Enhancing Credit Risk Assessment: Leveraging Machine Learning for Default Prediction in Lending Club Data

Subject: Risk Analysis

Lecturer: Nguyen Thi Lien

Trần Hải Nam

Student ID: 11219279

Class : DSEB 63

Enhancing Credit Risk Assessment: Leveraging Machine Learning for Default Prediction in Lending Club Data

Abstract:

This paper aims to explore the relationship between various demographic and financial variables and the payment behavior of consumer credit clients, particularly focusing on credit default risk. The objective of this study is to develop a predictive model capable of accurately assessing the likelihood of a customer defaulting on a loan, considering factors such as income, grade, and loan amount. This endeavor involves the collection and cleansing of data from Lending Club's publicly available datasets, followed by an in-depth exploration of the data to discern patterns. Subsequently, a machine learning algorithm will be meticulously selected and fine-tuned. The efficacy of the model will be assessed using performance metrics, including accuracy, precision, recall, f1-score and AUC. The findings from this research hold considerable implications for lending institutions, aiding them in making well-informed decisions regarding loan approvals and risk management.

1. Introduction

In recent years, the importance of risk management in credit has surged for both borrowers and lenders, particularly in developing nations. Consequently, financial institutions and banks have initiated a reevaluation of their lending policies. Credit lending activities encompass a multitude of functions, including the assessment of a customer's credit risk, decision-making regarding credit terms and limits, debt collection, monitoring of customer behavior, assumption of default risk, and financing of debtor investments (Summer and Wilson, 2000). This project focuses on the fourth function within credit lending activities. Herein, we employ machine learning techniques to construct a classification model capable of predicting whether a borrower is likely to default on a loan. These findings empower financial institutions to refine their lending strategies and mitigate credit default risks.

2. Theoretical Background

Banking portfolios, even in smaller banks, often possess substantial monetary value, rendering even minor enhancements significant. Consequently, the pursuit of improving model performance in credit default prediction is deemed invaluable. Thus, prior to embarking on model performance enhancement, it is imperative to review the models designated for classifying default customers.

2.1. Models

2.1.1. Logistic Regression (Traditional method)

Logistic Regression is a fundamental statistical technique used for binary classification tasks. Its application extends to fields like medicine, finance, and social sciences. The model estimates the probability that a given input belongs to one of two possible outcomes, usually denoted as 0 and 1. Logistic Regression models the relationship between the independent variables and the binary dependent variable using the logistic function, also known as the sigmoid function.

The core concept of Logistic Regression revolves around fitting a linear decision boundary to separate the classes in the feature space. It calculates the log-odds of the probability of the event occurring, which is then transformed into a probability value between 0 and 1 using the logistic function. The model parameters are estimated using maximum likelihood estimation, where the goal is to maximize the likelihood of observing the data given the model parameters.

Logistic Regression offers several advantages, making it a popular choice for binary classification tasks. Firstly, it is computationally inexpensive and requires minimal tuning of hyperparameters. This simplicity makes it easy to implement and interpret, even for those without a deep understanding of machine learning. Additionally, Logistic Regression provides probabilistic interpretations of predictions, which can be crucial in certain applications, such as medical diagnosis or risk assessment. Moreover, it can handle both numerical and categorical features, making it versatile for various types of data. Despite its simplicity, Logistic Regression often performs well on linearly separable datasets and serves as a baseline model for more complex algorithms.

The logistic function is of the form:

$$P(x) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x_1 + \cdots + \beta_k x_k)}}$$

2.1.2. Decision Tree

Decision Trees are non-parametric supervised learning models used for both classification and regression tasks. They are popular due to their simplicity and ability to handle both numerical and categorical data. Decision Trees recursively partition the feature space into regions that are as homogeneous as possible with respect to the target variable.

The concept behind Decision Trees involves partitioning the feature space based on the values of different features. At each node of the tree, the algorithm selects the feature and split point that maximize the information gain, Gini impurity, or entropy. This process continues recursively until a stopping criterion is met, such as reaching a maximum tree depth or minimum number of samples per leaf.

Decision Trees offer several advantages, including interpretability, ease of understanding, and robustness to outliers. They can capture complex nonlinear relationships in the data and are robust to noise. Decision Trees require minimal data preprocessing and are resistant to overfitting, especially with techniques like pruning and ensemble methods. Additionally, they provide feature importance scores, aiding in feature selection and understanding of the underlying data patterns.

2.1.3. Random Forest:

Random Forest is an ensemble learning method based on decision trees. It constructs multiple decision trees and aggregates their predictions through voting or averaging to improve generalization and robustness.

The concept behind Random Forest lies in the aggregation of multiple decision trees to reduce overfitting and improve prediction accuracy. Each tree in the ensemble is trained on a random subset of the training data and a random subset of features. During prediction, the output of each tree is combined through majority voting (for classification) or averaging (for regression).

Random Forest offers several advantages over individual decision trees. Firstly, it typically provides higher prediction accuracy due to the reduction of variance achieved through averaging multiple trees. Moreover, Random Forest is robust to noise and outliers and can handle high-dimensional data. It requires minimal hyperparameter tuning and provides estimates of feature importance, aiding in feature selection and understanding of the data.

2.1.4. XGBoost

XGBoost, short for Extreme Gradient Boosting, is a powerful gradient boosting algorithm known for its scalability and efficiency. It sequentially builds a strong ensemble of decision trees, optimizing a differentiable loss function at each iteration.

The concept behind XGBoost lies in the iterative construction of an ensemble of decision trees, where each tree corrects the errors of the previous ones. XGBoost employs a gradient boosting framework, where the objective is to minimize a differentiable loss function by adding weak learners (decision trees) to the ensemble. It incorporates regularization techniques such as shrinkage and tree pruning to prevent overfitting and improve generalization.

XGBoost offers several advantages over traditional gradient boosting algorithms. Firstly, it provides higher prediction accuracy and is more computationally efficient due to its optimization algorithms and parallelization techniques. XGBoost automatically handles missing values and supports parallel processing, making it efficient for training on multicore CPUs and distributed computing environments. Additionally, XGBoost provides a wide range of hyperparameters for fine-tuning and offers insights into feature importance, aiding in model interpretation and feature selection. Moreover, it has been widely adopted in machine learning competitions and real-world applications for its exceptional performance and robustness.

2.1.5. CatBoost

CatBoost is a gradient boosting algorithm specifically designed for handling categorical features. It automatically handles categorical variables without the need for pre-processing like one-hot encoding.

CatBoost incorporates innovative techniques for dealing with categorical variables within the gradient boosting framework. It optimizes the categorical features directly during training, avoiding the need for manual encoding or feature engineering. CatBoost utilizes an efficient algorithm for processing categorical features, including an ordered boosting scheme and a novel method for handling categorical labels.

CatBoost offers several advantages, particularly in scenarios with categorical data. By optimizing categorical features directly, CatBoost preserves the integrity of the data and captures valuable information from categorical variables. It automatically handles missing values and reduces the need for hyperparameter tuning, making it easier to use for practitioners. Additionally, CatBoost provides insights into feature importance and interactions, aiding in model interpretation and decision-making.

2.2. Weight of Evidence and Information Value

2.2.1 Weighted of Evidence

The WOE calculates how effective each property, or collections of attributes, is at distinguishing between good and bad accounts. It’s a metric for determining the proportion of good and bad in each feature (i.e, the odds of a person with that attribute being good or bad). The WOE depends on the log of odds calculation:

$$\left({}^{Distr\ Good}/_{Distr\ Bad}\right)$$

Which measures odds of being good.

A more user-friendly way to calculate WOE:

$$WOE = \ln\left(\frac{Distr\ Good}{Distr\ Bad}\right)$$

To compute the Weight of Evidence (WOE) for a continuous variable, the process begins by partitioning the data into 10 groups, or fewer depending on the data distribution. Subsequently, the occurrences of events and non-events are tallied within each group (bin), and the proportions of events and non-events are calculated accordingly. The WOE is then derived by taking the natural logarithm of the ratio between the percentage of non-events and the percentage of events.

For categorical variables, the procedure for computing WOE differs slightly. It is unnecessary to divide the data into distinct segments. Instead, the initial step is omitted, and the focus shifts directly to determining the occurrences of events and non-events within each group (bin). Following this, the proportions of events and non-events are computed for each group, and WOE is calculated using the natural logarithm of the ratio between the percentage of non-events and the percentage of events.

2.2.2 Information Value

The Information Value (IV) serves as a valuable tool for assessing the significance of variables within a predictive model. It aids in ranking variables based on their importance. The formula employed to compute IV is as follows:

$$IV = \sum_{i=1}^{n} (Distr\ Good - Distr\ Bad) \times \ln(\frac{Distr\ Good_i}{Distr\ Bad_i})$$

Based on this methodology, one rule of thumb regrading IV is:

- Less than 0.02: useless.
- 0.02 to 0.1: weak.
- to 0.3: medium.
- 0.3 to 0.5: strong.
- ≥ 0.5: very strong variable. This case may raise suspicions of being overly advantageous, potentially warranting further investigation.

3. Methodology

3.1. Data Source: LendingClub Loan Approval Dataset

The dataset utilized in this study was sourced from the public repository of Lending Club, a peer-to-peer lending platform facilitating direct loans between individuals or institutional investors. This platform serves as a bridge between borrowers and investors, offering a streamlined and cost-efficient alternative to traditional lending institutions. The dataset spans the timeframe from 2007 to 2018 and comprises over 1.3 million observations along with more than 150 features. Due to the vastness of the dataset, conducting analyses with all features and observations proves challenging.

Hence, a data cleaning process of 466,285 observations and 41 columns was undertaken, selecting specific features as predictor variables for the models while removing the remaining ones.

3.2. Data Exploration:

3.2.1 Target Variable

The "loan_status" variable represents the current state of the loan, indicating various statuses such as "Fully Paid," "Current," "Charged Off," "Late (31-120 days)," and "In Grace Period" within the dataset. For our analysis, we focused on two specific statuses: "Fully Paid" and "Charged Off." We encoded "Fully Paid" as 0 and "Charged Off" as 1 to facilitate our modeling process.

3.2.2 Independent Variables

Table 3.1: Independent Variables Description

No	Column	Description
1	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.
2	funded_amnt_inv	The total investment amount pledged by investors for the loan at that particular time.
3	term	The duration of the loan in terms of the number of payments, typically expressed in months and offering options of either 36 or 60 months.
4	int_rate	The interest rate applied to the loan.
5	installment	The monthly payment owed by the borrower if the loan originates.
6	grade	The loan grade assigned by Lending Club.
7	emp_length	The length of employment in years, ranging from 0 to 10, with 0 indicating less than one year of employment and 10 representing ten or more years of employment.
8	home_ownership	The borrower's reported home ownership status provided during registration, with categories including RENT, OWN, MORTGAGE, or OTHER.
9	annual_inc	The borrower's self-reported annual income provided during registration.
10	verification_status	Indicates whether the borrower's income was verified by Lending Club, not verified, or if the income source was verified.

11	issue_d	The month in which the loan was funded.
12	pymnt_plan	Indicates if a payment plan has been established for the loan.
13	purpose	A category provided by the borrower to describe the reason for the loan application.
14	addr_state	The state provided by the borrower in the loan application.
15	dti	A ratio computed by dividing the borrower's total monthly debt payments on all debt obligations (excluding mortgage and the requested LC loan) by the borrower's self-reported monthly income.
16	delinq_2yrs	The count of instances where the borrower's credit file indicates 30+ days past-due delinquency in the past 2 years.
17	earliest_cr_line	The month when the borrower's earliest reported credit line was opened.
18	inq_last_6mths	The count of inquiries made in the past 6 months, excluding auto and mortgage inquiries.
19	mths_since_last_delinq	The number of months since the borrower's last delinquency.
20	open_acc	The count of open credit lines in the borrower's credit file.
21	pub_rec	The count of derogatory public records.
22	revol_bal	The total revolving balance of credit.
23	revol_util	The revolving line utilization rate, representing the borrower's credit usage relative to all available revolving credit.
24	total_acc	The total count of credit lines currently in the borrower's credit file.
25	initial_list_status	The initial listing status of the loan, with possible values of "W" or "F.
26	out_prncp	The remaining outstanding principal amount for the total funded loan.
27	out_prncp_inv	The remaining outstanding principal amount for the portion of the total funded loan funded by investors.
28	total_pymnt	The total payments received to date for the total funded amount.
29	total_pymnt_inv	The total payments received to date for the portion of the total funded amount funded by investors.

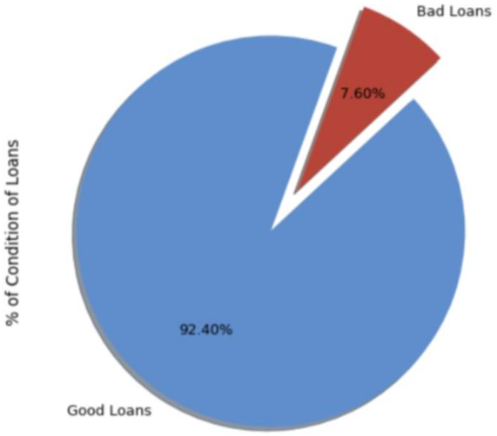
30	total_rec_int	The total interest received to date.
31	last_pymnt_d	The month of the most recent payment received.
32	last_pymnt_amnt	The total amount of the last payment received.
33	last_credit_pull_d	The most recent month when Lending Club pulled credit for this loan.
34	collections_12_mths_ex_med	The count of collections in the past 12 months excluding medical collections.
35	application_type	Indicates whether the loan application is individual or joint, involving two co-borrowers.
36	acc_now_delinq	The count of accounts where the borrower is currently delinquent.
37	tot_coll_amt	The total amount of collections ever owed.
38	tot_cur_bal	The total current balance across all accounts.
39	total_rev_hi_lim	The total revolving high credit or credit limit.
40	funded_amnt	The total amount committed to the loan at a given point in time.

3.2.3 Exploratory Data Analysis (EDA)

The primary objective of the Exploratory Data Analysis (EDA) phase is to delve into the dataset to discern the significance of various variables, present summary statistics, and employ visualization techniques. This phase serves as a crucial initial step in comprehensively understanding the dataset, unraveling hidden patterns, and establishing foundational insights.

We will start by exploring the distribution of "loan_status" target variable, we observe a notable data imbalance between good loans and bad loans. Specifically, the number of applications marked as defaulted (class 1) accounts for only 7.6%, while those who have not defaulted (class 0) composed for a majority of 92.4%.

Figure 3.1: Distribution of Loan Status



Credit grade scores are important metrics for assessing the overall level of risk. From the figure below, it's evident that lower-grade scores correspond to larger loan amounts, potentially indicating higher risk levels. Logically, customers with lower grades would likely face higher interest rates, leading to increased repayment burdens. Specifically, those with a grade of "C" exhibited a higher likelihood of loan default.

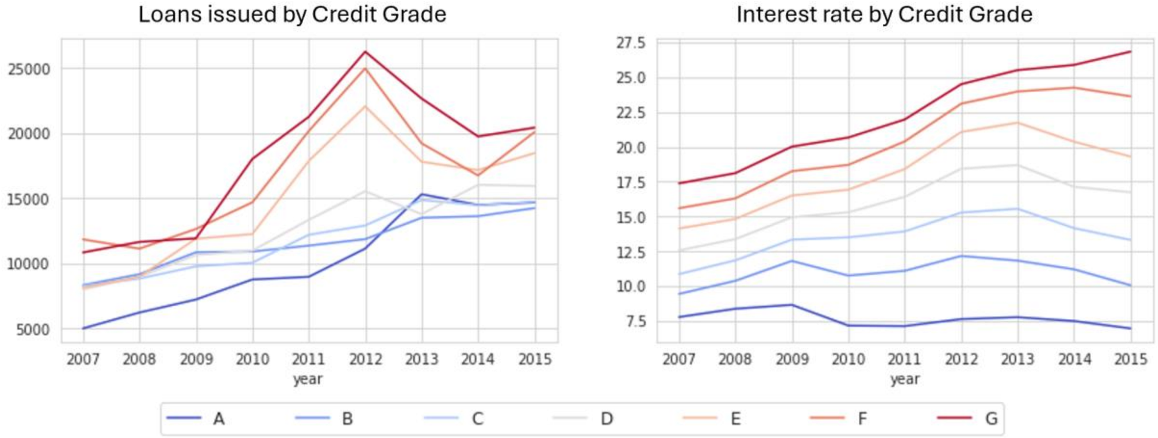


Figure 3.2: The Importance of Credit Grade

Upon investigate the relationship between interest rate and loan, our analysis reveals two key findings: loans with high interest rates, over 13.23%, are more likely to turn into bad loans, and loans with longer repayment periods, specifically 60 months, have a higher chance of defaulting. This highlights the importance of interest rates and loan duration in predicting the likelihood of loan default.

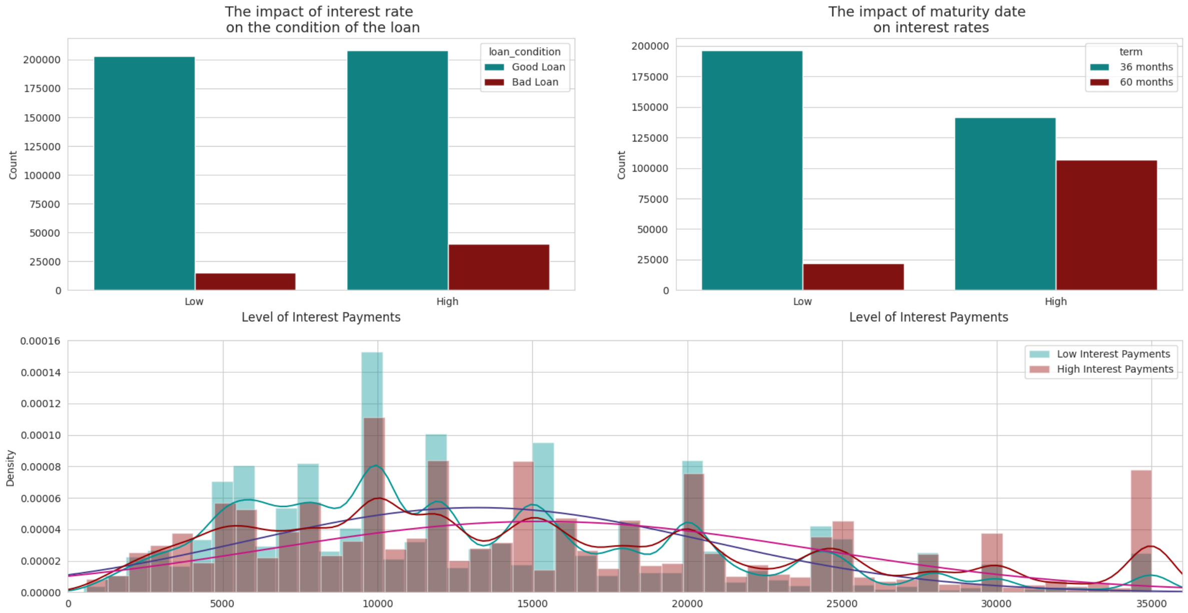


Figure 2.3: Interest rate and Defaulted Loan

When examining the aspect of home ownership in relation to loan performance, Mortgage was the type of home ownership condition that used the highest amount borrowed within loans that were considered to be bad.

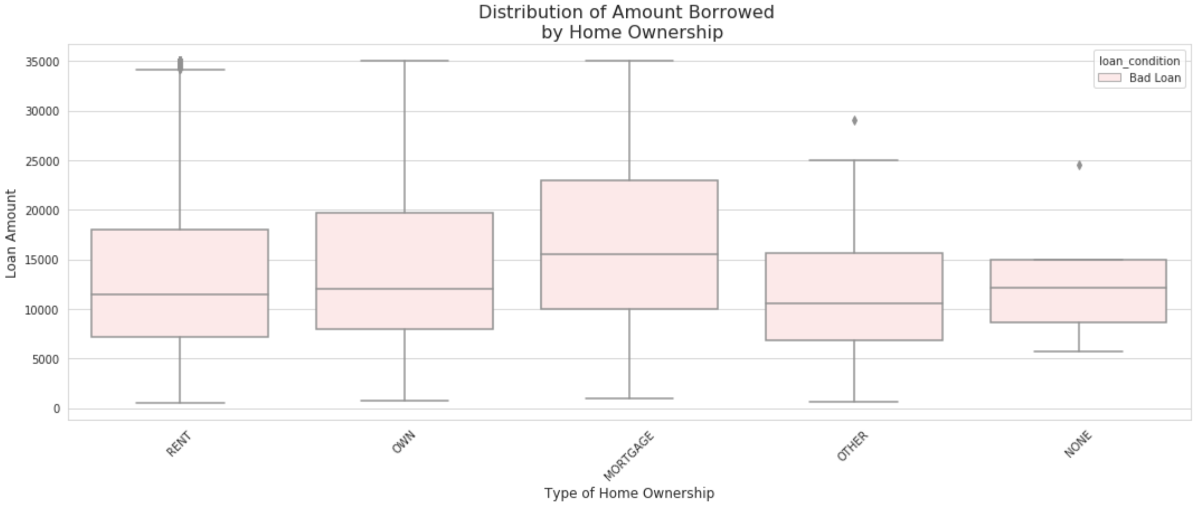


Figure 3.3: Home Ownership and Defaulted Loans

By exploring why people take out loans to see if certain reasons are riskier in terms of repayment. It points out that loans for education and small businesses tend to have a higher chance of not being repaid.

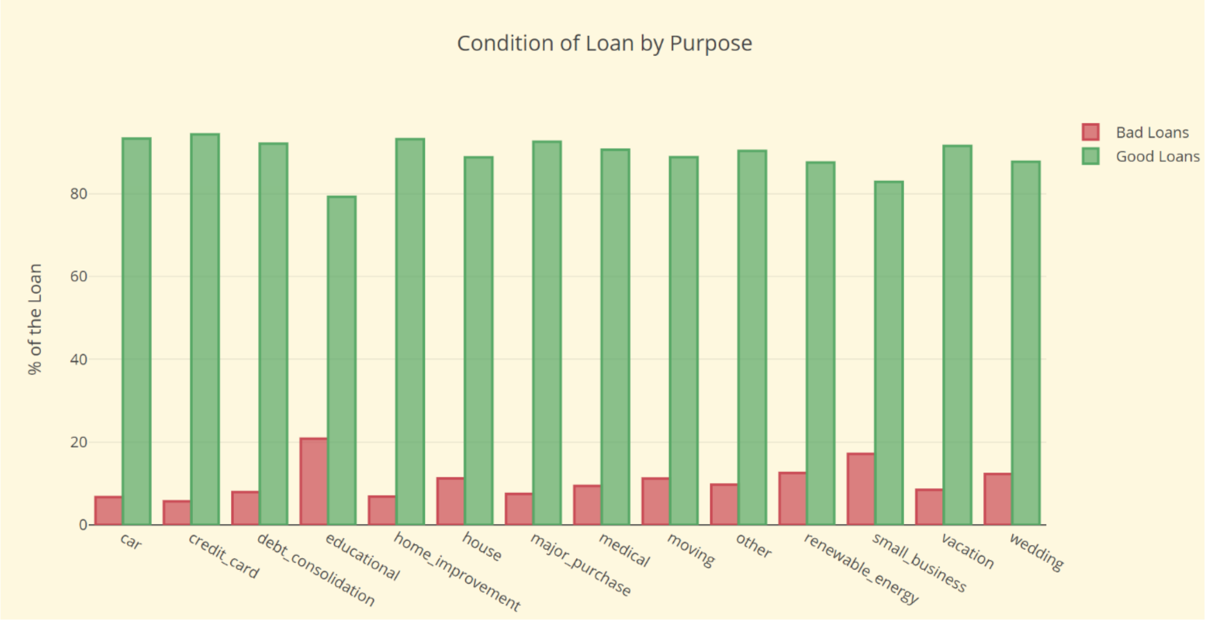


Figure 3.4: Purpose of Loan

In terms of Region, The Northeast region looks like a good place to lend money, while the Southwest and West regions have seen incomes rise slightly. People in the Southwest and West tend to have longer job tenures. And it seems like folks in the Northeast and Midwest aren't taking on much more debt compared to other areas.

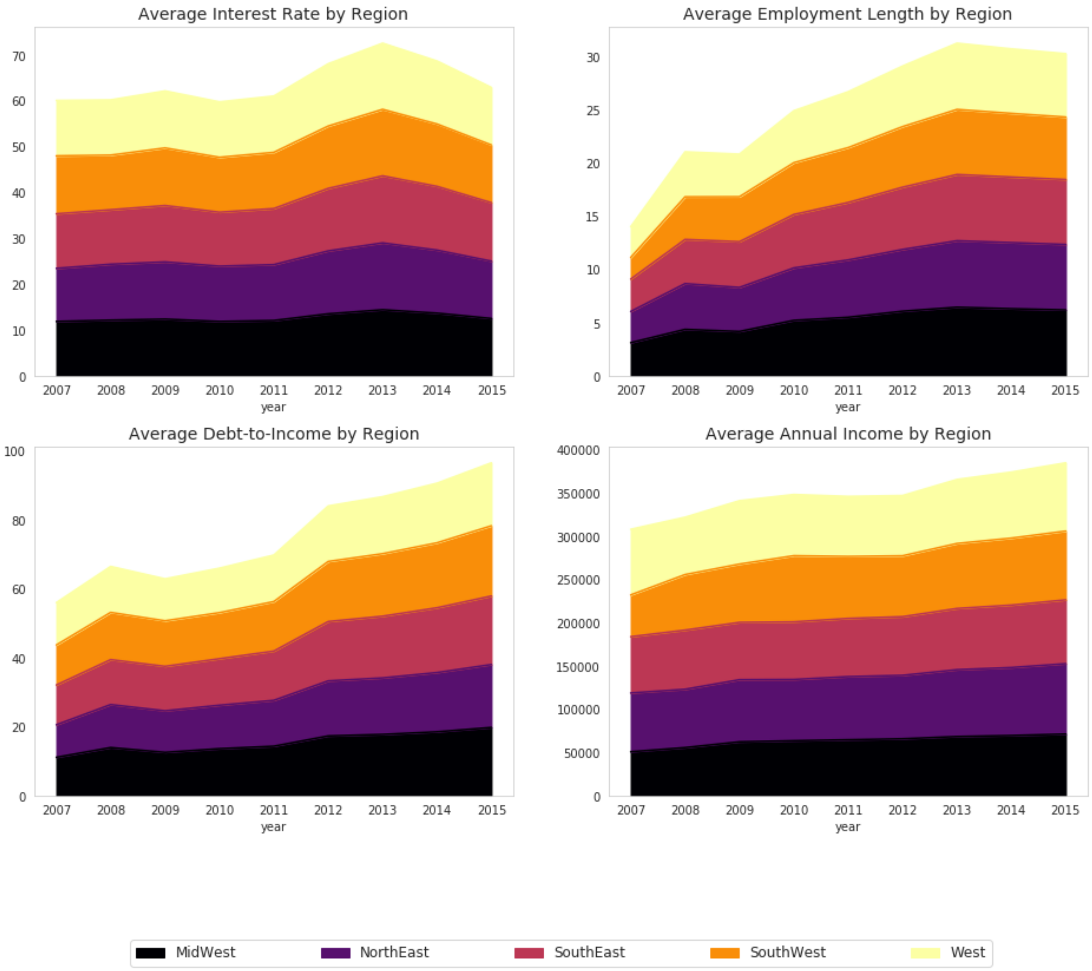


Figure 3.5: Loans by Region

3.3. Data Preprocessing

Having gained insights from our exploratory analysis, we are now moving on to the data preparation phase. Here, we conduct various preprocessing tasks to refine each variable in the dataset. Some of the key activities encompassed in this phase include:

3.3.1. Data Cleaning

In the Data Cleaning phase, it's reassuring to note that our dataset is already in excellent condition, requiring minimal intervention. With no null values present, we only focus on refining the data types, particularly for time-related variables, to ensure they align correctly with our analytical needs. This signifies a solid foundation for our subsequent analyses, allowing us to focus more on ensuring data integrity and consistency throughout our investigation.

3.3.2. Feature Selection

Feature selection is a critical step in data analysis as it allows us to enhance model performance by focusing only on the most relevant features. Not all features contribute equally to predictive accuracy, and including irrelevant or redundant ones can lead to overfitting and decreased model interpretability. Within the framework of this project, I have employed feature selection techniques based on two main methods: Correlation Analysis and Information Value (IV) assessment.

Correlation Analysis:

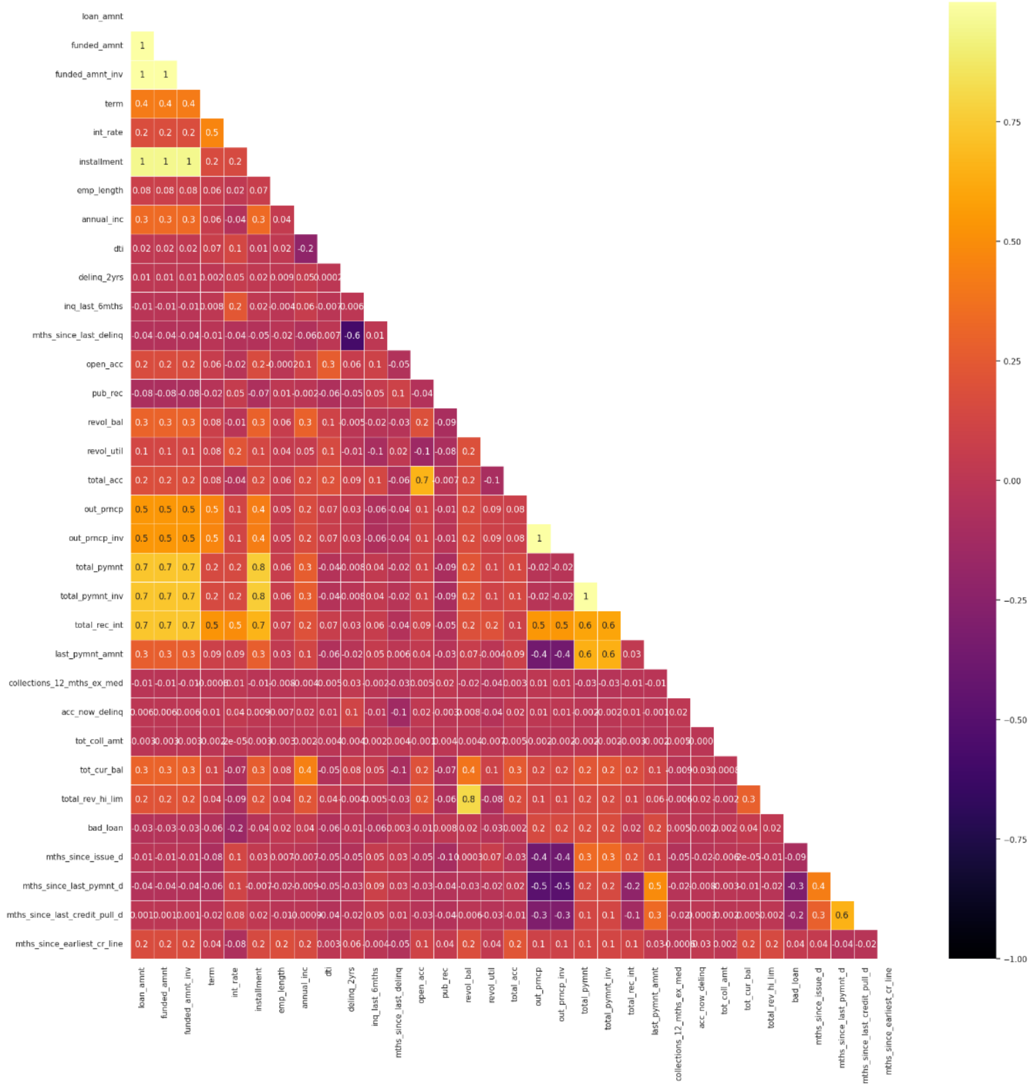


Figure 3.6: Correlation Matrix

The correlation matrix shows that there is a significant correlation of installment with “total_pymnt” and “total_pynmt_inv” (both equal 0.8), “revol_bal” with “total_rev_hi_lim” (0.8), “total_acc” with “open_acc” (0.7), as well as the perfect correlations of “installment” with “loan_amnt”, “funded_amnt” and “funded_amnt_int”. These strong correlations may result in multicollinearity, a condition in which predictors are highly correlated, potentially leading to instability in our model's coefficients and decreased interpretability. Thus, it is imperative to address this issue by considering the removal of one feature from each correlated pair to mitigate the impact of multicollinearity on our model's performance.

The following features will be dropped : “loan_amnt”, “revol_bal”, ”funded_amnt”, “funded_amnt_inv”, “installment”, “total_pymnt_inv”, “out_prncp_inv”, “total_acc”.

Information Value assessment:

An approach commonly used in credit risk problems, IV estimation was utilized in this study to determine the correlation between each attribute and the resultant variable (Zdravevski et al., 2014). For continuous variables, IV calculations were conducted after categorizing them into bins, and the findings are displayed in the provided table:

Table 3.2: IV of numeric features

Variable	IV
acc_now_delinq	0.00020
pub_rec	0.00050
collections_12_mths_ex_med	0.00073
tot_coll_amt	0.00074
delinq_2yrs	0.00104
mths_since_last_delinq	0.00249
open_acc	0.00450
emp_length	0.00717
revol_util	0.00886
total_rec_int	0.01111
total_rev_hi_lim	0.01884
mths_since_earliest_cr_line	0.02135
tot_cur_bal	0.02638
term	0.03548
annual_inc	0.03800
inq_last_6mths	0.04045
dti	0.04103
mths_since_issue_d	0.09054
mths_since_last_credit_pull_d	0.31306
int_rate	0.34772
total_pymnt	0.51579
out_prncp	0.70338
last_pymnt_amnt	1.49183
mths_since_last_pymnt_d	2.33119

The rule of thumb says that all variables with IV < 0.02 are not useful for prediction and IV > 0.5 have a suspicious predictive power. Therefore, the following 11 variables will not be included “acc_now_delinq”, “pub_rec”, “collections_12_mths_ex_med”, “tot_col_amt”, “delinq_2yrs”, “mths_since_last_delinq”, “open_acc”, “emp_length”, “revol_util”, “total_rec_int”, “total_rec_hi_lim”.

Throughout the feature selection process, we strategically discarded 15 features, resulting in a significant reduction in the input matrix size from 40 dimensions to 25 dimensions.

3.3.3. Data Transformation

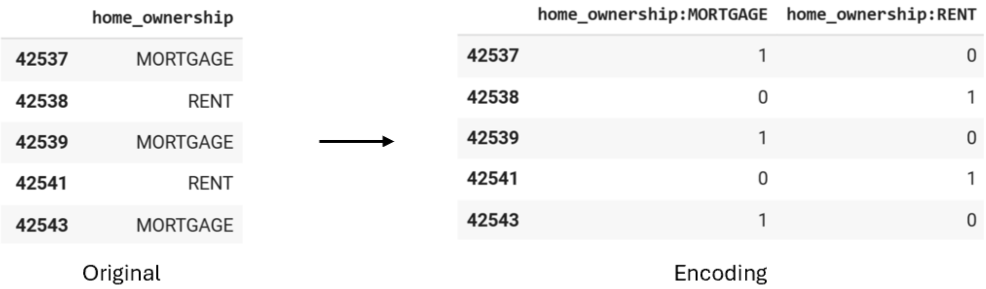
a. Categorical Features Transformation

In machine learning, many algorithms require numerical input data to operate effectively. Therefore, categorical features in machine learning models need to be transformed into a numerical format. To encode the nominal features in this project, I apply the One Hot Encoding method, also know as the dummy encoding.

One-Hot Encoding is a technique used to convert categorical variables into a format that can be provided to machine learning algorithms. It involves creating binary dummy variables for each category within a categorical feature.

E.g: Encoding categorical feature “home_ownership”:

Figure 3.7: Example of One hot Encoding on "home_ownership"



	home_ownership
42537	MORTGAGE
42538	RENT
42539	MORTGAGE
42541	RENT
42543	MORTGAGE

Original

	home_ownership:MORTGAGE	home_ownership:RENT
42537	1	0
42538	0	1
42539	1	0
42541	0	1
42543	1	0

Encoding

b. Numeric Features Transformation

In the domain of numeric data transformation, various methods have been meticulously developed to cater to the diverse needs of data preprocessing. This paper adopts Standardization as the chosen approach.

Standardization is pivotal for addressing scale discrepancies among variables, ensuring that no single feature dominates the analysis due to its magnitude. By employing Standardization, the data undergoes a transformation where the mean is centered at zero, and the standard deviation becomes one. This normalization process plays a crucial role in enhancing the coherence and cohesion of the dataset, setting the stage for effective analysis and modeling.

The standardization formular is expressed as follows:

$$x_{standardized} = \frac{x - \mu}{\sigma}$$

Where:

- $x_{standardized}$ is the standardized value of the original data point x.
- μ is the mean of the feature.
- σ is the standard deviation of the feature.

One practical implementation of Standardization is through the StandardScaler method provided by the sklearn.preprocessing module in Python. This method seamlessly standardizes the data by centering it around zero and scaling it to unit variance. By leveraging StandardScaler, the dataset undergoes a uniform transformation, enhancing the robustness and interpretability of machine learning models.

3.3.4. Data Splitting and SMOTE Implementation

In the process of data splitting, I employ the train-test split functionality from sklearn library facilitates the division of the dataset into training and testing subsets, following 80/20 ratio. Earlier, we examined the notable data imbalance between two classes in the target variable. This imbalance poses a significant challenge in the realm of machine learning, as models trained on such data tend to exhibit a bias towards the majority class. Therefore, addressing this issue is imperative to ensure the integrity and efficacy of the ensuing machine learning model.

To address this obstacle, I applied the SMOTE algorithm to my training dataset to handle class imbalance. SMOTE, short for Synthetic Minority Over-sampling Technique, is a well-known approach for managing class imbalances. This technique entails creating artificial instances for the minority class through interpolation of existing data points. By doing so, it aims to rectify the skewed class distribution and enhance the efficacy of machine learning algorithms. The mechanism behind SMOTE involves the random selection of a data point from the minority class, followed by the identification of its k-nearest neighbors. Subsequently, synthetic instances are generated by interpolating between the chosen data point and its neighbors. This iterative process continues until the desired balance between classes is achieved.

Upon implementing SMOTE, the distribution of the target variable "loan_status" exhibits an equal representation of both 0 and 1 values, indicating successful mitigation of class imbalance. The results of the SMOTE process on the training set are shown in the table below:

Table 3.2: Classes Distribution on Train and Train SMOTE

Loan Status	Original Train	SMOTE Train
Class 0 (Non-Defaulted)	131,140	129,830
Class 1 (Defaulted)	14,208	129,830

3.4. Modeling

Our primary objective is to develop a robust predictive model specifically designed to estimate borrowers' default risk. In this paper, the models will be trained on the train set after applying SMOTE to address class imbalance. As the mentioned above, five models will be used to examine: Logistic Regression, Decision Tree, Random Forest, CatBoost and XGBoost. Cross-validation will be employed to assess model performance on different subsets of the data, and hyperparameter tuning will be conducted using GridSearch CV to optimize model performance. The results of each model's performance will be compared to identify the most effective model for default prediction.

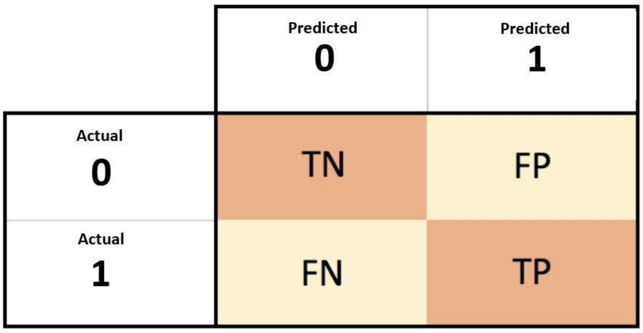
4. Results

4.1. Metrics

In evaluating the performance of credit default risk classification models, it is imperative to employ various metrics that provide insights into the model's effectiveness in distinguishing between creditworthy and defaulting borrowers. Here, we delve into five fundamental metrics: accuracy, precision, recall, F1-score, and AUC value. To understand these measures, we consider four key variables:

- TP: the count of correctly predicted positive values by the model.
- FN: the count of positive samples erroneously predicted as negative.
- FP: the count of negative samples erroneously predicted as positive.
- TN: the count of correctly predicted negative samples by the model.

Using the confusion matrix, we can visualize all 4 variables above in a single table:



	Predicted 0	Predicted 1
Actual 0	TN	FP
Actual 1	FN	TP

Figure 4.1: Confusion Matrix for Binary Classification

From the Confusion Matrix, several metrics can be calculated to evaluate the predictive performance of the classification model:

Accuracy serves as a foundational metric, offering a straightforward measure of the model's overall correctness in predicting credit default outcomes. It assesses the ratio of correctly classified instances to the total instances in the dataset.

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

Precision focuses on the reliability of positive predictions made by the model, emphasizing the proportion of correctly predicted positive instances out of all instances predicted as positive.

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

Recall, also referred to as sensitivity or true positive rate, measures the model's ability to capture all positive instances correctly, thereby minimizing the risk of overlooking potential defaulters.

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

Another method is F1-Score. The F1-score strikes a balance between precision and recall by computing the harmonic mean of the two metrics. This harmonic mean provides a single metric to assess the model's performance.

$$F1 - score = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

Our final metric is the AUC value, which stands for “Area Under the Curve”. It quantifies the discriminative ability of a model by calculating the area under the Receiver Operating Characteristic (ROC) curve. One of the key strengths of AUC lies in its ability to summarize a model's overall performance across all possible classification thresholds. This is particularly beneficial when dealing with imbalanced datasets, where the number of positive and negative instances might be unequal.

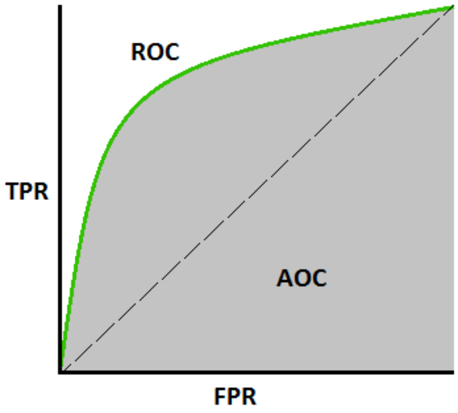


Figure 4.2: AUC ROC

4.2. Result

To find the best-performing model among the five classification models evaluated, we assess their performance on both the training and test datasets. This comparative analysis allows us to determine which model demonstrates superior generalization to unseen data, aiding in the selection of the most effective approach for the classification task. The experimental results on two sets are shown below:

Table 4.1: Summary Performance of Classification Models on Train set

Model	Accuracy	Precision	Recall	F1	AUC
Logistic Regression	0.909	0.909	0.909	0.909	0.909
Decision Tree	0.999	0.999	0.999	0.999	0.999
Random Forest	0.999	0.999	0.999	0.999	0.999
XGBoost	0.939	0.939	0.939	0.939	0.939
CatBoost	0.950	0.950	0.950	0.950	0.950

Table 3.2: Summary Performance of Classification Models on Test set

Model	Accuracy	Precision	Recall	F1	AUC
Logistic Regression	0.857	0.607	0.612	0.609	0.612
Decision Tree	0.846	0.629	0.683	0.647	0.683
Random Forest	0.897	0.720	0.699	0.711	0.699
XGBoost	0.891	0.695	0.699	0.697	0.699
CatBoost	**0.903**	**0.723**	**0.704**	**0.714**	**0.707**

The performance of the classification models on the train and test sets shows some interesting results and insights. Overall, CatBoost outperformed other algorithms in all criteria in the test set. This indicates that CatBoost has successfully learned patterns that generalize well to unseen data, making it a robust choice for this specific credit classification task. Random Forest and Decision Tree exhibited signs of overfitting, as their high sores on the train set (0.999 for all metrics) suggest that they may have captured noise and specific patterns unique to the training, leading to poor generalization, thus dropping scores significantly on the test set. Logistic Regression, despite its relatively lower performance on both sets, demonstrated more stability. XGBoost, a powerful boosting algorithm, its results are slightly better than Logistic Regression but not as impressive as CatBoost.

5. Conclusion and discussion

In this paper, we explored the effectiveness of machine learning techniques in improving default prediction using Lending Club data. Through the implementation of Synthetic Minority Over-sampling Technique (SMOTE), we successfully addressed the issue of class imbalance in the target variable, enhancing the predictive capability of our models. By generating synthetic samples for the minority class, SMOTE facilitated a more balanced representation of default and non-default instances, thereby mitigating the bias towards the majority class prevalent in the dataset.

Furthermore, our investigation revealed that tree-based methods outperformed traditional Logistic Regression in default prediction tasks. The decision trees, Random Forest, and Boosting models exhibited superior performance in capturing complex non-linear relationships present in the data, leading to more accurate predictions of borrower default likelihood. This finding underscores the importance of employing advanced ensemble learning techniques for credit risk assessment, particularly when dealing with intricate datasets like those from peer-to-peer lending platforms.

Future developments in this field might delve into comparing the efficacy of various machine learning models, including Support Vector Machines, K-Nearest Neighbors, Neural Networks, and Deep Learning, among others. Such endeavors could shed light on the optimal models for different contexts, potentially enhancing the accuracy and reliability of credit risk assessment. Furthermore, forthcoming research endeavors could explore the integration of alternative data outlets such as social media and online activities, offering potential avenues for bolstering credit risk analysis.

To summarize, this study underscores the effectiveness of machine learning methodologies in credit risk assessment. Leveraging the top-performing model, CatBoost, financial institutions can refine their credit risk analysis frameworks, leading to substantial long-term financial benefits. There exists ample opportunity for further exploration in this domain, including refining existing models and devising novel approaches that incorporate emerging data sources and features, thereby augmenting the precision and effectiveness of credit risk analysis.

References

Altman, E. I. (1968). Financial ratios, discriminant analysis and the prediction of corporate bankruptcy. The Journal of Finance, 23(4), 589–609. Altman, E.I., and Sabato, G. (2007). Modelling Credit Risk for SMEs: Evidence from The US Market. ABACUS, 43 (3), 332-357.

Breiman, Leo. 2000. Some Infinity Theory for Predictors Ensembles. Technical Report;

Berkeley: UC Berkeley, vol. 577.

Butaru, Florentin, Qingqing Chen, Brian Clark, Sanmay Das, Andrew W. Lo, and Akhtar Siddique. 2016. Risk and risk management in the credit card industry. Journal of Banking and Finance 72: 218–39.

Chen, T.; Guestrin, C., (2016), "XGBoost: A Scalable Tree Boosting System". 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD ’16.

Gahlaut, A., Tushar, & Singh, P.K. (2017). Prediction analysis of risky credit using Data mining classification models. 2017 8th International Conference on Computing, Communication and Networking Technologies (ICCCNT), 1-7.

Galindo, Jorge, and Pablo Tamayo. 2000. Credit risk assessment using statistical and machine learning: Basic methodology and risk modeling applications. Computational Economics 15: 107–43.

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., ... & Liu, T. Y. (2017). Lightgbm: A highly efficient gradient boosting decision tree. Advances in neural information processing systems, 30.