience

- https://towardsdatascience.com/naive-bayes-explained-9d2b96f4a9c0
- https://towardsdatascience.com/understanding-lightgbm-parameters-and-how-to-tune-them-6764e20c6e5b
- https://towardsdatascience.com/lightgbm-vs-xgboost-which-algorithm-win-the-race-1ff7dd4917d

[3] YouTube videos which cleared our concepts and helped us in implementing models are:

- https://www.youtube.com/watch?v=opQecq1pmWY
- https://www.youtube.com/watch?v=MxiktOPmhV8.

[4] Stackoverflow posts like:

- https://stackoverflow.com/questions/31344732/a-simple-explanation-of-random-forest#:~:text=A%20random%20forest%20is%20a,of%20number%20n_estimators%20in%20sklearn
- https://stackoverflow.com/questions/10059594/a-simple-explanation-of-naive-bayes-classification#:~:text=Naive%20Bayes%3A%20Naive%20Bayes%20comes,prior%20knowledge%20and%20independence%20assumptions.