Loan Default Prediction with LightGBM and XGBoost

Sandeep Sinha[1]
[1]Technical University of Munich (TUM), Arcisstr. 21, 80333 Munich, Germany
sandeep.sinha@tum.de
March 21, 2021

Abstract — Loan default prediction is one of the critical problem faced by the financial institutions such as banks, loan providers etc. and has significant impact on their profit.Although many traditional methods exist for mining information about a loan application, most of these methods seem to be under performing as there have been reported increases in the number of bad loans. In this article, I use Light Gradient Boosting Machine(LightGBM) and eXtreme Gradient Boosting(XG-Boost) algorithms to predict loan default. The prediction is based on historical data from Small Business Administration (SBA), a US government body which provides loan supports to entrepreneurs and small businesses. The data consists of historical information of loan applications over certain period of time. In the article important statistical evaluation metrics such as Accuracy, Recall, precision, F1-Score and ROC area of the analysis are presented. This article provides an effective basis for loan credit approval in order to identify risky stakeholders from a large number of loan applications using predictive modelling using the algorithms.

1 Introduction

The US Small Business Administration (SBA) was founded in 1953 on the principle of promoting and assisting small enterprises in the U.S. credit market. Small businesses play a significant role in the economy by creating jobs and fostering innovation. One way SBA assists these small business enterprises is through a loan guarantee program which is designed to encourage banks to grant loans to small businesses. SBA acts much like an insurance provider to reduce the risk for a bank by taking on some of the risk through guaranteeing a portion of the loan. In the case that a loan goes into default, SBA then covers the amount they guaranteed.The rate of default on these loans has been a source of controversy for decades. Critics are not in favour of a government body interfering in the loan process .Supporters of SBA guaranteed loans argue that the social benefits of job creation by those small businesses receiving government guaranteed loans far outweigh the costs incurred from defaulted loans.

Banks are faced with dilemma of approval of a loan since SBA guarantees for the partial amount of the loan, hence there are chances of losses to bank due to defaulting on a loan by the small businesses. One way could be making informed decisions based on analysing the historical data. [1].

The increase in the application for loans plus the rapidly growing competition means financial institutions must build effective models that can capture the information in the available data, and create robust predictive models that can help minimize the chances of bad credit. Through numerous modern predictive modeling, financial institutions can get insights into applicant's behavior, consumption patterns, default predictors and characteristics. Numerous studies have been conducted in order to identify the important factors that can affect the loan repayment, these studies are important as they help to maximize profit for financial institutions. In addition to identifying factors that may influence loan default, there is also a need to build robust and effective machine learning models that can help capture important patterns in credit data. The choice of model is of great importance as the chosen model plays a crucial role in determining accuracy, precision and efficiency of a prediction system. Numerous models have been used for loan default prediction and although there is no one optimal model, some models definitely do better than others. [2]

Li Ying (2018) [3] did a comparative study of three models (Random Forest, Logistic Regression and Support Vector Machines) on bank credit data and found out that Random Forest does better at the task. Kumar et al. (2018) [4] used a Neural Network to predict loan default on data from a lending club bank and inferred that a Neural Network performs better than most traditional models.

In this article, I use two gradient boosting algorithms LightGBM and XGBoost to study and analyze bank loan dataset and suggest some of the important factors/variables that may influence loan repayment. Evaluation statistics (Accuracy, Precision, Recall, Confusion Matrix, f1-Score and ROC

area) of the models are presented. The remainder of the article is organized as follows: section 2 provides a background and overview of Gradient Boosting, LightGBM and XGBoost. Section 3 explains the data collection, pre-processing steps and some basic description of the data. In Section 4 results are discussed and section 5 concludes.

2 Background

In this section the machine learning algorithms used for forecasting the loan default are discussed in context of supervised learning.In Supervised learning setting, a predictor/learner/estimator is presented with input-output pairs (x1, y1), (x1, y1), . . . , (xn, yn) for some function y = f(x). In this analysis, the supervised learning problem is posed as a classification problem since the target/output is composed of discrete binary variables (good/bad credit). There exist many machine learning algorithms such as Logistic Regression, Neural Networks, Vector Machines, Decision Trees etc. that can be used for this classification task but I use two techniques of ensembling called Gradient Boosting, LightGBM and XGBoost as these have been shown to outperform many traditional algorithms in structured data like the data from SBA.[1]

2.1 Overview of Boosting

Boosting is a popular ensemble technique evolved for reducing bias and variance. It is based on the question posed "Can a set of weak learners create a single strong learner?" A weak learner is defined to be a classifier that is only slightly correlated with the true classification (it can label examples better than random guessing). In contrast, a strong learner is a classifier that is arbitrarily well-correlated with the true classification.[5]

The concept of boosting is to correct the mistakes made by earlier learners and improving on those areas i.e. to construct new base learners that are more correlated to the negative gradient of the objective function. Boosting can also be seen as a kind of stage wise “additive modeling” in that it is an additive combination of a simple base estimator.Gradient Boosting is a type of boosting where the objective is treated as an optimization problem and training is done using weight updates by gradient descent.[6]

In the case of gradient boosted decision trees, successive models are found by applying gradient descent in the direction of the average gradient, calculated with respect to the error residuals of the loss function, of the leaf nodes of previous models. For example, assume that you have a custom base-learner h(x, θ) (such as decision tree), and a loss function $\psi(y, f(x))$; it is challenging to estimate the parameters directly, and thus, an iterative model is suggested such that at each iteration. The model will be updated and a new base-learner function h(x, θ_t) is selected, where the increment is guided by:

$$g_t(x) = E_y \left[\frac{\partial \psi(y, f(x))}{\partial f(x)} | x \right]_{f(x)=\tilde{f}^{t-1}(x)}$$

This allows the substitution of the hard optimization problem with the usual least-squares optimization problem:

$$(\rho_t, \theta_t) = \arg min_{\rho,\theta} \sum_{i=1}^{N} [-g_t(x_i) + \rho\, h(x_i, \theta)]^2$$

Algorithm 1 summarizes the Friedman algorithm.

Algorithm 1 Gradient Boost

1- Let $\hat{f}_0$ be a constant
2- For i= 1 to M
   a. Compute $g_i(x)$ *using eq()*
   b. Train the function $h(x, \theta_i)$
   c. *Find* ρ_i *using eq()*
   d. *Update the function*
   $$\hat{f}_i = \hat{f}_{i-1} + \rho_i h(x, \theta_i)$$
3- End

The algorithm starts with a single leaf, and then the learning rate is optimized for each node and each record.[7]-[8]

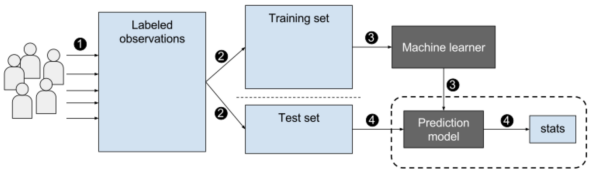

Figure 1 Steps in Gradient Boosting.[9]

Figure 1 visualizes the steps associated to working with a Gradient Boosting algorithm.

- Fit the model
- Tune the model’s parameters and Hyperparameters
- Make predictions
- Interpret the results

2.2 LightGBM

To reduce the implementation time, a team from Microsoft developed the LightGBM in 2017. LightGBM is an open-source framework for gradient boosted machines.The main difference is that the decision trees in LightGBM are grown leaf-wise, instead of checking all of the previous leaves for each new leaf. All the attributes are sorted and grouped as bins. This implementation is called histogram implementation. LightGBM has several advantages such as better accuracy, faster training speed, and is capable of large-scale handling data and is GPU learning supported.[10] By default LightGBM will train a Gradient Boosted Decision Tree (GBDT), but it also supports random forests, Dropouts meet Multiple Additive Regression Trees (DART), and Gradient Based One-Side Sampling (Goss).

2.3 XGBoost

XGBoost-Extreme Gradient Boosting is a scalable and highly efficient boosting system. It has been shown to achieve state-of-the-art results on many machine learning tasks. In XGBoost algorithm unlike the traditional gradient boosting, the process of adding weak learners does not happen sequentially; it approaches this phase in parallel using a multi threaded pattern, thereby resulting in proper utilization of hardware resources leading to greater speed and efficiency [11]. Some important features that make XGBoost more efficient than traditional boosting algorithms are:

- Sparse aware implementation.
- Weighted quantile sketch for approximate tree learning.
- Cache-aware access.
- Blocks for out-of-core computation.

3 Data

The article uses real data set provided by SBA. "National SBA" dataset (named SBAnational.csv) from the U.S. SBA which includes historical data from 1987 through 2014 (899,164 observations) and for which the outcome (paid in full or charged off/default) is known.[1]

Table 1 Description of 27 variables in the dataset.

Variable name	Data type	Description of variable
LoanNr_ChkDgt	Text	Identifier – Primary key
Name	Text	Borrower name
City	Text	Borrower city
State	Text	Borrower state
Zip	Text	Borrower zip code
Bank	Text	Bank name
BankState	Text	Bank state
NAICS	Text	North American industry classification system code
ApprovalDate	Date/Time	Date SBA commitment issued
ApprovalFY	Text	Fiscal year of commitment
Term	Number	Loan term in months
NoEmp	Number	Number of business employees
NewExist	Text	1 = Existing business, 2 = New business
CreateJob	Number	Number of jobs created
RetainedJob	Number	Number of jobs retained
FranchiseCode	Text	Franchise code, (00000 or 00001) = No franchise
UrbanRural	Text	1 = Urban, 2 = rural, 0 = undefined
RevLineCr	Text	Revolving line of credit: Y = Yes, N = No
LowDoc	Text	LowDoc Loan Program: Y = Yes, N = No
ChgOffDate	Date/Time	The date when a loan is declared to be in default
DisbursementDate	Date/Time	Disbursement date
DisbursementGross	Currency	Amount disbursed
BalanceGross	Currency	Gross amount outstanding
MIS_Status	Text	Loan status charged off = CHGOFF, Paid in full = PIF
ChgOffPrinGr	Currency	Charged-off amount
GrAppv	Currency	Gross amount of loan approved by bank
SBA_Appv	Currency	SBA's guaranteed amount of approved loan

NAICS (North American Industry Classification System): This is a 2- through 6-digit hierarchical classification system used by Federal statistical agencies in classifying business establishments for the collection, analysis, and presentation of statistical data describing the U.S. economy. The first two digits of the NAICS classification represent the economic sector.

The data is then processed and prepared for the analysis.The null values from the columns are dropped before starting the analysis. The columns which will be unnecessary for analysis have to be dropped including the non numerical information. Since these algorithms work well with categorical output on numerical input or through encoding the data to be compatible with the models. Classification model is needed to identify if a loan disbursed will default or not. LightGBM and XGboost will be used as ML models to work on the data. They can not handle categorical data hence, encoding is needed to be used on State and Industry (NAICS) columns hence an encoding is done on those columns based on the default rates.

Two new columns were created in order to provide more depth to the analysis. For example, SBA loans with over a 20 year Term have to be backed

Table 2 Description of the first two digits of NAICS.

Sector	Description
11	Agriculture, forestry, fishing and hunting
21	Mining, quarrying, and oil and gas extraction
22	Utilities
23	Construction
31–33	Manufacturing
42	Wholesale trade
44–45	Retail trade
48–49	Transportation and warehousing
51	Information
52	Finance and insurance
53	Real estate and rental and leasing
54	Professional, scientific, and technical services
55	Management of companies and enterprises
56	Administrative and support and waste management and remediation services
61	Educational services
62	Health care and social assistance
71	Arts, entertainment, and recreation
72	Accommodation and food services
81	Other services (except public administration)
92	Public administration

by real estate. Thereby creating a binary variable for RealEstate.

- RealEstate - 1 if loan is backed by real estate, 0 if not
- Recession - 1 if Fiscal Year of Approval is during a Recession, 0 if not

Additionally, Paid column is created by replacing MIS-STATUS column with a numerical column by making column for Paid in Full (1=Paid in full, 0 = no)

After the preparation and cleaning of data, it was divided into training and testing sets for training and testing the performance of the model. Training data consisted of 80% of total data.

4 Results and Discussions

Based on the cleaned and pre-processed data, I use the LightGBM and XGBoost classifier algorithm implemented in python programming language by feeding the training data. The classifiers were trained using numerical features.For evaluation,I use five metrics; Accuracy, F1-Score, Recall, Precision and ROC area.

4.1 Confusion Matrix

The confusion matrix is an important 2 dimensional matrix that contains information about the actual classes and the predicted classes of a classifier. Figure 2 shows the calculated confusion matrix of

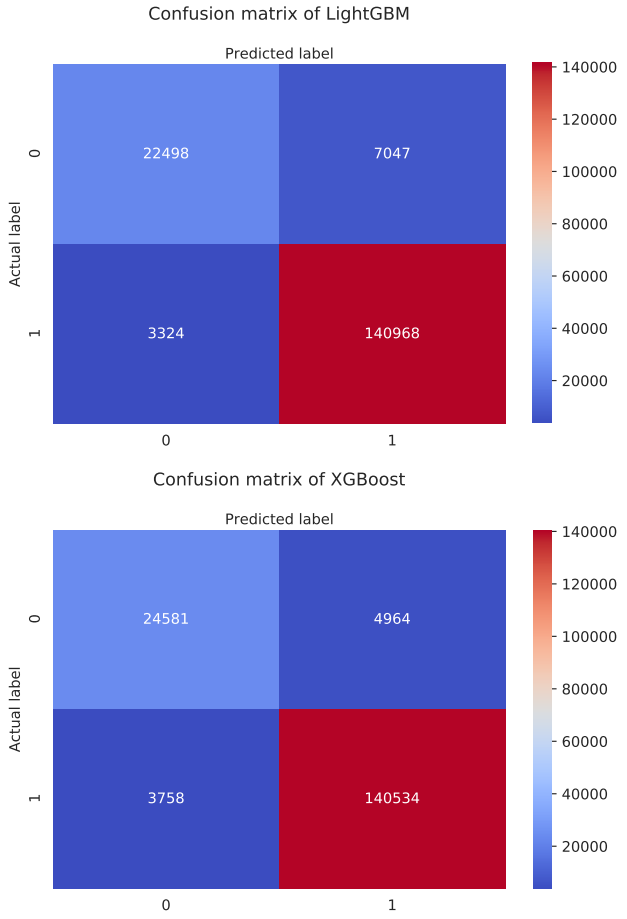

Figure 2 Confusion Matrix of the Models

LightGBM and XGBoost on the data. In the loan data column Paid used in this article, the number 0 represents the loan default category, and 1 represents the normal category depicted in the newly created Paid column.

4.2 Accuracy

Accuracy measures the proportion of correctly classified predictions; it is defined by the formula:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Where TP is the number of true positives, TN is the number of true negatives, FP is the number of false positives and FN is the number of false negatives.

4.3 Recall

Recall measures the ability of the classifier to predict correctly instances of a certain class; it is also called the TPR (true positive rate):

$$Recall = \frac{TP}{TP + FN}$$

4.4 Precision

Precision measures the proportion of predictions made by the classifier as positive that are actually positive:

$$Precision = \frac{TP}{TP + FP}$$

4.5 Recall

F1-score is the harmonic average of precision and recall:

$$F1_score = \frac{2 * Precision * Recall}{Precision + Recall}$$

Table 3 Performance metrics of the models

Metrics(Score%)	LightGBM	XGBoost
Accuracy	94.03	94.98
Precision	95.24	96.59
Recall	97.7	97.4
F1 Score	96.45	96.99

Table 3 summarizes the performance metrics of the both models.

4.6 ROC

ROC (Receiver Operating Characteristics curve) is a visualization technique for showing a classifier's performance. It represents the sensitivity and specificity of the classifier.The ROC curve is a two-dimensional curve with the FPR (false positive rate) as the X-axis and the TPR (true positive rate) as the Y-axis. The ranges of the ROC curve runs from (0, 0) to (1, 1). To compare models, area under the ROC curve (AUC) is calculated. The larger the AUC, the better the model.

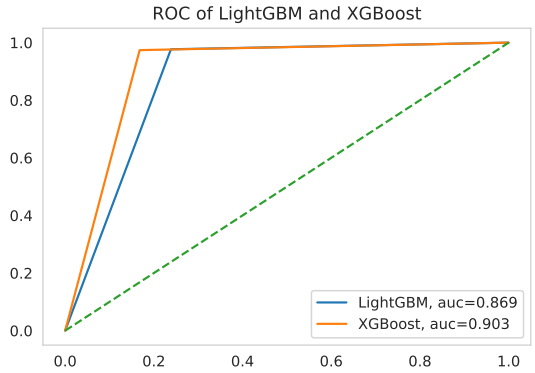

Figure 3 ROC area curve of LightGBM and XGBoost

4.7 Feature Importance

One of the important results of this analysis is to determine the important features that help the classifier to correctly predict loan defaults. This helps in doing business intelligence and decision making. Figure 4 shows the top 5 most important features.The Term of the loan and Approval financial year appear to be the two most important features from the result of the analysis by both the models.

5 Conclusions

In this article, I have successfully used the LightGBM and XGBoost for SBA loan default prediction. The task was to predict if a loan applicant will default in loan payment or not. The analysis was implemented in the python programming language, and performance metrics like accuracy, recall, precision, f1-score were calculated. They seem to be quite high and also confusion matrix shows higher amount for approved (or 1) loans for both the models. Hence a deeper analysis while splitting the data for training and testing the models should be done to improve on it. From the analysis, it was also found that the most important features used by the models for predicting if a customer would default in payment or not depends highly on the Term of the loan and ApprovalFY. From observing the ROC curve, XGBoost seem to be doing better because has slightly higher AUC. Although LightGBM was much faster compared to XGBoost while training the models. This article provides an effective basis for loan credit approval in order to identify risky stakeholders from a large number of loan applicants using predictive modeling.

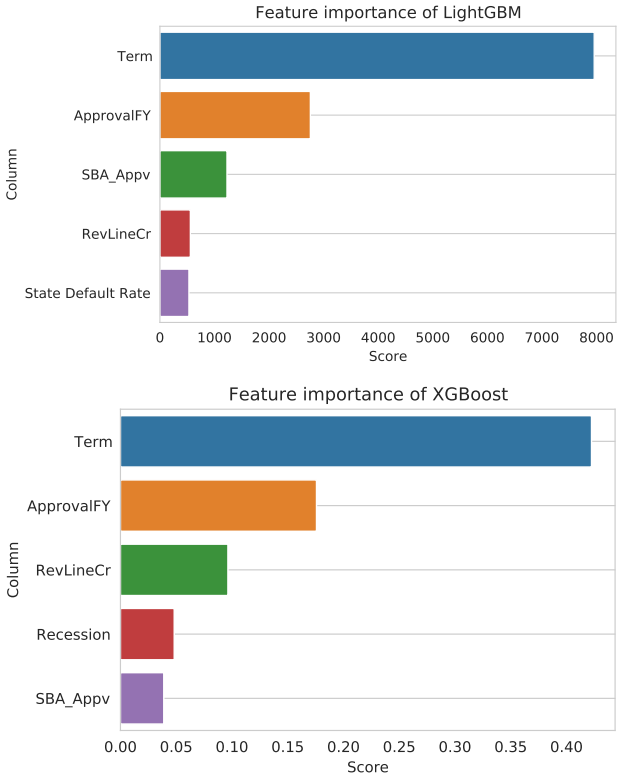

Figure 4 Feature importance of the Models

Acknowledgement

This article is written to fulfill the examination requirement for the course Python for Engineering Data Analysis - from Machine Learning to Visualization. I would like to thank Felix Mayr for help and guidance throughout the course even in a virtual set up.The project uses python on Pycharm IDE which is openly available software distributed by jet brains. The analysis was done using Pycharm pro licence freely accessible to students.

References

[1] M. Li, A. Mickel, and S. Taylor, ""should this loan be approved or denied?": A large dataset with class assignment guidelines," *Journal of Statistics Education*, vol. 26, no. 1, pp. 55–66, 2018.

[2] R. Odegua, "Predicting bank loan default with extreme gradient boosting," *arXiv preprint arXiv:2002.02011*, 2020.

[3] L. Ying, "Research on bank credit default prediction based on data mining algorithm," *The International Journal of Social Sciences and Humanities Invention*, vol. 5, no. 6, pp. 4820–4823, 2018.

[4] M. Kumar, V. Goel, T. Jain, S. Singhal, and L. M. Goel, "Neural network approach to loan default prediction," *International Research Journal of Engineering and Technology (IRJET)*, vol. 5, no. 4, pp. 4–7, 2018.

[5] Y. Freund, R. E. Schapire *et al.*, "Experiments with a new boosting algorithm," in *icml*, vol. 96. Citeseer, 1996, pp. 148–156.

[6] J. H. Friedman, "1999 reitz lecture," *Ann. Stat.*, vol. 29, no. 5, pp. 1189–1232, 2001.

[7] E. Al Daoud and H. Turabieh, "New empirical nonparametric kernels for support vector machine classification," *Applied Soft Computing*, vol. 13, no. 4, pp. 1759–1765, 2013.

[8] J. H. Friedman, "Greedy function approximation: a gradient boosting machine," *Annals of statistics*, pp. 1189–1232, 2001.

[9] "Gradient boosting classifiers in python with scikit-learn," https://stackabuse.com/gradient-boosting-classifiers-in-python-with-scikit-learn/, accessed: 2021-03-20.

[10] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu, "Lightgbm: A highly efficient gradient boosting decision tree," *Advances in neural information processing systems*, vol. 30, pp. 3146–3154, 2017.

[11] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, 2016, pp. 785–794.