Predicting Loan Outcomes: Fully Paid or Charged Off?

Arpit Dang

Data Science Institute at Brown University

Rocket Mortgage Case Study

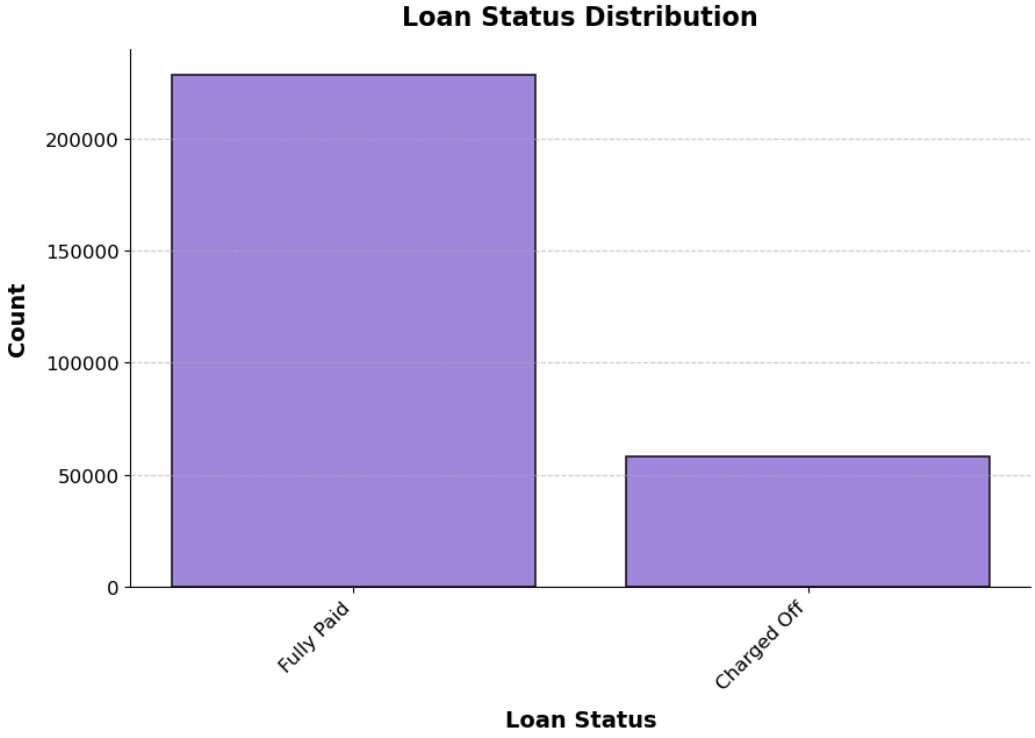
March 20th, 2025

https://github.com/adang66/Rocket_Mortgage/tree/main

Introduction and Exploratory Data Analysis

In the fast-growing fintech industry, effective loan approval decisions are crucial for optimizing business profitability while minimizing financial risks. This case study explores a data-driven approach to loan risk assessment, leveraging historical data of loan applicants to develop predictive models for default likelihood. The company aims to enhance its decision-making process by identifying key patterns that distinguish creditworthy applicants from those likely to default. By utilizing machine learning techniques on structured financial data, the objective is to design a robust predictive model that can assist in approving loans responsibly, adjusting interest rates for high-risk applicants, or rejecting applications that pose significant financial threats.

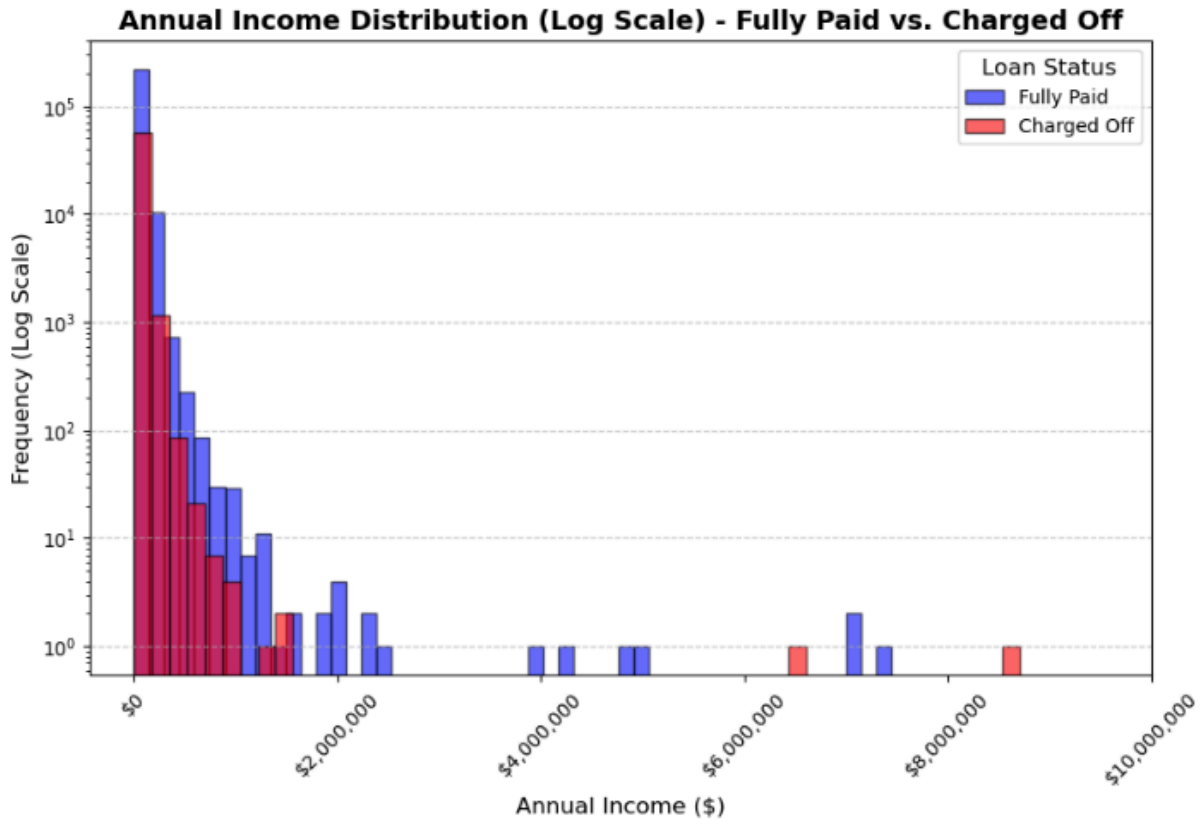
The dataset originally contained 316,970 data points, with some missing values in the columns ‘emp_title,’ ‘emp_length,’ ‘title,’ ‘revol_util,’ ‘mort_acc,’ and ‘pub_rec_bankruptcies.’ To address these missing values, ‘emp_title’ and ‘title’ were filled with ‘Unknown,’ while missing values in ‘emp_length’ were replaced with ‘< 1 year,’ assuming the individual had not yet secured employment. Data points with null values in ‘revol_util,’ ‘mort_acc,’ and ‘pub_rec_bankruptcies’ were removed, reducing the dataset to 286,584 data points. There were initially 26 variables. The target variable is ‘loan_status’ with a binary outcome of ‘Charged Off’ or ‘Fully Paid’. ‘Charged Off’ is represented by a 1, while ‘Fully Paid’ is a 0. This is a classification problem where we predict whether a client with Fully Pay the loan (0) or be Charged Off (1).



In the training dataset, 228,668 instances (79.8%) are labeled as ‘Fully Paid,’ while 57,916 instances (20.2%) are labeled as ‘Charged Off,’ indicating an imbalanced dataset. Given

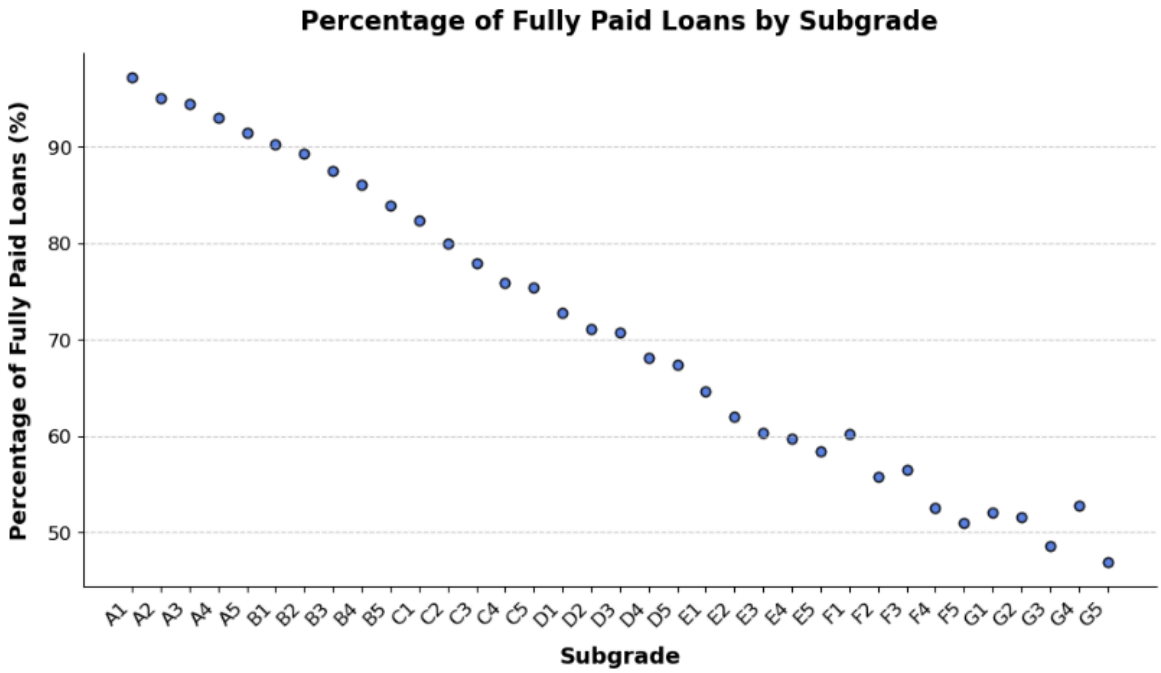
this imbalance, the F-score is an appropriate evaluation metric, as it focuses on the balance between precision and recall without being influenced by the number of True Negatives.

If the goal is to approve as many creditworthy clients as possible while minimizing lost business opportunities, a higher-weighted F-score (e.g., F1.5, F2, or F4) can be used, placing more emphasis on recall. Conversely, if the priority is to be highly precise in identifying clients who will reliably repay their loans, a lower-weighted F-score (e.g., F0.5) should be chosen, favoring precision over recall. For now, unsure in what position that bank is at in terms of their risk-taking ability, we will just use F1 score as an evaluation metric.

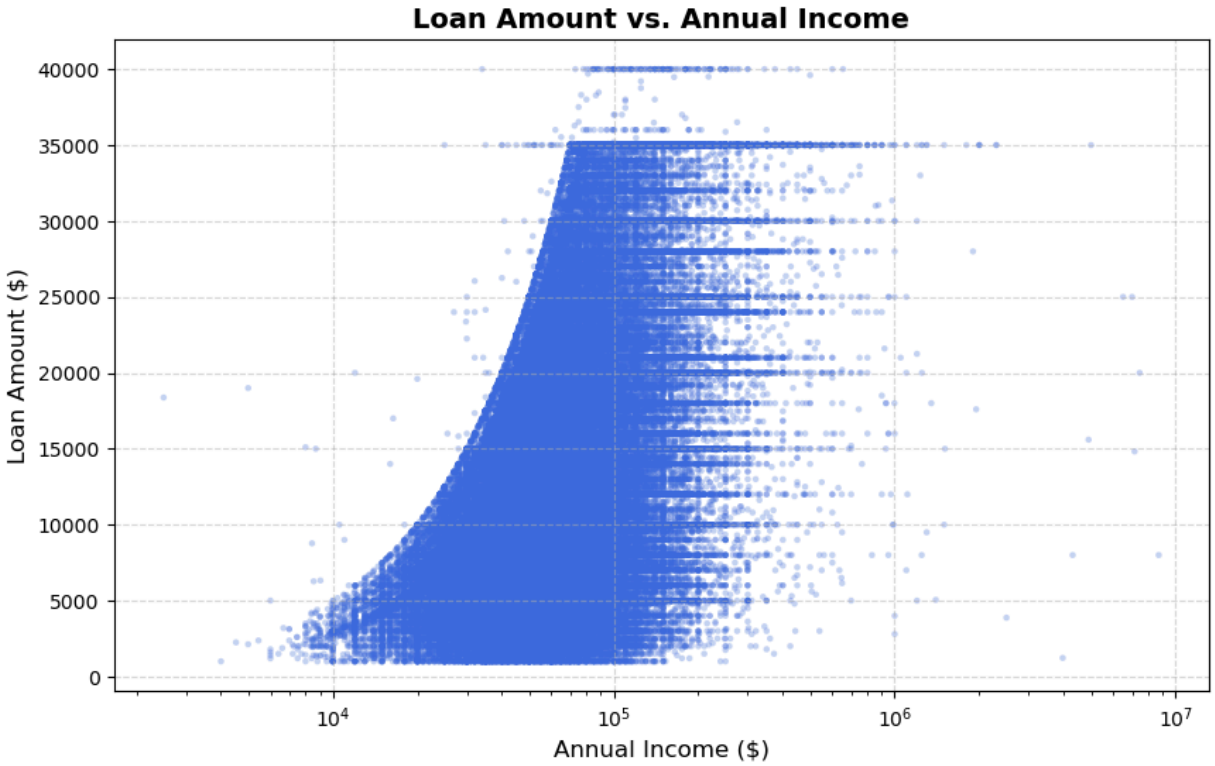


The histogram of annual income, displayed on a logarithmic scale, suggests significant income inequality. The majority of individuals earn relatively low incomes, as indicated by the concentration of values on the left side of the distribution. However, a long right tail, with sporadic high-income values, indicates the presence of a small number of individuals earning substantially more than the rest. This kind of right-skewed distribution is characteristic of income disparities seen in many economies, where wealth is concentrated among a few high earners while the majority earn far less. The log scale further emphasizes the vast differences in income levels, suggesting that policies addressing income distribution, such as progressive taxation or wage reforms, might be necessary to reduce economic inequality.

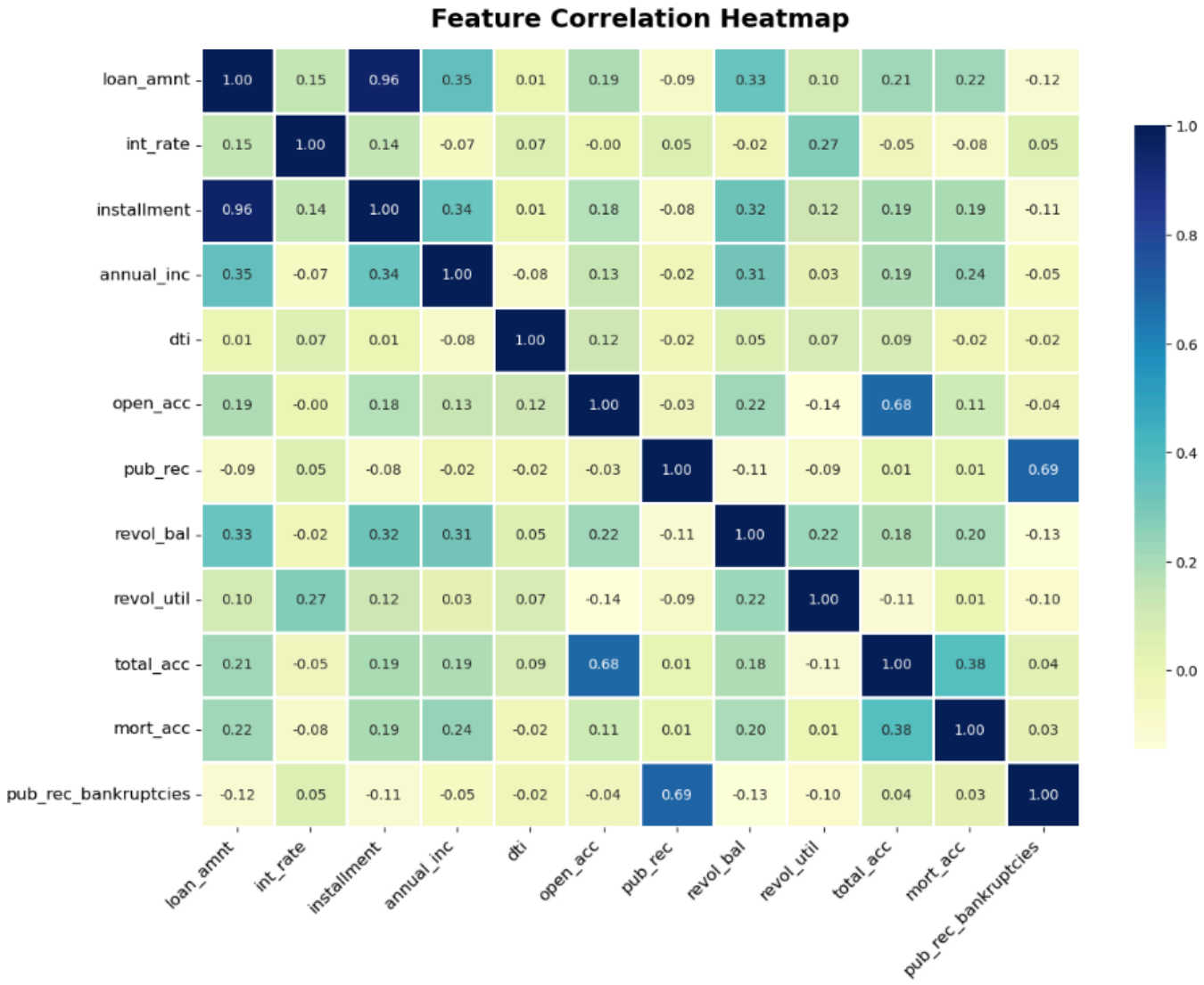
Even though some borrowers earn over $6,000,000 annually, some still failed to repay loans of less than $40,000. This suggests that annual income alone is not a strong determinant of whether a borrower will fully repay their loan.



The bar chart illustrates the percentage of fully paid loans across different subgrades, showing a clear downward trend from A1 to G5. Higher-rated subgrades have the highest fully paid percentages, indicating lower risk and better borrower reliability. As we move towards lower subgrades, the percentage of fully paid loans declines, suggesting increased risk and a higher likelihood of loan defaults. This trend aligns with expectations, as lower subgrades typically correspond to borrowers with weaker credit profiles and higher default rates.



The scatter plot illustrates the relationship between annual income and loan amount, revealing a positive correlation where higher-income individuals generally take out larger loans. However, this relationship is not linear, as most loans appear to be capped around $40,000, regardless of income level, suggesting lender-imposed limits. The logarithmic scale on the x-axis helps visualize income variations, showing that while some high-income borrowers take out large loans, many still opt for smaller amounts. Additionally, there are outliers, including low-income borrowers with relatively large loans and high-income individuals borrowing modest amounts, which may indicate risk-based lending criteria. This trend suggests that while income plays a role in loan approval, other factors such as creditworthiness, debt-to-income ratio, and lender policies also significantly impact borrowing limits.



The correlation heatmap shows that most features do not have a strong correlation, so we decided to keep all of them to ensure no potentially valuable information is lost. While loan amount and installment have a high correlation (0.96), both were retained because they capture different aspects of lending—loan amount represents borrowing behavior, while installment reflects affordability and repayment obligations. Removing one could impact model performance by omitting important financial context. Since no other features exhibit strong linear relationships, keeping the full feature set allows for better exploration of complex interactions in the data.

Methods

The ‘training.csv’ dataset was divided into a 60% training set, 20% validation set, and 20% test set. Due to the dataset’s imbalance, with only 20.2% of data points belonging to ‘Charged Off’, a stratified split was employed to ensure proportional representation of classes across all subsets. Due to a large dataset with 286,584 data points, a K-fold cross validation was not performed during splitting and training.

Type of Set	Total Data Points	Fully Paid	Charged Off
Training Set	171,951	137,217	34,734
Validation Set	57,316	45,738	11,578
Test Set	57,317	45,739	11,578

After addressing the missing data as previously described, we retained 286,584 data points across 26 features.

The target variable, ‘loan_status’, was transformed into a binary format, where ‘Charged Off’ was assigned 1 and ‘Fully Paid’ was assigned 0.

The variables ‘issue_d’ and ‘earliest_cr_line’ were initially stored as 'Mon-Yr' string values and were converted to datetime format. The number of months elapsed until March 2025 was then calculated for both variables, creating two new features: ‘cr_months_from_mar_2025’ and ‘issue_months_from_mar_2025’.

- ‘cr_months_from_mar_2025’ ranged from 137 months (11.4 years) to 974 months (81.1 years) in the past.
- ‘issue_months_from_mar_2025’ ranged from 99 months (8.25 years) to 156 months (13 years) in the past.

The ‘grade’ column was removed since its information was already captured in ‘sub_grade’.

The columns ‘address’, ‘emp_title’, and ‘title’ were also dropped due to their high cardinality. Specifically:

- ‘emp_title’ had 125,426 unique values
- ‘title’ had 28,417 unique values
- ‘address’ had 285,260 unique values

These columns contained too many distinct values to be effectively utilized in the analysis.

A data preprocessing pipeline was designed to handle continuous, ordinal, and categorical features effectively. MinMaxScaler was applied to 12 continuous features since their values fell within a defined range. OrdinalEncoder was used for two ordinal features, with their respective value orders explicitly defined. OneHotEncoder was applied to six categorical features to convert them into numerical representations. Additionally, StandardScaler was used for two features due to their wide range and presence of outliers.

Four machine learning models—logistic regression, random forest classifier, support vector classifier, and XGBoost—were implemented with hyperparameters detailed in Table below. Due to the dataset's large size and computational limitations, the Support Vector Classifier was trained, validated, and tested on only 15% of the dataset.

Machine Learning Algorithm	**Parameters**	**Balance Class Weights**
Logistic Regression	'C': [0.001, 0.01, 0.1, 1, 10, 100, 1000] 'penalty': ['l1', 'l2'], 'solver': ['liblinear']	class_weight='balanced'
Random Forest Classifier	'max_depth': [3, 10, 30, 100], 'max_features': [0.25, 0.5, 0.75, 1.0]	class_weight='balanced'
Support Vector Classifier* (Only performed on 15% of the dataset)	'C': [0.1, 1, 10, 100], 'kernel': ['linear', 'rbf'], 'gamma': ['scale']	class_weight='balanced'
XGBoost	'learning_rate': [0.01,0.1, 0.3], 'max_depth': [3, 10], 'n_estimators': [50, 200], 'min_child_weight': [1, 5], 'gamma': [0, 0.1, 1], 'reg_lambda': [1, 10]	scale_pos_weight = 4.95

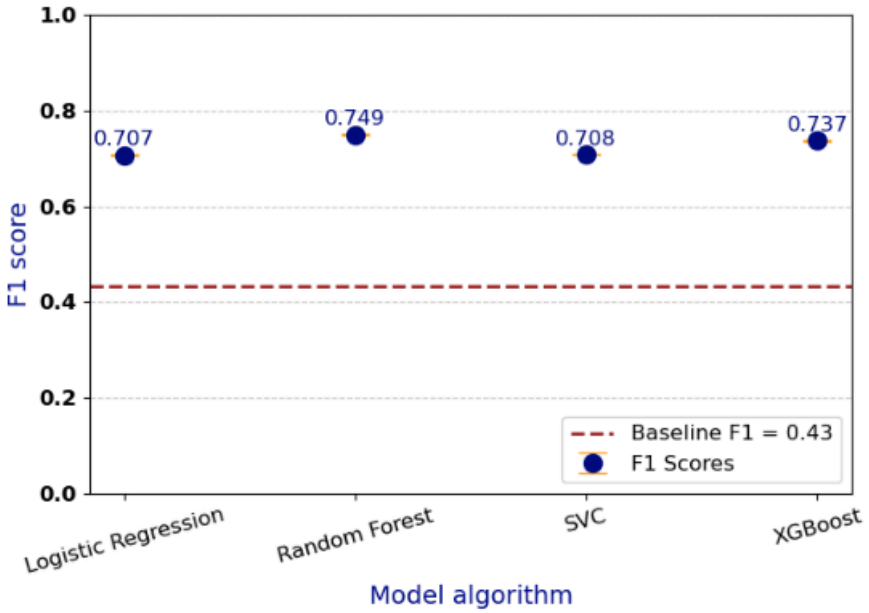
The F1 score was selected for this imbalanced dataset as it provides a balanced measure of recall and precision. Depending on the company's risk tolerance, it could be adjusted to F0.5 (favoring precision) or F2 (favoring recall). Unlike other evaluation metrics, the F1 score does not account for true negatives, focusing instead on the trade-off between false negatives and false positives.

If the goal is to approve as many creditworthy clients as possible while minimizing lost business opportunities, a higher-weighted F-score (e.g., F1.5, F2, or F4) can be used, placing more emphasis on recall. Conversely, if the priority is to be highly precise in identifying clients who will reliably repay their loans, a lower-weighted F-score (e.g., F0.5) should be chosen, favoring precision over recall. For now, unsure in what position that bank is at in terms of their risk-taking ability, we will just use F1 score as an evaluation metric.

The baseline F1 score is 0.43, based on the assumption that all predicted points belong to class 1 for the ‘Charged Off’ variable.

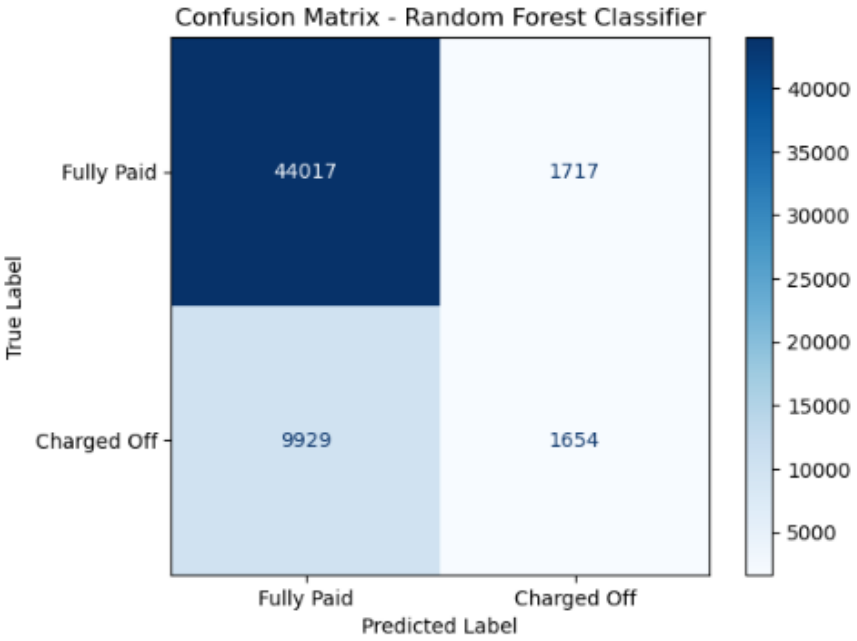
Each model was trained using 5 different random states, and the mean F1 score along with the standard deviation was calculated for evaluation. This approach eliminates any uncertainties caused by variations in the random state.

Results

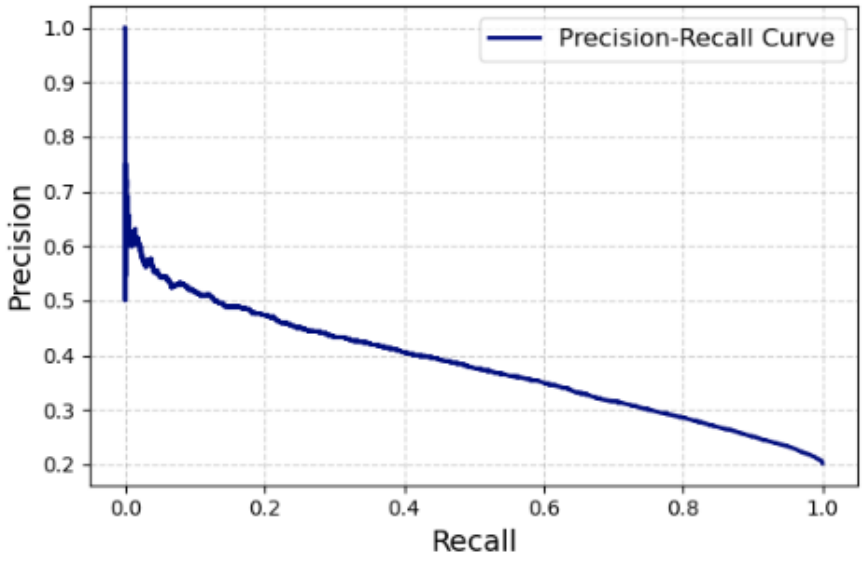


Based on the mean F1 score across the test set, the Random Forest Classifier achieved the highest performance with an F1 score of 0.749. While all models produced relatively similar F1 scores, Random Forest demonstrated a slight edge, as illustrated in the figure below. Additionally, the standard deviation of the F1 scores was very low, indicating consistent

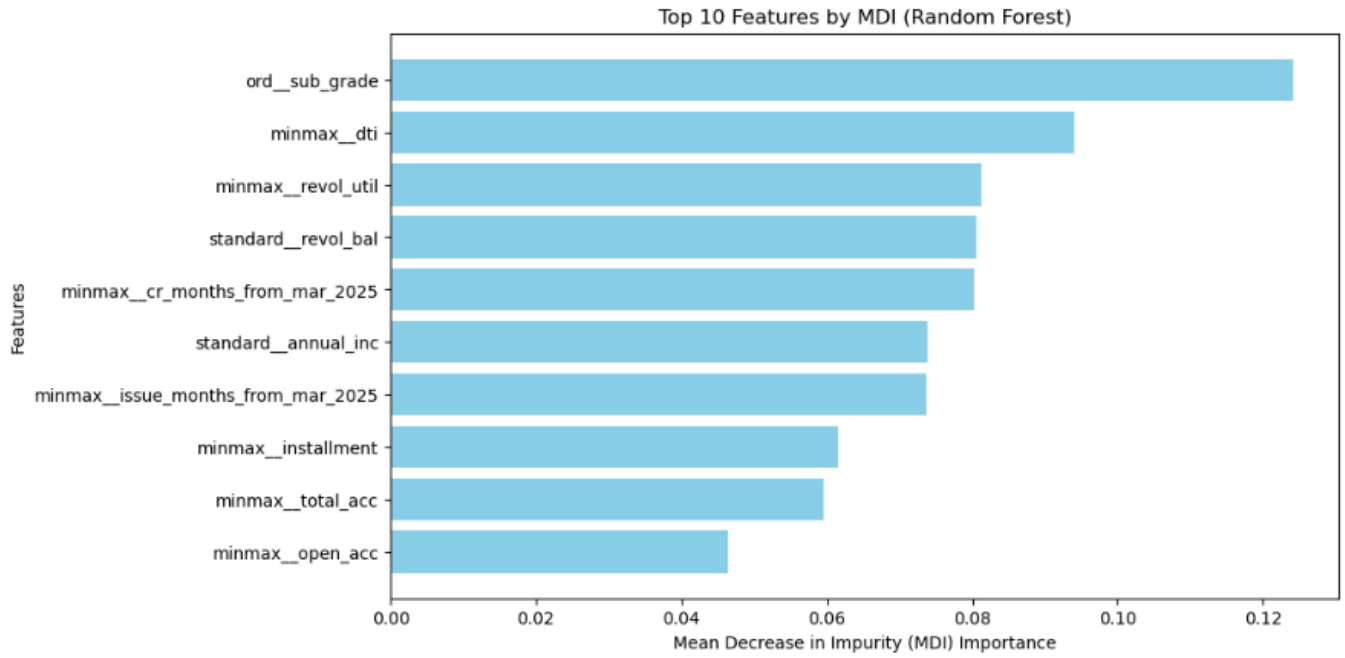
performance across different runs. Notably, all models significantly outperformed the baseline F1 score of 0.43.



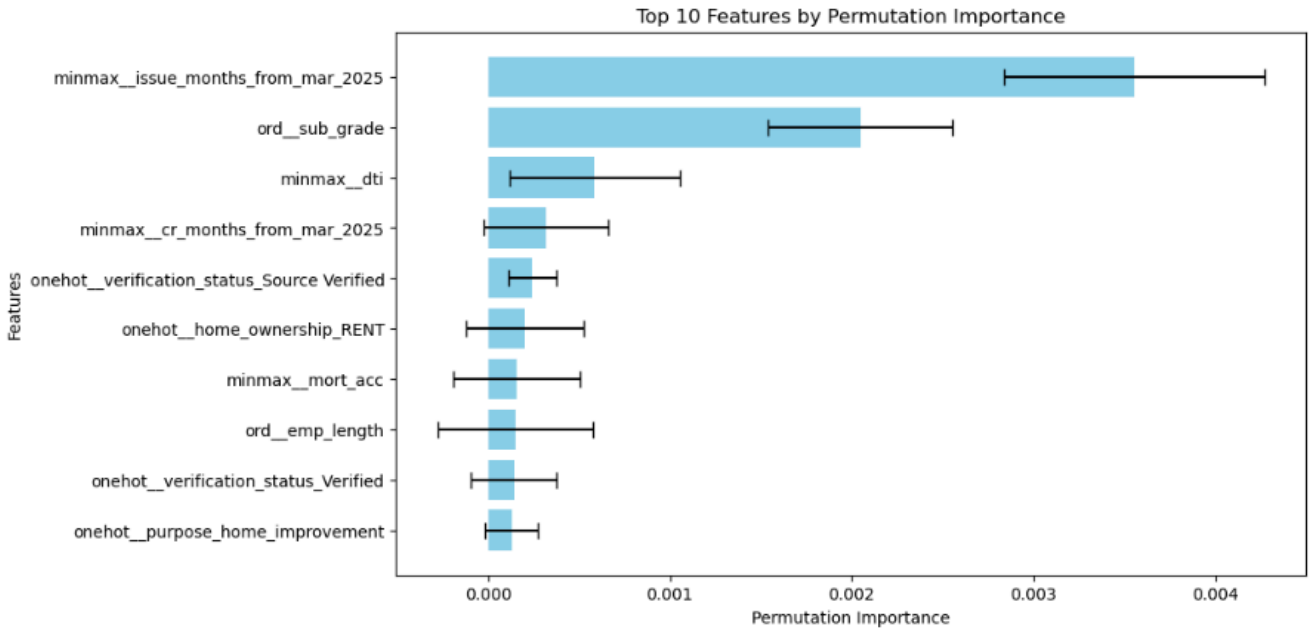
The Random Forest Classifier achieved its highest F1 score of 0.749 on the test set using the selected hyperparameters (max_depth = 30, max_features = 1.0). The model attained an accuracy of 79.7%, with a precision of 0.750 and a recall of 0.797. The balance between precision and recall suggests that the model effectively captures positive cases while maintaining predictive reliability. The confusion matrix further highlights the model's performance, showcasing its ability to distinguish between Fully Paid and Charged Off loans.



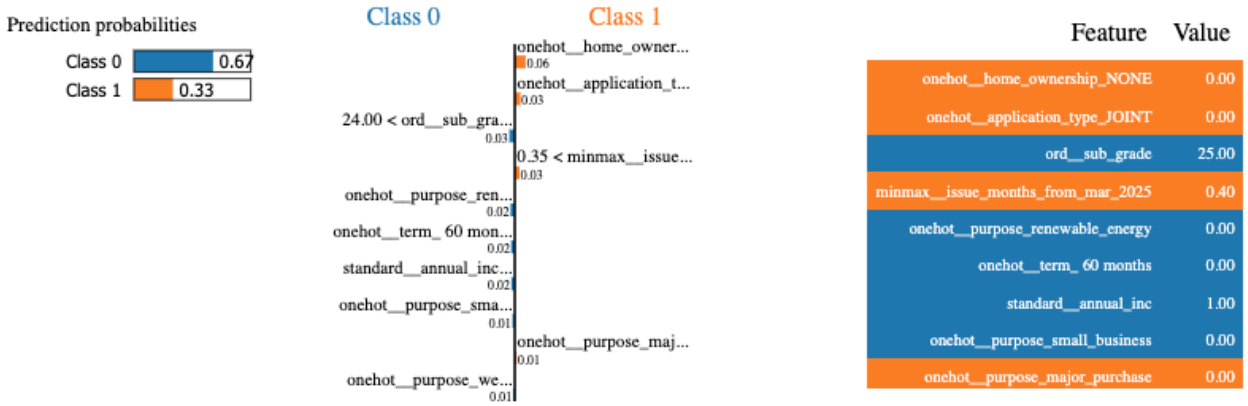
The Precision-Recall Curve shows a sharp drop in precision at low recall values, indicating that the model is highly confident in a small subset of positive predictions. As recall increases, precision steadily declines, suggesting that capturing more positive cases results in more false positives. This pattern is typical in imbalanced datasets, where increasing recall comes at the cost of lower precision. The optimal threshold depends on whether minimizing false positives (high precision) or false negatives (high recall) is the priority.



The bar chart displays the top 10 most important features in a Random Forest model, ranked by their Mean Decrease in Impurity (MDI), which measures the contribution of each feature in reducing uncertainty when making splits. The ordinal-encoded subgrade (ord__sub_grade) is the most influential feature, suggesting that loan subgrades significantly impact predictions. Other key features include debt-to-income ratio (minmax__dti), revolving utilization (minmax__revol_util), and revolving balance (standard__revol_bal), indicating that borrower financial stability plays a crucial role in the model's decision-making. Additionally, credit history-related features (minmax__cr_months_from_mar_2025, minmax__issue_months_from_mar_2025) highlight the importance of credit duration. The results suggest that credit risk assessment heavily relies on loan subgrades and financial behavior indicators.



The chart shows the top 10 features ranked by Permutation Importance in the Random Forest model. minmax__issue_months_from_mar_2025 and ord__sub_grade are the most critical features, indicating that loan issuance time and subgrade significantly impact predictions. Debt-to-income ratio (minmax__dti) and credit history duration (minmax__cr_months_from_mar_2025) also play key roles in assessing credit risk. Categorical features like verification status and homeownership type have lower importance but still contribute. Error bars show variability, with some features having more stable predictive power than others.



The model predicts a 67% probability for Class 0 (Fully Paid), indicating a stronger likelihood of this outcome based on the given features. Higher subgrade values (ord__sub_grade > 24.00) and a recent loan issuance date (minmax__issue_months_from_mar_2025 > 0.35) are the most influential factors pushing the prediction towards Class 0. Meanwhile, categorical features such as homeownership status (NONE) and application type (JOINT) have minimal impact, suggesting they are less predictive for this specific instance. The results imply that loan

subgrades and recency of issuance play a dominant role in the model's classification, while categorical attributes contribute only marginally.

Outlook

After determining the optimal parameters for the Random Forest Classifier (max_depth = 30, max_features = 1.0), we trained the model using 100% of the training data. The model achieved a high F1 score of 0.99, suggesting potential overfitting, but we proceeded with it for now.

The test set initially contained 79,060 data points with an unknown 'loan_status'. After applying the same missing data handling techniques as in the training set, the dataset was reduced to 71,430 data points. The same preprocessing pipeline was applied to handle continuous, ordinal, and categorical features. However, the 'purpose_educational' column was missing in the test set since no loans were categorized under this purpose. To maintain consistency with the training data, this column was added back with all values set to '0'.

The results are available in u_predicted_target.csv, where u represents the row number, and loan_status is the predicted outcome. 93.9% of the predictions were classified as 'Fully Paid', indicating a strong bias toward approving loans. Given this, a higher F-score variant (e.g., F2, F4) may be more appropriate, as the model appears to prioritize minimizing missed approvals over accurately identifying risk.

To improve the model's performance, we need to additionally try other machine learning algorithms like K-Nearest Neighbors (KNN), Naive Bayes, Bagging, and Boosting. These methods could help reduce false negatives and improve recall.

If more computational resources were available with more time, running an SVC on the entire dataset would be worth exploring. SVC is computationally expensive but can handle complex patterns, potentially improving recall and overall accuracy.

To better understand what drives the predictions, we could estimate feature importance using permutation methods or SHAP after dropping correlated variables. Pairwise permutations on two features at a time might also reveal key interactions. These steps would provide deeper insights into the model while helping fine-tune it for higher recall.

https://github.com/adang66/Rocket_Mortgage/tree/main