Home Credit: Credit Risk Model Stability

By: Jonathan Bugg (jtb2182),

Yunhao Luo (yl5444),

Yiyang Wu (yw4141),

Bruce Zhang (jz3731)

Table of Contents

I. Abstract……..……..…….……Page 3

II. Introduction……..…….…….……Page 3

III. Literature Review…..……..…….……Page 4

IV. Data Modeling……..…….…….……Page 5

V. Model Summary……..…….……Page 11

VI. Discussion……..……..Page 13

VII. Conclusion……..…….Page 16

VIII. Appendix……..……...….Page 17

IX. Bibliography……..…….…….….Page 23

Abstract

This study explores the development of predictive models aimed at improving loan accessibility and risk management. We utilize a dataset provided by Home Credit Group, supplemented with external data, to build and compare models using Logistic Regression, Random Forest, and XGBoost. Our analytical process incorporates various sampling methods to address the imbalanced dataset and the Boruta algorithm in robust feature selection. The performance and stability of the models in predicting loan defaults are evaluated by AUC, TPR, FPR, and Gini stability. This study highlights the role of machine learning techniques in improving financial risk management for loan providers and improving loan accessibility.

Introduction

The challenge posed by Home Credit Group in the Kaggle credit risk competition centers on addressing the limitations faced by individuals with little to no credit history. Often, these individuals are denied loans due to an absence of traditional data to assess their creditworthiness. This lack of data can result from various factors, such as young age or a preference for cash transactions, rather than credit. The competition focuses on creating a reliable predictive model that can accurately determine if a potential borrower will default, thereby making loans more accessible to them. Traditional banking systems often exclude these individuals from loan eligibility, limiting financial inclusion. This project highlights data science's potential to improve loan repayment predictability and expand credit access to underserved groups. Highlighting Home Credit's initiative to offer loans to those with little credit history, the project aims to improve the accuracy of predicting loan repayment probabilities, thereby supporting financial inclusion. The project addresses the need for predictive models to be regularly updated for accuracy, despite the delays in observing loan repayment behavior, striving for a balance between model stability and performance. Our research builds on four key papers covering aspects of predictive modeling in credit risk assessment, including handling imbalanced class distributions, comparing machine learning models, feature selection techniques, and managing missing data. By integrating these methodologies, we aim to enhance consumer finance providers' decision-making processes, contributing to both credit risk modeling advancements and the financial well-being of traditionally marginalized individuals.

The significance of this modeling exercise is multifaceted. Firstly, it has substantial implications for financial inclusion. By improving predictive accuracy regarding loan repayment, consumer finance providers can responsibly extend credit to a larger segment of the population that is traditionally underserved. This includes individuals without a credit history who may otherwise be excluded from the financial system. Secondly, enhancing the stability and reducing the need for frequent updates of these models can save valuable resources and reduce the risk of losses due to outdated or ineffective predictive tools. Finally, our work will contribute to the ongoing discussions and developments in the field of consumer finance, especially in how data science can be leveraged to foster safer and more inclusive lending practices. These advancements will not only benefit financial institutions by mitigating their risks but also empower consumers by expanding their access to needed financial services

Literature Review

In our literature review for the credit risk modeling project, we explore various methodologies that have been employed to overcome some of the data-preprocessing hurdles and enhance the performance of predictive models. A key piece of literature by Yuelin Wang is a comparative assessment of various machine learning models using bank loan data to predict default probabilities. The study highlights the Random Forest model as particularly effective, showing high precision and robustness in handling complex, high-dimensional data. This finding is crucial for our project, suggesting that Random Forest could lower predictive errors in our assessments due to its ability to process intricate and non-linear information without prior feature selection. As the study tackles a similar problem with data comparable to that of our project, their report on different model performances guided our model selection process.

Additionally, a comprehensive survey conducted, by Manoranjan Dash and Liu Huan, of thirty-two feature selection techniques relevant to large datasets delineates methods that optimize classifier performance by effectively distinguishing relevant predictors from noise. This framework is instrumental for our data preprocessing, ensuring efficiency and efficacy in our model; our dataset contains hundreds of features and demands a process of feature selection to optimize predictive performance. The Boruta package was used in the final feature selection process before fitting a model. The package is designed to be used for random forest models as "it iteratively removes the features which are proved by a statistical test to be less relevant than random probes" (Kursa). The Boruta algorithm is a robust feature selection method that identifies all relevant features by introducing randomness and extending the dataset with shadow attributes—randomized copies of real attributes. It uses a Random Forest classifier to compute the importance of each feature and compares these against the importance derived from the shadow attributes to distinguish between genuinely significant features and those affected by random fluctuations. This process involves iterative shuffling, classification, and statistical testing against the maximum importance score of the shadow attributes, and continues until all features are classified or a preset limit of iterations is reached. Highly regarded for its ability to reduce misleading random fluctuations and its scalability, Boruta is especially useful in complex domains like biology, where understanding every influential feature is crucial.

Lastly, the importance of handling missing data in machine learning is underscored by a review that examines various statistical techniques to manage missing entries across large datasets. This research is vital for our preprocessing steps, as implementing robust methods for missing data imputation ensures the reliability and integrity of our input data, thus enhancing the overall performance of our predictive models.

Data Modeling

1. Data Description

The data we are using is provided by Home Credit Group, an international non-bank financial institution based in the Netherlands that provides financial services to people with little or no credit history, ensuring a safe borrowing environment for both sides. Home Credit Group provides internal datasets available to them such as loan application data, personal data on applicants, deposit data, and debit card data. Additionally, these internal datasets are supplemented with external data from two credit bureaus and three tax registries. Together these sources form a more holistic analysis of an applicant's credit/financial history. The data spans almost two years of loan applications from 01/01/2019 to 10/05/2020.

Each dataset is categorized as either depth 0, 1, or 2. Depth 0 data includes static features that correspond to each case, whereas depth 1 has an associated historical record, indexed by num_group1. For example, the information about multiple people associated with a loan application is stored in a dataset with depth = 1.

case_id	**num_group1**	**…**	**classificationofcontr_13M**	**refreshdate_3813885D**
388	0	…	"4408ff0f"	null
388	1	…	"ea6782cc"	null
388	2	…	"a55475b1"	"2019-01-28"
…	…	…	…	…
2588481	0	…	"ea6782cc"	null
2588481	1	…	"ea6782cc"	null
2588481	2	…	"a55475b1"	"2019-07-07"
…	…	…	…	…

Table 1: Depth 1 example from train_credit_bureau_a_1_0

Depth 2 data also contains instances where a case has a historical record and is indexed by both num_group1 and num_group2. For example, each of the loan applicants may have multiple other loans, multiple credit cards, or multiple income sources, and this data is stored in a dataset with depth = 2.

case_id	num_group1	num_group2	…	subjectroles_name_541M	pmts_year_1139T
388	0	0	…	"a55475b1"	2018.0
388	0	1	…	"a55475b1"	2018.0
388	0	2	…	"a55475b1"	2018.0
…	…	…	…	…	…
2548729	2	24	…	"a55475b1"	2019.0
2548729	2	25	…	"a55475b1"	2019.0
2548729	2	26	…	"a55475b1"	2019.0
…	…	…	…	…	…

Table 2: Depth 2 example from train_credit_bureau_a_2_0

For modeling and predicting purposes, depth 1 and 2 data need further aggregation.

2. Data Preprocessing

To get an overview of the data, we started with exploratory data analysis on each of the 465 features across 32 training datasets. The process included inspecting each feature's data type, null value percentage, distribution, and whether it requires further aggregation (for depths 1 and 2). We paid particular attention to the categorical variables to examine potential ordinality since an encoding procedure would typically be mandatory (one-hot encoding for categorical variables; label encoding for ordinal variables) when assessing feature importance.

With the generated insights from EDA, we began our data-cleaning procedure. We first configured the data types by the suffixes of the features, parsed the data to appropriate formats, and cast them to the built-in variable types of Python. In particular, the suffixes of each column are named under the convention: “P” denotes days past due transformation; “M” represents masking categories transformation; “A” denotes amount transformation; “D” denotes date transformation; while “T” and “L”" are used for unspecified transformations. For Datetime objects, we transformed the records by computing the difference in the days from the specified “date_decision” column, thereby converting them into numeric features. This method is common for managing date variables in the finance sector, which allows us to impose statistical significance on dates to be incorporated into the model.

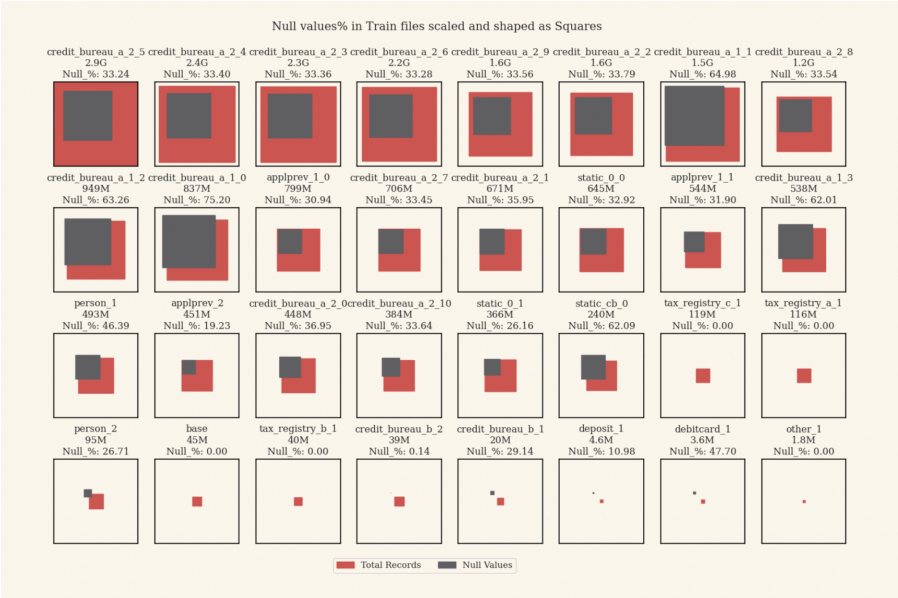
In addition, we introduced various helper methods to aggregate features for each instance in the dataset across different depths. This is particularly useful for handling panel data where summarization of historical records is imperative. Statistical functions such as max, min, median, mean, and last were applied to distinct subsets of data columns based on their transformation types (P, M, A, D, T, L). These

methods allowed us to aggregate and summarize the dataset and reduce it from potentially vast transactional / event-level data to case data for predictive modeling purposes.

Finally, we conducted data aggregation to analyze the entire dataset comprehensively. We augmented the base table by incorporating additional data frames from all depths and derived features from date information.

3. NaN Imputation

The graph below offered a visualization of the number of null entries with respect to the total number of data points in each file. We observed that there were several files where *NaN* dominated and specifically, there existed columns of which more than ninety percent were missing. Hence, it became imperative for us to either produce unbiased estimates for the missing values or discard those features when modeling.



Graph 1: Variable missingness (Saharovskiy, 2024)

To guarantee a robust analysis of the predictive power of each feature, we set a benchmark of seventy-five percent for the removal of variables with null entries, i.e. only retaining features with *NaN* percentage less than seventy-five. Moreover, after appropriate transformations, the dataset was enriched with the additional 700 columns and all of them were in the format of either floating-point numbers or categorical variables, which allowed us to apply various imputation techniques to handle *NaN*s.

For a dataset that includes millions of instances and possesses high variability, the tradeoff between optimal theoretical data-handling mechanisms and practical model complexity became a crucial factor under consideration when we attempted to implement them. In the financial industry, most *NaN*s are categorized by two main mechanisms for missingness - Missing Completely At Random (MCAR), and Missing Not At Random (MNAR).

MCAR:

Under the assumption that the data is MCAR, it suggests the missing values are not reliant on both the observed and the unobserved measurements. Therefore, for each individual feature, the remaining values can be considered as a random sample with replacement from the full distribution constructed from the available data. The method preserves the original distribution because selecting with replacement allows each non-missing value to be selected multiple times, ensuring the proportion of each value relative to others can remain consistent over time. Following the argument, we could establish our first strategy to address *NaN*s:

- For each column, we identified the non-null data as the representative sample of the potential values for the complete set of observations.
- Then we iterated through every null entry and assigned it a sampled value from the distribution.

MNAR:

In contrast, under MNAR, the missing data depends equally on the observed and unobserved values across variables, which, oftentimes, is impossible for us to compute as it is dependent on the unseen data. As a consequence, we introduced a few machine learning algorithms to attempt to handle NaNs based on data types, i.e. floating point numbers or categorical variables.

Floating-Point Numbers:

- Iterative imputation is applicable when there exist complex interdependencies between features and models each feature with missing values as a function of others in a round-robin fashion. In particular, we assumed a linear relationship among numeric columns and applied Bayesian Ridge regression to estimate the regression parameters.
- K-nearest neighbor imputation was feasible for the approximation of *NaN*s via finding the k closest points in the Euclidean space and replacing the missing value as the mean of those k neighbors.
- Support vector regression was the last resort we utilized for the estimation of *NaN*s. It fitted the best hyperplane within a predefined threshold value from the actual observed data for each feature.

Categorical Variables:

- Logistic regression was a starting point for fitting *NaN*s as it is capable of handling both numeric and categorical inputs. In our case, we iteratively selected each category as a baseline reference and modeled the log-odds of being in other categories relative to the reference. Similar to the iterative imputation we implemented, we assumed a linear relationship between the log-odds of the dependent variables and the independent ones.
- Decision tree and random forest classification algorithms were both implemented to handle categorical columns with missing entries. The underlying concept was almost identical for both methods: for every categorical feature, a decision tree was trained on the portion of the dataset

that had the observed value for the target feature and used tree splits to predict the missing ones; while random forest was constructed by averaging multiple deep decision trees which were trained on bootstrap samples of the dataset, thus reducing the variance of predictions.

Understanding the complexity and scale of our data, we tried to improve the computation efficiency by dividing the data into smaller batches, iteratively processing through each batch, and aggregating the results. However, all algorithms we employed under MNAR for the given dataset still required too much time and resources to be executed on a local machine. A future investigation direction will be experimenting with the methods on a cloud compute engine and comparing whether there is a significant improvement from the model under MCAR.

4. Feature Selection

In our integrated dataset, there are approximately 800 features; a systematic approach had to be used to select the relevant features. Taking note of a potentially high correlation between sets of variables, especially for the numeric ones that had been transformed from the datetime objects, we began our feature selection by Principal Component Analysis (PCA). To reduce to a set of approximately independent features, PCA essentially introduces a derived variable as a linear combination of a subset of the original variables based on variance maximization. Since PCA inherently works with continuous data, all categorical variables are supposed to be properly encoded before the analysis. However, we encountered significant numerical difficulties when we attempted to do so. Specifically, the transformation of categorical columns produced several sparse matrices within the data frame, which could lead to a notable increase in the computation overhead, algorithm complexity, and memory storage usage. Considering that after the data processing pipelines, the data frame retained a structure of more than 1.5 million rows and roughly 800 columns, it became impossible for us to further encode each categorical column and perform PCA to reduce the dataset's dimensionality on a local laptop.

The remedial approach we adopted for feature selection was the Boruta algorithm, which essentially performs a top-down search to find the variables that carry the most relevant information to the response variable, rather than the ones that have the strongest correlation. The strength of Boruta ultimately lies in its comprehensive and iterative approach to evaluating feature importance, i.e. usage of shadow copies: via a random shuffle of all features, Boruta maintains an identical distribution and type of data (thus allowing for categorical variables) for each feature while removing any meaningful relationship with the response. By using the maximum importance among all shadow copies of a feature as the benchmark, we effectively guarded against the selection of features that were less important than random chance. This dimensionality reduction process eventually selected 89 raw columns, for which we would expand to 128 by converting each categorical value into a new column and assigning a binary value of 0 and 1. Via one-hot encoding, we ensured that we did not introduce implicit order and priorities among different categories for a variable and removed any potential confounding effect of arbitrary numerical values.

4. Training and Testing Dataset Development

After performing variable selection and *NaN* imputation, the training and testing datasets were developed to train the various machine learning models. Given the large number of training instances in the data provided, the training and test sets were split 50/50 to decrease model training time and provide a robust

test set. In order to address the significant class imbalance of the dataset, three versions of sampling for the training dataset were explored: upsampling, downsampling, and standard sampling. Standard sampling did not resample the data at all, thus leaving the class imbalance in place. This resulted in 763,329 training examples (739,357 non-defaults and 23,972 defaults). Upsampling kept all the instances of the majority class and resampled the minority class (loan default) by creating copies of the minority class at a ratio of ~1:3, minority: majority. This resulted in 985,809 training examples (739,357 non-defaults and 246,452 defaults). Downsampling kept all the instances of the minority class and randomly sampled the majority class at a ratio of ~1:3, minority: majority. This resulted in 98,972 training examples (75,000 non-defaults and 2,3972 defaults). The test set was not resampled to provide an accurate representation of the model performance. The test set contained 763,330 examples (739,308 non-defaults and 24,022 defaults).

Model Summary

This study explored three models to predict loan default: logistic regression, random forest, and XGBoost. Logistic regression was chosen for its simplicity and interpretability, providing a baseline with which to compare more complex models. It is effective for binary classification problems and easily interpretable insights into the influence of predictor variables through its coefficients. Random forest was selected for its proficiency in handling large datasets with higher dimensionality without overfitting. Finally, XGBoost was included due to its dominance in many recent machine learning competitions. XGBoost is well-regarded for its scalability, efficiency, and capability to deliver superior results, even with less tuning of hyperparameters. A baseline random guess model is provided to put the modeling results into context. The following table summarizes the key performance metrics of the models when tested on an unseen testing dataset. TPR and FPR were chosen based on a threshold that minimized the differences between the two. AUC curves and confusion matrices are provided in the appendix.

Model	**AUC**	**TPR**	**FPR**	**Gini Stability**
Random Guess	**0.5**	**0.03**	**0.03**	**N/A**
Logistic Regression	**0.75**	**0.70**	**0.32**	**0.15**
Random Forest	**0.79**	**0.73**	**0.31**	**0.03**
XGBoost	**0.81**	**0.77**	**0.29**	**0.24**

Table 3: Model Comparison of key metrics

All models were initially trained on 89 unique features, which after preprocessing of categorical variables expanded to 128 features. After initial testing of the three different sampling methods, “upsampling” was determined to be the best due to a significant increase in performance across all models. All models used the same upsampled training set and testing set to ensure accurate comparisons.

Logistic Regression Model:

The logistic regression was initially trained with all 128 features. To refine the model, backward feature elimination was employed, methodically removing predictors to enhance model performance and simplicity. After each subsequent model run, the variable with the lowest p-value was removed from the model. Validation of the logistic regression model included comprehensive diagnostics to confirm no violation of the model's critical assumptions (included in appendix). Additionally, equivalence between training and testing performance metrics was verified, ensuring model robustness and generalizability.

Random Forest Model:

The random forest model was initially trained with all 128 features, applying the default parameters provided by the scikit-learn library. Subsequent to this baseline model establishment, hyperparameter tuning was performed. Hyperparameter tuning looked to refine the following variables: number of trees in

the forest, maximum number of features considered at each split, maximum depth of each tree, and minimum number of samples required at each leaf node.

XGBoost Model:

The XGBoost model was initially trained with all 128 features. Unlike the Random Forest model, hyperparameter tuning was not conducted for the XGBoost model. This decision was based on the inherent robustness of XGBoost against overfitting compared to traditional random forest models.

Discussion

The XGBoost model performed the best across the two most important metrics (AUC = 0.81 and Gini Stability = 0.24). Thus this was the final model selected from our analysis.

When determining the FPR and TPR of our models, we assumed an equal weighting between the two for general reporting. In the context of evaluating loan risk, the associated financial risk between an FP and FN may be extremely different. A False Positive (approving an individual for a loan when they end up defaulting) may represent a higher financial risk for Home Credit Group compared to a False Negative (not approving an individual for a loan when they would not end up defaulting) due to the structure of loans. Assume this situation: a risky candidate applies for a $20,000 loan at an annual interest rate of 5% to be repaid over 5 years. The simple interest of this loan (if accepted) would be $5,000 of revenue for Home Credit Group, but the potential loss of this loan would be $20,000 if the loanee defaulted immediately. Using this simple analysis as an example would significantly shift the decision threshold of the model that Home Credit Group would use in practice. The two following tables compare if TPR and FPR are equally weighted vs if the FPR is weighted at 4x compared to TPR (as shown in the above example).

CF using equal weighting	Predicted Label = 0	Predicted Label = 1
True Label = 0	523,187	216,121
True Label = 1	5,476	18,546

Table 4: Confusion Matrix of equal FPR and TPR

CF using equal $ weighting	Predicted Label = 0	Predicted Label = 1
True Label = 0	707,501	31,807
True Label = 1	17,244	6,778

Table 5: Confusion Matrix of non-equal FPR and TPR

Home Credit Group will need to consider this financial risk when applying any of the models chosen from the competition. While the stability metric that they used for the competition is important, balancing the financial risk of a model is also important for the company as a whole.

While validating the XGBoost model, we examined the feature importance of the model to ensure that the model was not influenced by noisy signals and interpreted the data and covariates correctly. Below are the top 20 important features that contributed to the XGB model (with no sign direction only magnitude):

Feature	Description	XGB Variable Importance
mean_pmts_overdue_1140A	Mean value of overdue payment for an active contract.	1
numrejects9m_859L	Number of credit applications that were rejected in the last 9 months.	2
min_dpdmaxdateyear_596T	Minimum value of year when maximum Days Past Due (DPD) occurred for the active contract.	3
mean_pmts_dpd_1073P	The mean value of days past due of the payment for the active contract.	4
maxdpdlast24m_143P	Maximal days past due in the last 24 months.	5
cat__last_education_1138M_P33_146_175	Category ‘P33_146_175’ of applicant's education level from their previous application.	6
days90_310L	Number of credit bureau queries for the last 90 days.	7
cat__min_education_1138M_P33_146_175	Category ‘P33_146_175’ of applicant's education level from their previous application.	8
median_dpdmax_139P	The median value of maximal days past due for active contract.	9
cat__education_1103M_6b2ae0fa	Category ‘6b2ae0fa’ of the level of education of the client provided by the external source.	10
maxdpdlast6m_474P	Maximum days past due in the last 6 months.	11
mobilephncnt_593L	Number of persons with the same mobile phone number.	12
cat__max_classificationofcontr_13M_ea6782cc	Category ‘ea6782cc’ of the active contract.	13
days120_123L	Number of credit bureau queries for the last 120 days.	14
days180_256L	Number of credit bureau queries for the last 180 days.	15
max_birth_259D	The maximum value of the date of birth of the person.	16
totalsettled_863A	The sum of all payments made by the client.	17
min_overdueamountmaxdateyear_2T	Minimum value of year when the maximum past due amount occurred for active contracts.	18
maxdpdtolerance_374P	Maximum number of days past due (with tolerance).	19
pmtnum_254L	Total number of loan payments made by the client.	20

Table 6: XGBoost Variable Importance

The five most important variables (e.g., "mean_pmts_overdue_1140A", “numrejects9m_859L”, “min_dpdmaxdateyear_596T”, “mean_pmts_dpd_1073P”, and “maxdpdlast24m_143P”) provide a holistic look at the creditor’s previous borrowing accounts, promptness of fulfillment, and default history, which are the most logical indicators when evaluating the creditor’s future default tendency.

The applicant's educational background (e.g., "cat__last_education_1138M_P33_146_175" and "cat__min_education_1138M_P33_146_175") is also important. Education level can indicate stability, earning potential, and financial literacy, all of which can affect one’s ability to pay back loans on time.

In addition, characteristics related to credit inquiries ("days90_310L", "days120_123L", "days180_256L") can give us insight into an applicant's recent credit inquiring behavior, which, if too many inquiries are made in a short period, may signal potential financial instability.

Other characteristics such as "mobilephncnt_593L" (the number of people with the same cell phone number) and "max_birth_259D" (the maximum value of an individual's date of birth) can indirectly contribute to credit risk assessment by serving as proxies for identity verification and fraud prevention.

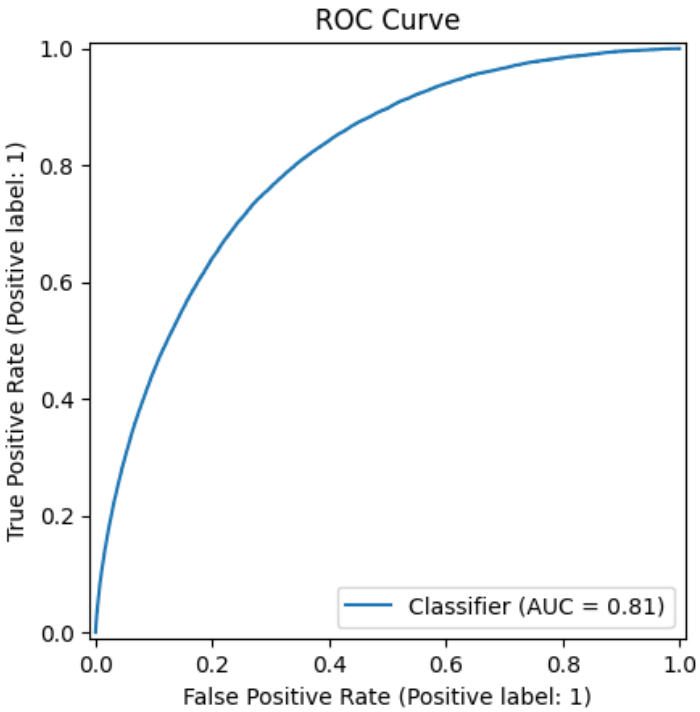
Overall, these characteristics are important for credit risk modeling because they provide valuable information about a borrower's financial behavior, stability, and ability to manage debt, as well as factors that may affect his or her creditworthiness and likelihood of default.

Conclusion

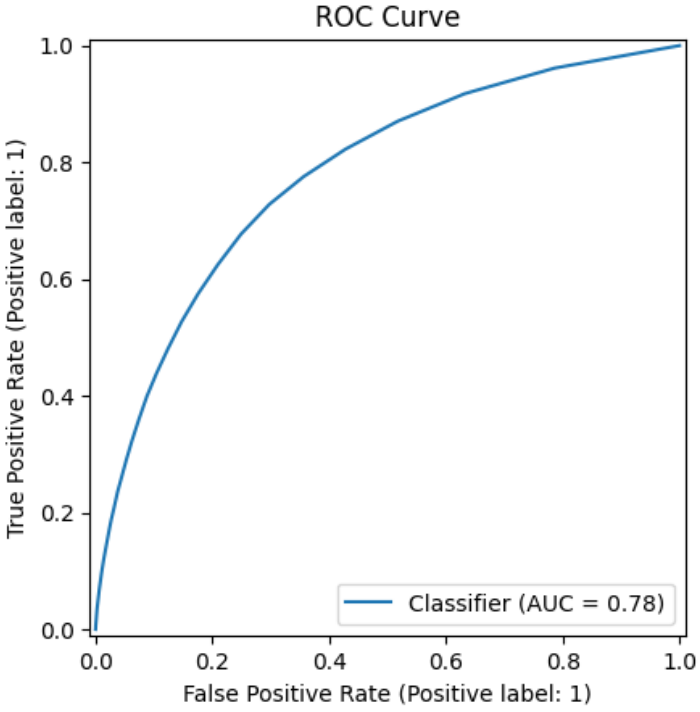
Through the employment of machine learning techniques such as Logistic Regression, Random Forest, and XGBoost, coupled with rigorous data preprocessing and feature selection methods like Boruta, we have developed predictive models that are both robust and effective in predicting loan defaults. The models are evaluated using metrics such as AUC, TPR, and Gini stability, revealing that the XGBoost model outperforms others in achieving higher AUC with high relative stability. The model also reveals that features that hold the most influence in the prediction process are ones related to creditors' previous loan activities and promptness in fulfilling their obligations. By enabling more accurate predictions of the likelihood of loan defaults, financial institutions can offer credit to a greater population, especially groups with low to no credit history, thereby promoting equity in financial services.

Appendix

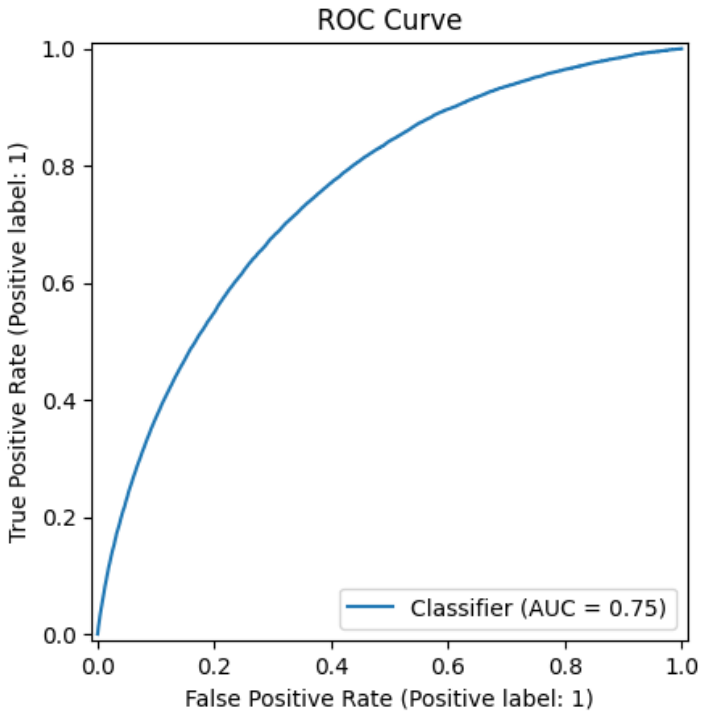
XGBoost Model



Random Forest Model



Logistic Regression Model



Dep. Variable:	y	**No. Observations:**	985809
Model:	Logit	**Df Residuals:**	985707
Method:	MLE	**Df Model:**	101
converged:	True	**Pseudo R-squ.:**	0.1408
Covariance Type:	nonrobust	**Log-Likelihood:**	-476310
		LL-Null:	-554350
		LLR p-value:	0

Table 6: Summary of Logistic Regression Fit

Predictor	coef	std err	z	P>|z|	[0.025	0.975]
const	-2.5044	0.304	-8.236	0	-3.1	-1.908
num__median_dpdmax_139P	0.1019	0.004	22.7	0	0.093	0.111
num__max_birth_259D	0.4685	0.006	79.652	0	0.457	0.48
num__mean_pmts_overdue_1140A	-0.0461	0.007	-6.601	0	-0.06	-0.032
num__disbursedcredamount_1113A	-0.0785	0.007	-10.844	0	-0.093	-0.064
num__last_mainoccupationinc_437A	-0.0268	0.003	-8.691	0	-0.033	-0.021
num__mean_pmts_dpd_1073P	-0.1185	0.006	-21.369	0	-0.129	-0.108
num__max_dateofcredend_289D	-0.0885	0.004	-24.446	0	-0.096	-0.081
num__price_1097A	0.0349	0.003	11.152	0	0.029	0.041
num__maxdebt4_972A	-0.0962	0.004	-23.442	0	-0.104	-0.088
num__secondquarter_766L	0.0093	0.004	2.661	0.008	0.002	0.016
num__max_firstnonzeroinstldate_307D	-0.0159	0.003	-5.04	0	-0.022	-0.01
num__maxdpdtolerance_374P	0.119	0.004	32.559	0	0.112	0.126
num__mean_annuity_853A	-0.0143	0.004	-3.596	0	-0.022	-0.006
num__fourthquarter_440L	-0.0114	0.004	-3.198	0.001	-0.018	-0.004
num__min_credamount_590A	-0.029	0.003	-8.31	0	-0.036	-0.022
num__maxdpdlast24m_143P	0.0131	0.004	2.958	0.003	0.004	0.022
num__max_totaloutstanddebtvalue_39A	-0.0825	0.01	-8.032	0	-0.103	-0.062
num__numrejects9m_859L	0.0663	0.003	20.109	0	0.06	0.073
num__applicationscnt_867L	0.018	0.004	4.811	0	0.011	0.025
num__min_overdueamountmaxdateyear_2T	0.1429	0.004	32.852	0	0.134	0.151
num__mean_overdueamountmax_155A	0.0559	0.007	8.351	0	0.043	0.069
num__max_numberofcontrsvalue_358L	-0.1088	0.003	-32.086	0	-0.115	-0.102
num__days120_123L	0.0424	0.006	7.545	0	0.031	0.053
num__last_firstnonzeroinstldate_307D	0.0557	0.004	12.875	0	0.047	0.064
num__max_pmtnum_8L	0.1133	0.003	36.065	0	0.107	0.119
num__days360_512L	0.0916	0.005	18.114	0	0.082	0.102
num__min_pmtnum_8L	0.0698	0.003	22.497	0	0.064	0.076
num__maxannuity_159A	0.1128	0.003	33.246	0	0.106	0.119
num__max_numberofoverdueinstlmax_1039L	0.1512	0.004	35.485	0	0.143	0.16
num__eir_270L	0.0987	0.003	35.337	0	0.093	0.104
num__max_mainoccupationinc_384A	-0.0072	0.003	-2.531	0.011	-0.013	-0.002
num__mobilephncnt_593L	0.3318	0.003	110.382	0	0.326	0.338
num__firstquarter_103L	0.0688	0.004	19.562	0	0.062	0.076

num__max_personindex_1023L	0.0319	0.004	8.168	0	0.024	0.04
num__thirdquarter_1082L	-0.0167	0.004	-4.715	0	-0.024	-0.01
num__min_dpdmaxdateyear_596T	0.2304	0.004	52.485	0	0.222	0.239
num__maxdpdfrom6mto36m_3546853P	0.0267	0.004	6.098	0	0.018	0.035
num__median_monthlyinstlamount_332A	-0.0158	0.007	-2.435	0.015	-0.029	-0.003
num__max_dpdmaxdateyear_596T	0.1159	0.004	31.318	0	0.109	0.123
num__dateofbirth_337D	-0.1156	0.006	-20.237	0	-0.127	-0.104
num__maxdpdlast3m_392P	-0.0301	0.004	-8.046	0	-0.037	-0.023
num__currdebt_22A	0.2287	0.003	75.195	0	0.223	0.235
num__min_refreshdate_3813885D	-0.0164	0.005	-3.422	0.001	-0.026	-0.007
num__annuity_780A	0.1081	0.006	18.981	0	0.097	0.119
num__pmtnum_254L	0.1057	0.004	27.495	0	0.098	0.113
num__totalsettled_863A	-0.3079	0.005	-56.496	0	-0.319	-0.297
num__min_pmts_year_1139T	-0.114	0.004	-30.507	0	-0.121	-0.107
num__min_dateofcredstart_739D	0.0557	0.004	15.803	0	0.049	0.063
num__min_dateofcredend_289D	-0.0266	0.003	-8.806	0	-0.033	-0.021
num__mean_credacc_credlmt_575A	-0.072	0.003	-23.854	0	-0.078	-0.066
num__maxdpdlast6m_474P	0.0244	0.004	6.211	0	0.017	0.032
num__credamount_770A	0.1602	0.008	19.092	0	0.144	0.177
num__max_num_group2	-0.0255	0.003	-8.138	0	-0.032	-0.019
num__max_dateofcredstart_739D	-0.0654	0.003	-19.454	0	-0.072	-0.059
num__homephncnt_628L	-0.0573	0.003	-16.49	0	-0.064	-0.051
num__min_firstnonzeroinstldate_307D	0.0199	0.004	4.51	0	0.011	0.029
num__max_annuity_853A	-0.0134	0.004	-3.593	0	-0.021	-0.006
num__days180_256L	0.0425	0.005	7.831	0	0.032	0.053
num__min_totaloutstanddebtvalue_39A	-0.0271	0.006	-4.634	0	-0.039	-0.016
num__days90_310L	0.0496	0.005	9.754	0	0.04	0.06
num__days30_165L	0.0936	0.003	28.614	0	0.087	0.1
cat__min_language1_981M_P209_127_106	-0.0244	0.005	-4.456	0	-0.035	-0.014
cat__min_language1_981M_a55475b1	-0.0936	0.018	-5.262	0	-0.128	-0.059
cat__description_5085714M_a55475b1	0.7202	0.012	59.213	0	0.696	0.744
cat__max_language1_981M_a55475b1	-0.128	0.014	-8.855	0	-0.156	-0.1
cat__max_postype_4733339M_P149_40_170	0.9326	0.299	3.119	0.002	0.347	1.519
cat__max_postype_4733339M_P169_115_83	1.4795	0.308	4.802	0	0.876	2.083
cat__max_postype_4733339M_P177_117_192	0.9101	0.298	3.054	0.002	0.326	1.494

cat__max_postype_4733339M_P217_110_186	1.0132	0.3	3.374	0.001	0.425	1.602
cat__max_postype_4733339M_P46_145_78	1.1356	0.298	3.811	0	0.552	1.72
cat__max_postype_4733339M_P60_146_156	1.0487	0.298	3.519	0	0.465	1.633
cat__max_postype_4733339M_P67_102_161	1.092	0.298	3.664	0	0.508	1.676
cat__max_postype_4733339M_a55475b1	1.0345	0.298	3.473	0.001	0.451	1.618
cat__min_education_1138M_P33_146_175	-0.1294	0.012	-10.976	0	-0.152	-0.106
cat__min_education_1138M_P97_36_170	0.0336	0.011	3.096	0.002	0.012	0.055
cat__max_education_1138M_P157_18_172	1.0938	0.191	5.713	0	0.719	1.469
cat__max_education_1138M_P17_36_170	0.4075	0.089	4.565	0	0.233	0.583
cat__max_education_1138M_P33_146_175	0.1839	0.059	3.118	0.002	0.068	0.3
cat__max_education_1138M_P97_36_170	0.412	0.058	7.066	0	0.298	0.526
cat__max_education_1138M_a55475b1	0.4464	0.058	7.673	0	0.332	0.56
cat__max_classificationofcontr_13M_ea6782cc	-0.3656	0.008	-45.447	0	-0.381	-0.35
cat__education_1103M_6b2ae0fa	-0.4212	0.014	-31.07	0	-0.448	-0.395
cat__education_1103M_717ddd49	-0.3012	0.015	-19.607	0	-0.331	-0.271
cat__education_1103M_a55475b1	-0.2369	0.014	-16.341	0	-0.265	-0.209
cat__education_1103M_c8e1a1d0	-0.1824	0.043	-4.244	0	-0.267	-0.098
cat__maritalst_385M_38c061ee	0.1821	0.02	9.19	0	0.143	0.221
cat__maritalst_385M_a55475b1	0.0474	0.009	5.555	0	0.031	0.064
cat__maritalst_385M_a7fcb6e5	0.1371	0.008	16.518	0	0.121	0.153
cat__maritalst_385M_b6cabe76	0.2066	0.015	13.549	0	0.177	0.237
cat__maritalst_385M_ecd83604	0.4801	0.04	12.051	0	0.402	0.558
cat__min_education_927M_P33_146_175	-0.4668	0.008	-55.86	0	-0.483	-0.45
cat__min_education_927M_a55475b1	-0.0971	0.015	-6.314	0	-0.127	-0.067
cat__max_subjectrole_93M_ab3c25cf	-0.1706	0.01	-16.318	0	-0.191	-0.15
cat__max_subjectrole_93M_be4fd70b	-0.153	0.023	-6.556	0	-0.199	-0.107
cat__max_subjectrole_93M_daf49a8a	-0.3219	0.029	-11.141	0	-0.379	-0.265
cat__max_collaterals_typeofguarante_359M_f4d8a027	-0.3194	0.067	-4.756	0	-0.451	-0.188
cat__last_education_1138M_P33_146_175	-0.065	0.02	-3.275	0.001	-0.104	-0.026
cat__last_education_1138M_P97_36_170	0.0857	0.019	4.529	0	0.049	0.123
cat__last_education_1138M_a55475b1	0.1601	0.019	8.227	0	0.122	0.198
cat__min_collater_typofvalofguarant_407M_8fd95e4b	-0.0719	0.009	-8.29	0	-0.089	-0.055

cat__min_collater_typofvalofguarant_407M_9a0c095e	0.1203	0.01	12.537	0	0.101	0.139

Table 7: Logistic Regression Predictor Summary

Bibliography

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic Minority Over-sampling Technique. *Journal of Artificial Intelligence Research*, 16, 321-357.

Dash, Manoranjan, and Huan Liu. "Feature selection for classification." *Intelligent data analysis* 1.1-4 (1997): 131-156.

Emmanuel, T., Maupong, T., Mpoeleng, D., Semong, T., Mphago, B., & Tabona, O. (2021). A survey on missing data in machine learning. Journal of Big Data, 8(1). https://doi.org/10.1186/s40537-021-00516-9.

Kursa, M. B., & Rudnicki, W. R. (2010). Feature Selection with the Boruta Package. *Journal of Statistical Software*, *36*(11), 1–13. https://doi.org/10.18637/jss.v036.i11.

Saharovskiy, S. (2024) *Home credit - credit risk model stability*, *Kaggle*. Available at: https://www.kaggle.com/competitions/home-credit-credit-risk-model-stability/discussion/473950 (Accessed: 09 February 2024).

Wang, Yuelin, et al. "A Comparative Assessment of Credit Risk Model Based on Machine Learning --a Case Study of Bank Loan Data." *Procedia Computer Science*, Elsevier, 27 July 2020, www.sciencedirect.com/science/article/pii/S1877050920315830.