Machine Learning Models for Loan Classification Using Real World Data

Natalya Sheremetyeva[1]

[1]*natalya.sheremetyeva@gmail.com*

(Dated: September 24, 2024)

Abstract

In this project, I developed a machine learning model for loan performance classification using XGBoost trained on real-world data from Freddie Mac's Single-Family Loan-Level dataset. Key features, such as Current Loan Delinquency Status (CLDS) and the here-engineered Estimated Loan-to-Value (ELTV) feature, emerged in the top ten most important features for the model's predictions based on SHAP values. The model achieved an average ROC AUC score of 0.94 on the unseen test set, indicating strong predictive ability. Class imbalance was addressed using undersampling techniques, improving the focus on predicting non-performing loans. While the model demonstrated robust performance, with an 82.9% profit advantage over a random baseline from 2014-2017, performance declined slightly in later years, highlighting the need for periodic retraining to maintain accuracy in evolving market conditions.

I. INTRODUCTION

To maintain the stability of financial institutions, lenders need to manage risk and allocate capital efficiently. A crucial part of a bank's credit risk assessment and valuation process is loan classification. Loan classification involves predicting the likelihood of a loan performing well *vs.* defaulting. By distinguishing between high-risk and low-risk loans, lenders can make informed decisions on loan approvals, interest rates, and credit limits. Additionally, robust loan classification models help in complying with regulatory requirements and mitigating financial losses. The ability to predict loan outcomes with high accuracy not only enhances the profitability of lending institutions but also contributes to overall economic stability.

In today's economic climate, accurate loan classification is more critical than ever. With fluctuating interest rates, inflation concerns, and ongoing economic uncertainties, financial institutions face significant challenges in managing credit risk. The current economic conditions echo past financial crises, such as the Great Recession of 2008, which underscored the consequences of inadequate risk management practices. More recently, the collapse of Silicon Valley Bank in 2023 and the bank run on First Republic Bank, which led to its buyout by JPMorgan, further highlighted the importance of robust risk assessment tools. These events underscore the necessity for advanced predictive models to safeguard the stability of financial institutions and the broader economy.

The primary objectives of this project are threefold. First, to train a machine learning

classifier that accurately predicts whether a loan will continue to perform well or become non-performing based on its origination and monthly performance data. Second, to optimize the performance of this classifier through hyperparameter tuning and model selection techniques. Lastly, to identify the key features that significantly influence the model's accuracy, thereby providing valuable insights into the factors driving loan outcomes. These objectives aim to contribute to more informed decision-making processes in the financial industry, ultimately enhancing risk management and economic stability.

This report is organized as follows: First, I discuss the design of the target variable and its connection to loan performance. Second, I present key performance metrics of the model, followed by an analysis of feature interpretability and stability across datasets. A significant portion of the report details the technical steps in the model development process, found in Sec.A. Additionally, concise explanations of key technical terms are provided in Sec.B.

II. TARGET ENGINEERING

The goal of this project is to predict whether a loan will perform well or become nonperforming within the following two years from any given monthly observation point. For the purposes of this study, a loan is classified as nonperforming if it reaches a delinquency status of 4 months or more (CLDS = 4). This approach leverages the Current Loan Delinquency Status (CLDS) reported monthly for each loan in the dataset.

Initially, the Zero Balance Code (ZBC) was considered as a potential target variable. The ZBC indicates the final outcome of a loan, such as whether it matured (was paid off) or defaulted. However, the ZBC is only populated once a loan's balance reaches zero, which occurs at the end of the loan's life. This makes ZBC impractical for predicting loan outcomes in a timely manner, as it does not provide early warning signals during the loan's active lifecycle.

To address this limitation, CLDS was used as a stand-in for default. An analysis of loans with ZBC values reported between 1999 and 2005 revealed that CLDS is highly correlated with loan performance, providing insight into a loan's delinquency trajectory before its balance reaches zero. The statistics in Fig. 1 summarize the relationship between CLDS and loan outcomes. These statistics reveal that the separation between matured and defaulted loans becomes more apparent as the CLDS increases, with CLDS = 4 providing the best

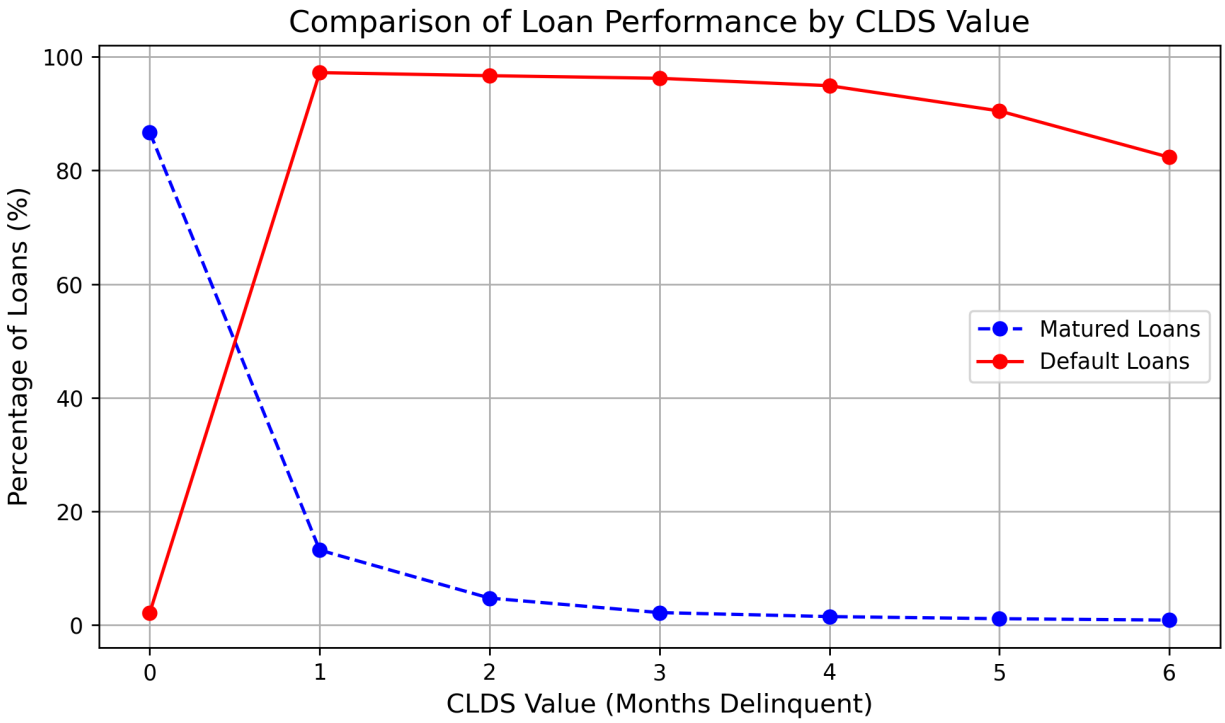

FIG. 1: The percentage of matured and default loans across different CLDS values (months delinquent). The CLDS=4 value offers a good separation between matured and default loans, as the gap between the two classes becomes substantial at that point, making it an effective threshold for defining non-performing loans.

distinction. By the time a loan reaches CLDS = 4 (four months delinquent), 94.9% of the defaulted loans have crossed this threshold, while only 1.5% of matured loans have done so. Higher CLDS values, such as CLDS = 5 and CLDS = 6, also show a clear distinction, but they encompass a smaller fraction of both defaulted and matured loans, making CLDS = 4 a more inclusive and actionable cutoff point for predicting defaults.

Given this strong correlation between CLDS and default outcomes, CLDS = 4 was chosen as a proxy for default because it effectively separates defaulting loans from those that will mature, while minimizing the risk of misclassifying matured loans.

The two-year prediction window was selected for several reasons. First, it provides an actionable timeframe for lenders to make decisions based on current and historical loan performance. A shorter prediction window would not allow sufficient time for loans to accumulate enough history to exhibit meaningful patterns of performance or delinquency. Conversely, a longer window might dilute the accuracy of predictions due to changing eco-

nomic conditions over time. The two-year period strikes a balance between capturing early risk signals and making actionable predictions, while still allowing enough performance data to be collected for a given loan.

In summary, using CLDS = 4 as a proxy for default within a two-year window provides a practical and effective means of predicting loan outcomes, helping lenders manage credit risk more proactively.

III. RESULTS AND DISCUSSION

Takeaway: An XGBoost classifier was trained on historical loan data from 1999 to 2011. To optimize model performance, a wide hyperparameter space was explored using Bayesian optimization to maximize the ROC AUC score on the tuning data set from 2012-2013. After identifying the optimal hyperparameters, the model was fit on the training set.

Next, the model's performance was evaluated on a hypothetical loan portfolio from the tuning set. Different probability thresholds were tested to determine the cutoff yielding the maximum portfolio profit. A threshold of 0.37 was found to be optimal (see Sec. A 8 for details), as it provided the best balance between false positives and false negatives. The maximum profit achieved by the model was 8.3% below that of a hypothetical "ideal model" with perfect knowledge of outcomes and 84.6% above a randomly guessing model.

Thus, the final model combines optimal hyperparameters and decision threshold, as determined using the tuning set. In the following, I will discuss the model's performance on the unseen test data, as well as the most important features driving its predictions. Finally, I will examine the model's stability against potential changes in data distribution over time.

A. Model Performance

Table I summarizes the model's performance as measured by the ROC AUC score. On the training set, an ROC AUC score of 0.999 was achieved, indicating that the model almost perfectly distinguishes between performing and non-performing loans during this period. While a high score on the training set is expected, a near-perfect score can indicate potential overfitting. On the tuning set (2012-2013), the ROC AUC dropped somewhat to 0.967. This decrease underscores that while the model generalizes well, some minor overfitting to the

Data sets	Training	Validation	Out-of-Time (Test)				
Year	1999-2011	2012-2013	2014	2015	2016	2017	2018
ROC AUC	0.999	0.967	0.948	0.946	0.933	0.931	0.804

TABLE I: ROC AUC scores of the final model evaluated across different datasets. The training set includes data from 1999 to 2011, while the validation (tuning) set consists of observations from 2012 to 2013. The test set, comprising unseen data, evaluates model performance from 2014 to 2018, with a noticeable decline in performance beginning in 2018.

training data may be present.

Performance on the unseen test data (2014-2018) shows a gradual decline in ROC AUC scores over time. In 2014 and 2015, the model maintained strong predictive ability, with scores of 0.948 and 0.946, respectively. However, beginning in 2016, the ROC AUC steadily decreased to 0.933 and 0.931 in 2017, reflecting a slight reduction in the model's ability to generalize as economic and market conditions may have shifted. By 2018, the ROC AUC dropped more significantly to 0.804, indicating that the model's ability to distinguish between performing and non-performing loans was notably weaker in more recent years, likely due to changing patterns in the loan market that were not captured in the training data. This suggests the need for periodic retraining or feature adjustments to maintain high model performance.

In addition, the model's performance is evaluated by applying it to a hypothetical loan portfolio and measuring the resulting profit. That metric can then be compared to the profit resulting from an ideal model with perfect knowledge, and a random model that guesses based on the class distribution in the training dataset. The profit calculation formulas are provided in Sec. A 7. The profit for an ideal model with perfect foresight is calculated as the sum of the interest earned on all correctly identified performing loans. This assumes no losses from non-performing loans since preemptive action can be taken. In contrast, the random model is a baseline model that makes predictions based on the true class proportions of the loans in the training set, randomly assigning each loan to either the performing or non-performing class. The gain from the random model is calculated by summing the interest earned from loans that are randomly predicted to be performing and actually perform well (true negatives), and subtracting the losses from loans that are randomly predicted to be

performing but actually are not (false negatives).

This analysis provides elucidates the model's performance in terms of financial returns, which are more relevant to business decision-making than traditional metrics like ROC AUC scores.

The percentage below the ideal model is calculated using the following equation:

$$\text{Below ideal model [\%]} = \frac{\text{Total} - \text{Ideal Model}}{\text{Ideal Model}} \times 100$$

The percentage above the random model is given by:

$$\text{Above random model [\%]} = \frac{\text{Total} - \text{Random Total}}{\text{Ideal Model} - \text{Random Total}} \times 100$$

Table II shows the resulting percentages. Compared to the ideal model, the trained model performs consistently well. During the training period (1999-2011), the model is only 1.2% below the ideal and 8.3% below the ideal for the tuning set period (2012-2013). Between 2014 and 2017, the model remains within 6-8% of the ideal, showing strong performance across these years. In the most recent period (2018), however, the trained model is 15.4% below, indicating a decline in predictive accuracy as market conditions evolve.

In comparison to the random model, the trained model shows a substantial improvement. It outperforms the random model by 97.9% in the training period and maintains an advantage of 81-85% in the test periods from 2014 to 2017. However, for 2018, the gap narrows to 64.2%, reflecting a decrease in model performance, similar to the comparison with the ideal model. Overall, the trained model significantly outperforms the random approach.

To sum up, the trained model demonstrates strong performance across all periods, but its predictive accuracy decreases as the test set years progress, with the largest drop occurring in the most recent period. This decline is expected, as market conditions and loan characteristics tend to evolve over time, making it more challenging for a model trained on historical data to accurately predict future outcomes. These results highlight the need for periodic retraining of the model to ensure it remains up-to-date with the latest data, allowing it to make more accurate predictions in the near future.

Next, I will examine the most important features the model uses to make predictions. In addition, understanding how features change over time will help explain the diminishing predictive power as time advances away from the training and validation period.

Data sets	Training	Validation	Out-of-Time (Test)				
Year	1999-2011	2012-2013	2014	2015	2016	2017	2018
Below ideal model [%]	-1.2	-8.3	-7.7	-6.7	-6.6	-5.5	-15.4
Above random model [%]	97.9	84.6	81.5	82.7	82.6	84.7	64.2

TABLE II: Performance comparison of the trained model relative to an ideal model and a random model across different time periods, based on calculated financial profit (see Sec. A 7. The "Below ideal model" row shows the percentage by which the trained model's profits fall short of the ideal model, while the "Above random model" row displays the percentage by which the trained model outperforms the random model. The calculations are based on the data from the training, tuning, and test sets, with the ideal model assuming perfect foresight and the random model making predictions based on true class proportions in the training dataset.

B. Interpretability

Interpretability is crucial in machine learning (ML), especially in domains like finance, where understanding the reasoning behind a model's predictions is as important as its performance. Interpretable models allow stakeholders to trust and act on the model's outputs by providing transparency into which features drive predictions. While machine learning models like XGBoost are powerful and complex, their black-box nature can make it difficult to understand their decision-making process directly.

To address this, various techniques can be used to interpret ML models. SHAP (SHapley Additive exPlanations) values are one of the most widely adopted methods for explaining complex models like XGBoost. SHAP values assign each feature a contribution value for a particular prediction, helping to identify how much each feature influences the model's output. They provide a consistent and mathematically sound way to explain feature importance at both the individual prediction level and the global model level.

The sum of all SHAP values for a given observation represents the model's predicted probability for that observation to fall into one of the target classes. In the context of binary classification, the SHAP values of each feature reflect how much they contribute to pushing the prediction toward class 0 (perfroming loan) or class 1 (non-perfroming loan).

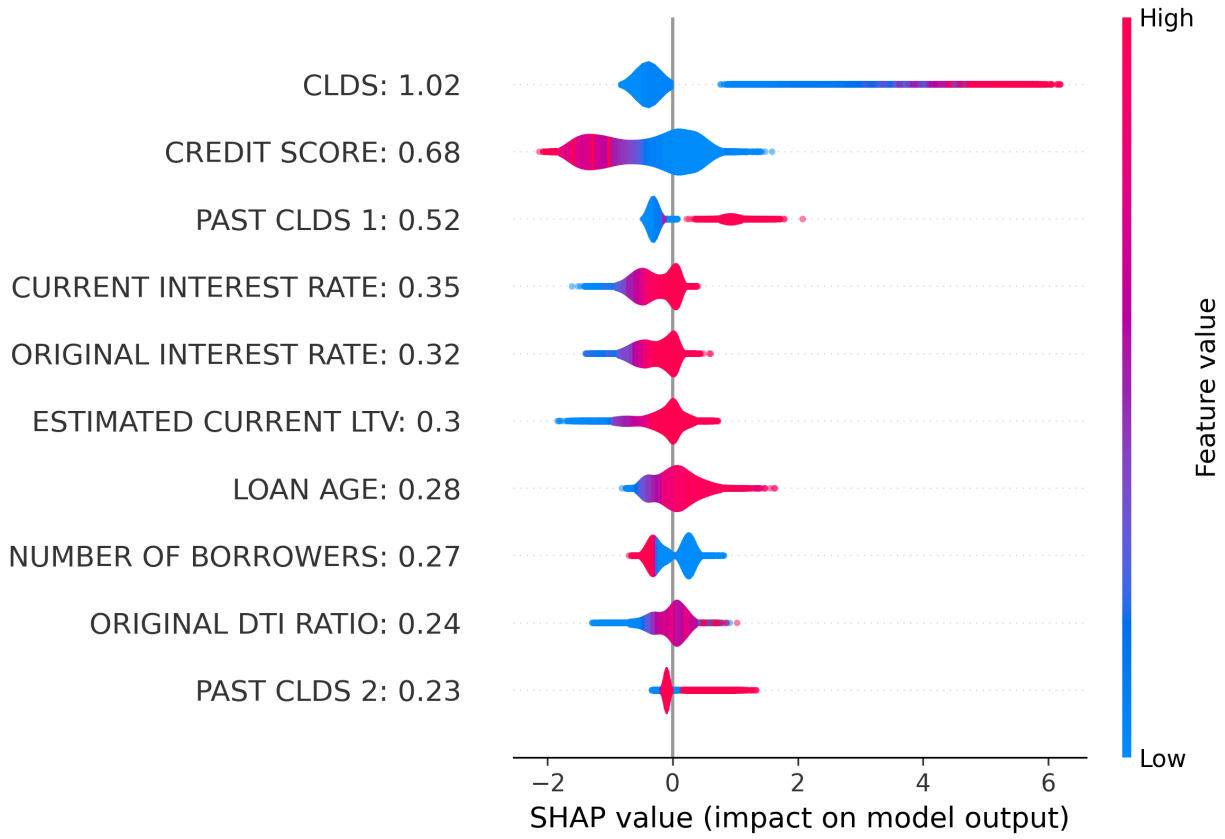

FIG. 2: SHAP summary plot for the top ten most important features influencing the model's predictions. The mean absolute SHAP values (next to each feature's name) quantify the average impact of each feature on the model output. Red represents higher feature values, while blue represents lower values. The x-axis shows SHAP values, indicating the contribution of each feature to the prediction of either class 0 (performing loan) or class 1 (non-perfroming loan). For instance, higher Credit Score values push predictions toward class 0 (negative SHAP values), while lower credit scores push toward class 1 (positive SHAP values). Features like CLDS, Current Interest Rate, and Original Interest Rate show a strong tendency toward the non-performing class when their values are high.

When these individual contributions are summed together, along with the model's baseline or expected value, the result is the predicted probability for the observation to belong to either class.

Figure 2 shows the SHAP summary plot and visualizes the top ten most important features that the model used to make predictions on the tuning set. The features are

ranked from top to bottom in descending order of their average absolute SHAP values, with Current Loan Delinquency Status (CLDS) being the most influential feature, followed by CREDIT SCORE, PAST CLDS 1 (indicating whether there was a CLDS=1 reported in the loan's history before the current reporting period), CURRENT INTEREST RATE, and ORIGINAL INTEREST RATE.

The number next to each feature name (e.g., 1.02 for CLDS) represents that feature's mean absolute SHAP value, which quantifies its average impact on the model's output. In this case, CLDS has the highest mean SHAP value (1.02). In contrast, PAST CLDS 2 has a lower mean SHAP value (0.23), meaning its influence is smaller.

Each point represents an individual loan from the tuning set, and its position along the x-axis shows how much the feature influenced that particular prediction. Higher feature values are represented in red, while lower values are shown in blue.

CLDS is the most influential feature with a mean SHAP value of 1.02. Higher values of CLDS (red) are strongly associated with positive SHAP values, indicating a tendency toward non-performing loan class (class 1), while lower values (blue) contribute less to the prediction.

Credit Score (mean SHAP value: 0.68) shows a clear pattern where higher credit scores (red) contribute negatively, moving the model's prediction toward the performing-loan-class (class 0), while lower scores (blue) push the prediction toward non-performing-loan-class (class 1).

PAST CLDS 1 and CURRENT INTEREST RATE also have meaningful impacts, with higher values generally pushing predictions toward non-performing-loan-class. The SHAP values for these features are less spread than for CLDS and Credit Score, indicating more consistency in their influence. Original Interest Rate, Estimated Current LTV, and Loan Age contribute less on average but still show a clear pattern where higher values (red) increase the probability of non-performing-loan-class, while lower values (blue) contribute to the performing-loan-class class. Number of Borrowers, Original DTI Ratio, and PAST CLDS 2 also influence the predictions, with similar trends—higher values leaning toward the non-performing-loan-class, and lower values favoring the performing-loan-class.

In this analysis, I utilized most of the features reported by Freddie Mac (see Sec. **??** for a list of all features used). However, before deploying any model in production, it typically must undergo review to ensure compliance with fair lending rules and regulations.

Fair lending refers to the legal requirements and ethical standards aimed at preventing discrimination in credit decisions based on protected characteristics, such as race, gender, marital status, or age. These rules ensure that all borrowers have equal access to credit and that no feature used in the model unfairly disadvantages a particular group.

For example, a feature like Number of Borrowers could potentially introduce bias by discriminating against single borrowers, who might receive less favorable predictions compared to multiple borrowers. As a result, this feature would likely need to be removed or adjusted to avoid any unfair impact and to comply with fair lending practices. This ensures that the model remains transparent, unbiased, and legally compliant.

C. Stability over time

In the earlier sections, we have seen that the model's performance declines over time, and we have also established the top ten important features for the model to make its predictions. In this section, I will examine the Population Stability Index (PSI) and Characteristic Stability Index (CSI). These indices measure the degree of change/shift in the distribution of the dataset over time, helping to assess whether a model's input data remains stable and reliable for prediction. While both these indices are calculated using the same formula (see Sec. B), the difference between the two is that the PSI is evaluated on the model's target or the model's performance score compared between the training and the test set, while the CSI is evaluated for specific features (characteristics).

Figure 3 shows the PSI for the target variable, comparing the distribution of the training set (1999-2011) to the tuning set (2012-2013) and test sets (2014-2018), separated by observation year. Typically, a PSI value below 0.1 indicates minimal change, values between 0.1 and 0.25 suggest moderate change and values above 0.25 indicate significant shifts that may warrant model recalibration. During the tuning set period (2012-2013) and early test set period (2014), the PSI remains very low, indicating that the target variable's distribution closely matches the training data. This suggests the model's performance remains stable, as it encounters data similar to what it was trained on.

Starting in 2015, the PSI gradually increases, peaking in 2017 at just under 0.2, indicating a moderate shift in the target variable's distribution. This suggests the model encounters data that diverges more from the training set, potentially leading to performance degra-

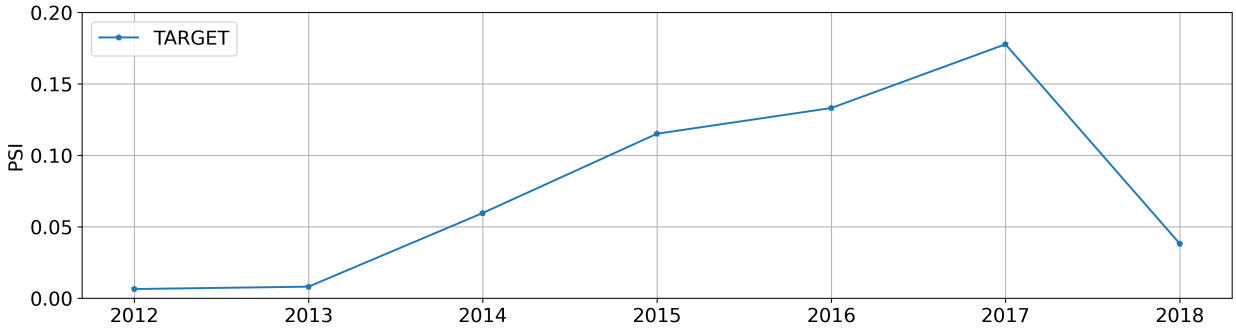

FIG. 3: Population Stability Index (PSI) for the target variable, comparing the distribution of the training set with the tuning set (2012-2013) and test sets (2014-2018).

dation. In 2018, the PSI drops to just below 0.05, still above the values from 2012-2014, suggesting a slight shift remains but with a more stable distribution.

Figure 4 shows the Characteristic Stability Index (CSI) for the top 10 most important features, as determined by SHAP values, comparing how each feature's distribution has changed from the training set to the tuning and test sets (2012-2018). Higher CSI values indicate more significant shifts.

Six of the top ten most important features, including the most important feature CLDS, remain stable over time with CSI values below 0.15. The CSI for CLDS shows a decreasing trend, while the Number of Borrowers feature remains constant. The Current Interest Rate exhibits an upward trend, which aligns with evolving economic conditions.

The Credit Score at Origination, the second most important feature, has a CSI below 0.25 from 2013 to 2016 but reaches 0.25 or above in 2017 and 2018, indicating a shift in its distribution. The Loan Age feature has a CSI consistently above 0.25, peaking at 0.35 in 2018, though its lower importance (7th) makes these shifts less critical.

The largest CSI values, above 0.5, are seen in Original DTI Ratio and Original Interest Rate. The Original DTI Ratio maintains CSI values above 1.5, peaking at nearly 3.5 in 2014, while Original Interest Rate shows a nearly linear increase from just above 0.5 to over 3.5 in 2018. While the large CSI for Original DTI Ratio is notable, the lower importance of this feature (9th) makes its impact on the model less significant. In fact, the model performed well in 2014 despite this shift. However, Original Interest Rate, the 4th most important feature, shows a steady rise in CSI, likely contributing to the model's diminished performance in 2018. As the Original Interest Rate is influenced by external economic factors, such as

Federal Reserve policies, these shifts highlight areas where the model's performance may degrade, reinforcing the need for monitoring and retraining.

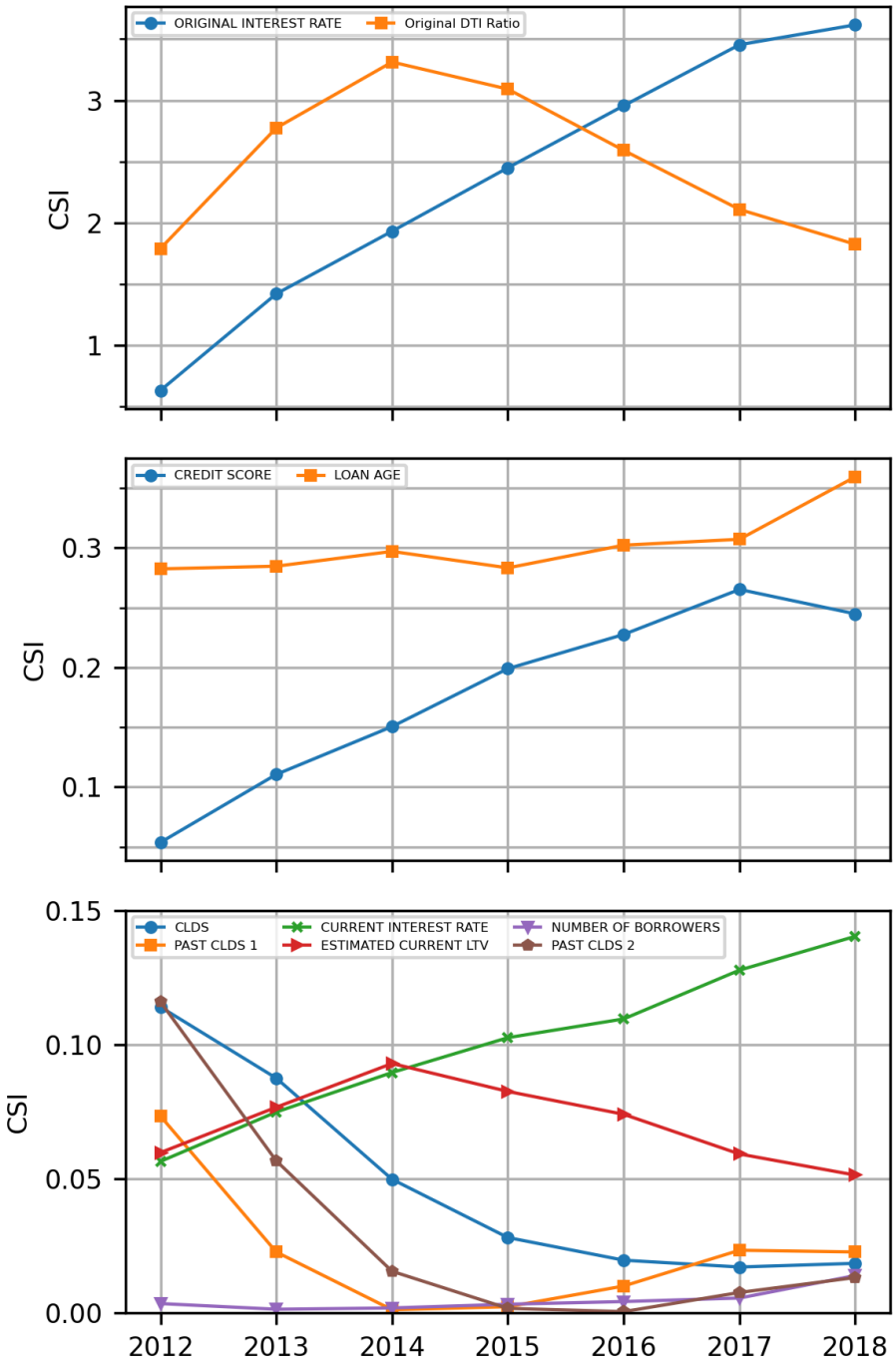

FIG. 4: Characteristic Stability Index (CSI) for the top 10 most important features, determined by SHAP values, comparing their distributional shifts from the training set to the tuning and test sets (2012-2018).

This analysis emphasizes the need for continuous monitoring of the data and regular retraining of the model.

IV. CONCLUSIONS & OUTLOOK

The trained model demonstrated strong performance in predicting loan performance, outperforming a random baseline by an average of 82.9% over the four years (2014-2017) following the time period included in the training and tuning dataset. The performance decline for later years highlights the need for periodic retraining to ensure the model stays aligned with changing market conditions. Incorporating additional external features, such as the Shiller Home Price Index, could enhance predictive power. At the same time, features like the Number of Borrowers may need to be removed to comply with fair lending regulations and prevent potential bias.

In summary, the model offers valuable insights into loan performance, but maintaining its relevance will require regular updates and compliance reviews.

Appendix A: Technical Details

1. Data Source

Freddie Mac, officially the Federal Home Loan Mortgage Corporation, is a government-sponsored enterprise (GSE) established in 1970 to expand the secondary mortgage market in the U.S. It does this by purchasing mortgages from lenders, bundling them into mortgage-backed securities (MBS), and selling them to investors, promoting homeownership and ensuring liquidity, stability, and affordability in the housing market.

The dataset for this project comes from Freddie Mac's publicly available Single-Family Loan-Level dataset, which includes detailed information on loan origination, performance, and borrower characteristics. Spanning from 1999 to the present, this dataset provides a comprehensive resource for analyzing and modeling loan performance, supporting improved risk assessment and management practices.

2. Dataset Preprocessing and Cleaning

The Freddie Mac dataset consists of two files for each loan: an origination file and a monthly performance file, organized by loan origination year (1999-2024). The origination file includes 32 features, such as the Loan Sequence Number (LSN), a unique identifier for each loan. The performance file also contains 32 columns, starting with the LSN, allowing for tracking each loan over time.

Each column was reviewed for value ranges, categories, and missing values. The dataset contains a mix of numeric, alphabetic, alphanumeric, and date-based fields, such as the first payment date and the monthly reporting period. To make the data usable for machine learning algorithms, which require numeric input, categorical variables were transformed using one-hot and ordinal encoding. One-hot encoding was applied to columns like Loan Purpose, converting distinct categories into binary columns. Ordinal encoding was used for columns like Current Loan Delinquency Status (CLDS), where certain values, like 'R', 'C', and 'XX', were mapped to numerical equivalents.

Missing values were filled with placeholders outside the typical value range to maintain dataset completeness. Columns that were incomplete for the majority of loans or redundant (e.g., derived date fields) were excluded from the training set. Table III lists all the features used in this work.

3. Target Engineering

This project's goal is to predict whether a loan will perform well or become nonperforming in the following two years from a given on monthly observation point. Here, nonperforming is defined as a loan reaching a delinquency status of 4 months or more. Initially, the Zero Balance Code (ZBC), which indicates the reason a loan's balance reaches zero, was considered as a target variable. However, ZBC is populated only at the end of a loan's life, making it impractical to predict loan outcomes within a timely window.

To construct a more actionable target, the Current Loan Delinquency Status (CLDS) from the monthly performance data file was used. A loan is labeled as nonperforming (class 1) if it reaches a delinquency status of 4 months or more within the following 24 months, and performing otherwise (class 0). This two-year prediction window is based on the observation

Feature Name	**Feature Name**
CREDIT SCORE	SELLER NAME
FIRST TIME HOMEBUYER FLAG	SERVICER NAME
MSA CODE	SUPER CONFORMING FLAG
MI PERCENTAGE	PROGRAM INDICATOR
NUMBER OF UNITS	RELIEF REFINANCE INDICATOR
OCCUPANCY STATUS	PROPERTY VALUATION METHOD
ORIGINAL CLTV RATIO	INTEREST ONLY INDICATOR
ORIGINAL DTI RATIO	MI CANCELLATION INDICATOR
ORIGINAL UPB	ORIGINAL VALUE
ORIGINAL LTV RATIO	CURRENT ACTUAL UPB
ORIGINAL INTEREST RATE	CLDS
CHANNEL	LOAN AGE
PPM FLAG	REMAINING MONTHS
AMORTIZATION TYPE	MODIFICATION FLAG
PROPERTY STATE	CURRENT INTEREST RATE
PROPERTY TYPE	CURRENT NON-INTEREST BEARING UPB
POSTAL CODE	STEP MODIFICATION FLAG
LOAN PURPOSE	PAYMENT DEFERRAL
ORIGINAL LOAN TERM	MODIFIED INTEREST BEARING UPB
NUMBER OF BORROWERS	ESTIMATED CURRENT LTV
PAST CLDS 1	PAST CLDS 2
PAST CLDS 3	TARGET

TABLE III: List of Features.

that loans typically reach default status within 80 months on average, whereas loans that are paid off tend to do so in about 48 months. By setting the prediction window to 24 months, we strike a balance between capturing early risk signals and making actionable predictions, while allowing for sufficient performance history to be collected for a given loan.

4. Additional Features Engineering

In addition to the features provided in the Freddie Mac dataset, several features were engineered in this work to enhance the model's ability to predict nonperforming loans.

1. **CLDS History**: For each loan, three additional features were created to record whether the loan had previously been delinquent by one, two, or three months before the current observation.

2. **Estimated Loan-to-Value (ELTV)**: Freddie Mac's dataset only provides ELTV data (based on the estimated current value of the property obtained through Freddie Mac's Automated Valuation Model (AVM)) starting in April 2017. To maintain consistency across the dataset and use this important feature for all periods, a crude approximation of the current ELTV for all loans was calculated. First, the "original value" of the property was calculated based on the provided Unpaid Principal Balance (UPB) and Loan-to-Value (LTV) ratio at origination, using the formula:

$$\text{Original Value} = \frac{\text{UPB}}{\text{LTV}}.$$

Then, the ELTV was estimated by assuming the current property value remains constant and is given by the original value. Thus, the estimated ELTV for any period is calculated as:

$$\text{ELTV} = \frac{\text{Current Actual UPB}}{\text{Original Value}}.$$

This is a rough approximation, as property values can change significantly over time. More sophisticated calculations could incorporate external data, such as the *Shiller Home Price Index* or other economic indicators. However, for the sake of conducting a controlled experiment, the present approach was limited to insights extracted solely from the Freddie Mac dataset without using external inputs.

5. Class Imbalance

Class imbalance occurs when one class in a dataset is significantly more frequent than the other. In binary classification, this often means that the majority class dominates, leading ML models to favor it, resulting in high overall accuracy but poor performance in predicting

the minority class, which is often the more critical outcome (e.g., predicting nonperforming loans).

Here, the original class distribution was highly imbalanced, with 4% nonperforming loans and 96% performing loans in the training and tuning sets, and an even more skewed 2% vs. 98% in the test years. To address this, I applied an undersampling technique, retaining only 10% of the majority class (performing loans). This reduced the training set from 16,200,436 to 2,201,796 samples, and the tuning set from 4,128,251 to 559,853 samples. After undersampling, the class distributions shifted to 29.4% nonperforming loans and 70.6% performing loans in the training and tuning sets, and 16.5% vs. 83.5% in the test set. This adjustment balances the dataset, allowing the model to focus more effectively on identifying patterns related to nonperforming loans while still retaining sufficient data on performing loans.

The model trained in this way may still perform well in detecting nonperforming loans in datasets with the original class proportions, but it could lead to a higher rate of false positives (predicting non-performing for loans that will perform).

6. Hyperparameter tuning

Hyperparameter tuning was performed using a Bayesian optimization search with 500 initial exploration points selected at random in the specified ranges for hyperparameters, and 100 additional optimization steps.

The following hyperparameter ranges were explored:

```
# Define a range of hyperparameters to search
param_bounds = {
'n_estimators': (500, 1000),  # Number of trees
'learning_rate': (0.01, 0.1),  # Learning rate
'max_depth': (2, 18),  # Maximum depth of trees
'gamma': (0., 5.),  # Regularization term for tree split
'subsample': (0.45, 1.0),  # Fraction of samples used for fitting trees
'colsample_bytree': (0.45, 1.0),  # Fraction of features for tree construction
'reg_alpha': (0., 5.),  # L1 regularization term
'reg_lambda': (0., 5.),  # L2 regularization term
```

}

Note that an additional hyperparameter called `min_child_weight` (Minimum sum of instance weight in a child) was set to its default value of 1, found to be optimal in preliminary testing.

7. Portfolio Profits and Losses Evaluation

The following describes how gains and losses are calculated from a portfolio of loans based on the model's predictions.

a. Fixed Monthly Payment Calculation

The fixed monthly payment for a loan is calculated using a standard mortgage formula. Given the loan's principal balance (P_0), the annual interest rate (r), and the loan term in months (n), the formula for the monthly payment is:

$$\text{Monthly Payment} = \frac{P_0 \times r_{\text{month}} \times (1 + r_{\text{month}})^n}{(1 + r_{\text{month}})^n - 1}$$

Where $r_{\text{month}} = \frac{r}{12 \times 100}$, which converts the annual interest rate into a monthly rate expressed as a fraction.

b. Outstanding Loan Balance After a Given Number of Payments

To calculate the remaining balance on a loan after m monthly payments have been made, the following formula is used:

$$\text{Remaining Balance} = \frac{P_0 \times ((1 + r_{\text{month}})^n - (1 + r_{\text{month}})^m)}{(1 + r_{\text{month}})^n - 1}$$

This calculation takes into account how much of the loan has been paid off and how much remains based on the number of payments made and the loan's original terms.

c. Interest Gained on Performing Loans

For loans that are perfroming, the interest gained is calculated over a fixed period of 24 months. The total gain from such a loan is the sum of the 24 monthly payments minus the difference between the current unpaid balance (UPB) and the balance remaining after 24 months. This difference accounts for the fact that monthly payments include both interest and principal, so the calculation reflects the actual interest earned over the 24-month period.

The gain from a performing loan is expressed as:

$$\text{Interest Gain} = (\text{Monthly Payment} \times 24) - (\text{Current UPB} - \text{UPB after 24 months})$$

This allows the calculation of the interest the lender would have earned from a loan if it is performing as expected.

d. Loss on Non-performing Loans

For loans that were predicted to be performing but ended up non-performing (false negatives), the assumption is that the lender loses 15% of the loan's remaining unpaid balance (UPB). This simplified approach assumes a fixed loss rate.

The loss is calculated as:

$$\text{Loss} = 0.15 \times \text{Current UPB}$$

This loss only applies to loans that the model incorrectly predicted to be performing (false negatives) but actually were non-performing.

e. Total Gain/Loss Calculation

The total financial result for the portfolio is computed by summing the interest gains from the loans predicted to be performing and subtracting the losses from the non-performing loans (false negatives). The overall calculation is:

$$\text{Total Portfolio Gain} = \sum(\text{Interest Gains}) - \sum(\text{Losses on Non-performing loans})$$

This approach helps evaluate the financial impact of the model's predictions in terms of gains from correctly predicted performing loans and losses from incorrectly predicted non-performing loans. The model's performance can then be measured by the net financial gain or loss.

8. Threshold tuning

Using the gain calculation described in the previous section, the decision threshold of the model (the default value being 0.5) can be tuned to maximize the overall financial gain. Figure 5 shows the calculated gain as a function of the selected probability threshold for the model to classify loans as either performing or non-performing. The gain function follows an inverted parabolic shape, with the maximum gain occurring at a threshold of 0.37.

A threshold of 0.37, lower than the default 0.5, indicates that the model is more conservative in predicting loans to be performing. By lowering the threshold, the model classifies more loans as potential non-performers, which reduces the number of false negatives (loans incorrectly predicted to be performing but that default) at the expense of increasing false positives (loans incorrectly predicted to default but actually performing). This adjustment is beneficial in situations where the financial impact of missed non-performers is higher than the cost of false alarms, thus maximizing overall portfolio gain.

For reference, Fig. 5 shows the gain of an ideal model with perfect foresight. If such a model were possible, its gains would be 8.3% higher than those achieved by the current model with the optimized probability threshold.

Appendix B: Glossary

XGBoost Classifier: Machine learning algorithm based on gradient boosting designed for speed and performance. It builds an ensemble of decision trees sequentially, where each tree corrects the errors made by the previous ones. XGBoost is often chosen for its efficiency and its regularization techniques that prevent overfitting. The model outputs probabilities for classification tasks, which can be interpreted as the likelihood of an event occurring, such as whether a loan will default.

Hyperparameters: parameters of a machine learning model that are not learned from

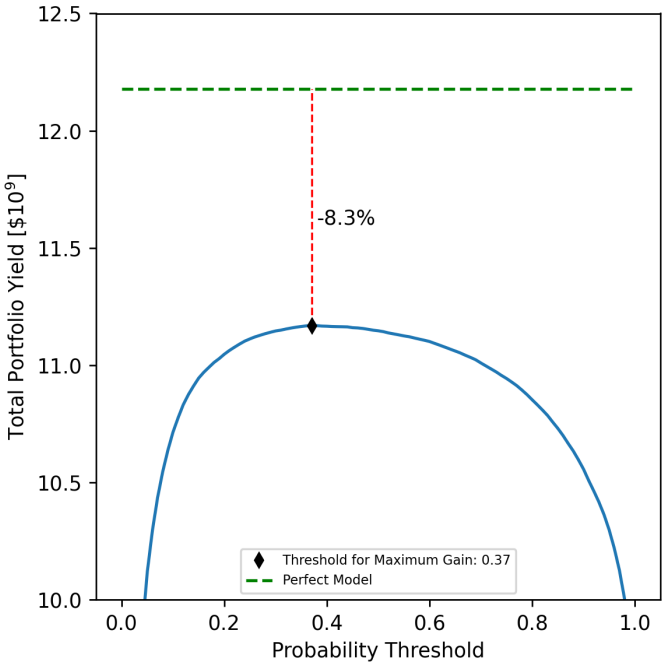

FIG. 5: Financial Gain from a hypothetical loan portfolio based on the tuning set data (2012-2013) as a function of the probability threshold. The maximum portfolio gain is achieved at a threshold of 0.37, which is lower than the default value of 0.5. The green dashed line indicates the portfolio gain that would be provided by an ideal model with perfect foresight, if such a model existed.

the data, but set before the training process begins. For an XGBoost classifier, key hyperparameters include the learning rate (controls how quickly the model adjusts), max depth (the maximum depth of each decision tree), and subsample (the fraction of the training data used to build each tree). A separate tuning set (also called validation set) is used to avoid overfitting to the training data.

Bayesian Optimization: a method used to optimize the hyperparameters of a model by building a probabilistic model of the objective function and using it to choose the most promising hyperparameter values. It works by initially sampling hyperparameters at random, and then iteratively choosing hyperparameters based on prior results to focus on the most promising regions of the hyperparameter space. Key technical features to set include the number of initial random steps, the acquisition function (which balances exploration and exploitation), and the number of optimization steps. Unlike grid search, which exhaustively tests every combination of hyperparameters, Bayesian optimization efficiently narrows down

the search space, leading to faster convergence and more effective tuning.

Receiver Operating Characteristic Area Under the Curve (ROC AUC) score: a performance metric for binary classification models. It measures how well a model distinguishes between classes across all possible thresholds. A higher ROC AUC indicates better performance, with a score of 1.0 representing perfect classification and 0.5 representing random guessing. Unlike accuracy, which is sensitive to class imbalance, ROC AUC provides a more comprehensive view of model performance. It is particularly useful when the costs of false positives and false negatives differ. A large ROC AUC means that the model can effectively rank predictions by the probability of class membership.

Threshold Tuning: the process of selecting the probability cutoff point that determines whether a predicted probability is classified as a positive or negative class. For binary classification models like XGBoost, the default threshold is often 0.5, but this may not always be optimal, especially if the classes are imbalanced or the costs of false positives and false negatives differ. By adjusting the threshold, one can optimize for different objectives, such as maximizing precision, recall, or portfolio gain. For example, in loan perfromance prediction, tuning the threshold to maximize portfolio gain ensures the model provides the best possible financial outcome for the business.

Feature Importance: contribution of each feature in making predictions in a model. In XGBoost, feature importance measures how much each feature improves the model's decision-making when splitting data in the decision trees. It provides interpretability from understanding which features are most influential in the model's predictions. Having interpretability is important because it builds trust in the model, allows for better communication of results, and provides insights that can drive decision-making. Understanding feature importance is particularly valuable in highly regulated industries, like finance, where transparency is required.

SHapley Additive exPlanations values (SHAP) Values: a method for explaining the output of machine learning models. They provide insight into how much each feature contributes to a particular prediction. SHAP values quantify the impact of each feature on the prediction for each individual instance, rather than in aggregate. SHAP values ensure that the contribution of each feature is calculated in a fair, consistent manner, following principles from cooperative game theory.

Population Stability Index (PSI): a metric used to measure the shift in the distri-

bution of a variable between two different datasets, typically a training dataset and a more recent dataset. It is commonly used to monitor the stability of a model's input features or predictions over time, ensuring that the model continues to perform well as the underlying data distribution changes. PSI is computed by dividing the data into bins and comparing the proportion of data in each bin for both the original and new dataset. The PSI formula is:

$$PSI = \sum \left((\text{Original\%} - \text{New\%}) \times \ln \left(\frac{\text{Original\%}}{\text{New\%}} \right) \right)$$

A PSI value below 0.1 indicates minimal change, a value between 0.1 and 0.25 suggests moderate change, and a value above 0.25 signals a significant population shift, potentially requiring model recalibration.

Characteristic Stability Index (CSI): similar to PSI but is specifically used to track the stability of individual features (characteristics) over time. CSI measures whether the distribution of a particular feature in the dataset has shifted between the training period and a more recent dataset. Like PSI, CSI is calculated by binning the values of the feature, comparing the proportions between the original and new datasets, and using the same formula as PSI. CSI is particularly useful for identifying features whose distribution has changed significantly, which can degrade model performance and may require retraining or recalibration.

To emphasize the difference between PSI and CSI, PSI focuses on the model's predictions or risk scores to assess whether there is population drift in the model's predictions over time or across datasets, while CSI looks at individual features to detect drift in the variables used for predictions.

Appendix C: Acknowledgments

Useful discussions with and peer-review by Michael Zlotnikov are gratefully acknowledged.