Predictive modeling and interest rate forecasting for Lending Club's credit evaluation

Gobber Giacomo - Loero Elena - Lo Verde Giulio Surya

May 31, 2024

1 Problem statement

We are working with Lending Club (LC), an online peer-to-peer lending platform.

Our goal is to create a model that helps the company determine the probability that a current client, who has already been granted a loan, will default over a period of 31-120 days (from now on we will refer to it as PD). Understanding the PD is crucial for LC to proactively manage potential losses and take measures to mitigate them.

For clients in default, we are also interested in providing a model to predict the expected amount of money recovered. This allows us to assess the overall credit risk associated with LC's loan portfolio and encourages measures to mitigate potential losses. Accurately predicting recoveries highlights the importance of estimating future cash flows, ensuring financial solidity for all stakeholders.

Additionally, when a new client applies for a loan, it is crucial for LC to decide the appropriate interest rate to apply. To achieve this, we aim to forecast the Federal Reserve Bank (FED) interest rate for the upcoming months, which is crucial for several reasons. Firstly, it impacts both debtors and investors behaviour influencing borrowing trends and it also directly affects our institution's borrowing costs, guiding our loan interest rate decisions in order to maintain profitability. Lastly, FED rates are linked to broader economic factors like inflation, employment, and economic growth, aiding informed decision-making aligned with macroeconomic conditions.

2 Datasets

To develop the algorithm for the estimation of the PD and the expected recoveries, we used a cross-section dataset gathered from LC that includes 142 data variables, including those showed in Table 1. These informations have been collected from 2.925.493 clients of LC in the US.

Variable	Description	Type
debt_settlement_flag	Debt settlement indicator	Binary
default	Loan default indicator	Binary
fico_range_low	Lower bound of FICO score range	Integer
funded_amnt	Total amount funded	Real
int_rate	Interest rate of the loan	Real
installment	Monthly installment amount	Real
last_pymnt_amnt	Last payment amount received	Real
loan_amnt	Loan amount requested	Real
mort_acc	Number of mortgage accounts	Integer
num_bc_sats	Number of satisfactory bankcard accounts	Integer
num_bc_tl	Number of bankcard accounts	Integer
out_prncp	Remaining outstanding principal amount	Real
recoveries	Post charge-off recoveries	Real
revol_util	Revolving line utilization rate	Real
term	Loan term duration in months	Categorical
title	Purpose of the loan	Categorical
tot_cur_bal	Total current balance of all accounts	Real
total_pymnt	Total payments received	Real
total_rec_int	Total interest received	Real
total_rec_late_fee	Total late fees received	Real
total_rev_hi_lim	Total revolving high credit limit	Real
verification_status	Income verification status	Categorical

Table 1: Description of *Lending Club* dataset: the most relevant variables.

Initially, we removed variables with a significant amount of missing values and those that logically contained same informations as other existing variables. To verify our approach, we computed a correlation matrix to identify and confirm the redundancy. Then, we deleted outliers and observations with at least one missing value: this step helped in maintaining the integrity of our dataset.
We converted the target variable `loan_status`, used in modeling the PD, into a binary format because our analysis aims to predict whether a loan should be granted or not:

- A value of 1 was assigned to loan statuses indicating financial distress: *"Charged Off"*, *"Default"*, *"Does not meet the credit policy. Status: Charged Off"*, or *"Late (31-120 days)"*;
- Conversely, we assigned a value of 0 for all other loan statuses.

Through these steps, we prepared a clean and reliable dataset, which forms the foundation for our subsequent data analysis and predictive modeling efforts.

In the PD analysis, we divided the dataset into three equal parts: train, test, and validation. This enabled us to train the model, determine the optimal threshold for classifying defaulted loans and evaluate its performance on unseen data.

Recoveries represent the efforts made by creditors to recoup losses when a loan defaults. Given that, we utilized only a subset of the observations with `loan_status` modalities *"Charged Off"* and *"Does not meet the credit policy. Status: Charged Off"*. In this case we used a 70-30 split between training and test sets.

For our last objective, in order to predict FED rates we used a time series dataset obtained from the US Federal Reserve; it contains 836 monthly observations from July 1954 to February 2024. These interest rates, set by the Federal Reserve, central bank of the United States, represent the interest rate at which banks lend money to each other. Adjustments to these rates are aimed at stabilizing the economy by either stimulating growth or controlling inflation, depending on the prevailing economic conditions.

3 Methods

The primary objective in our classification model is to label borrowers as either default (1) or non-default (0). To achieve this, we first calculate the PD and then we classified clients into one of these two categories using a threshold value (the determination of the threshold is discussed in Section 4).

Given the binary nature of `default`, a Bernoulli random variable effectively represents this scenario. Let Y_i be a Bernoulli random variable describing the default event: the probability of default for the i-th individual is the conditional expected value $E(Y_i|X_i) = p_i$, where X_i is a vector of borrower and loan characteristics.
We explored three different prediction methods: linear model, logistic regression and probit model. Logistic and probit model are particularly suitable for binary outcomes because they provide bounded predictions between 0 and 1. The linear model, although simpler, may produce probabilities that fall outside this interval and thus don't respect the bounded support; however, it serves as a useful baseline for comparison purposes due to its straightforward interpretation and ease of implementation.
In this analysis, it was deemed appropriate to switch from the simplest model (linear model) to a more complex and less interpretable one (logit or probit) as it was believed to bring economic advantages by leading to improved risk assessment and cost-saving measures in lending decisions.

1. **Linear model** assumes a linear relationship between predictor variables and probability of default. Linear probability model can be expressed as:

$$E(Y_i|X_i) = X_i'\beta \tag{1}$$

   where β is the vector of coefficients.
   Linear regression is estimated using Ordinary Least Squares (OLS) method by using `lm` function in R.

2. **Logistic regression** is a widely used statistical method for binary classification problems and it uses logistic function as its link function. The model is defined as:

$$E(Y_i|X_i) = \frac{1}{1 + e^{-X_i'\beta}} \tag{2}$$

3. **Probit model** is another popular approach for binary classification. In this case, the probability of default is modeled using the cumulative distribution function of the standard normal distribution. Model equation is expressed as:

$$E(Y_i|X_i) = \Phi(X_i'\beta) \tag{3}$$

Both logit and probit models are estimated using Maximum Likelihood (ML) method via the `glm` function in R.

After addressing the classification task, we shifted our focus in predicting `recoveries`. Given that Y is defined only on $\mathbb{R}^+$, we employed two different approaches:

1. **Linear Model with Log Transformation**: we applied a logarithmic transformation to the `recoveries` variable, defining a new variable distributed over the entire real numbers ($\mathbb{R}$). This transformation is necessary as `recoveries` follows a right-skewed distribution. By transforming datas logarithmically, we make the distribution more symmetric and more appropriate to linear regression analysis. The model can be expressed as following:

$$\log(\text{Recoveries}) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_k X_k + \epsilon \tag{4}$$

Throughout our analysis, we experimented with various specifications of the model, discussed in Section 4 along with their results.

2. **GLM with Gamma Distribution and Log Link Function**: we used this method as Gamma distribution is defined only on $\mathbb{R}^+$. GLM uses a Link Function transforming $E(Y|X)$ to the linear predictor such that it constraint the linear predictor into the appropriate space. The general relation is $g(\alpha) = X_i'\beta$; in this specific case, the link function is:

$$g(\alpha) = log(\alpha) \tag{5}$$

and consequentially

$$\alpha = \exp(X_i'\beta) \tag{6}$$

Gamma distribution is parameterized by α and a dispersion parameter β. The

probability density function for a Gamma distribution is:

$$f(x;\alpha,\beta) = \frac{\Gamma(\alpha)}{\beta^{\alpha}} x^{\alpha-1} e^{-\beta x} \tag{7}$$

In addition to our classification and regression tasks, we forecasted **FED interest rate** for the next 12 months employing two popular time series models: ARIMA and GARCH. These models are well-suited for capturing temporal dependencies and volatility patterns present in financial time series data.

ARIMA model combines autoregressive, differencing and moving average components to capture underlying patterns in the data. The model equation of an ARIMA(p,d,q) can be expressed as:

$$\Delta^d Y_t = \beta_0 + \sum_{i=1}^{p} \beta_i \Delta^d Y_{t-i} + \varepsilon_t + \sum_{j=1}^{q} \theta_j \epsilon_{t-j} \tag{8}$$

where:

- $\Delta^d Y_t$ is the value of the time series differenced d times at time t;
- β_0 is the constant term;
- $\sum_{i=1}^{p} \beta_i \Delta^d Y_{t-i}$ represent the autoregressive part of the model, in which β_i are autoregressive coefficients and $\Delta^d Y_{t-i}$ are past values of the differenced time series
- ϵ_t is the error term at time t, assumed to be white noise with zero mean and constant variance.
- $\sum_{j=1}^{q} \theta_j \varepsilon_{t-j}$ represent the moving average part of the model, in which θ_j are moving average coefficients and ϵ_{t-j} are past error terms

GARCH model is specifically designed to capture volatility clustering and time-varying volatility present in financial time series. GARCH(p, q) model can be expressed as:

$$\sigma_t^2 = \omega + \sum_{i=1}^{q} \alpha_i \varepsilon_{t-i}^2 + \sum_{j=1}^{p} \beta_j \sigma_{t-j}^2 \tag{9}$$

where:

- σ_t^2 is the conditional variance at time t;
- ω is a constant,
- α_i are the coefficients of ARCH terms, they weight the importance of squared past errors ϵ_{t-i}^2 in determining the current conditional variance,
- β_j are the coefficients of GARCH terms, they weight the importance of the past conditional variances σ_{t-j}^2 in determining the current conditional variance,

In our analysis, we employed a GARCH+ARFIMA model; **ARFIMA** (Autoregressive Fractionally Integrated Moving Average) model is a generalization of ARIMA that allows for fractional differencing.

4 Analysis

To determine the most appropriate model for PD, we began by identifying potential predictor variables through exploratory data analysis. We initially discarded variables generated after the borrower defaults (such as `collection_recovery_fee`, which represents a fee charged to the borrower for debt recovery services). Once we eliminated unusable variables, we proceeded by gradually including variables that we believed could best predict the PD. After several attempts, the most effective variables for this task were identified as:

$$X_i = \{\text{funded\_amnt}, \text{int\_rate}, \text{fico\_range\_low}, \text{total\_pymnt}, \\ \text{total\_rec\_int}, \text{out\_prncp}, \text{revol\_util}\}$$

This set of covariates is used to produce model (1), (2) and (3).
Next, we turned our attention to determine the optimal threshold for PD model: if the predicted PD is less than or equal to the threshold, we consider the customer in default; otherwise, we do not.
Since a decision threshold was not provided a priori by the company, it was necessary to identify an optimal one: we've used F1 score as a metric to evaluate discrimination ability at different thresholds (alternative decision methods are shown on R). F1 score is particularly useful when dealing with imbalanced datasets, where one class is much more frequent than the other: in our situation, we had an imbalance because there were more observations characterized by non-default (0) than default (1).
F1 score is defined as follows:

$$F1 \text{ score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{10}$$

where precision is the ratio of correctly predicted defaults to the total predicted default and recall (or sensitivity) is the ratio of correctly predicted defaults to the actual defaults. Since it is not possible to maximize recall and precision simultaneously, F1 score provides a balanced measure. By maximizing F1 score, we find a classification threshold that optimally balances precision and recall. This approach helps avoid the extremes of having too many false positives (high precision, low recall) or too many false negatives (low precision, high recall).
We determined a distinct threshold for each model: 0.12 for the linear model, 0.19 for the logit and 0.16 for the probit.
Subsequently, to evaluate the models and select the best one, we used different metrics: Area Under Curve (AUC) [Hosmer & Lemeshow, 2004], Brier Score (Brier, 1950) and sensitivity. Table 2 shows the metric values for each model.
Sensitivity (also known as recall or true positive rate) measures the proportion of actual

positives that are correctly identified by the model. It's crucial in our analysis because we want to avoid predicting that a person is not in default when he actually is, because this can result in significant financial losses for the company.

	AUC	Brier Score	Sensitivity
Linear	0.967	0.035	0.934
Probit	0.971	0.0062	0.942
Logit	0.971	0.0061	0.947

Table 2: Comparison of Model Performance Metrics

As evident from Table 2, logit model demonstrates the most satisfactory results, thus serving as our final model.

After completing PD model, our focus shifted to construct a linear model for `recoveries`. We first attempt to model the variable using the natural logarithm, as explained in Section 3. Initially, we eliminated variables deemed not useful to predict our dependent variable and then applied the stepwise forward selection method to perform an informative feature selection. In all the following described models we also transformed the variable `total_pymnt` with the natural logarithm: this transformation was employed to address the right-skewed distribution of the variable.
Next, we explored an alternative modeling approach by logically selecting variables based on our knowledge of the dataset, obtaining the following set:

$$\begin{aligned} X_i = \{&\text{loan\_amnt}, \text{term}, \text{int\_rate}, \text{fico\_range\_low}, \log(\text{total\_pymnt}), \text{tot\_cur\_bal}, \\ &\text{mort\_acc}, \text{num\_bc\_sats}, \text{num\_bc\_tl}, \text{debt\_settlement\_flag}, \\ &\text{total\_rec\_prncp}, \text{total\_rec\_int}, \text{total\_rec\_late\_fee}\} \end{aligned}$$

Afterwards, we explored the efficacy of LASSO regression, which applies a penalty to the absolute size of regression coefficients, encouraging simpler and more interpretable models. Note that in LASSO model the log transformation was not necessary because the predictions in this case were all defined in $\mathbb{R}^+$.

$$\begin{aligned} X_i = \{&\text{loan\_amnt}, \log(\text{total\_pymnt}), \text{total\_rec\_prncp}, \text{total\_rec\_int}, \\ &\text{total\_rec\_late\_fee}, \text{debt\_settlement\_flag}\} \end{aligned}$$

Following these attempts, we also investigated the application of a GLM with a Gamma distribution for `recoveries`. This distribution is often appropriate for modeling positively skewed data, such as in our particular situation. The covariates obtained through these

method are:

$$X_i = \{\text{loan\_amnt}, \text{term}, \text{int\_rate}, \text{verification\_status}, \text{fico\_range\_low}, \log(\text{total\_pymnt}), \text{total\_rec\_prncp}, \text{total\_rec\_int}, \text{total\_rec\_late\_fee}, \text{last\_pymnt\_amnt}, \text{debt\_settlement\_flag}\}$$

To assess the performance of the models, we employed RMSE metric and R-squared. As shown in Table 3, LASSO model achieved the lowest RMSE.

	RMSE	R-squared
Stepwise	318.46	0.5381
Logical	312.80	0.5364
LASSO	67.07	0.4877
Gamma	1280	0.0586

Table 3: Comparison of Model Performance Metrics. Note that to compute RMSE of the model with a log-transformed dependent variables it was necessary to back-transform the predictions. For stepwise, logical e Lasso model we computed Adjusted R-squared, while for Gamma is a Mc-Fadden R-squared (these two different metrics are not comparable in absolute value).

Analysis of **time series** was conducted adopting the Box-Jenkins methodology (Box & Jenkins, 1970), which involves three main steps: model identification by analyzing autocorrelation (ACF) and partial autocorrelation (PACF) functions, parameter estimation, and model validation by ensuring that residuals are white noise (uncorrelated).
To achieve covariance-stationarity, we first-differenced the time series. The stationarity of the first-differenced series was confirmed by using the Augmented Dickey-Fuller test (Dickey & Fuller, 1979). As the first order of differencing made the series stationary, no higher order was necessary.
Orders p and q of our ARIMA model were chosen by qualitatively assessing ACF and PACF of the model residuals. We also estimated the parameters of various ARIMA models and select the model that minimize the Akaike Information Criterion (AIC), ensuring a balance between model fit and complexity.
To further refine our model selection, we also considered an ARMA(1,1) model applied to the logarithm of the time series, instead of differencing.
Based on the overall analysis and several considerations, we selected an ARIMA(1,1,1) model as the most appropriate for our data: this model effectively balances complexity and fitting, providing a robust framework to forecast the time series.
Finally, we validated our chosen model by ensuring that the residuals were white noise, i.e., uncorrelated. This step is crucial to confirm that the model adequately captures the patterns in the data. After observing that the squared residuals of our model did not exhibit white noise characteristics, as evidenced by the ACF plot, we recognized the

presence of residual correlation. Consequently, we opted to incorporate a GARCH model to address this correlation and better model the variance.
Additionally, we decided to use an ARFIMA model to capture the persistence without over-differencing; then we determined the optimal fractional differencing parameter, d.
With this refined model, we divided the time series into training and test sets, with test set starting from January 2017 onwards. We first fitted the model on training datas and used the last observation to predict h-months ahead. We then iteratively extended the training set by one observation, refitted the model on the new training set and produced the next h-months-ahead prediction.
To compute prediction intervals we first needed to specify a distribution for the residuals. Accurate prediction intervals require in fact a proper understanding of the residual distribution, as this directly affects the interval estimation.
After several trials, we identified that residuals followed a standardized Student's t-distribution with four degrees of freedom. This choice was based on the distributional characteristics of the residuals, such as their heavy tails, which are better captured by a t-distribution than a normal one.
We calculated upper and lower bounds of our predictions by finding appropriate quantiles for the given degrees of freedom. This approach ensured that our prediction intervals properly reflected the variability and uncertainty in the data, leading to more reliable and robust forecasts. A graphical representation of the result can be seen in Figure 1 (Left).
Finally, as always shown in Figure 1 (Right), we computed the actual forecast point predictions and intervals for 12 months ahead, using a model fitted on all available observations. This horizon balances the long-term outlook (given that our loans are 36 or 60 months) with the need to capture short-term rate fluctuations. Furthermore, a 12-month forecast aligns with annual budgeting, aiding in planning for revenue, expenses and profitability over the fiscal year.

5 Discussion

We consider our prediction of PD satisfactory. However, continuous improvement and validation with new data will be essential to maintain and enhance its accuracy over time. Regularly updating training data is crucial to capture any changes in the characteristics of defaulters, ensuring the model remains effective.

In estimating predictions for the recoveries, we are interested just in point predictions and not in prediction intervals; therefore, our work focused solely on the expected value (we did not consider the variance). However, if in the future LC wishes to include prediction intervals in their estimations, we would need to extend our model to account for the variance in order to obtain correct prediction intervals.
For this reason, in our current models we corrected the variance-covariance matrix for

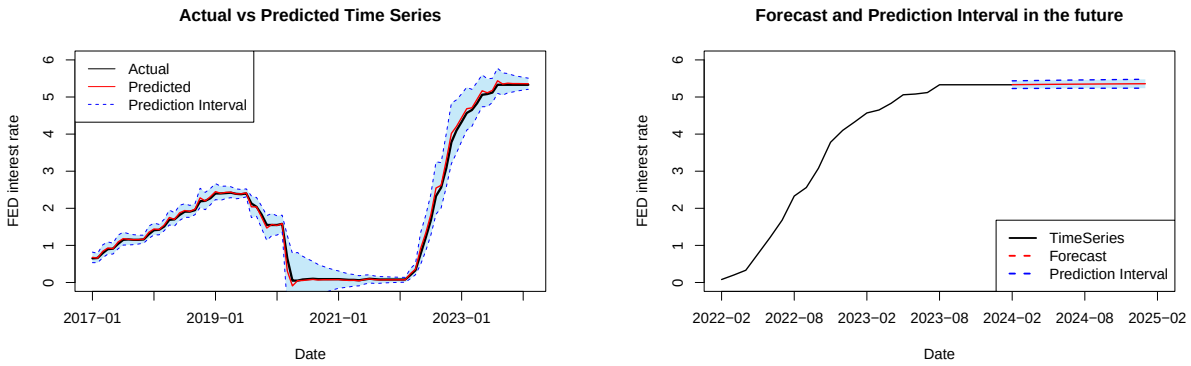

Figure 1: (Left) This plot illustrates the comparison between the actual historical monthly FED Interest Rate and the predicted series obtained from the test set using the ARFIMA+GARCH model. The data spans from January 2017 to February 2024. Additionally, prediction intervals at 95% are shown to provide a visual representation of the uncertainty associated with the predictions.
(Right) This graph illustrates the forecast for the time series extending from February 2024 to February 2025, covering the next 12 months, along with 95% prediction intervals. The forecast is obtained using the ARFIMA+GARCH model.

heteroskedasticity (White, 1980) to ensure accurate calculation of confidence intervals for the expected value.
We've also observed that Gamma model metrics don't show a good fit nor predictive accuracy. Therefore, we could consider ways to improve its performance and lead to enhanced predictive performance.

Regarding our forecast for FED interest rates, to increase the precision of our model we could consider adding covariates by using ARIMAX model. Federal Reserve rates might be better predicted by taking into account the overall environment in which they are set. For example, including macroeconomic variables such as previous months' inflation rates, unemployment rates, GDP growth, and consumer sentiment indices could provide a more comprehensive understanding of the factors influencing interest rate decisions. This approach could enhance the model's ability to capture the complexities of the economic landscape and improve forecast accuracy.
Furthermore, it is worth mentioning that our data extended up to February 2024, so our future predictions include estimates that have already been realized (as we are writing in May 2024). Indeed, the Federal Reserve has kept interest rates stable from February to May, confirming our forecast for these three months.

Bibliography

Box, G. E., Jenkins, G. M., & Reinsel, G. (1970). Time series analysis: forecasting and control Holden-day San Francisco. *BoxTime Series Analysis: Forecasting and Control Holden Day, 1970.*

Brier, G. W. (1950). Verification of forecasts expressed in terms of probability. *Monthly weather review, 78(1), 1-3.*

Dickey, D. A., & Fuller, W. A. (1979). Distribution of the estimators for autoregressive time series with a unit root. *Journal of the American statistical association, 74 (366a), 427-431.*

Hosmer Jr, D. W. and Lemeshow, S. (2004). Applied logistic regression. *John Wiley & Sons.*

White, H. (1980). A heteroskedasticity-consistent covariance matrix estimator and a direct test for heteroskedasticity. textitEconometrica: journal of the Econometric Society, 817-838.