Risky or Reliable?

Predicting Loan Outcomes

Final Project

Ania Shaheed

QST BA 305 B1
Professor Gerry Tsoukalas
May 4, 2025

I. Introduction

LendingClub is a peer-to-peer lending platform that connects borrowers directly with investors, bypassing traditional banks. While this model offers borrowers access to credit and investors the opportunity to earn interest, it also introduces significant risk: the possibility that borrowers will fail to repay their loans. A "bad/risky" borrower may make late payments, default entirely, or be charged off (cases in which the lender concludes repayment is unlikely and records the loan as a loss). These outcomes not only result in financial losses for individual investors, but also erode trust in the LendingClub platform, reduce future investment, and penalize responsible borrowers through higher interest rates. In this analysis, we use LendingClub's dataset to investigate the characteristics and behaviors associated with borrower risk, aiming to better understand and potentially predict loan performance outcomes.

To predict borrower risk on LendingClub, we analyze a dataset comprising loan records from 2012 to 2019, totaling over 2.26 million records and 145 features that capture a wide range of demographic, financial, and loan-specific attributes. Each row reflects a borrower's status as of February 2019. The target variable we chose was *loan status*, which indicates whether or not a borrower is paying responsibly. See ***Table 1*** for a summary of categories in this variable. While the majority of loans are "good" (87%), a significant amount (13%) represent "bad" loans, as can be seen in ***Figure 1***.

The goal of this project is to attempt to provide a useful predictive model to investors in order to detect risky borrowers and make smarter lending decisions. For our first experiment, we modeled riskiness at different stages. Our first version used features prior to issuance of a loan, while the second version used real-time features given a snapshot of the borrower. For our second experiment, we evaluated and compared different predictive models. This included logistic regression, decision trees, random forests, gradient boosting, a stacked model, and finally a Convolutional Neural Network (CNN). Beyond mere accuracy, we wanted to specifically evaluate the recall of "risky" borrowers. The focus also included interpretability of each model and their determining top predictors as well as practical application in the real-world.

II. Methodology

A large portion of time spent on this project was spent cleaning the data and making meaningful decisions to prepare the dataset for modeling. In order to make appropriate decisions,

Exploratory Data Analysis (EDA) was required. In this process, univariate analysis helped us understand the dataset better and identify empty or constant features. We also discovered that there were 33.2% missing cells, prompting exploration of missing values where we quantified and visualized nulls per feature and per row. We investigated features with >10% missing values as well as a small subset of rows with excessive missing data (23–30 null features), but these could not be explained by borrower attributes. Lastly, we computed and visualized a correlation heatmap among numeric features to address multicollinearity. Using a lower limit threshold of 0.7, we extracted feature pairs. These features were often redundant and suitable for removal, aiding dimensionality reduction and avoiding multicollinearity downstream in modeling.

Following EDA, extensive data pre-processing was conducted to prepare for modeling. Based on insights from the EDA, a significant number of features were dropped due to irrelevance, redundancy, collinearity, or high missingness. In this feature selection process, we also excluded specifics related to joint applications and hardship or settlement plans, only keeping binary features indicating whether or not these circumstances applied to the borrower.

Notably, a few columns were dropped as they were direct indicators of *loan_status* (see ***Table 2***). We decided to keep *loan_amount*, but combining this data with total amount paid/unpaid in principal/interest would allow the model to reverse-engineer the outcome based on how much of the loan was repaid or still owed. This is particularly true considering that transformed target variable is derived from prior classifications based on payments. Additionally, any non-zero value in *recoveries* confirms the loan failed. Including such features would appear highly accurate on historical data but perform poorly in real-world deployment, where future payment behavior is unknown. For example, the hypothetical model may not detect a risky borrower based on inflated payment history. In contrast, excluding these outcome-dependent signals forces the model to detect subtle indicators of borrower risk that persist even when recoveries are zero or payment history appears favorable. This approach leads to a more powerful and useful model.

To address incomplete records, rows with over 30% missing values were removed, which resulted in the deletion of approximately 64,000 rows. For remaining missing values, median imputation was applied to numerical columns. This method was chosen for its low computational cost compared to alternatives like KNN. Columns indicating months since various negative

credit events were deferred for consolidation later in the process. The results of feature selection are summarized in ***Table 2***.

Several types of variables were transformed. Ordinal features were mapped to numerical scales using custom dictionaries. Nominal features were one-hot encoded, dropping a class to avoid collinearity. The *addr_state* variable was converted to broader region categories for simplicity and then one-hot encoded. Date columns were converted into "months since" using the current date as a numeric representation of the data. Two-class categorical features were encoded with 0/1 binary mappings. To simplify our classification task, we converted the target variable into a binary indicator of borrower quality. A summary of this process can be seen in ***Table 1***. To note, even loans labeled "Does not meet the credit policy. Status: Fully Paid" were classified as bad because despite full repayment, they represent exceptions to LendingClub's screening standards and carry elevated risk. Our goal is to train models that prioritize safe investments.

After cleaning the data, we moved on to modeling. To ensure fair comparison and robust model evaluation, we used Stratified K-Fold Cross-Validation (5 splits), preserving class distribution across training and testing sets. Within each training fold, we applied normalization and SMOTE (Synthetic Minority Over-sampling Technique) to address class imbalance. We constructed two feature sets: *pre-loan*, which excludes any information that would not be available at the time of loan approval, and *post-loan*, which includes all available features. The excluded features for pre-loan modeling can be found on ***Table 2***.

Reusable functions were built to streamline training, tuning, and evaluation across different models. These include: *prepare*() for applying SMOTE, test train split and scaling to a feature set and target variable; *model*() for training and generating probabilities and predictions; *tune*() for hyperparameter optimization using *GridSearchCV*; *evaluate*() to generate classification metrics. We used SMOTE (Synthetic Minority Over-sampling Technique) instead of random oversampling to avoid exact duplicates of minority class examples, which can lead to overfitting. SMOTE generates synthetic examples by interpolating between existing ones. We also preferred it over class weighting, as it allowed us to maintain balanced class distributions. Due to this step, our naive baseline became 50% rather than the majority rule of 87%.

As a starting point, we trained a Logistic Regression model on both pre- and post-loan datasets to examine the impact of post-loan variables. Model performance was assessed using classification metrics and ROC curves.

Using the post-loan feature set, we subsequently trained and fine-tuned: Logistic Regression, Gaussian Naive Bayes, Decision Tree, Random Forest, and XGBoost. In this process, rather than using the 2M original records, we created a balanced subsample of 10,000 records reflecting the original class proportions to reduce computational complexity and runtime. This size struck a balance between representativeness and feasibility for training more complex models. However, there are drawbacks to this approach. Metrics from the smaller sample may have slightly higher variance and potentially lower generalizability. Definitive models should ideally be retrained and validated on the full dataset. All further tuning and evaluation was conducted on this optimized sample.

Using the best-performing models from individual learners, we constructed a stacked ensemble to leverage complementary strengths, using Logistic Regression as the meta model. Additionally, we developed a Convolutional Neural Network (CNN) to compare a simple deep learning approach. Though CNNs are traditionally used for spatial data like images, we reshaped our tabular data to a 1D format to explore its ability to extract local patterns. The CNN architecture included a 1D convolutional layer with ReLU activation, followed by max pooling, batch normalization, dropout regularization, flattening, and two fully connected layers, ending with a sigmoid activation for binary classification. We compiled the model with the Adam optimizer and binary cross-entropy loss, and trained it over 10 epochs with a batch size of 32, using 20% of the training data as a validation set.

All models were compared based on precision, recall, F1, accuracy, and AUC. We particularly wanted to optimize recall of risky borrowers rather than overall accuracy.

III. Results

Table 3 summarizes model performance before and after the addition of post-loan features. Across all metrics, post-loan data led to substantial performance improvements. Most notably, the F1-score for risky borrowers increased from 0.337 to 0.657, a relative improvement of 94.9%. This suggests that post-loan variables provide critical predictive value in identifying defaults. Similarly, the overall accuracy improved from 67.7% to 88.9%, indicating that the

model became substantially more reliable with access to post-loan behavior signals. The ROC curves in Figure 2 help visualize the dramatic increase in the model's power. These results are not surprising considering that borrower behavior is extremely pertinent to risky outcomes.

Table 4 compares the performance of various classification models using the full post-loan feature set and the smaller sample. The Neural Network achieved the best results across nearly all metrics, with an AUC of 1.00 (see ***Figure 3***). XGBoost followed closely, Decision Trees and Random Forests also performed strongly, Logistic Regression performed moderately well, and Naive Bayes performed the worst. Stacking models did not improve performance, instead decreasing the power of the strongest learner by introducing noise.

These results suggest that complex, non-linear models (CNN and XGBoost) are best suited for capturing the patterns in post-loan behavior that predict default. This makes sense because post-loan signals introduce intricate, high-dimensional interactions better captured by flexible model architectures.

Figure 4 displays the top 10 predictive features for XGBoost. The most important features focus primarily on the borrower's payment history and loan characteristics. The amount of the last payment and the time elapsed since the last payment are critical in understanding the borrower's recent financial behavior, with longer gaps or smaller payments potentially signaling higher risk. The loan term also plays a significant role; perhaps shorter-term loans pose different risks compared to longer ones. Additionally, the number of months since the most recent installment gives insight into the borrower's credit activity, while the debt settlement flag indicates past struggles with managing debt.

IV. Conclusion

The analysis of LendingClub's dataset has provided valuable insights into the factors that determine borrower risk and how this risk can be predicted effectively. By focusing on post-loan features such as payment history, loan term, and debt settlement flags, we were able to identify key patterns that differentiate risky borrowers from responsible ones. The models, particularly XGBoost and the Convolutional Neural Network, demonstrated substantial improvements in performance when post-loan data was incorporated. This underscores the importance of considering borrower behavior after the loan is issued, as it provides a more accurate prediction of loan default than pre-loan features alone.

Furthermore, the modeling approach used in this analysis can help investors make more informed decisions, reducing the risk of financial losses due to defaults. By prioritizing the recall of risky borrowers, the models were optimized to identify those who pose the greatest threat to loan repayment. As a result, these findings can directly influence strategies for loan approval, repayment monitoring, and risk mitigation, ultimately benefiting both investors and responsible borrowers by creating a more reliable lending environment.

Tables & Figures

Table 1 - Loan Status Variable

Loan Status Category	Description	Class
Fully Paid	Loan has been completely repaid by the borrower.	1
Current	Borrower is actively making on-time payments.	1
In Grace Period	Borrower is slightly late but within the acceptable grace period.	1
Late (16-30 days)	Borrower has missed a payment and is more than two weeks overdue.	0
Late (31-120 days)	Borrower is significantly overdue; high risk of default.	0
Charged Off	Loan has been written off as a loss by the lender due to nonpayment.	0
Default	Borrower has failed to meet loan obligations; considered in default.	0
Does not meet the credit policy. Status: Fully Paid	Borrower was considered a higher risk but repaid the loan in full.	0
Does not meet the credit policy. Status: Charged Off	Borrower did not meet standard credit criteria and failed to repay loan.	0

Table 2 - Feature Selection

Decision	Reason	Columns
Remove	Empty, irrelevant, or constant	id, member_id, url, zip_code, desc, emp_title, title, issue_d, acceptD, expD, ils_exp_d, reviewStatusD, creditPullD, initial_list_status, initialListStatus, policy_code, deferral_term
Remove	Apply only to loans in hardship	hardship_length, hardship_type, hardship_status, hardship_start_date, hardship_end_date, hardship_amount, hardship_dpd, hardship_loan_status, hardship_reason, hardship_payoff_balance_amount, hardship_last_payment_amount
Remove	Apply only to settlement	debt_settlement_flag_date, settlement_status, settlement_date, settlement_amount, settlement_percentage, settlement_term
Remove	Apply only to joint applications	annual_inc_joint, dti_joint, verification_status_joint, revol_bal_joint, sec_app_earliest_cr_line, sec_app_inq_last_6mths, sec_app_mort_acc, sec_app_open_acc, sec_app_revol_util, sec_app_open_act_il, sec_app_num_rev_accts, sec_app_chargeoff_within_12_mths, sec_app_collections_12_mths_ex_med, sec_app_mths_since_last_major_derog
Remove	High null count	orig_projected_additional_accrued_interest, payment_plan_start_date, next_pymnt_d, il_util, all_util, open_acc_6m, total_cu_tl, inq_last_12m, open_act_il, open_il_12m, open_il_24m, total_bal_il, open_rv_12m, max_bal_bc, inq_fi, open_rv_24m
Remove	High collinearity or redundant information	funded_amnt, funded_amnt_inv, installment, open_acc, out_prncp_inv, total_pymnt_inv, total_rec_prncp, collection_recovery_fee, open_il_12m, open_rv_12m, total_rev_hi_lim, tot_cur_bal, revol_util, num_actv_bc_tl, num_bc_tl, num_op_rev_tl, num_rev_accts, num_rev_tl_bal_gt_0, num_sats, num_tl_30dpd, num_tl_op_past_12m, percent_bc_gt_75, tot_hi_cred_lim, total_bal_ex_mort, total_bc_limit, total_il_high_credit_limit, grade
Remove	Direct indicators	out_prncp, recoveries, total_pymnt, total_rec_int
Remove	Consolidated	mths_since_recent_bc, 'mths_since_last_delinq, mths_since_recent_revol_delinq, mths_since_last_major_derog, mths_since_recent_bc_dlq, mths_since_last_record
Remove (pre-loan only)	Repayment information	total_rec_late_fee, last_pymnt_d, last_pymnt_amnt, last_credit_pull_d, hardship_flag, debt_settlement_flag, neg_cr_event, disbursement_method_DirectPay, joint_flag, loan_status_binary
Keep	Relevant features (54)	loan_amnt, term, int_rate, sub_grade, emp_length, annual_inc, pymnt_plan, dti, delinq_2yrs, inq_last_6mths, pub_rec, revol_bal, total_acc, total_rec_late_fee, last_pymnt_amnt, collections_12_mths_ex_med, acc_now_delinq, tot_coll_amt, mths_since_rcnt_il, acc_open_past_24mths, avg_cur_bal, bc_open_to_buy, bc_util, chargeoff_within_12_mths, delinq_amnt, mo_sin_old_il_acct, mo_sin_old_rev_tl_op, mo_sin_rcnt_rev_tl_op, mo_sin_rcnt_tl, mort_acc, mths_since_recent_inq, num_accts_ever_120_pd, num_actv_rev_tl, num_bc_sats, num_il_tl, num_tl_120dpd_2m, num_tl_90g_dpd_24m, pct_tl_nvr_dlq, pub_rec_bankruptcies, tax_liens, hardship_flag, debt_settlement_flag, home_ownership, verification, purpose, disbursement_method, addr_state, earliest_cr_line, last_pymnt_d, last_credit_pull_d, joint_flag, neg_cr_event, loan_status

Table 3 - Pre vs. Post Loan Model Performance

Phase	precision_0	recall_0	f1_0	precision_1	recall_1	f1_1	accuracy
Pre	0.227707	0.649757	0.337232	0.93058	0.680561	0.786171	0.676661
Post	**0.540203**	**0.839585**	**0.657414**	**0.974716**	**0.896413**	**0.933926**	**0.889218**
Difference	0.312496	0.189828	0.320182	0.044136	0.215852	0.147755	0.212557

Table 4 - Model Performance

Model	precision_0	recall_0	f1_0	precision_1	recall_1	f1_1	accuracy
Logistic Regression	0.519879	0.815956	0.635106	0.970919	0.890759	0.929113	0.881288
Naive Bayes	0.595238	0.197472	0.29656	0.893935	0.980534	0.935234	0.881388
Decision Tree	0.668345	0.838863	0.743958	0.975743	0.939654	0.957359	0.926893
Random Forest	0.738959	0.740126	0.739542	0.962318	0.962098	0.962208	0.933993
XGBoost	0.852895	0.860979	0.856918	0.979819	0.978472	0.979145	0.963596
Stacking	0.833333	0.832347	0.97583	0.975495	0.975663	0.957496	0.957496
Neural Network	**0.929811**	**0.973144**	**0.950984**	**0.99608**	**0.989351**	**0.992704**	**0.987299**

Figure 1 - Loan Status Distribution

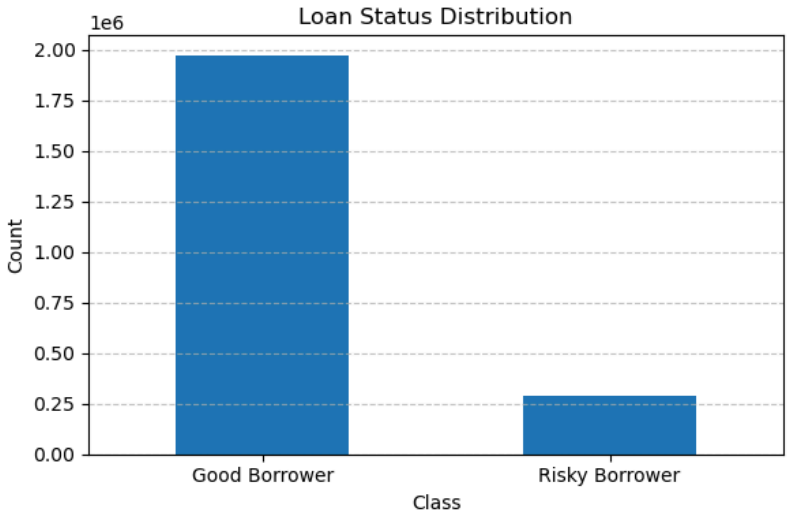

Figure 2 - Pre-Loan vs. Real-Time ROC Curves and AU

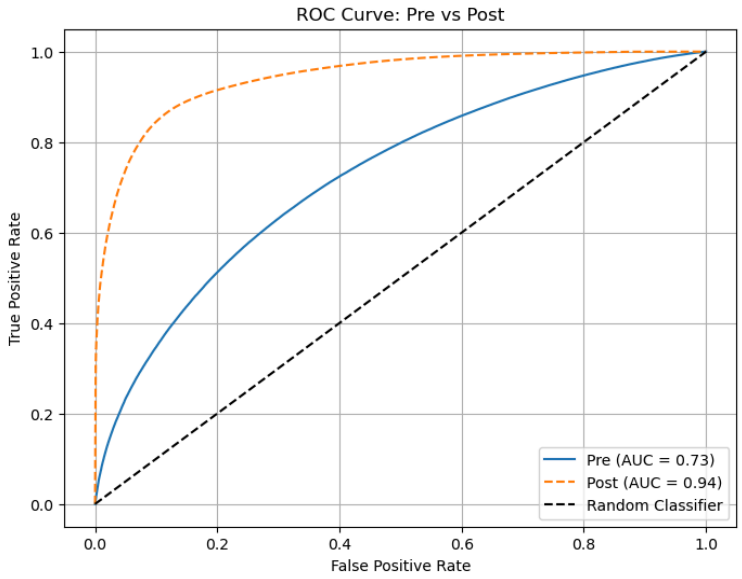

Figure 3 - Comparing all Models

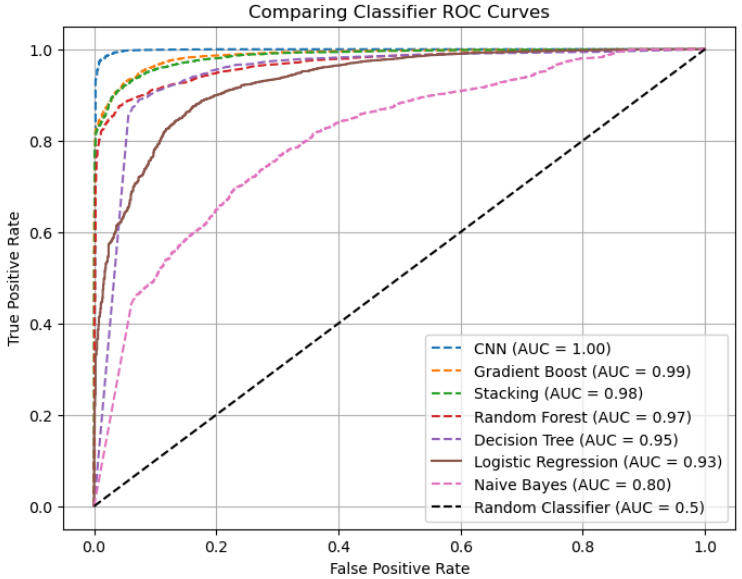

Figure 4 - Top 10 Features