Home Credit Default Risk Prediction

TEAM: Sp@rKliNG newBies

G.Rajeev
MT2020075
M.Tech, IIITB
Rajeev.Gamidi@iiitb.org

N. Siva Harish Dutt
MT2020043
M.Tech, IIITB
Siva.HarishDutt@iiitb.org

B. Vamsi Srivathsav
MT2020164
M.Tech, IIITB
vamsi.srivathsav@iiitb.org

Abstract : The objective of this project is to use historical loan application data to predict whether an applicant will be able to repay a loan. Main focus is kept on analysis of 122 features with good amount of missing entries over the main data table. Several machine learning models were trained and applied on the given data. Algorithms like, XGBoost , LightGBM and Random forest have given the best performance when weighted on average.

I. Problem Statement

Home Credit Group 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.

The objective of this project is to use historical loan application data to predict whether 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)

In this project, the goals achieved are:

- Data Exploration routines are designed and implemented to do Statistical analysis and Visualization.
- Classification models such as Naive Bayes, Logistic Regression,Random Forest, and Gradient Boosting Machine (Xgboost) are built to predict whether or not an applicant will be able to repay a loan.
- Evaluated Classification models by ROC – AUC .

II. DATASET

The data used in this project is provided by Home credit project hosted on Kaggle. Predicting whether

or not a client will repay a loan or have difficulty is a critical business need, and Home Credit wants to unlock the full potential of their data to see what sort of machine learning/deep learning models can be develop to help them in this task.

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 which is loan id. 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.

III. DATA EXPLORATION

The goal of Data Exploration is to learn what our data can tell us. It generally starts out with a highlevel overview, then narrows into specific areas as we find intriguing areas of the data. The findings may be interesting in their own right, or they can be used to inform our modeling choices, such as by helping us decide which features to use. First, we find the size of the 7 tables which are stored in CSV files using python library pandas as follows:

```
Size of application_train:  (199882, 122)
Size of application_test:  (107629, 121)
Size of credit_card_balance:  (3840312, 23)
Size of installments_payments:  (13605401, 8)
Size of bureau:  (1716428, 17)
Size of bureau_balance:  (27299925, 3)
Size of POS_CASH_balance:  (10001358, 8)
Size of previous_application:  (1670214, 37)
```

Figura 1: Sizes of each table

1) Examining the Distribution of the Target Column:

The target is what we are asked to predict: either a 0 for the loan was repaid on time, or a 1 indicating the client had payment difficulties. We first examine the number of loans falling into each category and plot the pie-chart for count percent.

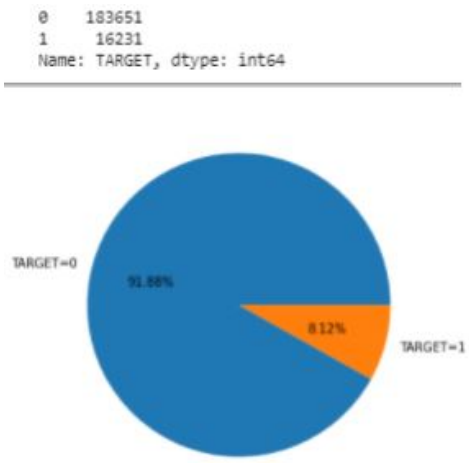

Figura 2: Target variable distribution

So, here we can conclude that the data is imbalanced.

2) Examining missing values: We look at the number and percentage of missing values in each column. In the application_train data, there are 122 columns and 67 columns of them have missing values. Below image shows the output along with top 10 columns that have missing values with the percent of missing values.

	Total	Percent
COMMONAREA_MEDI	139817	69.949770
COMMONAREA_AVG	139817	69.949770
COMMONAREA_MODE	139817	69.949770
NONLIVINGAPARTMENTS_MODE	138850	69.465985
NONLIVINGAPARTMENTS_MEDI	138850	69.465985
NONLIVINGAPARTMENTS_AVG	138850	69.465985
FONDKAPREMONT_MODE	136876	68.478402
LIVINGAPARTMENTS_MEDI	136707	68.393852
LIVINGAPARTMENTS_MODE	136707	68.393852
LIVINGAPARTMENTS_AVG	136707	68.393852

Figura 3: Top 10 features have most missing values

3) Encoding Categorical Variables: We need to deal with categorical variables as not all machine learning models can deal with categorical variables. Therefore, we have to find a way to encode (represent) these variables as numbers before handing them off to the model. There are two main ways to carry out this process:

- Label encoding : assign each unique category in a categorical variable with an integer. No new columns are created.
- One-hot encoding: create a new column for each unique category in a categorical variable. Each observation receives a 1 in the column for its corresponding category and a 0 in all other new columns.

The problem with label encoding is that it gives the categories an arbitrary ordering. The value assigned to each of the categories is random and does not reflect any inherent aspect of the category. We use Label Encoding for any categorical variables with only 2 categories and One-Hot Encoding for any categorical variables with more than 2 categories.

4) EDA for some important features:

CODE_GENDER: The percentage of Man having a difficult in repaying loan is high compared to female. Female applicants are more than male applicants.

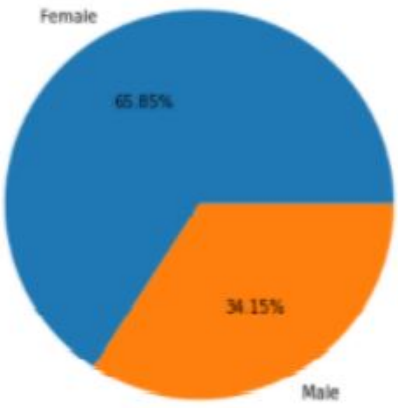

Figura 4: Data distribution in CODE_GENDER

DAYS_BIRTH: Below plot shows as the DAYS_BIRTH (Age) is decreasing, the percent of Target=1 is decreasing, so with the decrease in DAYS_BIRTH (Age), more loans are being repaid.

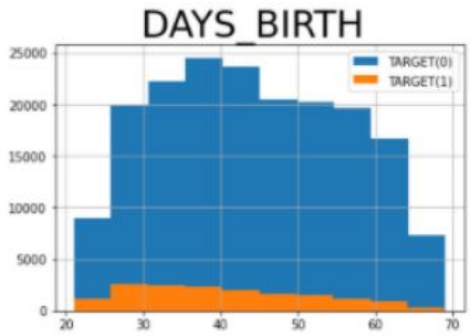

Figura 5: Data distribution in DAYS_BIRTH

EMERGENCYSTATE_MODE: It has "NO" for 98.55% datapoints so it is not much useful for model.

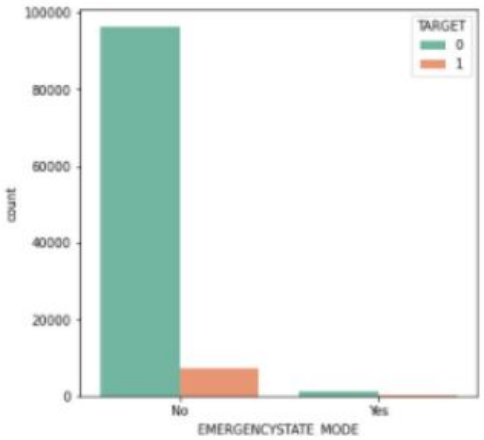

Figura 6: Data distribution in EMERGENCYSTATE_MODE

DAYS_EMPLOYED: This feature has some outliers , data around saying that client is employed for 1000 years so removing them and filling with median.

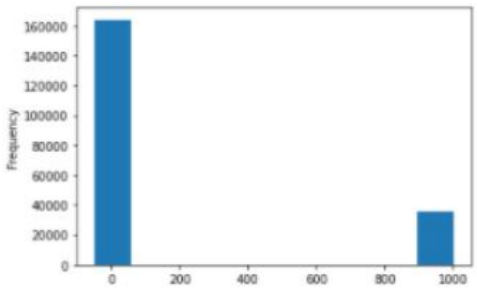

Figura 7: Data distribution in DAYS_EMPLOYED

5)Feature Engineering:

5.1)New Features

Credit Card

AVERAGE OF DAYS PAST DUE PER CUSTOMER

NO OF INSTALMENTS PAID BY CUSTOMER PER LOAN

AVERAGE DRAWING PER CUSTOMER

Installments

MAX Days Extra Taken to pay installment

MAX INSTALMENT difference (paid less)

POS_CASH:

AVERAGE OF DAYS PAST DUE PER CUSTOMER

TOTAL_PAID_INSTALMENTS.

Bureau :

This data talks about the Loan data of each unique customer with all financial institutions other than Home Credit For each unique SK_ID_CURR we have multiple SK_ID_BUREAU Id's, each being a unique loan transaction from other financial institutions availed by the same customer and reported to the bureau.

1.Status of the Credit Bureau (CB) reported credits-creating separate feature for active and closed accounts.

2. summed up the no.of total credits a person took previously.

3. calculated the average of past loans per type.

4. calculated the average of total debt credit ratio.

Bureau Balance

This is 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.

1. Represented days past due in terms of months for easier computation lest say if dpd is 0-31 then 1 is shown and 31-60 then 2 is shown likewise.

2. status of previous credit is shown whether they are active or closed.

3. summed up the monthly balance for a single person.

Previous application

1.Calculated aggregate values that is sum,average,median to all payments involved in this dataset.

2.Counted total number of previous applications the person took.

3.Took this previous credit approved or refused or cancelled or unused feature which is important in seeing why is it refused/cancelled/unused if credit is refused/cancelled/unused.

IV. MODEL TRAINING

The following Machine learning algorithms and built Classification models for our supervised classification task :

1) Naïve Bayes

2) Logistic Regression

3) Random Forest

4) Xgboost

5) LightGBM

6) Ensembling Weighted Average for Random Forest, Xgboost and LightGBM.

Out of the above, the first two are Baseline models and last four are improved models.

V. RESULTS

a) Evaluation Metrics

Confusion Matrix

A confusion matrix shows the number of correct and incorrect predictions made by the classification model compared to the actual outcomes (target value) in the data. The matrix is NxN, where N is the number of target values (classes). Performance of such models is commonly evaluated using the data in the matrix. The following table displays a 2x2 confusion matrix for two classes (Positive and Negative).

Confusion Matrix		Target			
		Positive	Negative		
Model	Positive	a	b	*Positive Predictive Value*	a/(a+b)
	Negative	c	d	*Negative Predictive Value*	d/(c+d)
		Sensitivity	*Specificity*	**Accuracy** = (a+d)/(a+b+c+d)	
		a/(a+c)	d/(b+d)		

- Accuracy : the proportion of the total number of predictions that were correct.
- Sensitivity or Recall : the proportion of actual positive cases which are correctly identified.
- Specificity : the proportion of actual negative cases which are correctly identified.

b) Comparison of different machine learning techniques:

LightGBM :

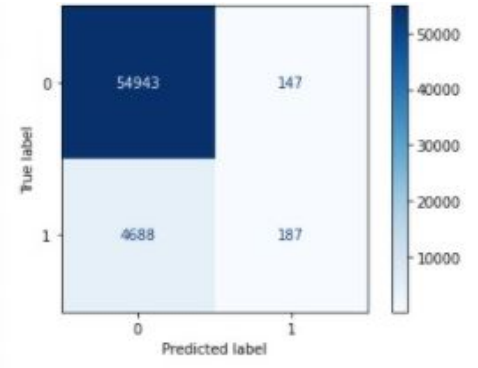

Figura 8: Confusion matrix for LightGBM

TN = 54943, FP = 147 , FN= 4688, TP = 187

Accuracy = (TP+TN) / (TP+TN+FP+FN)	0.91936963
Sensitivity, Recall or TPR = TP / (TP+FN)	0.03835897
Specificity or TNR = TN / (TN+FP)	0.99733

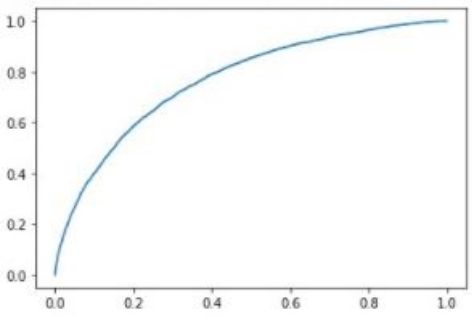

Figura 9: ROC Curve for LightGBM

roc_auc score : 0.7680329381757591

XGBoost :

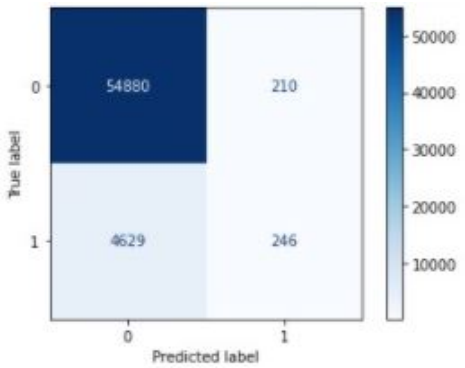

Figura 10: Confusion matrix for XGBoost

TN = 54880, FP = 210 , FN= 4629, TP = 246

Accuracy = (TP+TN) / (TP+TN+FP+FN)	0.919302926
Sensitivity, Recall or TPR = TP / (TP+FN)	0.05046153
Specificity or TNR = TN / (TN+FP)	0.99618805

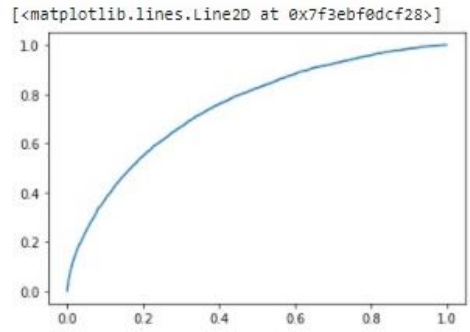

Figura 11: ROC Curve for XGBoost

roc_auc score : 0.748406990891362

Random Forest :

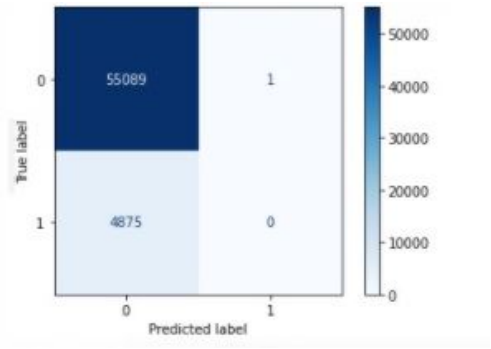

Figura 12: Confusion matrix for Random Forest

TN = 55089, FP = 1 , FN= 4875, TP = 0

Accuracy = (TP+TN) / (TP+TN+FP+FN)	0.91868590
Sensitivity, Recall or TPR = TP / (TP+FN)	0.0
Specificity or TNR = TN / (TN+FP)	0.99998

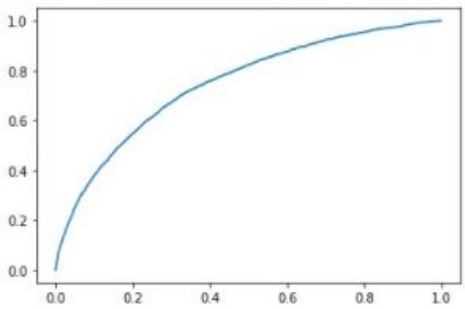

Figura 13: ROC Curve for Random Forest

roc_auc score : 0.7468546034228374

The Datamodels and their respective generated scores are listed below:-

1) Random Forest - 0.7323

2) Xgboost - 0.73596

3) LightGBM - 0.7627

4) Ensembling Weighted Average for Random Forest(0.1), Xgboost(0.3) and LightGBM(0.6) - 0.78105

.

VI. CONCLUSION

After the analysis of results of all training algorithms, and of the important models used, we found that Ensembling weighted average for Random Forest, Xgboost and LightGBM with ratio 1:3:6 of weights, provided the best results. Giving high weightage to the LightGBM is because of the high accuracy it obtained individually.

VII. ACKNOWLEDGEMENT

We would like to thank Professor G. Srinivas Raghavan and our Machine Learning Teaching Assistants, Tushar Masane, Vibhav Agarwal, Arjun Verma, Nikhil Sai Bukka, Tanmay Jain, Divyanshu Khandelwal, Tejas Kotha, Shreyas Gupta for giving us the opportunity to work on the project and help us whenever we were struck by giving us ideas and resources to learn from. We would also like to thank all other teams in Kaggle for being a great competitor and setting a benchmark time by time for the rest of us which acted as a driving fuel for us to constantly work hard and surpass them. We would gladly say that we had a great learning experience while working on the project. Leaderboard was great motivation to work on the project and the competition forced us to read up various articles and papers which gave us ideas and enthusiasm for the project.