Loan Risk Prediction

Shiva Keerthan Jalla - Aravindh Gowtham Bommisetty - Krishna Mallik Nanduri - Koushik Billakanti

Summary:

Due to weak or non-existent credit records, many people have difficulty obtaining loans. Because of this, untrustworthy lenders frequently take advantage of this group. A crucial task for every lending organization in the banking sector is to evaluate whether an applicant has the capability to repay the loan or if the applicant is a potential loan defaulter. Since the global financial crisis, risk management has been playing a crucial role in the banking sector. There are many underlying features that come to play if credit should be approved for the given client. The Data is provided by Home Credit, a service that provides lines of credit to the unbanked group of population.

Our Goal of the project is to report the most key features in deciding whether an applicant can repay the loan or not. Secondly, Visualize interesting patterns in the data. And finally, to predict how probable the applicant can repay the loan. This will help reduce the rejections of the client who can pay the loan which provides a safe borrowing experience.

Methods used to work on this data include, tidying to analyze the data. We have used visualization packages to understand the distribution of target variable and understand different features impacting the risk of loan default. Adding new columns based on credit, converting values of categorical variables to numerical values where the data can be used for analysis and used for predicting target variable by feeding the data to Machine Learning Algorithms. Plotting of explanatory variables with predictor variable to extract interesting patterns. Modeling the data with Logistic Regression, Decision Trees, Random Forest, Extra Trees, XGBoost, LGBM Classification techniques to predict the probability of a person whether he is a loan defaulter. Upon considering Area Under Curve (AUC), the XGBoost and LGBM Classifiers have performed best on the data in this scenario. Because of the unbalanced distribution of the target values, we scored our models based on area under the curve and not accuracy.

Methods:

We have used methods for collecting, preparing, modeling and presentation of our data. Each step of the process is explained below.

1. **Data Pre-Processing**:

   Data for the loan default risk prediction is distributed into 7 files:

   They are:

   1. Application Data
   2. Bureau Data
   3. Bureau balance data.
   4. POS Cash Balance
   5. Credit Card balance
   6. Previous Applications
   7. Installment Payments.

To move forward to work with data, we have joined the data from all the above-mentioned files and pre-process the data. Merging of bureau balance into bureau is done using SK_ID_BUREAU key, then merging bureau into train data using SK_ID_CURR key. All other tables are merged directly into train using SK_ID_CURR key. we have found very less percentage of NA or

NULL values. We have found out that the missing values are MCAR and imputed them with medians of the respective columns. There were some columns which are encoded in inappropriate datatypes. During analysis we have found the respective columns manually and converted them relevant datatypes. There were some outliers present in the data. We have used boxplots for the data to detect and remove the outliers. There are some features with inconsistent encoding of values in the cell, we have deleted that value in the cell and imputed that value with median value of that cell. For Categorical variable missing values have been imputed with “Unknown”.

For the initial analysis we have tried out to find relationships between the continuous features by scatter plot whether it has a positive or negative correlation. Secondly, we have included categorical variables with continuous features to find out the patterns. Thirdly, we have included target variable vs explanatory variables to know any interesting patterns emerge to specifically find out if any feature can explain the loan defaulter perfectly. As this is a real-world data, we are not able to find a variable which can clearly distinguish the target variable. Though, we have found some interesting facts from this initial analysis.

2. Exploratory Data Analysis:

The Exploratory Data analysis helps us in analyzing to find out different underlying patterns and summarize how each feature behave. Though we have visualized many distributions. The below are the plots are found to be interesting and important for the modeling part of the project.

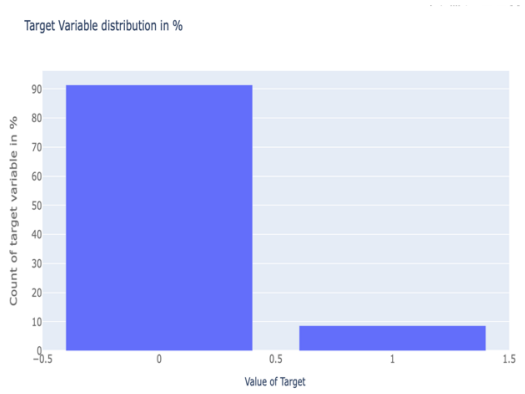


Fig. (a)

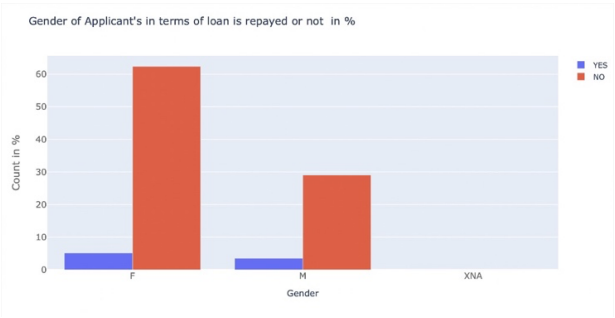


Fig. (b)

The Fig. (a) represents Target Label distribution, where we have 0’s (Loan Repayers) around 90 percentage and 1’s (Loan Defaulters) has 10 percentage of data. This visualization shows that the data is highly imbalanced. The Fig. (b) represents gender distributions about which gender group has taken more credits. Where most of the loans were taken under the name of female, and the figure also shows the percentage of gender who are loan defaulters. It was found that there are same proportion of people who are loan defaulters in the data.

Fig. (c)

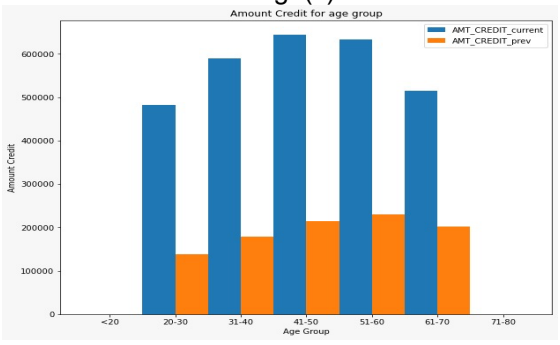


Fig. (d)

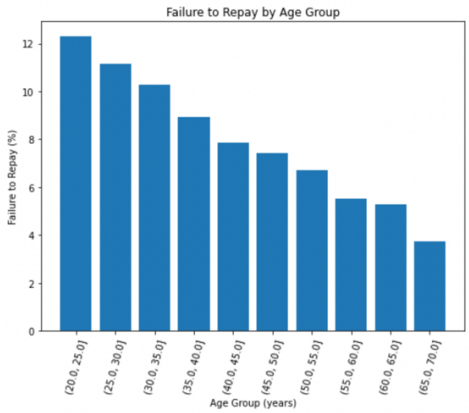


Fig. (e)

In the Fig. (c) shows that Amount of Credit given to the different age group of customers which visualizes the previous sanctioned credit and currently sanctioned credit. From the Fig. (c) we can infer that the credit amount granted for previous application and the current applications has been tripled. The Previous Highest credit grant was for the age group 51-60 and current highest credit grant was for the age group 41-50.

The Fig. (d) above plot shows the distribution of borrower ages and the target variable. From the Fig. (d) plot, we can visualize that the people who took the credit in their 20's are highly probable to be loan defaulters, and probability to be loan defaulters decreases with age and increases the probability to pay their loan.

The Fig. (e) plot shows age group vs percentage of failure to pay back the loan. This plot supports the inference taken from the Fig. (d) that as Age increases the probability of paying back the loan increase. Here the age group 20-25 customers are highly probable that they will be loan defaulters.

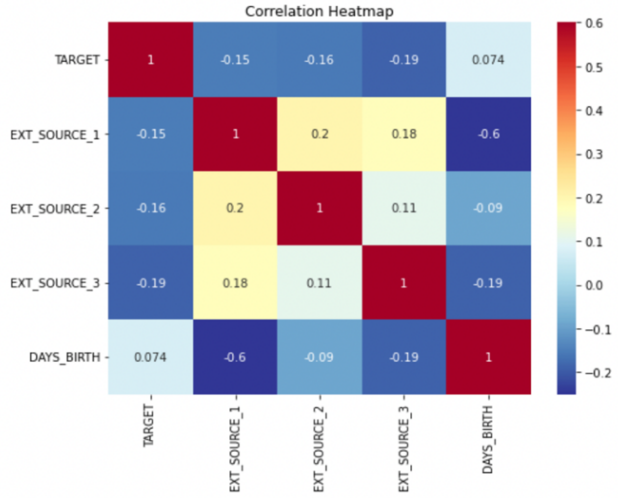


Fig. (f)

The Fig. (f) shows the correlation plot for 4 important features with how the correlate with target variable. We can infer that from the 3 variables with the strongest negative correlations with the target are EXT_SOURCE_1, EXT_SOURCE_2, and EXT_SOURCE_3. These represent "normalized

score from external data source" which means, they are a cumulative sort of credit rating made using numerous sources of data.

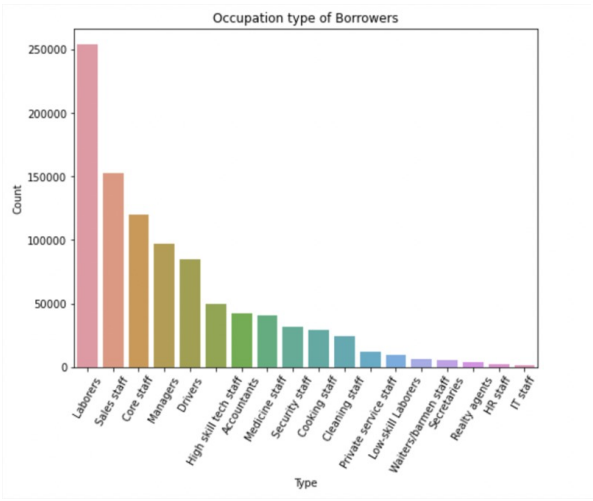


Fig (h)

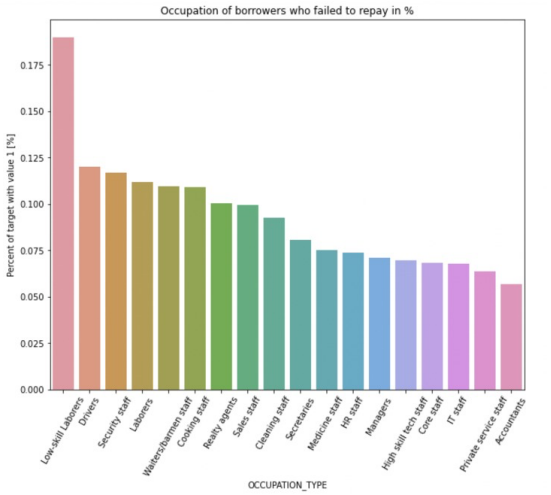


Fig. (i)

Fig. (h) shows customers occupation who borrowed the loan in the dataset.

Fig.(i) shows occupation type of borrowers Laborer's are highest and surprisingly IT Staff are in lowest number in seeking credit.
We can Infer that "Low Skill laborers" are 6th lowest to borrow, yet highest to default the loan.
Also, the least percent of defaulters are working at highly skilled jobs such as IT, Accountants, etc.

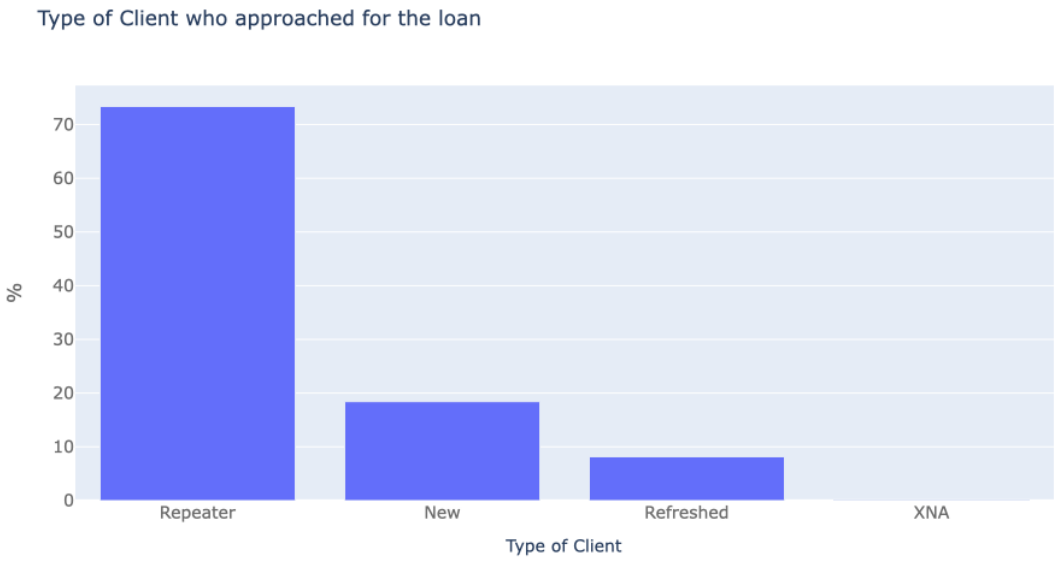


Fig. (j)

The Fig. (j) plot shows type of clients who have seeked the credit.
We can Infer that highest number of people who are taking loan have taken the credit before and second highest people taking credit are the fresh applications.

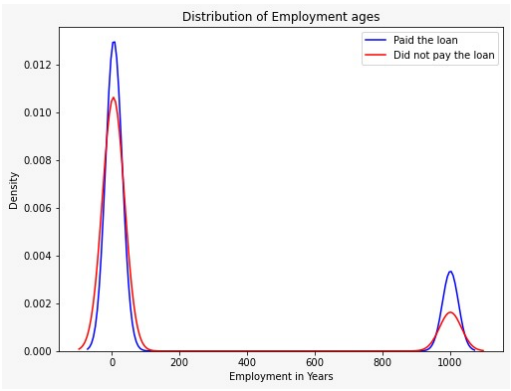


Fig. (k)

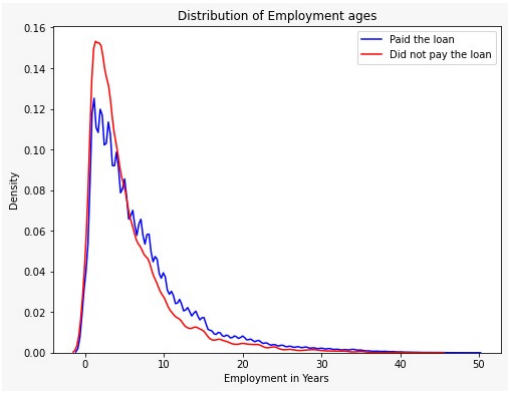


Fig. (l)

The Fig. (k) shows the distribution plot of customers employment in years.
The Fig. (l) shows that the plot after removing outlier.
From this plot we can see that there is an outlier with experience of 1000 years in employment.

3. Feature Engineering and Modeling

3.1 Cleaning data

Cleaning data before feeding it to the models is of utmost importance. Column names of almost all the datasets were slightly modified for better understanding. In the previous application dataset, quite a few columns had 365243 as value which was replaced to be a missing value. 'DPD' and 'DBD' columns in installment_payments dataset were modified for dealing with these features more efficiently.

3.2 Feature Engineering and Merging

Domain Specific Features

Introducing new features can make a whole lot of difference in the performance of machine learning models on the data. One of the efficient ways to do this is applying the domain knowledge to create relevant features. We have referred to relevant sources to gain more financial knowledge specific to Loans given out by banks and financial institutions and have created quite a few new features, out of which few played an important role in the predictive performance of our final model. 5 Ratio features like proportion of debt that is overdue were introduced in bureau dataset. 3 more new features such as Interest paid were created in previous_application dataset. Similarly, novel features were created in other datasets as well. In total, 3 Boolean features and 27 numerical features were created newly.

Group by Features

One of the challenges we faced during merging the datasets was some datasets had multiple records of the same person. To solve this, we have created numeric features by grouping on SK_ID_CURR (ID of the loan) and finding the group means. Doing this creates single record for each unique ID.
Similarly, we wanted to tackle the same issue with categorical features and hence one-hot encoded the categoric features. After this we created new features by grouping these one-hot encoded features on SK_ID_CURR and taking the group means. This was repeated for every dataset.

Merging

As there were quite a few datasets, merging was a bit complicated. But we talked about how we tackled some of our issues in the above sections. Initially, bureau_balance was merged into bureau using SK_ID_CURR key. And this merged dataset will be merged directly into application_train/test dataset later. All other tables will be merged into application_train/test dataset using SK_ID_CURR key.
Final merged dataset had 307,511 records and 413 features, out of which 398 are numerical features and 15 categorical features.

3.3 Train and Test dataset Preparation

From the train data (application_train dataset) we held back 25% of the data for validation. This proportion was unchanged for all the models. Also, for splitting the data, we used stratified splitting which retains same representation of target variable in all the splits. Two pipelines were created to deal with train and test data. The numerical pipeline had Median imputer and standard scaler normalizer. While the pipeline meant for categorical features had constant value ('Unknown') imputer for missing values and one-hot encoder as part of the pipeline. Numerical features and categorical features were identified and later sent through relevant pipelines which outputs our final train and test datasets which can now be fed into machine learning models.

3.4 Classification Models

When all the necessary information pertaining to a particular loan seeker is given, classification models should be able to predict whether this individual can repay the loan on time or will he default on the payments. As outlined above, we have the final train and test datasets split in the ratio of 75:25. 10-fold cross validation was used for validation step to estimate the optimal parameters. For choosing best hyper parameters, we have performed grid search only for few of the parameters for every algorithm to speed up the process. Optuna was also used for Random Forest and LightGBM models. Given that data is imbalanced, we have used ROC-AUC score as the evaluation metric which is more robust than other metrics for evaluating performance on imbalanced datasets.

3.4.1 Logistic Regression

It is a classification model that's simple to implement and delivers excellent results with linearly separable classes. The logistic regression model does not classify data; instead, it models the likelihood of output in terms of input. It can, however, be used to build a classifier by simple cutoff-based rules. To get the best performance out of Logistic regression, we tried both lbfgs and saga solvers and got faster performance and slightly higher scoring out of lbfgs. We also experimented with maximum iterations parameter as low as 200 but did not see any degradation of accuracy below 400. The best hyperparameters we got were C = 0.005 and penalty = l2. AUC score on test data is 0.769 which is a decent score.

3.4.2 Decision Tree

Decision Tree is a Supervised Machine Learning method in which data is continuously split according to a specific parameter. Recursive partitioning is a heuristic used to construct decision trees. This method divides the data into subsets, which are then repeatedly divided into even smaller subsets, and so on and so forth until the process ends when the algorithm determines that the data within the subsets is sufficiently homogeneous, or another stopping criterion is met. After grid search on the data and bit more manual analysis, the best parameters that we could find were max_depth = 8 and min_samples_leaf = 10. Though this is not the best model,

but it serves as benchmark for Random Forest and Extra Trees Models. This model performance was satisfactory as the AUC score on test data was equal to 0.731.

3.4.3 Random Forest

Random Forest makes use of ensemble learning, which is a technique that combines many weaker classifiers to solve complex problems. It is made up of a large number of decision trees. The random forest algorithm's 'forest' is trained using bagging or bootstrap aggregation.
The outcome is determined by the algorithm based on the predictions of the decision trees. It predicts by averaging the output of various trees. Optuna was used to find optimal values for max_depth and min_samples_leaf. We used max_depth = 39 and min_samples_leaf = 37 in the final model. Anyways the model is not very much influenced by a wide range of choices for both the hyperparameters. Hence, any pair of values between 30-40 yields more or less the same results. AUC score on test data is 0.765 which bettered the performance of decision tree model.

3.4.4 Extra Trees

ExtraTreesClassifier is an ensemble learning method that uses decision trees as its foundation. ExtraTreesClassifier, like Random Forest, randomizes specific decisions and subsets of data to prevent overlearning and overfitting from the data. But Extra Trees differs in two major ways: it does not bootstrap observations and nodes are split on random splits rather than optimal splits. Best hyper parameters were min_samples_split = 8, min_samples_leaf = 18 and n_estimators = 300. AUC Score is 0.76 and so the performance is similar to that of Random Forest.

3.4.5 XG Boost

The XGBoost ensemble model is a gradient boosting ensemble model based on decision trees. Gradient boosting attempts to predict a target variable by combining the estimates of several simpler and weaker models. Some of the features of XGBoost are as follows: Parallel tree structure, pruning trees with a depth-first approach, regularization is done to avoid overfitting. Best hyperparameters found by grid search were gamma = 0, learning_rate = 0.05, n_estimators = 300, max_depth = 6. AUC score of the model on test data is 0.78 which bettered the performance of all the models so far.

3.4.6 LightGBM

LightGBM is a gradient boosting framework based on decision trees to increase the efficiency of the model and reduces memory usage. LightGBM adopts two novel techniques Gradient-based One-Side Sampling (GOSS) and Exclusive Feature Bundling (EFB) which helps in increasing information gain and dealing with high-dimensional data efficient. Optuna was used to find optimal parameters which were learning_rate = 0.02, lambda_l1 = 2, lambda_l2 = 6. AUC score on test data is 0.7881. So, lightGBM is the best performing model.

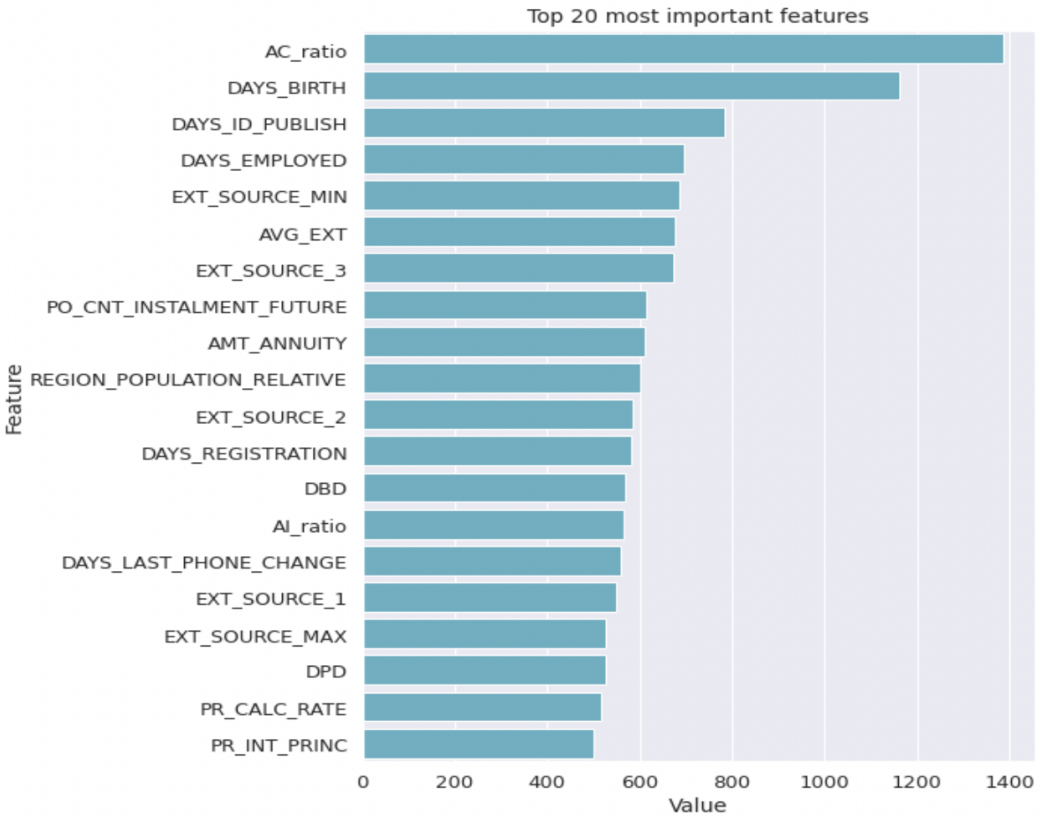
Results:

Comparative Analysis

Model Name	AUC- Train	AUC- Test
Logistic regression	0.762	0.769
Decision Tree	0.726	0.732
Radom Forest	0.757	0.766
Extra Trees	0.752	0.761
XG Boost	0.774	0.781
Light GBM	0.782	0.791

We have achieved the best results on the test data using Light GBM. While, the worst performing model was decision tree but even it performed decently. Logistic regression despite being a simple model performed well and it is the fastest of all the six models we employed for classification.

Feature Importance



Feature importance analysis informs the banks which features are more important in deciding whether to give an individual loan or not. AC ratio is the debt percentage of a client, it is calculated by the total annuity divided by total credit amount. Calculating this AC Ratio determines the debt-to-income ratio. AC ratio was introduced in the feature engineering part which tells us feature engineering should not be ignored at all.

Discussion:

Results demonstrate which are the most defining factors to play an important role in issuing credit to the unbanked population who are taken advantage by most of money lenders. These Analysis/results at bare minimum give additional advantage to customers who have bad credit score and/or not issued loan by other banks. From these results, both customers and Organization (Home Credit) can be benefitted. Previously there used to be very a smaller number of parameters which can decide that a customer can be granted a credit or not. But with the Huge data we have and with different set of parameters which can describe almost every attribute of person, can we detect underlying patterns and key features. This Analysis and Modeling can boost the efficiency of lending credits for an organization in a most effective way.

Future Enhancements:

1. Try ensemble algorithms which is mix of XGBoost, LGBM and logistic regression for the data to further improve the performance.
2. Try more models like catboost and Neural Networks for the data.

Statement of Contributions:

- **Shiva Keerthan Jalla**: Performed Data Preprocessing, Exploratory Data Analysis, Implemented Logistic Regression Algorithm. Worked on Presentation and Report.
- **Aravindh Gowtham Bommisetty**: Performed Feature Engineering, Implemented XGBoost and Extra Trees Algorithm. Worked on Presentation and Report.
- **Krishna Mallik Nanduri**: Performed Data Preprocessing, Exploratory Data Analysis, Implemented Light Gradient Boost Algorithm. Worked on Presentation and Report.
- **Koushik Billakanti**: Performed Feature Engineering, Implemented Decision Tree Classifier and Random Forest Classifier. Worked on Presentation and Report.

References:

[1] Dataset Source, https://www.kaggle.com/c/home-credit-default-risk/data

[2] Exploratory Data Analysis, https://towardsdatascience.com/exploratory-data-analysis-8fc1cb20fd15

[3] Logistic Regression, https://scikitlearn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html

[4] XGBoost Algorithm, https://towardsdatascience.com/xgboost-python-example-42777d01001e

[5] Light Gradient Boost, https://lightgbm.readthedocs.io/en/latest/

[6] Extra Trees Classifier Algorithm, https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.ExtraTreesClassifier.html

[7] Decision Trees Classifier, https://towardsdatascience.com/a-guide-to-decision-trees-for-machine-learning-and-data-science-fe2607241956

[8] Metrics for Model Evaluation, https://medium.com/@MohammedS/performance-metrics-for-classification-problems-in-machine-learning-part-i-b085d432082b

[9] Applying Standard Scaler to Dataset, https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html

[10] Seaborn for visualization, https://seaborn.pydata.org

Appendix:

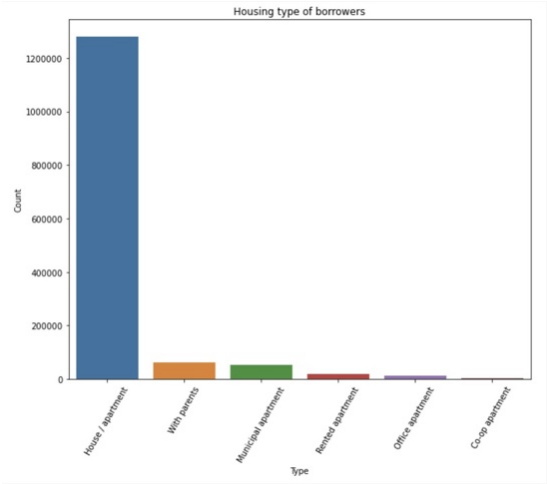


Fig. (1)

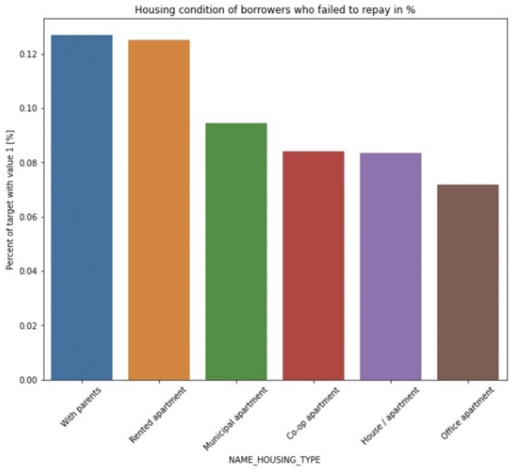


Fig. (2)

From Fig. (1) represents the count of type of housing of borrowers and Fig. (2) shows the type of housing of borrowers why failed to repay the loan.
We can infer that the people with rented apartment are relatively highly probable that they will fail to repay the loan and borrowers with Office apartment are relatively reliable compared to other categories.