CS5617 Project- Loan Default Classification

H.R Nisal Asanga Waruna
University of Moratuwa (Department of CSE)
E-mail: nisal.23@cse.mrt.ac.lk

Introduction

Increasing growth of economy leads to find new strategies to compete with each other. Here we are interested in Loan Default Classification which is one of the challenging tasks that every Bank/Financial Institution face. Since the importance of this predicting task directly impact to the health of the banking system, conducting research in this area is very important.

This paper describes the implementation of a Loan Default Classification model using machine learning techniques. The data used for this implementation based on a published dataset in Kaggle, related to banking sector [1]. The purpose of this paper is to classify a particular customer whether default or not. This paper will help to identify a good model for better management to minimize the risk of a bank/ Financial Institution.

Data

Data Dictionary

Collaborators /Owner	**License**	**File Name**	**Format**
Prateek Upadhyay	Unknown	train_indessa, test_indessa	csv
No of attributes	**No of records**	**Data source provider**	**Size**
45	887379	Unknown	239 MB

Description of attributes

Below table contain some important feature variables.

Attribute	**Data type**
loan_amnt	Metric continuous
funded_amnt	Metric continuous
term	Metric discrete
int_rate	Metric continuous
acc_now_delinq	Metric Continuous
emp_length	Categorical ordinal
annual_inc	Metric continuous
mths_since_last_delinq	Metric Continuous
loan_status	Categorical nominal

Complete description of the dataset and attributes contain in [1]

Algorithms

To identify a good model, experiments were conducted using three classification algorithms namely logistic regression, Decision tree and XGBoost.

Logistic Regression

Logistic regression is basically an extension of liner regression restricting the outcome of the model to a binary variable.

Let's consider a model with n number of predicator variables and one binary response variable Y(i.e., 0 or 1). where we can denote

$$\pi = P\ (Y{=}1)$$

$$\text{I.e., } Y \overset{iid}{\sim} \text{Ber}(\pi)$$

Logistic regression assumes that a linear relationship between the predictor variables and the log-odds of the event that Y=1

We can write the linear relationship as,

$$\log\left(\frac{\pi}{1-\pi}\right) = \beta_0 + \beta_1 X_1 + \ldots + \beta_n X_n$$

Where β_i represent the coefficient of the ith predictor variable and β_0 represent the intercept of the model.

Where i = 1,2, 3…, n

Decision Trees

Decision trees are one of the powerful methods commonly used in various fields.

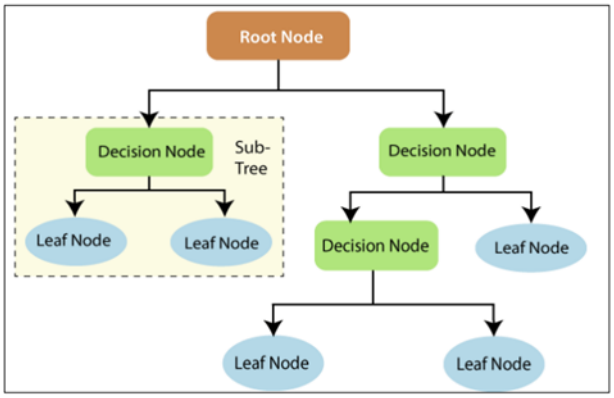


Figure 1: Decision tree Architecture [2]

Decision Trees are a tree-structured classifier, where internal nodes represent the features of a dataset, branches represent the decision rules, and each leaf node represents the outcome.

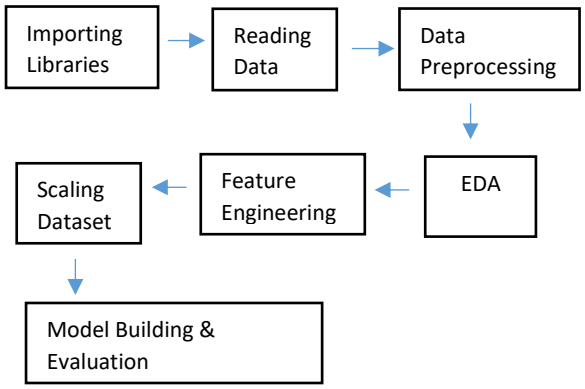
There are several Types of DT algorithms such as: Iterative Dichotomies 3 (ID3), Successor of ID3 (C4.5), Classification and Regression Tree (CART).

XGBoost

XGBoost (Extreme Gradient Boosting) is a popular machine learning algorithm that uses a gradient boosting framework to build and improve predictive models. It is designed to be highly efficient, scalable, and flexible, and is widely used for various types of regression and classification problems, including structured data, text, and images. XGBoost works by combining the predictions of multiple weak models, typically decision trees, to create a stronger and more accurate model.

Experiments

Process



In the first step necessary libraires were imported for data wrangling, visualization, modeling, evaluations & etc. Then using pandas read_csv function data were imported.

There were two datasets initially called ‘train_indessa’ and ‘test_indessa’. train_indessa’ had a shape of (532428, 45) and test_indessa’ had a shape of (354951, 44). But the test_indessa didn’t had ‘loan_statues’ column which was what needed to be predicted. Therefore test_indessa dataset had no use and train_indessa dataset used for all training and testing.

Then duplicated records were checked and there were no such records. After that Null values checked and below table represent part of the summarization of null value count and percentage.

Column Name	Null value count	%Null count
verification_status_joint	532123	100
desc	456829	86
mths_since_last_record	450305	85
mths_since_last_major_derog	399448	75
mths_since_last_delinq	272554	51
batch_enrolled	85149	16

Intially 'desc','verification_status_joint','member_id', 'emp_title' columns were dropped.

After that missing values in ‘mths_since_last_delinq’, ’mths_since_last_record’,’mths_since_last_major_derog ’ feature variables were filed with zeros.

Late payments and delinquencies are negative indicators of creditworthiness and a longer period since 'last_delinq' may viewed as riskier. Therefore, a lower value for "mths_since_last_delinq" could indicate a lower credit risk, while a higher value may suggest a higher risk. Assuming 'mths_since_last_delinq', 'mths_since_last_record' null values as customers that has no delinquency, Null values were filled with zeros. 'mths_since_last_major_derog' is also an important variable therefore setting null values with zero to indicate customers with high values with high derog effect.
Since there is no any logical way to fill null values in remaining columns those records removed to get a clean dataset.

Then Categorical ordinal variables encoded using LabelEncoder() in sklearn.preprocessing and categorical nominal variables encoded using OneHotEncoder() in sklearn.preprocessing. Along the way some string type variables were processed/dropped using different techniques.

After that Exploratory data analysis carried out and found that target variable was not balanced.

Also, correlation plot was graphed. (figure 3)

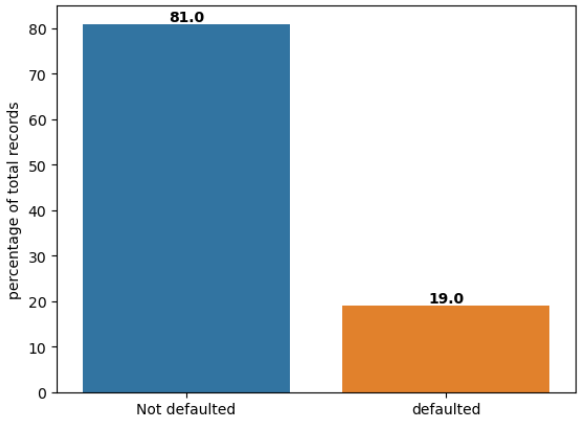


Figure 2: classes in Target variable

Next Feature engineering was done and created several new variables 'loan_to_income', 'bad_state', 'avl_lines' and 'int_paid'.

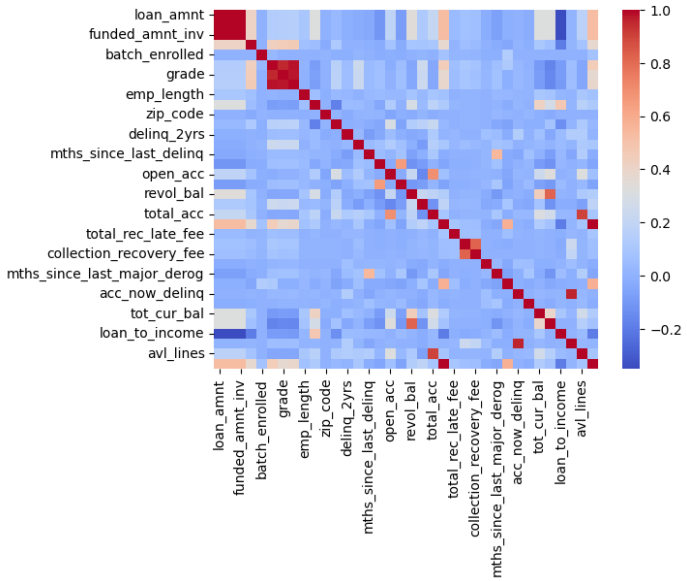
'loan_to_income' variable explain How big the loan with respect to his earnings, annual income to loan amount ratio. 'bad_state' column gives a magnitude of how much the repayment has gone off course in terms of ratios. 'avl_lines' variable explain total number of available/unused 'credit lines'. 'int_paid' variable explain how much interest paid so far.

Since variables in the dataset had different scales, scaling was necessary. Therefore, using the StandardScaler() in sklearn.preprocessing all non-encoded variables scaled.

In the final phase model building was done and initially scaled dataset divided in to 80% training and 20% testing.

Also, principal component analysis also carried out but when fitted data with the PCA didn't produce much good result. Therefore, initial scaled dataset used for model building.

Figure 3: correlation plot



Since target variable is not balanced instead of removing records, what done was in the optimization function high weightage given to the class with small number of records.

This was done by setting 'class_weight' to 'balanced' in logistic regression and Decision tree model. In XGBoost 'scale_pos_weight' set to 20 since the ratio of the two classes equal to roughly 20.

To find the optimum parameters, sklearn.model_selection GridSearchCV used.

Below table represent the summarization of parameters used in the models.

Model	Parameters
Logistic regression	class_weight = 'balanced', max_iter=1000, C= 0.0015, penalty ='l2'
Decision trees	max_depth=5,class_weight = 'balanced',min_samples_split =2,min_samples_leaf=2
XGBoost	n_estimators=100, learning_rate=0.1, scale_pos_weight=20

Model	Accuracy_train	Accuracy_test
Logistic regression	0.7321	0.7338
Decision trees	0.5749	0.5749
XGBoost	0.6131	0.6064

Model	Recall_test	Precession _test	F1-measure_test
Logistic regression	0.7095	0.3926	0.5055
Decision trees	0.7880	0.2820	0.4154
XGBoost	0.9801	0.3253	0.4885

Also, all the models trained to get a higher recall score since identifying a customer who is going to default is more important than identifying a customer who is not going to default.

Conclusion

From the above results clearly see that XGBoost model had higher recall score than other two models. But accuracy was near 60%. Logistic regression on the other hand had accuracy and recall both in 70% level. In terms of recall we can use XGBoost model but logistic regression also provided accuracy and recall both in an acceptable level.

References

[1] Loan Default Classification Dataset

[2] Jijo, Bahzad Taha, and Adnan Mohsin Abdulazeez. "Classification based on decision tree algorithm for machine learning." evaluation 6 (2021): 7.

Appendices

[i] Python Working