Predicting Loan Defaulting using Machine Learning

Ashhar Aziz
ashhar21137@iiitd.ac.in

Lakshya Goel
lakshya21469@iiitd.ac.in

Sanmay Sood
sanmay21095@iiitd@ac.in

Srimant Mohanty
srimant21207@iiitd.ac.in

Abstract

This project aims to predict the likelihood of people defaulting on loans. In this project, we will implement various algorithms on our data such as Random Forest, Logistic Regression, Decision Trees, Support Vector Machines and Multi-Layer Perceptron Classifiers .

1. Motivation

The rapid growth of India's banking sector has increased loan applicants, amplifying the challenge of accurately predicting loan defaults. Recent data from the Reserve Bank of India (RBI) reveals a concerning trend. Loan write-offs by banks surged to an alarming Rs 209,144 crore in the fiscal year ending March 2023, which is a 19.53 % increase from FY22. Although existing strategies encompass collateral evaluation, income validation, and scrutiny of borrowing records, a transformative approach is imperative. Through harnessing the potential of machine learning (ML), we aim to formulate an advanced predictive model to bolster the precision of loan default prediction. When the company receives a loan application, the company has to make a decision for loan approval based on the applicant's profile. Two types of risks are associated with the bank's decision:

- If the applicant is likely to repay the loan, then not approving the loan results in a loss of business to the company.
- If the applicant is not likely to repay the loan, i.e. he/she is likely to default, then approving the loan may lead to a financial loss for the company

This innovation holds the potential to equip Indian banks with proactive measures against potential defaults, mitigating financial losses and fortifying the lending landscape.

2. Literature Review

Loan default prediction is a critical issue for financial institutions, as it directly impacts their profitability and risk management. In recent years, the application of machine learning algorithms to predict loan defaults has gained significant attention due to their potential to enhance predictive accuracy and risk assessment. We have found several papers tackling a similar problem statement. Through these works, we have gained a better understanding of implementation of machine learning techniques in the specific setting. We also intend to use some of the models used in these studies as a benchmark for our models. Some relevant studies and literature on loan default prediction:

- **"Predicting Credit Card Defaults Using Machine Learning Techniques" by Wei et al. (2009):** This study compared different types of ML algorithms, like neural networks and decision tree, for predicting credit card defaults. The authors found that ensemble methods, such as random forests, were the most effective for this task.
- **"Credit Scoring and Loan Default" by Thomas et al. (2016):** This book provides an overview of credit scoring and loan default prediction. It covers the traditional statistical methods, such as logistic regression and discriminant analysis, as well as more recent machine learning techniques.
- **"Loan Default Prediction Using Bayesian Networks: A Comparative Study" by Azevedo et al. (2019):** This study compared Bayesian networks with other ML techniques, like SVM and decision-tree, for predicting loan defaults. The authors found that Bayesian networks outperformed the other methods.
- **"A study on predicting loan default based on the random forest algorithm" by Lin Zhu et al. (2019):** This paper used data from the Lending Club company and applied SMOTE to cope with imbalanced classes. Their experiments concluded that Random Forest algorithm outperforms logistic regression, decision tree and other machine learning algorithms in predicting default samples.

- **"Credit Risk Assessment Using Machine Learning Techniques: A Review" by Sathyadevan et al. (2021):** This review article covers various machine learning techniques for credit risk assessment, including loan default prediction. The authors discuss the strengths and weaknesses of different methods and provide recommendations for future research.

- **"Loan Default Prediction Using Machine Learning Techniques" by E. Praynlin et al. (2023):** This study proposed a K-NN model to predict potential loan defaulters and tried to compare and evaluate the performance of this model against the decision tree model. It also found age, income, loan length, and loan amount to be the most crucial variables for prediction.

Overall, our survey found that ensemble approaches like the random forest approach tend to perform better for default prediction. Moreover, key indicators of default include variable like income, loan size, and loan length.

3. Dataset : Lending Club Loan Dataset

The data contains complete loan data for all loans issued through the 2007-2015 by the Lending Club company. LendingClub is a US peer-to-peer lending company, headquartered in San Francisco, California. This dataset offers a multifaceted view of loans, including critical attributes such as loan amount, interest rates, employee title, employment years, loan purpose, and credit history, among others.

3.1. Dataset Overview

The dataset comprises an extensive 2,260,668 rows and 145 columns. Our analysis initially contemplates this comprehensive dataset, though subsequent preprocessing reduced its dimensions. Among these 145 features, a subset proved pivotal in our analytical journey. Thus, the huge dataset was later reduced using various pre-processing techniques. Some of the key features and variables which are particularly important and relevant for our models are described in Fig.1.

Feature	Description
loan_amnt	The listed amount of the loan applied for by the borrower. If at some point in time, the credit department reduces the loan amount, then it will be reflected in this value.
funded_amnt	The total amount committed to that loan at that point in time.
funded_amnt_inv	The total amount committed by investors for that loan at that point in time.
term	The number of payments on the loan. Values are in months and can be either 36 or 60.
int_rate	Interest Rate on the loan
installment	The monthly payment owed by the borrower if the loan originates.
grade	Lending Club assigned loan grade
sub_grade	Lending Club assigned loan subgrade
emp_length	Employment length in years. Possible values are between 0 and 10 where 0 means less than one year and 10 means ten or more years.
annual_income	The self-reported annual income provided by the borrower during registration.
dti	A ratio calculated using the borrower's total monthly debt payments on the total debt obligations, excluding mortgage and the requested LC loan, divided by the borrower's self-reported monthly income.
delinq_2y	The number of 30+ days past-due incidences of delinquency in the borrower's credit file for the past 2 years
open_acc	The number of open credit lines in the borrower's credit file.
pub_rec	Number of derogatory public records
revol_bal	Total credit revolving balance
revol_util	Revolving line utilization rate, or the amount of credit the borrower is using relative to all available revolving credit.
total_acc	The total number of credit lines currently in the borrower's credit file
recoveries	post charge off gross recovery
acc_now_delinq	The number of accounts on which the borrower is now delinquent.
tot_cur_bal	Total current balance of all accounts
total_bal_il	Total current balance of all installment accounts
all_util	Balance to credit limit on all trades
total_cu_tl	Number of finance trades
avg_cur_bal	Average current balance of all accounts
bc_util	Ratio of total current balance to high credit/credit limit for all bankcard accounts.
delinq_amnt	The past-due amount owed for the accounts on which the borrower is now delinquent.
num_sats	Number of satisfactory accounts
pub_rec_bankruptcies	Number of public record bankruptcies
hardship_flag	Flags whether or not the borrower is on a hardship plan
State	US State to which the borrower belongs

Figure 1. Features

3.2. Data Types and Descriptive Statistics

The dataset exhibits two primary data types: float (representing numerical values) and object (representing categorical values). We conducted a comprehensive statistical analysis, encompassing mean, median, mode, value counts, and percentile values across all variables. This procedure provided us with a foundational understanding of the dataset's characteristics.
Given our objective of classifying loans as either 'good' or 'bad,' we scrutinized the distribution of loan statuses. Rows with a 'current' loan status were excluded from consideration due to their inconclusive nature. This helped us conclude that since the goal of this project is to classify potential loan data into good loans and bad loans, rows with loan status as 'current' should be dropped/not considered.

3.3. Categorical Feature Insights

To gain insights into categorical features, we employed pie charts to visualize their distributions. This facilitated the comprehension of variable proportions within each categorical feature.

3.3.1 Feature and Target Variable Relationship

We probed the relationships between different features and their alignment with the target variable (loan status). This analysis aimed to identify features that could potentially enhance our modeling efforts. Notable observations included the propensity to offer loans to individuals with stable income sources, such as teachers, managers, and business owners. Loan amounts predominantly fell within the range of $5,000 to $25,000. Loans featuring lower interest rates exhibited higher repayment rates, while those with interest rates exceeding 20% experienced a substantially elevated default rate. Common loan purposes encompassed debt consolidation, credit card refinancing, and home improvement.

Correlation heatmaps were employed to investigate relationships among numerical features. Notably, the high correlation (correlation coefficient of 1) between 'loan amount,' 'funded amount,' and 'funded amount inv' indicated their near-identical nature, allowing for the removal of redundant columns.

Boxplots elucidated key statistics such as medians and averages for annual incomes, loan amounts, installments, and interest rates. Additionally, they revealed the presence of outliers in these features. These boxplots were juxtaposed with the target variable, highlighting that higher annual incomes corresponded to a greater ability to repay loans, resulting in higher loan approval rates.

This comprehensive exploratory analysis lays the foundation for subsequent modeling efforts, offering invaluable

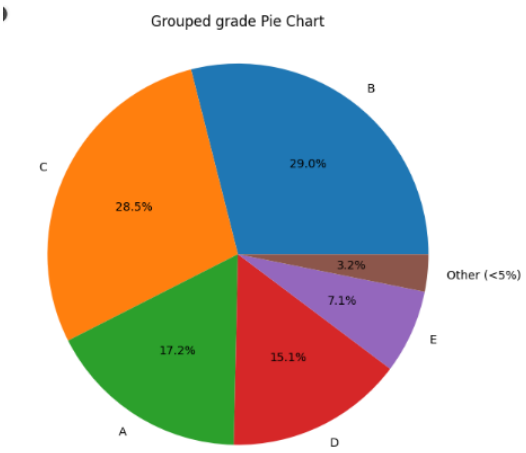

Figure 2. Loan Grade

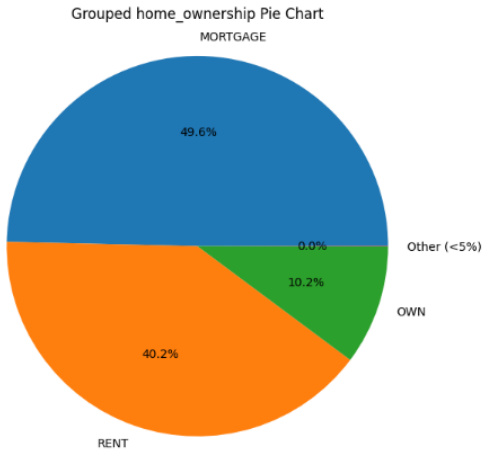

Figure 3. Home Ownership

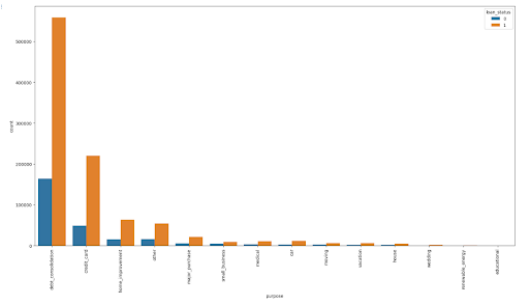

Figure 4. Loan Purpose Distribution

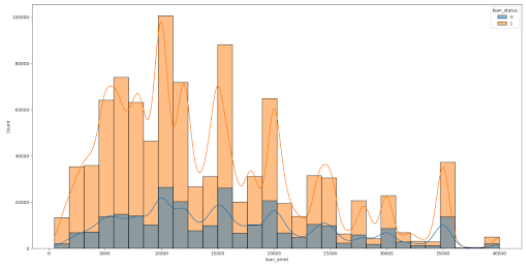

Figure 5. Loan Amount Distribution

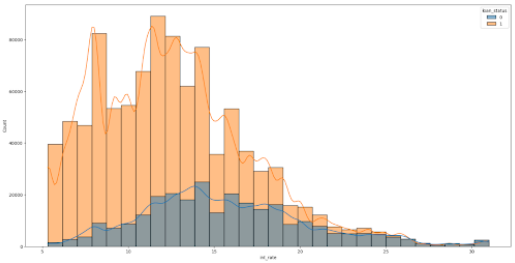

Figure 6. Interest Rate Distribution

insights into the dataset's nuances and key features influencing loan outcomes.

4. Data Preprocessing

Data Preprocessing is the first step in the machine learning pipeline. It involves cleaning, transforming, and organizing raw data into a format suitable for analysis or model training. The following are the typical steps in data preprocessing that were used.

4.1. Handling Missing Data

4.1.1 Identification

All samples tested for missing target variables. None found missing target value. Dropped columns having more than 50 percent missing values. Post removal, 101 out of 145 columns remain. Overall, missing values formed 9.87 percent of total values. All samples tested for missing any categorical data. All samples found missing one or more were dropped. Post removal, 2,070,597 samples out of 2,260,668 remain.

4.1.2 Imputation

Common imputation strategies include replacing by mean, median, mode, or a more complex method like regression or interpolation. To ensure if the mean would be a good estimate, the distribution of data was studied via qq plots for all numerical features. As the data didn't exhibit a normal distribution, 'median' was chosen for imputation. Total missing values post removal and imputation - 0

4.2. Data Cleaning

4.2.1 Duplicate Removal

The data was checked for duplicates. Columns 'title' and 'purpose' had the same categorical data, of which data under 'purpose' was better presented. Hence, 'title' was dropped.

4.2.2 Data Relevance

The target variable 'loan_status' contained several categorical values. But, as per the problem statement, only two classes of data were relevant. These were 'Charged Off' and 'Fully Paid'. The loans that could not be paid back and were 'Charged Off' were considered as positive samples, and loans that were paid back 'Fully Paid' were considered as negative samples, represented by 1 and 0 respectively. Samples related to all other classes were dropped

4.3. Data Transformation

4.3.1 Encoding categorical variables

Categorical variables were converted into numerical representations, using techniques like one-hot encoding, label encoding and manual mapping.The distribution of categorical variables was studied using piecharts and the most efficient way of encoding was selected. The 22 categorical columns were manually divided into the following categories-

- **Label Encoding:** term, grade, emp_length, subgrade
- **One-hot Encoding:** home_ownership, verification status, purpose, application_type, disbursement_method, initial_list_status, addr_state
- **Boolean:** debt_settlement_flag, pymnt_plan, hardship_flag

4.3.2 Dropped Features

1. **zip_code :** The remaining features from the dataset were dropped due to various reasons. For example, in the feature 'zip_code'. the last two digits were hidden (12XX), hence it didn't convey any specific information.
2. **title :** It was duplicate of 'purpose'
3. **emp_title :** Due to the extremely high number of unique values, it was realistically not possible to encode it manually or through one-hot encoding. Label encoding could not be done since different professions shared no progressive relation with each other.

   Post-encoding all categorical variables, a total of 174 features remained.
4. **Others :** A lot of the remaining features, such as 'total_rec_late_fee ', 'open_il_12m', 'open_acc_6m', 'next_pymnt_d' etc. were also dropped because they either contained information that would only be available after the loan was given or they were similar to the other selected features.

5. Modelling and Results

After the preprocessing steps, we did a 70-30 traintest split of the data. Feature selection was done on the basis literature review of previous works, relevancy, and information available about the features. We attempted a number of different (so far) machine learning models to address our task. We also standardize the remaining non-encoded columns. We calculated the accuracy, F1 score, precision, and recall after training the models.

5.1. Logistic Regression

We performed logistic regression on our dataset and tested the performance when using L1 and L2 regularization.

5.1.1 With L2 Regularization :

We used the Logistic Regression implementation of the sklearn library with the solver set as 'NewtonCholesky', which is recommended when the number of samples is much greater than number of features. The accuracy on the train set was 93.103% and on test set it was 93.041%.

The overall precision, recall and ROC-AUC score for the test-set are as follows :

Precision	Recall	F1-Score	ROC-AUC
99.877	65.128	78.843	90.031

Class-specific scores are as follows :

Class	Precision	Recall	F1-Score	Support
0	92	100	96	251949
1	10	65	79	62624

5.1.2 With L1 Regularization :

To use L1 regularization, we chose the solver 'saga', as it supports the use L1 regularization unlike other solvers. The accuracy on the train set was 92.751% and on test set it was 92.687%.

The overall precision, recall and ROC-AUC score for the test-set are as follows :

Precision	Recall	F1-Score	ROC-AUC
99.888	63.339	77.522	89.782

Class-specific scores are as follows :

Class	Precision	Recall	F1-Score	Support
0	92	100	96	251949
1	100	63	78	62624

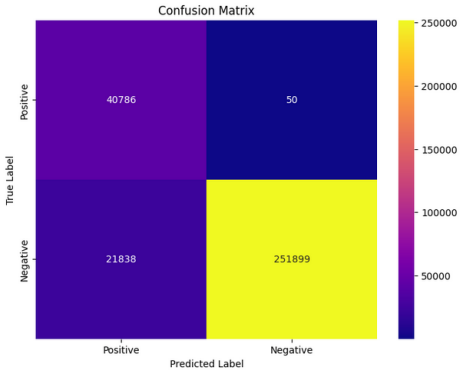

Figure 7. Logistic Regression with L2 Reg.

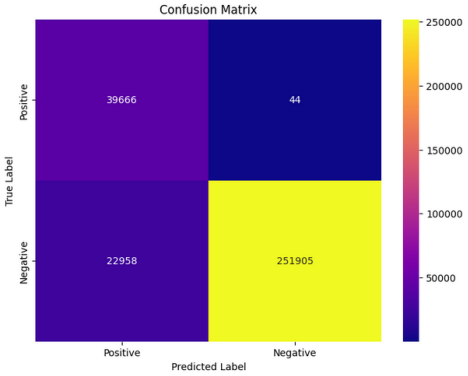

Figure 8. Logistic Regression with L1 Reg.

5.2. Decision Trees

We also tested the performance of the Decision Trees. After testing various values for the hyperparameters, we found the best results to when the max depth was set to 8. The number of leaves can out to be 125 and number of nodes to be 249. To select the best attribute at each node, we have used Gini Index which is the default criteria. The accuracy on train set was 93.678% and on test set it was 93.77%.
Overall precision, recall, F1-score and ROC-AUC :

Precision	Recall	F1-Score	ROC-AUC
99.981	68.254	81.126	91.667

Class-specific scores are as follows :

Class	Precision	Recall	F1-Score	Support
0	93	100	96	251949
1	100	68	81	62624

Feature importances of some features for Decision Tree :

Feature Name	Importance
recoveries	0.9659
sub_grade	0.0113
all_util	0.0068
annual_inc	0.0007
loan_amnt	0.0003
dti	0.0016

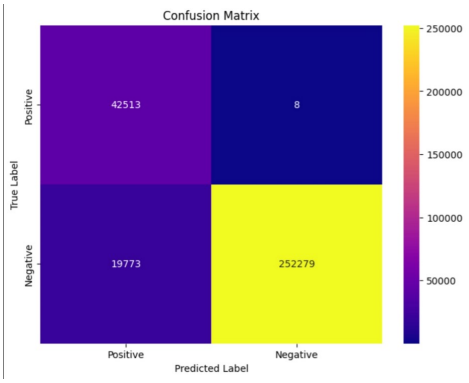

Figure 9. Confusion Matrix for Decision Tree

5.3. Random Forest

We also tested Random Forest on the dataset as in previous works it was shown to give the best performance for this task. Here, we set the number of trees as 150, max depth as 8, minimum samples per split as 2 and minimum samples per leaves as 1. Here, also we used Gini-index for attribute selection. The accuracy on train set was 93.676% and the accuracy on test set was 93.7127%. Overall precision, recall, F1-score and ROC-AUC :

Precision	Recall	F1-Score	ROC-AUC
100	68.246	81.126	91.128

Class-specific scores are as follows :

Class	Precision	Recall	F1-Score	Support
0	93	100	96	251949
1	100	68	81	62624

Feature importances of some features for Random Forest :

Feature Name	Importance
recoveries	0.8060
grade	0.0515
sub_grade	0.0422
all_util	0.0034
avg_cur_bal	0.0047
loan_amnt	0.0029
dti	0.0066

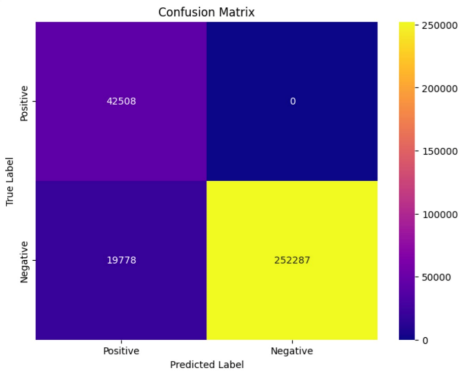

Figure 10. Confusion Matrix for Random Forest

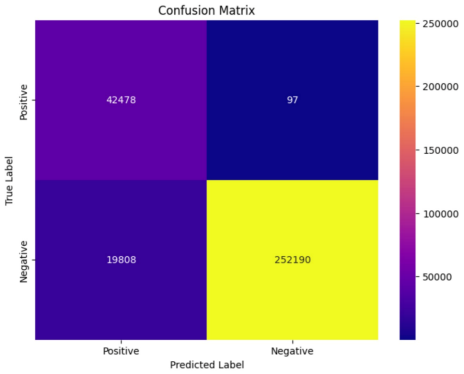

Figure 11. Confusion Matrix for XGboost

5.4. XGboost

XGboost is an implementation of gradient-boosted decision trees designed for speed and performance. It is useful for predictive modeling in regression and classification tasks, especially when dealing with tabular data. The accuracy on train set was 93.71% and on test set it was 93.67%.

Precision	Recall	F1-Score	ROC-AUC
99.772	68.198	81.017	92.461

Class-specific scores are as follows :

Class	Precision	Recall	F1-Score	Support
0	93	100	96	251949
1	100	68	81	62624

Feature importances of some features for XGboost :

Feature Name	Importance
recoveries	0.9335
grade	0.0085
sub_grade	0.00091
all_util	0.0014
term	0.0055
encoded_w	0.0010
encoded_WV	0.0008
encoded_MS	0.0007

XGboost gave importance to location (state) feature unlike other methods.

6. Conclusion

- **Best Precision:** Random Forest achieved the highest precision, making it the safest choice when minimizing false positives is crucial, ensuring minimal unnecessary loan rejections.
- **Best Recall:** Decision Tree and XGBoost had the highest recall, indicating their effectiveness in capturing most default cases. They are suitable when identifying potential loan defaults accurately is a priority.
- **Best Overall Performance:** XGBoost demonstrated the highest ROC-AUC score, showcasing the best overall discrimination ability. It is the top choice for achieving a balanced and robust performance in loan default prediction.

In summary, while Random Forest offers unmatched precision, Decision Tree and XGBoost provide a slightly better balance between precision and recall. XGBoost stands out with the highest ROC-AUC score, making it the optimal choice for accurate and reliable loan default predictions.

7. Future Work

We will try other boosting methods like AdaBoost and LightGBM Boost and try hyperparameter tuning for these and XGboost. Next, we will evaluate the performance of Support Vector Classification and ANNs on this task. We will also try incorporating various fairness measures in the classification process and test models on the same.

8. Contributions

- Ashhar Aziz : Feature Selection, Logistic Regression, XGboost, Report
- Lakshya Goel : Exploratory Data Analysis, Data Visualization, Data Preprocessing, Report
- Sanmay Sood : Literature Review, Decision Trees, Random Forest, XGboost
- Srimant Mohanty : Exploratory Data Analysis, Data Visualization, Data Preprocessing, Literature Review