Loan Club Report

Noah Brandes
ECON 484

June 12, 2025

0 Problem Context and Objectives

This project concerns a set of compiled data from Lending Club, a fintech firm that facilitates peer-to-peer lending. There are two provided dataframes, one with loan applications that were accepted, and one with loan applications that were rejected. The accepted dataframe has many features with many NAs, while the rejected dataframe only has only a few features. There are four primary objectives:

(A) Simulate all independent variables in the accepted set that are not in the rejected set.

(B) Using my own parameters, simulate whether individuals default and, if they defaulted, the proportion of their loan recovered by the lender.

(C) Reverse engineer the loan rejection rule used by Lending Club.

(D) Develop a rule to accept loans in such a way as to profit maximize.

This report explains my workflow, the major judgment calls made in completing this project, and my recommendations for loan issuance. Each section is a major segment of the project and, if relevant, is labeled with the relevant goal.

1 Data Cleaning

There is a very high density of NAs in the set, and some variables are sparse. I could filter out all sparse columns, but I am suspicious that the presence of an NA in certain columns may be important for prediction. For example, an NA in *joint_app_inc* indicates that an application was not made jointly, and no variable describes which applications are joint.
So, for all columns $x \in X_{Acc}$ with any NA values, I define a factor describing if x_i is NA:

$$x\_na_i = \begin{cases} 1, & \text{if } x_i \text{ is unobserved} \\ 0, & \text{else} \end{cases}$$

Then I calculate the sample mean of the observed x and redefine x as

$$x_i = \begin{cases} \bar{x}, & \text{if } x_i \text{ is unobserved} \\ x_i, & \text{else} \end{cases}$$

I also find that in the rejected set there are many missing risk scores. I handle this differently than above, choosing to simulate risk scores instead. This is necessary such that the rejected features match the accepted ones (where all risk scores are present).

I also sample reasonably sized subsets of each set to proceed with, as there are too many observations to feasibly model on with my computing resources.

2 Simulate Defaults (B)

I specify the following true data model for defaulting, where all features are standardized.

$$y_i = -8 - 4 \times inc + \frac{1}{2} \times dti + 2 \times amnt - 3 \times \frac{fico\_low + fico\_high}{2}$$

$$P(defaults_i) = \frac{1}{1 + e^{y_i}}$$

All coefficients have a sign that aligns with my intuition, and the magnitude reflects my assumptions about their relative importance. The logistic function ensures the result is a probability. A Bernoulli experiment with the set probability is performed to obtain the $default$ factor variable.

3 Variable Selection (A)

I lasso $defaults$ on all features of the accepted set. Lasso here is of the binomial family, as $defaults$ is a dummy variable. All variables with nonzero coefficients are selected. Notably, I over-sample from the observations that defaulted, such that I can obtain reasonable coefficient estimates in feasible runtime.

4 Simulate Independent Variables (A)

For each of the 42 variables selected by lasso, I simulate that variable conditioned on the rejected observations. To do so, for each variable z_i:

1. If z_i is continuous, fit an RF, a boosted model (after tuning lambda), and OLS.

   If z_i is a factor, fit logistic regression, LDA, and NB.

2. If z_i is continuous, choose the model that minimizes cross-validated MSE.

   If z_i is a factor, choose the model that minimizes misclassifications with cross-validation.

3. If z_i is continuous, predict all $\hat{z}_i$ conditioned on X_{new} with the chosen model.

   If z_i is a factor, predict all $\hat{P}(z_i)$ conditioned on X_{new} with the chosen model.

4. If z_i is continuous, inject noise by pulling values from $z\_sim_i \sim N(\hat{z}_i, \sqrt{MSE_z})$.

   If z_i is a factor, take a multinomial sample with $\hat{P}(z_i)$ to simulate the class z_i.

5 Simulate Defaults in Rejected Set (B)

I use the same simulation model as in (2) to fill in simulations for the rejected set. It turns out that the default rate here is about 0.05, which is much higher than 0.01, which is the default rate in the accepted data. I find this encouraging, because it implies that there is a meaningful difference between the accepted and rejected set.

6 Predict Rejected Applications (C)

I merge the accepted dataset with the rejected one, now armed with the simulated valued for all variables. With this dataset, I fit a tree on all independent variables to predict whether an applicant is accepted or rejected, and cross-validated to prune the tree.

The tree is $\hat{f}(x) = \begin{cases} 1, & fico\_low < 661.5 \\ 1, & fico\_low \geq 661.5, \ dti \geq 39.995 \\ 0, & \text{else} \end{cases}$

The tree achieves 93% correct classifications, and still performs well under cross-validation. In the accepted set, $\min(fico\_low) = 610$. While not exactly at the tree's cutoff, I consider this to be strong evidence that a tree-like model was used to accept or reject loan applications.

7 Predict Defaults and Recoveries (B)

To predict $default$, I consider LDA, NB, Logit, and a classification RF as feasible models. I use cross validation to choose the best model, which turns out to be logit. I then train logit on the full dataset to estimate $\hat{P}(def \mid X)$.

To predict $recovery$, which is continuous, I only consider an RF and a boosted model. This is because these models will enforce $0 < recovery < 1$ given that all of the data is in that range. This is our best estimate of $\hat{E}[recovery \mid X]$.

8 Loan Issuance Strategy (D)

We want to issue loans if and only if the expected profit $\hat{E}[\pi \mid X] > 0$. That is,

$$\begin{aligned} E[\pi \mid X] &= E[rev \mid X] - E[loss \mid X] \\ \hat{E}[\pi \mid X] &= (1 - \hat{p}(def \mid X)) \times interest - \hat{p}(def \mid X)(1 - 0.7 \times \hat{E}[recov \mid X]) \times amt \\ &= (1 - \hat{p}(def \mid X)) \times 0.15 \times amt - \hat{p}(def \mid X)(1 - 0.7 \times \hat{E}[recov \mid X]) \times amt \end{aligned}$$

This expectation is to be calculated with the models estimated in (7), alongside amt. Though, amt can be divided out to model the expected proportional return on the loan. If our models predict positive profits from issuing the loan, we ought to do so, rather than relying on overly simplistic heuristics, like the tree that filtered applicants for Loan Club.