Calculating creditworthiness for rural India

1. Introduction

➔ Here, we can reformulate the given problem as regression problem. Our goal is to build a regressor model that can predict the *loan_amount* from the given features for the given tabular data. To achieve this, first, we need to do the data analysis and data cleaning, data preparation for training, and then model building. After this, based on the performance of the model and to get the expected result, we need to change the model and sometimes it may require data analysis and data pre-processing again.

2. Dataset

2.1. Data Information

➔ *trainingData.csv* contains the information of loan applications from the customers of rural area. Data has many variables and it also contains missing values. So, before passing the data to model we need to do some analysis and handle these missing values to prepare data for training.

2.2. Data Analysis*

➔ First, data is explored by checking some statistics of the data, i.e., size of the data; number of columns; types of all the columns; some info of all the columns, e.g., mean, max, count, etc.; check for missing values; percentage of missing value in each column; etc.

2.3. Data Cleaning

➔ First, we need to remove all the column which contains missing datapoints equal or more than 30% of all the data. Because if any column exist with kind of data than this feature does not provide meaningful information and replace this much amount of data is also not a feasible task.

➔ After that, convert the categorical data to numrical data. But before that we need to handle missing data. So, find the categorical data and check for the number of datapoints which has missing value.

***:** All the details of data analysis is included in the code with proper comments.

➔ Now, to handle these missing value 2 methods are explored:

1. Drop the missing value
2. Replace the missing value with 0

➔ After handling the missing value categorical data is converted to numerical value (specifically, labels are converted to numbers) for both the methods.

➔ After that, feature importance graph is plotted for further process, i.e., to handle missing value for numerical variables, to check any if we can drop any variable.

1. For method - 1, no missing value remaining in data because the datapoints with missing value for numerical data are already dropped. So, it does not need further process.
2. For method - 2, there are still missing value for numerical data. Hence, *Feature importance* is plotted, which is shown below:

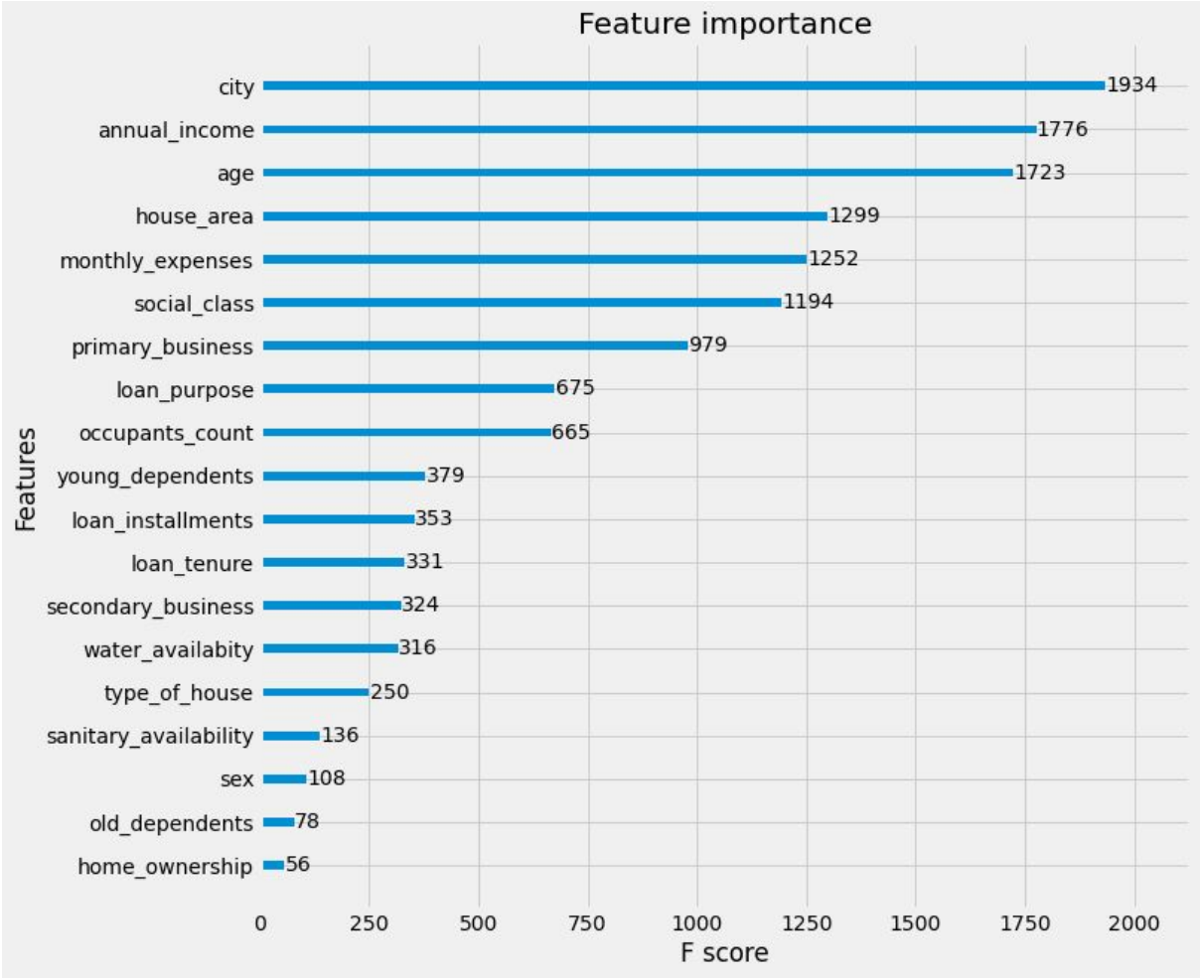

Figure 1. Feature Importance Graph for Method - 2

➔ For method - 2: From the above analysis, we can say that out of 4 features - *water_availabity*, *home_ownership*, *sanitary_availability*, *monthly_expenses*; which have missing values, only 1 variable has high importance, i.e., monthly expenses. To handle missing value, for monthly expenses, let's replace it with linear interpolation and for other 3 replace it with 0.

➔ Now, data is ready for both the methods.

2.4. Data Preparation

➔ In data preparation, first, target variable (i.e, *loan_amount*) is separated from the data and then data is divided in into 80-20% for training and testing, respectively, for both the methods.

3. Model building and Evaluation

3.1. Evaluation

➔ As this is regression problem, evaluation of all the models are done using following 3 metrics:

1. Mean Square Error (MSE)
2. Root Mean Square Error (RMSE)
3. Pearson correlation coefficient (PCC)

3.2. Linear Regression

➔ Linear Regression (LR) is a basic and commonly used type of predictive analysis. Linear regression attempts to model the relationship between two variables by fitting a linear equation to observed data.

➔ Following are the results of both the methods:

1. Method - 1
   MSE: 216675950.65
   RMSE: 108337975.33
   PCC: 0.176
2. Method - 2
   MSE: 221130801.94
   RMSE: 110565400.97
   PCC: 0.190

➔ From the above results, we can conclude that performance of LR is very less. Let's analyze why? Check how the *loan_amount* is distributed - Let's focus on the target variable (*loan_amount*). Create

a histogram of the target variable. (Note: As performance of both the methods are same, below analsis is done for only 1 method.)

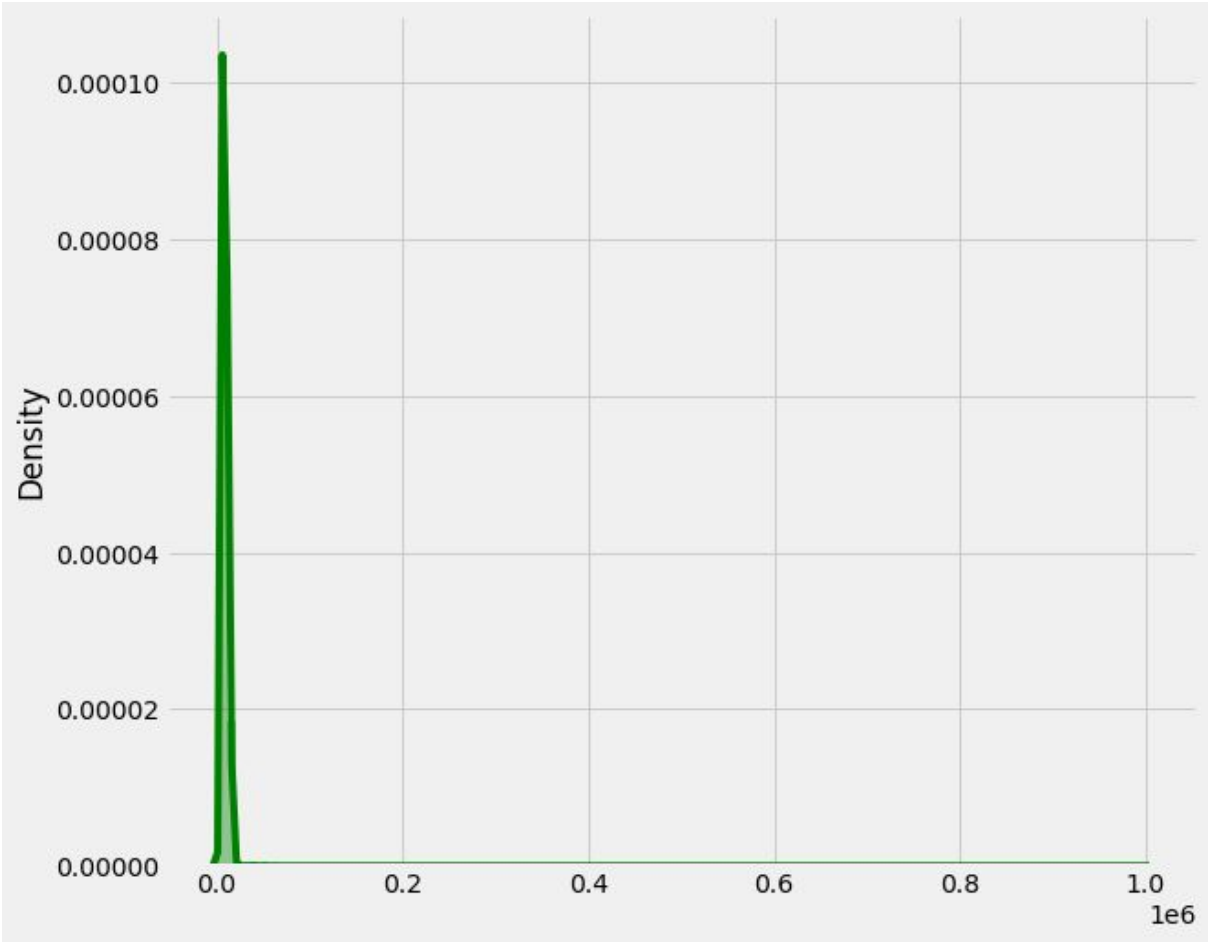

Figure 2. Distribution of Target Variable (*loan_amount*)

➔ From the above diagram we can say that target variable is not Normally distributed, it has some amount of skewness. If we want to create any linear model, it is essential that the features are normally distributed. This is one of the assumptions of multiple linear regression.

➔ Let's explore more about target variable - relationship between feature and target variable.

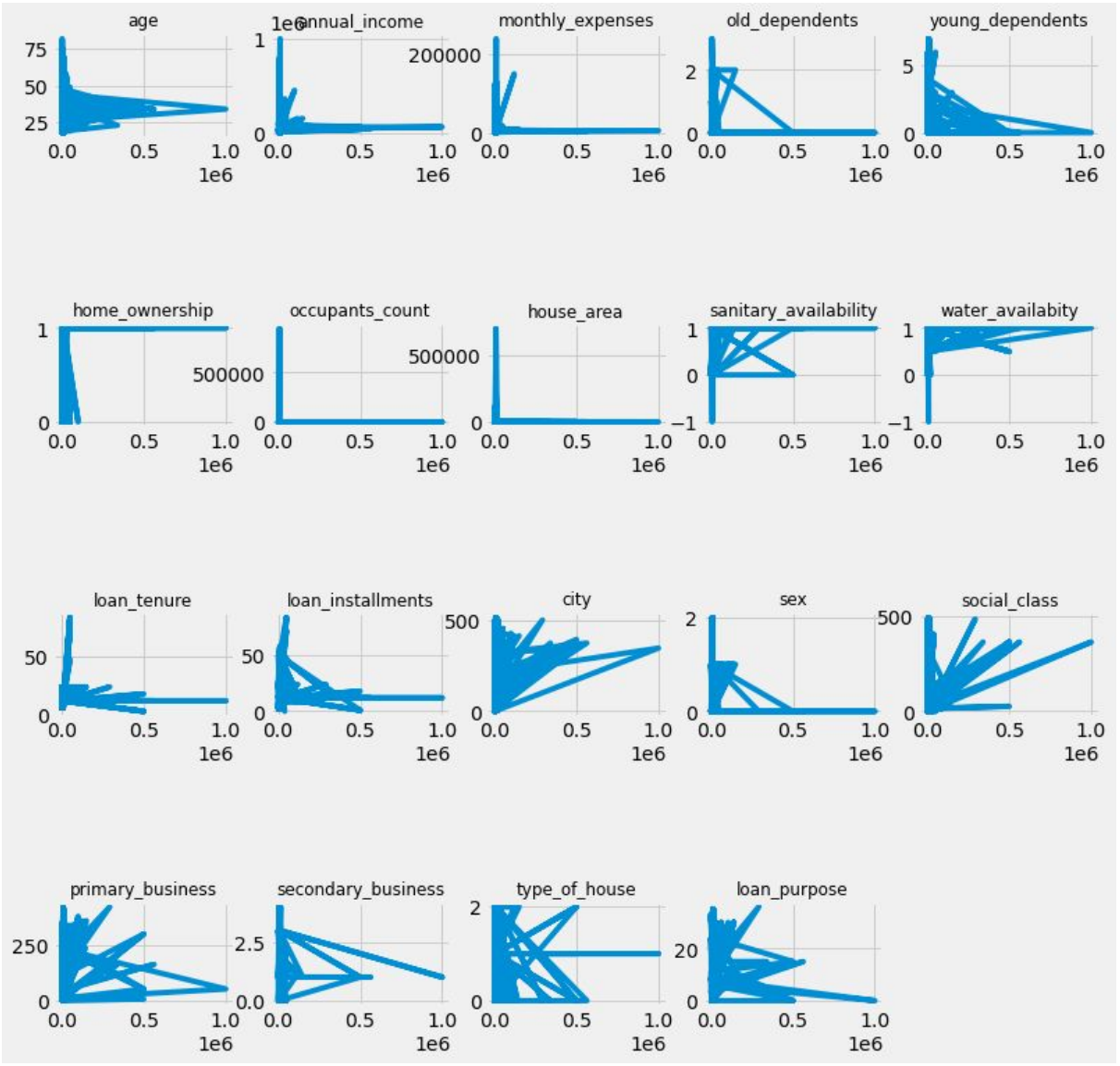

Figure 3. Relation between all the feature and target variable.

➔ From the above diagrams, we can conclude that our target variable shows an unequal level of variance across most predictor (independent) variables, which indicates linear regression model is not an efficient algorithm for this problem.

3.3. RandomForest

➔ A forest is comprised of more than 1 decision trees. It is said that the more trees it has, the more robust a forest is. Random forests creates decision trees on randomly selected data samples, gets prediction from each tree and selects the best solution by means of voting.

➔ Following are the results of both the methods:

1. Method - 1
   MSE: 107817910.41
   RMSE: 53908955.21
   PCC: 0.869
2. Method - 2
   MSE: 173246361.89
   RMSE: 86623180.95
   PCC: 0.527

➔ From the results of the above algorithm (i.e., RandomForest), we can conlcude that method - 1 is better than method - 2.

3.4. Ensemble Technique

➔ Let's use enesemble method to combine the Linear Regression and RandomForest for method - 1. Results are as following:

MSE: 154540668.85

RMSE: 77270334.42
PCC: 0.820

➔ From the above result, we can conclude that because of poor performance of Linear Regression (here, it is weak classifier), in ensemble technique we did not get higher performance comapred to both the models.

➔ So, it is better to go for another strong algorithm compared to Ensemble technique. (Note: As we have seen method - 1 is working better comapred to method - 2, but it is always good to check new algorithm for both the methods. So, we are going to explore new algorithm for both the methods.)

3.5. XGBoost

➔ XGBoost is an efficient algorithm based on the the gradient boosted trees algorithm. XGBoost minimizes a regularized (L1 and L2) objective function that combines a convex loss function (based on the difference between the predicted and target outputs) and a penalty term for model complexity (in other words, the regression tree functions). The training proceeds iteratively, adding new trees that

predict the residuals or errors of prior trees that are then combined with previous trees to make the final prediction.

➔ Following are the results of both the methods:

1. Method - 1
   MSE: 72398991.89
   RMSE: 36199495.95
   PCC: 0.932
2. Method - 2
   MSE: 281576684.97
   RMSE: 140788342.49
   PCC: 0.429

➔ From the above algorithm, we can conclude that data we have created using method - 1 is more efficient way compared to method - 2.

➔ Plot of the predicted data and original data of XGBoost algorithm and method - 1.

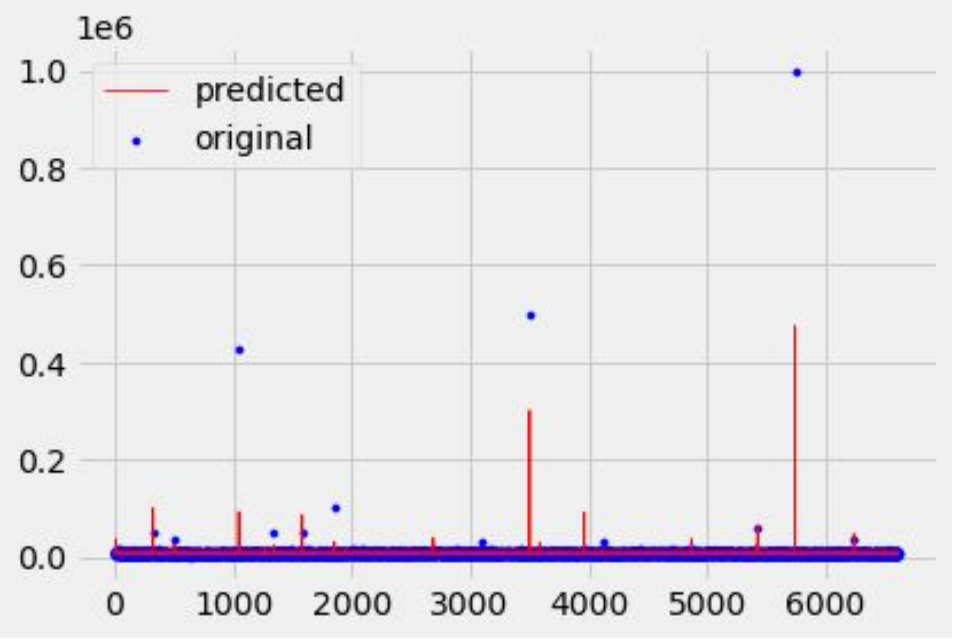

Figure 4. Predicted and original data of XGBoost algorithm and method - 1

4. Q&A: Disucssion

- Do a descriptive analysis of all the variables.

  All the data analysis part is included in the code.

- There is a new customer who needs a loan. Which models will be best suited to predict the loan_amount that can be granted to the customer?

  As we have seen above, XGBoost is giving best performance with PCC = 0.93. So, this model can give best results for new customer data.

- Build a model to predict the maximum loan_amount that can be granted to the customer. Which all variables are good predictors?

  '*It is always better to use all the features.*' If any feature is less important in prediction than instead of removing a feature (variable), it is always better to use that feature by providing it less weight. Motivated by this, all the features are use in model building.

- Is loan_purpose a significant predictor? The business has insisted on using loan_purpose as a predictor. If it is not already a significant contributor, can we still modify the model to include it?

  As we have seen in Figure 3., we can observe that there is no specific relation between loan_purpose and loan_amount. So, it is not a significant predictor. However, every feature add some information and help in prediction, we can include this feature in model.

- How will you measure the fitness of the model? Which metrics (accuracy, recall, etc.) are most relevant?

  As mentioned above following 3 metrics are used to measure the fitness of the model:
  1. Mean Square Error (MSE)
  2. Root Mean Square Error (RMSE)
  3. Pearson correlation coefficient (PCC)

5. Conclusion

➔ Out of 40,000 datapoints, we have 6935 datapoint where at least one categorical feature value is missing, I.e., at least one NaN value is present (around 1/5 data is missing). I have devised two approaches to handle the

values: 1) consider NaN values as UNKNOWN category, and 2) Remove them. From the experiments, we can observe that replacing NaN values method has poor performance compare to the removing them. One of the reasons behind this is that it may happen that because of replaced value (which is not 100% true data) model can interpret data wrongly or replaced value may introduce outliers in data. Because of limited time, this scope of study only provide two approaches to handle missing data, however, there are various ways to handle this and one can explore more techniques for better results in future. For statistical significance, I have done cross-validation for two models - Linear Regression and XGBoost. You can see the results above. In future, we can try different cross-validation methods to make sure that our model is working fine with low variation on different data patterns.