POLITECNICO
MILANO 1863

Financial Engineering 2024/25

A Neural Network Approach for the Estimation of Mortgage Prepayment Rates

Andrea Catina, Ansh Kharbanda
Group 5A
5th June, 2025

Abstract

Prepayment risk in mortgage portfolios, especially under fixed-rate contracts, poses a significant challenge in interest rate risk and asset-liability management. This project replicates and implements the neural network model presented in "A Neural Network Approach for the Estimation of Mortgage Prepayment Rates" (Baccaglini et al., 2021). We construct the network using fixed-rate mortgage data, integrate both contractual and market features, and compare forecasting performance against logistic regression benchmarks. The findings demonstrate the enhanced predictive power and flexibility of the neural network in modeling client behavior.

Contents

1 Introduction

Prepayment modeling plays a vital role in the management and valuation of mortgage-related financial instruments. The decision of borrowers to repay their loans early, whether driven by interest rate changes, refinancing opportunities, or individual financial circumstances, can significantly affect the cash flows and risk profile of mortgage-backed securities and other fixed-income products.

Understanding and predicting prepayment behavior is therefore crucial for financial institutions, investors, and analysts. It enables more accurate pricing, improved risk management, and better strategic decision-making. Traditional models have relied on statistical techniques or domain-specific rules, but recent advancements in machine learning offer promising alternatives that can capture complex, non-linear relationships within the data.

This project explores data-driven approaches to prepayment prediction using both traditional and modern machine learning models. By leveraging historical loan data, we aim to develop predictive models that can forecast prepayment behavior with improved accuracy. The remainder of this report outlines the methodology used including model selection, training & testing strategy and comparison of model performance.

We replicate the Feedforward Neural network(FNN) framework to estimate CPR for fixed-rate mortgages, focusing on:

- Implementation of the sequential NN with He initialization and Adam/Adagrad optimizers.
- Feature engineering using contractual, time and market variables (e.g., IRS rates, Euribor etc),
- Performance comparison against logistic regression.

The code is structured to align with Sections 3 and Appendix A of the original paper, with validation on provided datasets.

2 Modeling Early Repayments for Mortgages

2.1 The Prepayment Phenomenon and Estimation Factors

Mortgage prepayment occurs when borrowers repay all or part of their loan ahead of the scheduled maturity. This behavior is influenced by a combination of financial incentives and personal circumstances. Lower prevailing interest rates often encourage borrowers to refinance existing mortgages, while other non-financial factors such as relocation or changes in income can also motivate prepayments.
From a modeling perspective, identifying relevant drivers is essential. These include:

- **Contractual features** such as initial and current outstanding loan amounts, interest rate, and time to maturity.
- **Time-related variables** including the age of the loan and seasonal patterns in prepayment activity.
- **Market conditions**, particularly interest rate dynamics, such as the spread between current market rates and contract rates.

- **Behavioral factors** which account for idiosyncratic borrower decisions not directly linked to financial indicators.

Understanding these dimensions allows for more accurate forecasting of cash flows and better risk management.

2.2 Conditional Prepayment Rate (CPR)

The Conditional Prepayment Rate (CPR) is a key measure in mortgage modeling that quantifies the proportion of loans expected to be prepaid in a given time period, typically one month. It serves as the primary target variable in many prepayment risk models.
Formally, it is defined as:

$$\mathrm{CPR}_t = \frac{V\mathrm{prep}, t}{V_{\mathrm{out},t}}$$

where $V_{\mathrm{prep},t}$ is the volume of loans prepaid in month t and $V_{\mathrm{out},t}$ is the total outstanding loan balance at the beginning of t.
To facilitate long-term modeling and comparison, the CPR can be annualized. For instance, the annualized CPR for a constant monthly prepayment rate is computed as:

$$\text{Annualized CPR} = 1 - (1 - \mathrm{CPR}_t)^{12}$$

This transformation allows institutions to estimate and manage long-term risks in their mortgage portfolios by accounting for cumulative prepayments.

3 Dataset Description

The dataset employed for this analysis consists of fixed-rate mortgage loans, each observed over time through monthly snapshots. Each loan is characterized by the following variables:

- **Current Outstanding Amount**: The remaining principal balance to be repaid as of the observation date.
- **Initial Outstanding Amount**: The original loan amount disbursed by the bank at contract inception.
- **Contract Full Rate**: The fixed interest rate applied to the mortgage, defined at origination.
- **Contract Spread Rate**: In this dataset, this value is always zero, as only fixed-rate mortgages are considered.
- **Distribution Channel**: The originating institution or bank through which the loan was issued.
- **Flag Belonging**: A binary variable indicating customer loyalty (`Y` for loyal clients, `X` otherwise).
- **Reference Date**: The date of observation used to evaluate whether a prepayment event occurred.
- **Time from Contract Origination (months)**: The elapsed time, in months, since the loan's inception.

- **Original Maturity (months)**: The total contractual term of the loan in months.
- **Residual Maturity (months)**: The remaining duration, in months, until contractual maturity at the observation date.
- **Prepayment Flag**: A categorical variable indicating the type of event observed. Values are interpreted as follows:
  - 0: No event
  - 1: Total repayment
  - 2: Mortgage subrogation
  - 3: Partial repayment
  - 4: Extra-contractual renegotiation
  - 5: Renegotiation of the TAEG rate

  Only flags **1, 2 & 3** are considered as genuine prepayment events, as they entail a change in cash flows.
- **Volume Associated to an Event**: In the presence of a prepayment event, this variable tracks the amount of principal repaid at the observation date.
- **Disbursement Month**: The date of the original contract signing and loan issuance.

To enrich the feature set with macroeconomic variables, the analysis also incorporates a secondary dataset containing historical market interest rates. Specifically, this includes EURIBOR rates for 1, 3, and 12-month maturities, as well as IRS swap rates for 10, 15, 20, 25, and 30-year tenors.

4 Feature Engineering

The set of features used in this study replicates the selection strategy adopted in the reference literature. In particular, the input variables were grouped into three categories: contractual features, market features and time feature.

4.1 Contractual Features

The following loan-specific contractual attributes were included:

1. **Current Outstanding Amount**: The residual principal balance at the observation date.
2. **Initial Outstanding Amount**: The original amount disbursed at contract inception.
3. **Time from Contract Origination**: The number of months elapsed since loan origination.
4. **Contract Full Rate**: The fixed interest rate agreed upon at the signing of the loan.
5. **Original Maturity**: The contractual term of the mortgage, expressed in months.

4.2 Market Features

To incorporate macroeconomic dynamics and monetary conditions, four market-related variables were added:

1. **Full Rate on New Fixed-Rate Contracts**: The volume-weighted average of contract rates for new fixed-rate mortgages originated during the previous month.
2. **3-Month EURIBOR Rate**: The monthly average of 3-month EURIBOR fixings observed in the previous month.
3. **IRS Rates**: Monthly averages of Interest Rate Swap (IRS) fixings for tenors of 10, 15, 20, 25 and 30 years, computed over the last month.
4. **Deviation from Historical Minimum of IRS**: The difference between the current IRS rate and the historical minimum observed prior to the last 12 months. This feature accounts for client reaction delays to market changes.

4.3 Time Feature

Seasonality effects were modeled using a binary indicator variable equal to 1 if the observation month is August and 0 otherwise. This adjustment reflects the empirically observed drop in prepayment activity during holiday periods.

4.4 Target Variable

The binary target variable indicates the occurrence of a prepayment event. It assumes value 1 when the prepayment flag (`flg_oper`) corresponds to one of the following values: `1` (Total repayment), `2` (Subrogation) or `3` (Partial repayment). All other values are mapped to 0, indicating no prepayment.

5 Neural Network Approach

A feedforward neural network was implemented following a sequential architecture to model the prepayment binary flags. The model learns complex nonlinear relationships between input features and the prepayment indicator by stacking fully connected layers, each receiving input from all neurons in the previous layer.

5.1 Model Architecture

The network consists of an input layer matching the number of features, followed by a variable number of hidden dense layers with ReLU activations. The number of layers and neurons per layer were subject to hyperparameter tuning, allowing the model complexity to adapt to the data. The final output layer contains a single neuron with a sigmoid activation to predict the probability of prepayment.

Weights were initialized using the He normal initializer to promote stable gradient flow during training. The model was compiled with the Adam optimizer, and learning rate was also optimized via tuning.

5.2 Custom Volume-Weighted Loss Function

A custom volume-weighted binary cross-entropy loss was implemented to account for the differing financial impact of contracts based on their outstanding volumes. Each sample's contribution to the loss is weighted proportionally to its loan outstanding amount, ensuring the model prioritizes accuracy on economically significant contracts.

5.3 Data Preparation and Temporal(Vertical) Splitting

The dataset was split chronologically to avoid look-ahead bias, respecting the temporal order of contracts. The first 80% of observations formed the training set, the next 10% validation and the last 10% test set. Missing values were imputed using the training set means and the features were standardized accordingly. The target variable was binarized from multi-class flags into a binary indicator.

5.4 Hyperparameter Tuning with KerasTuner Hyperband

We use automated hyperparameter tuning using Keras Tuner's Hyperband algorithm to systematically identify the optimal neural network configuration based on validation loss. The tuning procedure explored different numbers of hidden layers (1 to 3), neurons per layer (32 to 128 in steps of 32), and learning rates (0.01, 0.001, 0.0005), while using ReLU activation for hidden layers and sigmoid activation for the output layer. Hyperband's adaptive resource allocation and early stopping enhanced efficiency by focusing training on the most promising configurations and avoiding overfitting. Although a more exhaustive approach, tuning additional parameters like batch size and activation functions, was initially tested, it was discarded due to high computational costs and minimal performance improvements. The final strategy balances exploration breadth with computational feasibility, enabling reproducibility and reducing manual bias.

5.5 Model Training and Evaluation

The best model from tuning was then evaluated on the test set. Predictions were then generated for the entire dataset. For benchmarking, a logistic regression model was trained on the same input features and evaluated similarly.

5.6 Monthly Aggregation and Residuals Analysis

Predictions and actual prepayment flags were aggregated by calendar month to assess temporal alignment. Residuals (actual minus predicted) were analyzed over time and via histograms to evaluate model bias and error distribution. Both analyses indicated that the neural network consistently outperformed logistic regression, with smaller, more symmetric residuals and fewer systematic errors.

This methodological framework closely follows the code implementation, emphasizing the tuning process that enabled selecting a parsimonious but effective model architecture tailored to the dataset.

6 Results

To better replicate the reference paper approach the same types of graphic visualizations were applied

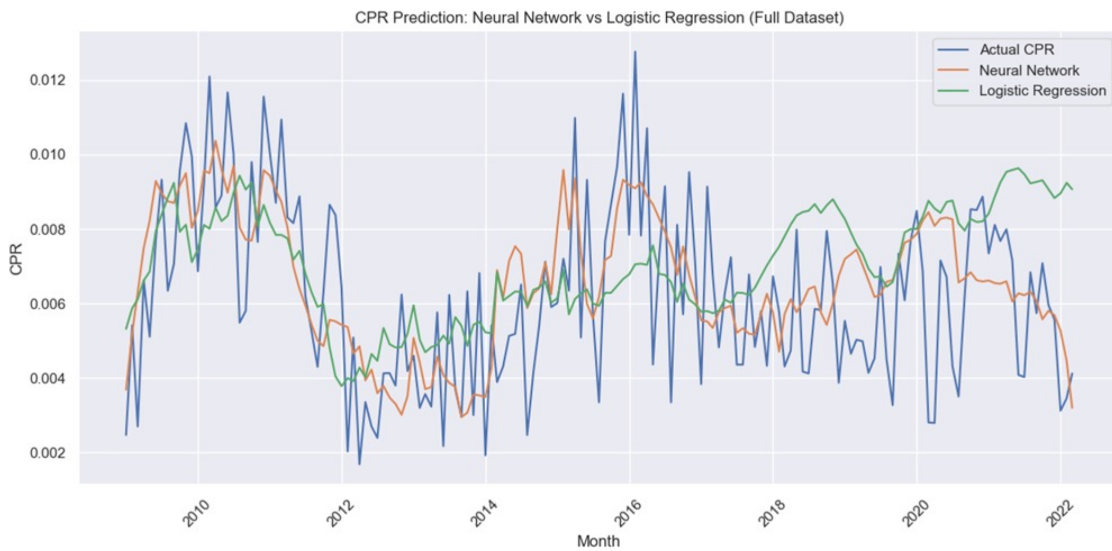

Figure 1: Monthly CPR: Actual vs Predicted Prepayment Rates.

The monthly aggregation plot (Figure 1) compares the actual prepayment rates (CPR) against the neural network predictions. The model closely follows the temporal trend of observed prepayments, capturing seasonality and market fluctuations better than a simple baseline.

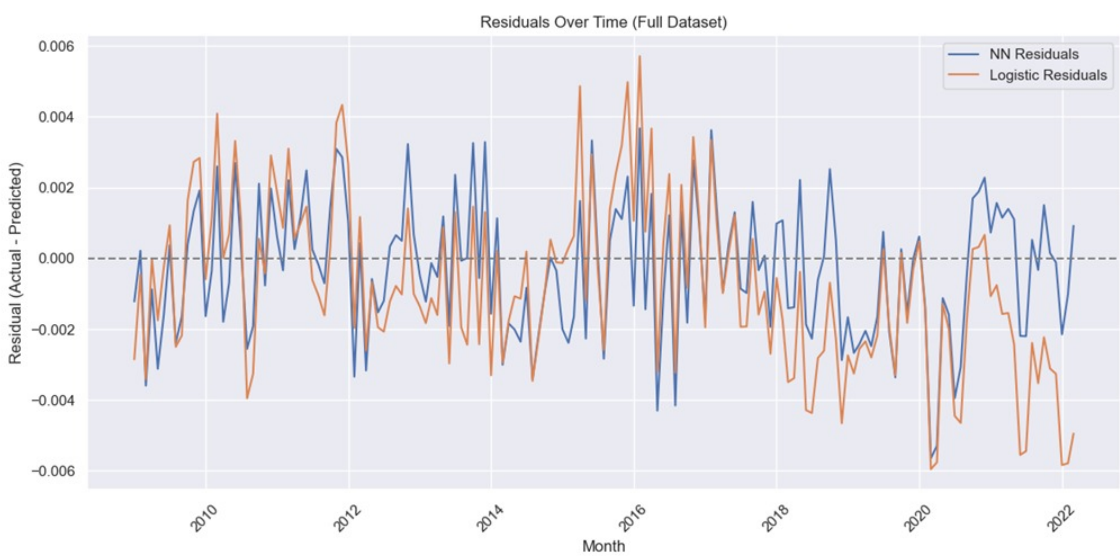

Figure 2: Monthly Residuals: Actual CPR minus Predicted CPR.

Figure 2 shows the residuals over time, highlighting periods where the model slightly under- or over-estimates prepayment. The residuals remain mostly centered around zero, indicating low bias and good temporal stability.

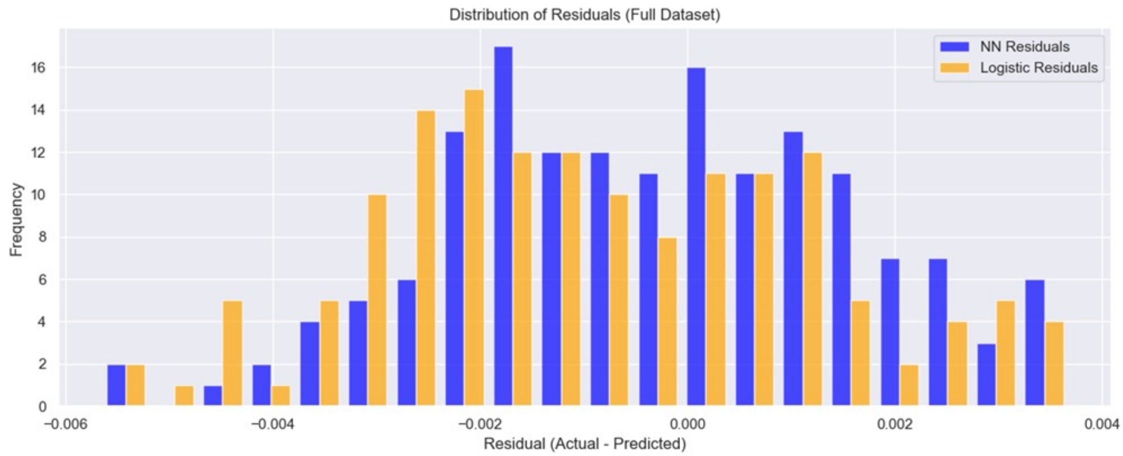

Figure 3: Histogram of Residuals: Neural Network vs Logistic Regression.

The residual distribution histogram in Figure 3 reveals that the neural network residuals are more tightly concentrated around zero with a symmetric spread, whereas logistic regression shows a wider and slightly skewed error distribution. This confirms the neural network's superior fit and lower systematic error.

Summary of Results:

- The neural network achieves better alignment with actual prepayment behavior across months compared to logistic regression.
- Residual analyses indicate reduced bias and more stable prediction errors for the neural network.
- The tuning process enabled selecting a model that balances complexity and generalization effectively.

Overall, the results confirm the effectiveness of the neural network approach combined with volume-weighted loss and hyperparameter tuning, improving predictive accuracy and robustness in prepayment modeling.

7 Considerations & Remarks

1. **Class Imbalance and Optimizer Choice**: The dataset is notably imbalanced, with far fewer prepayment events than non-prepayments. Oversampling was explored as a potential solution, however, the results did not show a clear improvement. Therefore, we opted to proceed with the original dataset and used the Adam optimizer due to its adaptive learning rates, which help the model converge faster and handle sparse gradient updates.
2. **Motivation for Volume-Weighted Loss**: Not all loans pose equal financial risk. Contracts with higher outstanding balances matter more. A volume-weighted binary cross-entropy loss was used to give higher importance to such loans, better aligning the model's learning with economic impact.
3. **Loss Function Equivalence**: The custom loss function `VolumeWeightedBCE` matches the paper's design intent. While the paper defines $\mathcal{L} = -\sum_c \bar{y}_c \log(z_c) \cdot u_c$ (sum over contracts), the code computes

$\mathcal{L} = \frac{1}{N}\sum_c \mathrm{BCE}(y_c, z_c)\cdot u_c$ (mean over batch)*(BCE stands for Binary Cross-Entropy)*. This difference in aggregation (sum vs. mean) introduces a constant scaling factor $\frac{1}{N}$ but preserves the relative volume-weighting mechanism ($u_c = V_{\text{out},c}/\sum V_{\text{out},c}$). Since optimization gradients depend only on relative weighting, the two formulations are functionally equivalent for training purposes.

4. **Splitting Strategies**: Chronological splitting simulates real-world forecasting, while stratified horizontal splitting tests generalizability. Both the methods were used but the final choice was to use the chronological split, since it is the one used in real financial applications.

5. **Role of Feature Scaling and Imputation**: To manage missing values and heterogeneous feature scales, mean imputation and standard scaling were applied. These steps stabilize neural network training by ensuring consistent input distribution and preventing issues like gradient instability.

6. **Code execution time**: Executing the code from scratch initiates a full hyperparameter optimization routine using the keras-tuner, which may require approximately 22 minutes to complete, as it systematically searches the hyperparameter-space rather than using predefined or manually selected values.

8 Conclusion

This project implemented and validated a deep learning approach for modeling fixed-rate mortgage prepayment behavior using a volume-weighted loss formulation. The neural network architecture successfully captured complex relationships between market and contract-level features, outperforming logistic regression in both classification accuracy and temporal prediction stability.

The findings underscore the potential of neural models in enhancing prepayment risk assessment, particularly when large-scale mortgage datasets are available. Future work could involve experimenting with recurrent or attention-based architectures to better model time dependencies and extending the analysis to floating-rate mortgage portfolios.

References

[1] Baccaglini, D., Lazzarin, M., & Savino, S. (2021). *A Neural Network Approach for the Estimation of Mortgage Prepayment Rates*.