BT4221 Advanced Analytics with Big Data Technologies
AY 2024/25 Semester 2

Predictive Modeling for Loan Default
Video Presentation: https://shorturl.at/LlJTl

Group 2 Final Report

Members:	**Matriculation Numbers:**
AARON TEO YUAN CAI	A0269646L
ALLAN CHRIS	A0277031L
LEE WEI KIAT	A0273289N
LIANG SHI YIN, MARCUS	A0277877B
TAN KEE XIANG	A0273340M
TONY KOO YE LONG	A0269756H

1. Problem Statement

With an increasing reliance on credit to fund purchases such as houses, automobiles, or even daily essentials, it is becoming imperative for firms offering credit lines to manage their risks accordingly. Organisations offering credit lines must handle their risks appropriately, given the growing dependence on credit for purchases such as homes, cars, and daily necessities. One primary approach is to target specific consumers based on their characteristics and adapt the loan line to their requirements and ability to repay while increasing possible returns and avoiding risks. Otherwise, firms risk the danger of facing similar historical global financial crises (GFC) such as the GFC 2008, where most firms took advantage of repackaging intrinsically bad credit loans into seemingly investable graded mortgage-backed securities (MBS) and sold them to firms and the general public. Consequently, this caused a cascade of firms to collapse, such as Bear Stearns and Lehman Brothers. However, if more light had been shed on borrowers' repayment abilities and their ability to repay the lender in the future, such unfortunate events might have been avoided.

In this project, our group has decided to focus on the peer-to-peer (P2P) lending firms domain. Particularly, our focus in this project is on Lending Club (LC), one of the world's largest P2P firms. LC's primary business operations from 2007 to 2020 focused on providing loans to individuals and companies. Given that LC's portfolio was mostly unsecured loans at about 91% (Figure A1), the group believed that such an arrangement offered a more realistic picture of what elements drive borrowers' inability to make repayments when compared to banks where borrowers are expected to submit collaterals and are rigorously sieved out under tight criterias. At the same time, LC was chosen because it was a leading player in the credit industry, and its data was more readily accessible than banks' data, which is more obscure and difficult to obtain.

Ultimately, our group aims to forecast whether a borrower will default or be able to make timely payments. This slightly differs from our original proposal due to insufficient delayed payments data, because late payments took up only 0.75% of the total loan statuses. Employing the analysis of such data, the group will be able to build a clearer image of the traits motivating a person's success or causing the company to run into financial trouble. By analysing such information, the group can paint a clearer picture of the characteristics driving up a borrower's probability of default as well as hidden trends driving the profits and losses. Such insights could give financial institutions like LC useful data for strategic planning, including identifying the thin boundary between profit and risk. In foresight, this project will carefully make use of Exploratory Data Analysis (EDA) practices, optimisation techniques, and Machine Learning (ML) models such as Logistic Regression (LR) and Random Forest (RF) to extract useful information to help LC gain an edge over its competitors.

2. Dataset

2.1 Source of dataset:

https://www.kaggle.com/datasets/ethon0426/lending-club-20072020q1/data

2.2 General description of the dataset

As a start, our group obtained the LC's dataset from Kaggle on its loans from 2007 to 2020 Q3. The dataset contains information about LC''s customers with dimensions spanning 2,925,493 x 142 (rows, cols) and a size of 1.7 GB. Each row represents a loan listing. The dataset consists of a multitude of features that illustrate the characteristics of the loan, such as the loan amount and the interest rate, as well as information on the borrower, such as their work, the length of their employment, and whether or not they are experiencing financial difficulties. At its core, the dataset allows us to investigate the patterns, trends, and potential factors influencing the possibility of default, such as focusing on borrowers' marital commitment and the loan's structure.

2.3 How is this dataset appropriate for the problem statement

The dataset provided by Kaggle is suitable for our problem statement as it allows us to use the raw data to help us identify whether a borrower will successfully make their payments or default. As the dataset contains an abundance of information on the loan and borrower's characteristics, the group can paint a clearer picture of a person who may default on their payments.

2.4. Exploratory Data Analysis (EDA)

For EDA, our group began by exploring the dataset, encompassing its structure and features (Figure A5.1 & A5.2). One of the objectives of the EDA was to derive behavioural insights of lending institutions and borrowers alike. To understand the patterns behind LC's lending behaviour, we explored a plot of revolving credit utilisation against credit limits. After scaling the axes to remove outliers, the scatter plot generated (Figure B1) suggests that individuals with a lower credit limit have a much higher tendency to have a revolving utilisation percentage exceeding 100%. A value exceeding 100% indicates that an individual has borrowed more than their designated credit limit. Beyond a credit limit value of $125,000, the spread of individuals with a utilisation exceeding 100% diminishes rapidly. This validates the tendency of lending institutions are reluctant to offer additional money to individuals who have no credit available when the absolute amount lent exceeds a certain point, reducing the impact of default.

To explore individuals ' borrowing behaviours, we examined the relationship between annual income and the borrowed loan amount. After taking a sample to generate a visible scatterplot, the plot (Figure B2) validates the expected trend that individuals with higher annual income generally loan more significant amounts. However, some individuals in the lower annual income range (under $50,000) with loan amounts high in proportion to their income remain. These are likely individuals at higher risk of defaulting.

Besides these insights, since the primary objective of our group's analysis is to determine whether a borrower will default on loan payments, our EDA also sought to analyse the relationships of potentially key predictor variables with the 'loan_status' variable to gain insights into defaulters. Before examining these predictors, after reducing the 11 categories in 'loan_status' down to 2, our preliminary analysis of the variable distribution indicates a heavy skew towards loans paid on time (86.9% vs 13.1%), which may require additional handling of the class imbalances (Figure B3).

In identifying the relationship of potential predictor variables with loan defaults, our group analysed trends in the following variables: 'purpose', 'grade', and 'dti' against loan default rates. A quick peek on the analysis of loan counts grouped accordingly to purpose and default status (Figure B4.1) indicates that debt consolidation makes up the majority of loan purposes, accounting for the highest number of defaults. When analysing the default rate by loan purpose (Figure B4.2), the highest default rates are those of educational loans (36.1%) and small businesses (20.9%). Despite debt consolidation (14.1%) and credit card loans (10.7%) reflecting among the middling and lower default rates, the distribution of the number of loan defaults in Figure 4.1 indicates that most defaults stem from these purposes.

An analysis of the loan counts grouped by grade and default status (Figure B5.1) indicates that most loans are issued in the top few grades (A - C), with declining counts for grades D and below, showcasing how most of LendingClub's loans have a lower default risk. However, most loan defaults by grades are also clustered between grades B and D. The analysis of the default rate by loan grade (Figure B5.2) reflects the expected trend, with lower default rates (3.8%) for higher grade loans, ranging upward to the highest default rates (45.1%) for the lowest graded loans.

An analysis of the loan counts grouped by debt-to-income (DTI) ratio and default status (Figure B6.1) indicates that the highest number of defaulters and individuals are between a DTI ratio of 0-30. The further analysis of the default rate by the DTI ratio of individuals (Figure B6.2) indicates an upward trend in the default rate (11.1%-16.4%) for DTI ratios between 0-40. For DTI ratios upwards of 40, the default rate trends downwards, subverting expectations that individuals with a higher DTI and higher relative debt will tend to default more. However, as observed in Figure 6.1, individuals with DTIs exceeding 40 make up a small portion of the total individuals. As such, this anomaly may reflect insufficient statistical power/sampling bias, where the data of this subgroup may not be a generalisable trend.

In preparation for data processing and cleaning, our group also analysed the feature cardinality of all the features(Figure C1.1 & C1.2) and extracted features with high missing or null values (>=80%) (Figure A4).

3. Data Preparation & Cleaning

3.1 How the Dataset was Cleaned

Before making any changes, the group carefully evaluated each feature based on contextual and statistical knowledge. We used the following principles as indicators to help us make better decisions. A summary has been included in the appendix for reference (Figure A2).

3.1.1 Missing Data

Specific columns in the dataset were missing before 2017 because LC had performed an outer join with the data, resulting in the loss of numerous columns of data from 2007 to 2016. Therefore, we have decided to discard data before 2017 to ensure that our data is consistent and complete.

3.1.2 Adding Flags

As some features contained NULL but were intended to be represented as “0” or “Not applicable”, we reproduced additional columns to ameliorate this issue and prevent model training instability. An example would be the “mths_since_last_delinq” column, whereby null means the user has never been delinquent.

3.1.3 Standardization

In some columns, some values meant the same thing but were represented differently. For instance, in the Verification Status column, “Verified” was represented as “Source Verified” and “Verified”. Inconsistencies like this were standardised to avoid misleading the model during training.

3.1.4 Removing Vague Values

Some values provided in the dataset were vague, lacked precise meaning or had irregular values. For example, the value “ANY” appeared in the Home Ownership column, which could represent any home ownership status. Entries as such were removed to ensure that only well-defined categories were used for model training.

3.1.5 Fixing Format

As many columns had inconsistent formatting, our group had to tidy them before they were usable for model training. Several columns, such as int_rate, were stored as strings due to the presence of “%” symbols. Ultimately, these symbols were removed, and the respective column was converted to appropriate numeric types, such as DoubleType, to enable accurate computations and modelling. Additionally, several feature engineering steps were taken to enhance model performance. For example, the mths_since_last_delinq column was binned into categorical intervals to reduce the number of dimensions. Also, we created a binary flag to indicate the presence or absence of delinquency.

3.1.6 Feature Engineering

To enhance our models’ predictive powers, we engineered additional features to capture critical aspects of borrower behaviours and financial health (Figure A3). These features offer a more holistic and nuanced perspective of borrowers’ risks, enabling our ML models to perform informed and accurate predictions.

3.2 Deciding Factors to Keep or Discard Columns

The first deciding factor to discard certain features is based on dropping missing values. Since there were numerous features with missing values greater than 80%, most of which depend on whether the borrower is facing hardships, the majority class did not require it and, hence, was dropped (Figure A4). However, we still need one feature to give importance to the hardship plan to capture the minority likely to default. Hence, we decided to keep one of the features, hardship_reason, while dropping the remaining columns with more than 80% missing values.

Moreover, the second deciding factor is based on our domain knowledge and online research. Looking at the data dictionary of each feature, features that do not help predict a person’s defaulting on a loan are dropped(Figure A5.1 & A5.2). Features such as dti, delinq_2yrs, il_util, and pub_rec_bankruptcies were ultimately chosen as the group inherently felt that these features contribute to a borrower’s probability of defaulting on the loan (Figure A6).

The third factor is based on whether the columns are considered post-event variables, which means the variables would be available at the prediction point. For example, the columns “last_pymnt_amnt” and total_rec_prncp(principal received to date) were removed as such features would not be known at the time of prediction. Retaining these features would lead to data leakage, artificially inflating the model’s performance and causing it to not generalise well to unseen data.

The last deciding factor is to leverage the correlation matrix to drop multicollinear features. Features with a correlation of more than 0.8 were checked between the respective features and determined based on which feature was more important to keep, with the other feature being dropped. Intrinsically, the rationale for discarding highly correlated features before performing PCA is to address the curse of dimensionality. Consequently, performing PCA after highly correlated variables are discarded can help relieve the burden on the PCA and avoid capturing noise. According to Jolliffe (2002), multicollinearity can obscure the interpretation of principal components and lead to components being dominated by highly correlated variables. If highly correlated features were not dropped before PCA, the effectiveness of dimensionality reduction would be reduced. Hence, removing highly correlated features beforehand improves the clarity and efficiency of PCA.

3.3 Deciding Factors to Choose the Target Variable

Since we aim to predict customer defaults, selecting *Loan Status* as the target variable would be suitable. It directly reflects whether a borrower has repaid or defaulted, enabling the subsequent models to investigate historical patterns and make accurate predictions.

3.4 How Categorical Columns Were Handled

The categorical columns were converted to numerical values first using StringIndexers and afterwards one-hot encoded using OneHotEncoding. Doing so ensures no inherent rankings between the different categories, preventing the model from misinterpreting categorical variables as ordinal features.

3.5 Principal Component Analysis (PCA)

Due to the dataset's large volume and high dimensionality, we have performed PCA as part of the dataset cleaning and preprocessing step. PCA helps to reduce the number of features by identifying the most essential features that explain the most variance in the data. The threshold selected was 95%, meaning we retained enough components to explain 95% of the variance in the dataset. By doing this, we have reduced the total number of features from 83 to 62, addressing the curse of dimensionality while preserving most of the original information.

3.6 Handling Class Imbalances

Most borrowers do not default, which makes sense unless the business fails to sustain itself as a credit company; the minority class would be borrowers who default. As observed in our EDA (Figure B3), the defaults comprise roughly 10% of the dataset. Training directly on an imbalanced dataset risks model bias toward majority class predictions, which can inflate overall accuracy while severely underestimating the model’s actual capability to detect defaulters. As such, handling class imbalance would be imperative to ensure that models could effectively identify potential defaulters to prevent potential reduced profitability. To mitigate such risks, our group has employed the following resampling strategies:

3.6.1 Synthetic Minority Over-sampling Technique (SMOTE)

SMOTE is an oversampling technique based on the K-nearest neighbours algorithm to generate synthetic samples of the minority (default) class. By generating synthetic neighbours in the feature space, SMOTE helps to increase the density of minority instances without having to replicate existing records. Consequently, SMOTE assists classifiers in learning better decision boundaries for minority instances. This improves model sensitivity (recall) and performance on evaluation metrics such as PR AUC rather than optimising solely for accuracy (Chawla, Bowyer, Hall, & Kegelmeyer, 2002). Such improvements can be particularly beneficial for classifiers like Logistic Regression (LG), Random Forest (RF), and Support Vector Machines (SVM), which are often sensitive to class imbalance.

3.6.2 Adaptive Synthetic Sampling (ADASYN)

Similarly, ADASYN is an oversampling technique focusing on minority class instances that are harder to learn, such as borderline or noisy samples. This technique can be effective, particularly for cases where borrowers' profiles are ambiguous between default and non-default. This is because ADASYN shifts the focus onto such ambiguous instances to enhance recall without having to oversample well-represented minority instances.

3.6.3 Tomek Links (TOMEK)

On the other hand, Tomek Links is an undersampling technique that removes overlapping majority class samples close to the minority class. This reduces ambiguity at class boundaries, which can help models like RF and SVM form more precise decision boundaries. This potentially reduces false positives and improves precision, enhancing PR AUC.

3.6.4 Edited Nearest Neighbours (ENN)

Moreover, ENN is an undersampling technique to remove both majority and minority instances whose class labels disagree with those of their nearest neighbours. ENN can eliminate mislabeled and noisy records that may otherwise cause erratic decision boundaries for models such as decision trees. As a result, ENN can enhance the model's calibration and generalisation to unseen borrowers to improve recall at low false-positive rates.

3.6.5 Why Random Under/Over Sampling May Be Superficial

Random oversampling techniques on the surface level just duplicate existing minority data randomly. It provides no new information to the dataset and may result in overfitting. As for random undersampling, the dataset's majority data is being reduced randomly to match the ratio of the minority data better. This may result in information loss on the majority of the dataset. Hence, random undersampling and oversampling are not used as the techniques mentioned above are used to help address the limitations.

3.7 Sampling

Due to the limitations of Google Colab Pro and limited access to higher processing clusters, after trial and error, our group has concluded that randomly sampling 5-10% of the data (approx. 100k rows) is the sweet spot to train our models before running into memory errors. We will be using SMOTE, ADASYN, TOMEK, ENN, SMOTE & ENN, and SMOTE & TOMEK for modelling and comparing the results of each sampling method.

4. Machine Learning (ML) Models

Our group has decided to focus on the following six ML models for their varying algorithmic approaches to provide a more diverse range of insights. These models include Logistic Regression, Naïve Bayes, Decision Tree, Random Forest, Gradient-Boosted Trees, and Support Vector Machine. In this section, the group will explain our rationale for choosing the respective models, their advantages, and their potential drawbacks.

Our group adopted a structured model validation strategy to evaluate the upcoming models fairly. PR_AUC, a commonly used metric for imbalanced classification problems, will be the primary metric to assess the model's performance. This metric effectively allows the team to analyse the model's ability to pick up potential defaulters correctly while minimising false alarms. Our secondary metric will be recall; this is due to the nature of the project to identify potential defaults, which we felt takes precedence given the possible severity of missing out on false positives, which could potentially harm credit companies' ongoing concerns. Precision also remains vital to avoid falsely rejecting applicants who would pay back, resulting in a loss of profits.

To interpret these metrics, A high PR_AUC indicates that the model can better identify defaults with minimal false flags. At the same time, a high recall suggests that the model can detect most of the defaulters. In contrast, high precision means that among all the customers predicted to default, a high percentage are actual defaulters. Ideally, we would want all three metrics to be as high as possible, but there will likely be situations where we must compromise one for the other. Therefore, selecting the best model involves metrics balancing to minimise financial loss and ensuring profit maximisation.

Additionally, three-fold cross-validation will be performed for all of our models' training to ensure a reliable and accurate estimate of the evaluation metrics. Three-fold cross-validation splits the data into three subsets, changing the training and validation sets across iterations. It then takes the average of the performance metrics, giving a more robust and unbiased estimate of the model's effectiveness.

4.1 Logistic Regression

Since Logistic regression is commonly used for classification tasks, our group has decided to use it as the fundamental model because of its simplicity and speed. It works by assigning a weight to each feature and applying the sigmoid function to output a probability between 0 and 1. Logistic regression uses gradient descent to find the optimal weights, updating its weights iteratively to minimise the loss function. Regularisation techniques such as Ridge, LASSO, and Elastic Net can also improve generalisation and prevent overfitting.

The base parameters used for logistic regression are: regParam = 0.01, elasticNetParam = 0.5, threshold = 0.5, and maxIter = 100. Table D1 shows the results for each logistic regression model using the different sampling methods. Based on the results shown in Table D1, the model performs the best when using hybrid sampling (SMOTE & ENN).

Although the accuracy is high for the base random sampling method at 0.9128, its recall is very low at 0.0007, suggesting that the model is most likely biased towards predicting everything as payment on time. This trend can also be observed for undersampling techniques like TomekLinks and ENN, indicating that the undersampling may not be viable alone. As for SMOTE & ENN, the model achieved a high recall of 0.8655 and PR_AUC of 0.2116, but with a low accuracy of 0.4874 and a precision of 0.1308. This approach is suitable for capturing possible defaults but comes at the cost of falsely flagging out individuals. For SMOTE, ADASYN, and SMOTE & TomekLinks, the recall achieved is lower, ranging from 0.7187 to 0.7441, but with slightly higher precision, ranging from 0.1519 to 0.1561, and higher PR_AUC, ranging from 0.2122 to 0.2154, compared to SMOTE & ENN. These approaches reduce the number of false flags but also capture fewer defaults. Since our primary focus is capturing defaults while maintaining a reasonable trade-off with precision, the most suitable sampling method would be SMOTE & ENN.

Using hybrid sampling with SMOTE & ENN, a grid search was used to find the best set of hyperparameters for logistic regression. The grid is defined as follows: regParam = [1, 0.1, 0.01, 0.001] and elasticNetParam = [0, 0.5, 1], and threshold = [0.3, 0.5, 0.7]. The best results were when regParam = 0.001, elasticNetParam = 0, and threshold = 0.3, as shown in Table D2. The recall improved significantly at the cost of a drop-off in precision. This means the model predicts more defaults at the cost of falsely flagging individuals paying on time.

4.2 Naïve Bayes

Furthermore, our group has explored the Naïve Bayes classifier to supplement our predictive prowess further. Since Naïve Bayes is based on conditional probability and essentially the classifier infers each feature as independent, it may not work well with extracting insights from the LC dataset. This is because LC dataset's borrower characteristics, such as income, FICO scores, and loan purposes, are often linked. As a result, the independence assumption may not hold. Nevertheless, the model remains a valuable benchmark for its computational efficiency, interpretability, and ability to uncover initial trends within the data.

To ensure that the Naïve Bayes model performs optimally, we conducted hyperparameter tuning on the best model using Grid Search with the smoothing parameter of [0.1, 0.5, 1.0, 1.5, 2.0] and model evaluation using three cross-validation folds. In hindsight, the diverse range of smooth parameters helps to mitigate the impact of zero probabilities by adjusting the probability estimates. At the same time, the three-fold cross-validation ensures the generalisability of the selected smoothing parameter while reducing risks of overfitting and ensuring that computational efficiency is not strained.

After extracting the performances from the model (Table D3), we can observe that Naïve Bayes struggles to identify correct positive cases (precision) more than Logistic Regression, with ranges of 0.1260 to 0.1560 and recall ranging from 0.1070 to 0.7770 along with an accuracies of 0.4874 to 0.9128. Additionally, Naïve Bayes obtained PR AUC scores ranging from 0.08830 to 0.08840. In essence, while Naïve Bayes demonstrated stability across different sampling methods, its precision and PR AUC performance remains comparatively weaker than Logistic Regression. This reflects the importance of selecting models that can better capture feature interdependencies when modelling borrowers' behaviour.

After conducting hyperparameter tuning using Grid Search and three-fold cross-validation on the best-performing sampling method (SMOTE & ENN), we extracted the performances from the tuned Naïve Bayes model (Table D4). The model achieved a precision of 0.1260, a recall of 0.7770, and a PR AUC of 0.08840. While the recall remains high to identify potential defaulters, the lowered precision may suggest a trade-off with increased false positives. Nevertheless, Naïve Bayes maintains computational efficiency and may be suitable where identifying defaults is prioritised over precision.

4.3 Decision Tree

With its tree-like structure, the decision tree model is intuitive when interpreting a classification problem. The more critical the feature, the closer it will be to the tree's root. The features at the top of the tree would gain the most information in distinguishing who will likely default and who will pay on time. Moreover, Decision trees do not make assumptions about the data distribution, making them suitable for big data with complex, non-linear relationships like our dataset.

The base parameters used for the decision tree are: maxDepth = 5, minInstancesPerNode = 1, and impurity = gini. Table D5 shows the results for each decision tree model using the different sampling methods. The accuracy of the base model is high at 0.9128 but has a very low recall of 0.0055, suggesting that the model is more biased towards predicting on time than default. This trend is also similar for the undersampling method as the base and undersampling model is trained on significantly more on time label than default label. As for SMOTE, ADASYN, and SMOTE & TomekLinks have a lower accuracy than the base and undersampling model ranging from 0.5580 to 0.5977 but higher recall ranging from 0.6671 to 0.7409. Lastly, SMOTE & ENN although has a lower accuracy of 0.4448, it scored a significantly higher recall at 0.8565 with slightly lower precision of 0.1209 than the oversample and SMOTE & TomekLinks model. With a high recall and slightly lower precision, SMOTE & ENN has a PR_AUC of 0.1299 which is between the range of PR_AUC of oversampling and SMOTE & TomekLinks ranging 0.1237 to 0.1365. In conclusion, the model performs best when using hybrid sampling (SMOTE & ENN) as it has the best recall and second best PR_AUC.

Using SMOTE & ENN, gridsearch was used to find the best hyperparameters. The grid is defined as follows, and it is checked for the best model for each maxDepth with these configurations: MaxDepth = [3, 5, 8], minInstancesPerNode = [1, 5, 10] and impurity = ['gini', 'entropy']. After running the model, the best results were when maxDepth = 8, minInstancesPerNode = 5 and impurity = gini, maxDepth = 3, minInstancesPerNode = 1 and impurity = gini and the original base model at maxDepth = 5, minInstancesPerNode = 1, and impurity = gini.

Table D6 shows that the grid search for maxDepth = 8 has increased the PR_AUC but decreased the recall. This is likely to be caused by the decision tree trying to overfit the training data, which results in making the model more complex and making more splits. More depth in the tree allows the decision tree to learn the specific data of the on-time, which is generally easier to predict than that of defaulters. Hence, having more depth in the decision tree makes it likely to have learned the on-time pattern, resulting in a less effective model in identifying default cases and a decrease in recall.

For maxDepth = 3, the overall metrics for all are lower than the base model of maxDepth = 5, making it worse. This means that at maxDepth 3, the tree is too shallow and cannot capture the critical distinctions between the defaulters and on-time. Hence, the best model would still be the base model at maxDepth = 5, giving the most balanced result with higher recall than maxDepth = 8.

4.4 Random Forest (RF)

Random forest builds on the decision tree model by creating multiple decision trees during training. Instead of relying on a single tree, RF aggregates the predictions of many trees, making it more robust and accurate. This makes it suitable for our problem, as it helps to address the significant imbalance of default and on-time payments by helping prevent overfitting and improving the model's generalisation to unseen data. It is also suitable since it can capture complex patterns in borrower characteristics.

The base parameters used for the random forest model are (numTrees: 50, maxDepths: 10, maxBins: 32, featureSubsetStrategy: "auto", impurity: "gini"). Table D7 shows the results for each random forest model using the different sampling methods. Based on the results, the model performs best using hybrid sampling (SMOTE & ENN) with the best recall and decent precision. It has a recall of 0.8242, F1_Score of 0.2305 and PR_AUC of 0.1883.

Using hybrid sampling with SMOTE & ENN (Table D8), gridsearch was used to find the best set of hyperparameters for logistic regression. The grid defined is as follows: numTrees = [25, 50], maxDepth = [5,10], maxBins = [16,32], featureSubsetStrategy = ['auto',' sqrt','log2'], impurity = ['gini',' entropy']. The best results were when numTrees = 50, maxDepth = 10, maxBins = 16, featureSubsetStrategy = 'sqrt' and impurity = 'gini'. Virtually, the PR_AUC increased slightly (0.0007), recall improved from 0.8242 to 0.8261 (0.0019), and precision dropped by 0.0003. Unfortunately, due to memory limitations, the numTrees and maxDepth hyperparameters cannot run values greater than 50 and 10, respectively.

4.5 Gradient Boosted Tree (GBT)

GBT (GBTClassifier) builds an ensemble of decision trees, but each new tree is trained to fix the errors of the previous trees. Unlike RF, which trains trees independently and aggregates their results, GBT trains trees sequentially, with each tree focusing on the residuals of the previous ensemble. This implementation in PySpark uses stochastic gradient boosting and subsampling of previous trees to reduce overfitting. It minimises a loss function by using gradient descent across the trees. This sequential learning approach can help predict rare cases (default) better and boost the influence of minority cases during training.

The baseline models are trained using default parameters to evaluate which sampling technique is the best. Hence, in Table D10, it was that SMOTE & ENN is the most optimal because while all resample techniques results with the PR_AUC ranges from 0.1681 to 0.2381, SMOTE & ENN has the highest recall, while the other resampling techniques that have higher PR_AUC have recall less than 0.1, meaning the model was unable to detect defaulters. Though using SMOTE & ENN sacrifices accuracy, we aim to spot the defaulters. Hence, the hybrid sampling SMOTE & ENN was deemed the most optimal.

Using hybrid sampling with SMOTE & ENN, a grid search was used to find the best set of hyperparameters for GBT. The grid is defined as maxDepth = [5,7], maxIter = [20,30] and stepSize =[0.05, 0.1]. The maxDepth controls the maximum depth of each tree, increasing this to capture more complex patterns but risking overfitting. maxIter controls the number of boosting iterations, where more iterations lead to better performance but increase training time and risk of overfitting. Lastly, stepSize refers to the learning rate, which controls how much each tree contributes to the final prediction; a too high learning rate can overshoot the optimal solution.

After Grid Search, these hyperparameters: (maxDepth = 7, numTrees = 30, stepSize = 0.1), gave the optimal PR_AUC (0.1841). The final GBTClassifier, as seen in Table D11, achieved a recall (0.7451), indicating that it is highly effective at identifying potential defaulters. However, this comes at the cost of precision (0.1376), suggesting a higher rate of false positives. The F1-score of 0.2324 reflects this trade-off, which is acceptable given the goal of minimising undetected defaults.

4.6 Support Vector Machine (SVM)

Support Vector Machines (SVMs) seek an optimal hyperplane that maximises the margin between classes in a high-dimensional space. They are effective in high-dimensional spaces, such as PCA-reduced data, while resisting overfitting. However, since PySpark's LinearSvc is designed for linear classification, it cannot model non-linear relationships between features and the target variable. As such, if non-linear relationships exist within the PCA-transformed feature space, other models may be better suited to capturing them.

The base model had the following hyperparameters (maxIter=50 and regParam=0.1): The maxIter parameter controls the maximum number of optimisation iterations, balancing model convergence with training times. The regParam parameter controls the regularisation strength of the model, penalising significant coefficients to prevent overfitting. This balances the smoothening of decision boundaries with model overfitting.

As shown in Table D12, the base model performance of a high accuracy (0.9131) but near-zero recall (0.0062) indicates severe bias toward the majority class, demonstrating the model's ineffectiveness in predicting defaulters. The SMOTE and SMOTE & TomekLinks models have almost identical results, sharing a similar training dataset. SMOTE & TomekLinks removed a negligible 21 samples from the majority class, reflecting their near-identical results with the SMOTE model.

Among the models (Table D12), the SMOTE model stands out with its balance between recall (0.6999) and accuracy (0.6352) while also having the best PR_AUC value (0.1874). The SMOTE & ENN model also has a comparable PR_AUC value (0.1706) with the SMOTE model while boasting a significant recall advantage (0.9068 vs 0.6999) over the SMOTE model.

Using both SMOTE and a hybrid sampling of SMOTE & ENN, a grid search was conducted with the following hyperparameter values: maxIter [50,100] and regParam [0.05, 0.1, 0.5], validated with a three-fold cross-validation.

Both models (Table D13) had the same best hyperparameters of maxIter=100 and regParam=0.5, and both models improved in the PR_AUC metric. Despite SMOTE and ENN having a very high recall score, they have low precision and are less practical in a real-world setting. As there is a significant tradeoff in denying profitable loans, the SMOTE model is better suited with its balance of precision and superior overall PR_AUC.

4.7 Best ML Model Among the Six

4.7.1 Selecting the Best Model (Random Forest)

After analysing the best performances for each model type in Table D14, the group concluded that the **Random Forest** model performs best. While models like Logistic Regression and Decision Tree displayed their ability to capture defaulters with a high recall, their precision was simply too low. The scores suggest that the model overpredicts the minority class, which is the *default class* in this case, resulting in many false alarms. This one-sided behaviour exhibited makes it unsuitable for real-world applications. As expected, because of its independence assumption, Naive Bayes has the worst performance overall, achieving low recall and precision scores. As for Gradient-Boosted Tree and Support Vector Machines, while their precisions are higher, their recalls are not ideal compared to Random Forest, suggesting that they cannot capture default classes well. After careful analysis of the results, the team believes that the Random Forest model has the most balanced performance, being able to capture most defaulters with minimal false alarms. Therefore, the team decided to make further improvements and analysis to the Random Forest model.

4.7.2 Feature Selection for Best Model (Random Forest)

Performing feature importance on the Random Forest model would return these Principal Components(PC) in Figure D15. However, since each PC is a linear combination of each original column, we cannot evaluate what each PC means. Therefore, we start by extracting the PCA loadings (Figure D16), indicating how strongly each original feature contributes to each PC. However, the categorical values are not represented well since it was one-hot encoded. Henceforth, the sum of the absolute loadings of each categorical value was grouped back into their original features. For example, grade_encoded_0, grade_encoded_1, and grade_encoded_2 will be combined and represented as "grade" instead. This transformation allows us to easily identify the most essential features in the PC, giving us a clearer understanding of which features are the most important.

Looking at the highest importance PC, PC5, the top five representations are mths_since_last_delinq (0.274), grade (0.492665), hardship_flag (0.355), hardship_reason (1.192), and home_ownership (0.294). From PC5, it is clear that hardship_reason is the most dominant feature with the most extensive loading. This indicates that PC5 is primarily influenced by factors related to a borrower's hardship situation. Grade and home_ownership also contribute to PCs, but at a lower rate. PC5 could be interpreted as "Borrower's Hardship" since it contains hardship-related characteristics.

Next, using the formula of PC importance and PCA Loadings, the results were multiplied to attain the overall weighted importance for each original feature (Figure D17). This avoids treating all PCs equally, ensuring that features influencing more important PCs are weighted more, resulting in a more accurate and meaningful ranking for the feature importance. From the results, hardship_reason, purpose, grade, mths_since_last_delinq, and home_ownership are the most influential features for the random forest model.

After determining feature importance, a random forest model was re-trained without the bottom 10 features shown in Figure D17. Removing the least essential features could be helpful as their presence does not contribute to the model's predictive power and could even introduce noise, reducing the model's accuracy. Moreover, it takes up additional computational resources, leading to longer training time. Table D9 shows the results of the model trained without the bottom 10 features. Based on the table, even though there is a slight increase in accuracy, F1_Score, and PR_AUC, it led to a drop in recall. Even though the model without feature selection is more balanced overall, recall is critical in our context, and failing to identify potential defaulters could lead to significant financial risk. Therefore, the original model is preferred since it achieves a higher recall and can catch more defaulters.

5. Discussion of Results

5.1 Insights

5.1.1 Evaluation of "Test" Results

To ensure a fair and unbiased evaluation of the model performance, we retrieved 20% (200,807 rows) of the unused data from the remaining 95% left from sampling and used it for evaluation purposes. As shown in Table 18, the performance on the final test set is similar to the original test set. Key evaluation metrics such as PR_AU, recall, and precision remain identical, suggesting that the model could generalise well and perform well against unseen data. Following this, we plotted the predictions to analyse the correct and incorrect predictions for the top features identified earlier.

5.1.2 Hardship_reason (Figure E1)

Starting from hardship_reason, which has the highest weighted importance, we notice that most predictions are under the "NA" category. Despite the skew, the model can correctly classify less frequent categories like income_curtailment and unemployed, suggesting that it can accurately predict hardship.

5.1.3 Loan Purpose Prediction (Figure E2)

The following important feature, loan purpose, shows many true negatives in categories like credit_card and debt_consolidation. However, the model makes many default predictions, leading to many false positives, indicating misclassification of on-time borrowers as defaulters. However, false negatives are minimal across the categories, supporting the model's high recall.

5.1.4 Grade Prediction (Figure E3)

Higher grades like "A" have the model predicting mainly on time. Even though the model attempted to predict defaults, it could not accurately do so for grade A. In contrast, grades B, C, and D show cases where the model could predict defaults more accurately at the cost of increased false positives. The imbalance between false and accurate predictions highlights the model's tendency to over-predict default; however, given the context, it may be acceptable, as false positives are less costly than false negatives.

5.1.5 Mths_since_last_delinq (Figure E4)

The results performed the same across all delinquency categories, with the most predictions on false positives, followed by true negatives, true positives, and very few false negatives. This suggests that historical delinquency alone may not be sufficient for determining defaulters, and new features may be needed to improve the model.

5.1.6 Home_ownership (Figure E5)

The RENT group has the highest number of true positive from the homeownership categoriess, suggesting that the model effectively predicts defaulters who rent their houses. The model is better at predicting on-time customers in the MORTGAGE group, while not performing as well in the own group.

5.2 Challenges

5.2.1 Computational Limits

Google Colab's out-of-memory error was a common challenge, especially when using a large dataset or complex models. Colab only offers limited resources, such as RAM, to free-tier users, which will disconnect or crash the runtime when the limits are exceeded. To mitigate this issue, the data for model training was sampled, variable references were deleted before training the next model, and/or multiple sessions were created to ensure the limit was not reached unexpectedly.

5.2.2 Limitations of Fine-Tuning the Models

Similarly, due to computational limits, hyperparameter tuning for some models was difficult, as some hyperparameters could not be computed, leading to crashes or indefinite run times. Thus, the parameter grid had to be minimised for computational reasons.

5.2.3 Lack of PySpark Tools for Imbalanced Data

There is an issue of imbalanced class when it comes to identifying loan defaults. In the dataset, the majority of loans were classified as usual, while defaults were of a much smaller proportion. To tackle this, a resampling technique needed to be done. Python libraries like imbalanced-learn offer a variety of techniques, such as SMOTE, ADASYN, and hybrid methods, to address this issue. However, similar libraries are not integrated into PySpark. Thus, to implement this, the Spark DataFrame had to be converted into a Pandas DataFrame and then converted back into a Spark DataFrame for the modelling after the resampling was completed.

5.3 Assumptions

5.3.1. Late Payments Are Categorised as Defaults

Due to insufficient delayed payments data, where late payments took up only 0.75% of the total loan statuses, we decided to consolidate them together with default, as late payments often foreshadow impending defaults.

5.3.2. Assumptions Regarding PCA and Information Preserved

Our project uses PCA for dimensionality reduction, assuming that the principal components capturing the most variance also retain the most helpful information for predicting loan default. By retaining 95% of the variance, we can preserve the essential predictors in categorising default and on-time borrowers.

5.3.3. Data Sampling Due to Computational Constraints

Due to the computation limitation, our decision to sample 5-10% of the data was necessary to relieve computational burdens. Given the sample, we assume that while using the full dataset would be ideal, the chosen random sampling is sufficient to capture underlying patterns and train reasonably robust models.

5.3.4 Dataset Accuracy

Our project assumes the dataset is accurate, particularly regarding whether a borrower has defaulted. These labels are critical for the modelling process, and any mistakes in these labels can potentially affect the model's reliability and performance.

5.4 Improvements

5.4.1. Leveraging the Full Dataset

Since our current analysis is based on a 5% sample due to the computational constraint, utilising the entire dataset would enable a more comprehensive and representative understanding of the underlying patterns. More data would also allow our model to learn a more robust and generalised relationship, capturing subtle but essential signals that might be missed in smaller samples. Hence, the increase in data would allow for more reliable model training and evaluation, leading to more confident and accurate predictions of loan default risk.

5.4.2. Explore More Complex Models

While traditional machine learning models like Logistic Regression and Random Forest can capture many patterns from the data, exploring more complex models such as neural networks offers the potential for even more sophisticated modelling capabilities, better suited to learn the increasingly complex and non-linear relationship between features

5.4.3. Threshold Optimisation

A more detailed analysis of the optimal threshold for the classification model could be conducted based on the precision-recall curve. The optimal threshold should be set to address the Lending Club's business needs and risk tolerance. This data-driven approach will ensure the model's predictions are utilised to maximise value and minimise potential negative impacts.

6. How can these insights help Lending Club?

Combining everything, our findings can provide LC with robust, data-driven insights to enhance risk management. With the Random Forest model (Table D8) achieving a recall of 0.8261 and a PR AUC of 0.1890, the model offers LC a channel to identify potential defaulters early for improved loan approval processes. Since the analysis revealed that borrowers in lower loan grades, such as grade G, face default rates as high as 45.1%, in contrast to 3.8% in higher loan grades (e.g., grade A), it is recommended for LC to focus its efforts to tailor accordingly for lower-graded borrowers. Similarly, as default risks increase with debt-to-income (DTI) ratios, peaking at 16.3% between 30 and 40, it provides an avenue for LC to explore potential strategies that can improve both returns and risk exposure. Fundamentally, these insights point to an opportunity for LC to adopt a tiered risk-pricing model to offer competitive rates for different types of borrowers. To paint a clearer picture, risk mitigation can be achieved either by tightening eligibility criteria or applying higher interest rates to riskier segments. Essentially, this approach can improve profitability while mitigating LC's credit risks.

As features such as home_ownership, loan_purpose, and hardship_flags emerged as significant predictors of default, they provide a glimpse of the various borrower and loan characteristics that LC should focus on to manage risks more effectively. For instance, customers who rent instead of owning their own homes showed the highest true positive rates for default. On the other hand, hardship indicators like income_curtailment were simultaneously flagged out by the model. Instrumentally, Lending Club can leverage on these findings to develop more granular borrower profiles. Afterwards, the company can refine its targeting strategies accordingly. To further add value, LC can prioritise applicants with mortgage ownership or stable employment, along with those borrowing for small business or educational purposes, as these profiles may offer better risk-return balances. By integrating these insights into its customer acquisition processes, LC can reduce non-performing loans, improve investor confidence, while creating a more sustainable, risk-adjusted lending portfolio.

Ultimately, our end-to-end ML pipeline, from EDA to model optimisation and finally, evaluation, demonstrates how ML can empower credit firms like LC to make more informed decisions. By translating raw data through a series of intense refinements, pure numerical and categorical data can be transformed into value-adding, actionable insights. Despite the numerous limitations our group has faced, we were able to produce insights that can set the ball rolling for credit firms such as LC to mitigate risks more effectively. If given access to higher computational resources, we firmly believe that our group could produce industrial-level insights that add immense value to credit firms. Ultimately, our group reckons that by adopting data-driven strategies like ours, credit firms such as LC can set the tone of a firm that is able to draw the delicate balance between profits and risks to prevent another financial catastrophe like the 2008 crisis from happening again.

References

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic Minority Over-sampling Technique. Journal of Artificial Intelligence Research, 16, 321–357. https://doi.org/10.1613/jair.953

Jolliffe, I. T. (2002). Principal Component Analysis (2nd ed.). Springer. https://doi.org/10.1007/b9883

Juez-Gil, M., Arnaiz-González, Á., Rodríguez, J. J., López-Nozal, C., & García-Osorio, C. (2021). Approx-SMOTE: Fast SMOTE for Big Data on Apache Spark. Neurocomputing, 464, 432–437. https://doi.org/10.1016/j.neucom.2021.08.086

Swastik. (2025, April 4). Overcoming class imbalance using SMOTE techniques. Analytics Vidhya. https://www.analyticsvidhya.com/blog/2020/10/overcoming-class-imbalance-using-smote-techniques/

Appendix A: Dataset

LENDINGCLUB CORPORATION
LOANS AND LEASES HELD FOR INVESTMENT
(In thousands)
(Unaudited)

	September 30, 2022	December 31, 2021
Unsecured personal	$ 3,642,254	$ 1,804,578
Residential mortgages	197,776	151,362
Secured consumer	180,768	65,976
Total consumer loans held for investment	4,020,798	2,021,916
Equipment finance [1]	167,447	149,155
Commercial real estate	372,406	310,399
Commercial and industrial [2]	246,276	417,656
Total commercial loans and leases held for investment	786,129	877,210
Total loans and leases held for investment	4,806,927	2,899,126
Allowance for loan and lease losses	(303,201)	(144,389)
Loans and leases held for investment, net	$ 4,503,726	$ 2,754,737

(1) Comprised of sales-type leases for equipment.
(2) Includes $89.4 million and $268.3 million of Paycheck Protection Program (PPP) loans as of September 30, 2022 and December 31, 2021, respectively. Such loans are guaranteed by the Small Business Association and, therefore, the Company determined no allowance for expected credit losses is required on these loans.

Figure A1: LC's portfolio as of Q3 2022
https://s24.q4cdn.com/758918714/files/doc_financials/2022/q3/LendingClub-3Q22-Earnings-Release.pdf

Data_Cleaning_Summary

Columns Affected	What Changed	Rationale
issue_d	Parsed into 'issue_d_parsed' and filtered for >=2017	Focus only on relevant, recent loans and drop missing dates
hardship_flag & hardship_reason	Removed rows where hardship_flag='Y' and hardship_reason=NULL, then filled NULLs with 'NA'	Rows with inconsistency were dropped; remaining NULLs treated as Not Applicable
emp_length	Cleaned text, created employment flag, converted to numeric (0–10 scale)	Standardized employment duration and separated unemployed borrowers
home_ownership	Filtered only RENT, OWN, MORTGAGE, OTHER	Dropped ambiguous/rare categories like ANY and NONE
verification_status	Merged 'Source Verified' into 'Verified', removed anomalous '38000'	Fix labeling inconsistencies and remove dirty records
purpose	No cleaning required	Values are already standardized
avg_cur_bal	Dropped rows with NULL avg_cur_bal	NULLs treated as outliers and removed due to small missing proportion
dti	Dropped rows with NULL dti	NULLs treated as outliers
il_util	Dropped rows with NULL il_util	NULLs treated as outliers
mths_since_last_delinq	Created 'had_delinquency_flag'; grouped into bins (never, 0-6 months, etc.)	Capture delinquency history meaningfully and group missing values as 'never'
pct_tl_nvr_dlq	No action needed (no NULLs)	Already clean
percent_bc_gt_75	Dropped rows with NULL percent_bc_gt_75	NULLs treated as outliers
int_rate	Removed '%' and cast to double	Standardize numeric type for modeling
revol_util	Removed '%' and cast to double, dropped NULLs	Standardize numeric type and ensure clean input
bc_util	Dropped rows with NULL bc_util	NULLs treated as outliers
loan_status (outcome)	Mapped loan_status into binary outcome: 1 = problematic (default/late/charged-off), 0 = on time	Simplify into binary classification for clearer modeling
Feature Engineering	Created 'credit_util_ratio', 'income_to_loan_ratio', 'installment_to_income_ratio', and 'delinq_flag'	Derived additional predictive features for better model performance

Figure A2: A summary of the data cleaned

New feature	Formula	Rationale
credit_util_ratio	*Tot_cur_bal / tot_hi_cred_lim*	Measures how much of their available credit a borrower is using. A higher ratio may indicate a higher credit risk, as the borrower is closer to their credit limit. Helpful in assessing financial stress levels.
income_to_loan_ratio	*Annual_inc / loan_amnt*	Evaluates the borrower's ability to repay the loan. A higher ratio suggests the borrower earns significantly more than the loan amount requested, implying a lower risk of default.
installment_to_income_ratio	*Instalment / annual_inc*	Assesses the burden of the monthly loan instalment relative to monthly income. A higher value indicates that a larger portion of the borrower's income is tied to loan repayment, signalling potential difficulty in repayment.

Figure A3: Newly Engineered Features

Column	Missing_Count	Missing_Percentage
hardship_loan_status	2782082	95.09788606569902
hardship_reason	2781861	95.09033178339514
hardship_status	2781858	95.09022923657653
hardship_dpd	2781856	95.09016087203081
deferral_term	2781855	95.09012668975794
hardship_start_date	2781855	95.09012668975794
hardship_end_date	2781855	95.09012668975794
payment_plan_start_date	2781855	95.09012668975794
hardship_length	2781855	95.09012668975794
hardship_type	2781853	95.09005832521218
orig_projected_additional_accrued_interest	2746253	93.87316941110439
hardship_amount	2743417	93.77622848525019
hardship_payoff_balance_amount	2743417	93.77622848525019
hardship_last_payment_amount	2743417	93.77622848525019
sec_app_revol_util	2730906	93.34857406939616
verification_status_joint	2730706	93.34173761482253
revol_bal_joint	2727669	93.23792605212181
sec_app_fico_range_low	2727669	93.23792605212181
sec_app_fico_range_high	2727669	93.23792605212181
sec_app_earliest_cr_line	2727669	93.23792605212181
sec_app_inq_last_6mths	2727669	93.23792605212181
sec_app_mort_acc	2727669	93.23792605212181
sec_app_open_acc	2727669	93.23792605212181
sec_app_open_act_il	2727669	93.23792605212181
sec_app_num_rev_accts	2727669	93.23792605212181
sec_app_chargeoff_within_12_mths	2727669	93.23792605212181
sec_app_collections_12_mths_ex_med	2727669	93.23792605212181
dti_joint	2714984	92.80432392078873
annual_inc_joint	2714978	92.80411882715153
mths_since_last_record	2498046	85.38889000930783

Figure A4: Features with Missing Values (>= 80%)

LoanStatNew	Description
acc_now_delinq	The number of accounts on which the borrower is now delinquent.
acc_open_past_24mths	Number of trades opened in past 24 months.
addr_state	The state provided by the borrower in the loan application
all_util	Balance to credit limit on all trades
annual_inc	The self-reported annual income provided by the borrower during registration.
annual_inc_joint	The combined self-reported annual income provided by the co-borrowers during registration
application_type	Indicates whether the loan is an individual application or a joint application with two co-borrowers
avg_cur_bal	Average current balance of all accounts
bc_open_to_buy	Total open to buy on revolving bankcards.
bc_util	Ratio of total current balance to high credit/credit limit for all bankcard accounts.
chargeoff_within_12_mths	Number of charge-offs within 12 months
collection_recovery_fee	post charge off collection fee
collections_12_mths_ex_med	Number of collections in 12 months excluding medical collections
delinq_2yrs	The number of 30+ days past-due incidences of delinquency in the borrower's credit file for the past 2 years
delinq_amnt	The past-due amount owed for the accounts on which the borrower is now delinquent.
desc	Loan description provided by the borrower
dti	A ratio calculated using the borrower's total monthly debt payments on the total debt obligations, excluding mortgage and the requested LC loan, divided by the borrower's self-reported monthly income.
dti_joint	A ratio calculated using the co-borrowers' total monthly payments on the total debt obligations, excluding mortgages and the requested LC loan, divided by the co-borrowers' combined self-reported monthly income
earliest_cr_line	The month the borrower's earliest reported credit line was opened
emp_length	Employment length in years. Possible values are between 0 and 10 where 0 means less than one year and 10 means ten or more years.
emp_title	The job title supplied by the Borrower when applying for the loan.*
fico_range_high	The upper boundary range the borrower's FICO at loan origination belongs to.
fico_range_low	The lower boundary range the borrower's FICO at loan origination belongs to.
funded_amnt	The total amount committed to that loan at that point in time.
funded_amnt_inv	The total amount committed by investors for that loan at that point in time.
grade	LC assigned loan grade
home_ownership	The home ownership status provided by the borrower during registration or obtained from the credit report. Our values are: RENT, OWN, MORTGAGE, OTHER
id	A unique LC assigned ID for the loan listing.
il_util	Ratio of total current balance to high credit/credit limit on all install acct
initial_list_status	The initial listing status of the loan. Possible values are – W, F
inq_fi	Number of personal finance inquiries
inq_last_12m	Number of credit inquiries in past 12 months
inq_last_6mths	The number of inquiries in past 6 months (excluding auto and mortgage inquiries)
installment	The monthly payment owed by the borrower if the loan originates.
int_rate	Interest Rate on the loan
issue_d	The month which the loan was funded
last_credit_pull_d	The most recent month LC pulled credit for this loan
last_fico_range_high	The upper boundary range the borrower's last FICO pulled belongs to.
last_fico_range_low	The lower boundary range the borrower's last FICO pulled belongs to.
last_pymnt_amnt	Last total payment amount received
last_pymnt_d	Last month payment was received
loan_amnt	The listed amount of the loan applied for by the borrower. If at some point in time, the credit department reduces the loan amount, then it will be reflected in this value.
loan_status	Current status of the loan
max_bal_bc	Maximum current balance owed on all revolving accounts
member_id	A unique LC assigned Id for the borrower member.
mo_sin_old_il_acct	Months since oldest bank installment account opened
mo_sin_old_rev_tl_op	Months since oldest revolving account opened
mo_sin_rcnt_rev_tl_op	Months since most recent revolving account opened
mo_sin_rcnt_tl	Months since most recent account opened
mort_acc	Number of mortgage accounts.
mths_since_last_delinq	The number of months since the borrower's last delinquency.
mths_since_last_major_derog	Months since most recent 90-day or worse rating
mths_since_last_record	The number of months since the last public record.
mths_since_rcnt_il	Months since most recent installment accounts opened
mths_since_recent_bc	Months since most recent bankcard account opened.
mths_since_recent_bc_dlq	Months since most recent bankcard delinquency
mths_since_recent_inq	Months since most recent inquiry.
mths_since_recent_revol_delinq	Months since most recent revolving delinquency.
next_pymnt_d	Next scheduled payment date
num_accts_ever_120_pd	Number of accounts ever 120 or more days past due
num_actv_bc_tl	Number of currently active bankcard accounts
num_actv_rev_tl	Number of currently active revolving trades
num_bc_sats	Number of satisfactory bankcard accounts
num_bc_tl	Number of bankcard accounts
num_il_tl	Number of installment accounts
num_op_rev_tl	Number of open revolving accounts
num_rev_accts	Number of revolving accounts
num_rev_tl_bal_gt_0	Number of revolving trades with balance >0
num_sats	Number of satisfactory accounts
num_tl_120dpd_2m	Number of accounts currently 120 days past due (updated in past 2 months)
num_tl_30dpd	Number of accounts currently 30 days past due (updated in past 2 months)
num_tl_90g_dpd_24m	Number of accounts 90 or more days past due in last 24 months
num_tl_op_past_12m	Number of accounts opened in past 12 months
open_acc	The number of open credit lines in the borrower's credit file.
open_acc_6m	Number of open trades in last 6 months
open_il_12m	Number of installment accounts opened in past 12 months
open_il_24m	Number of installment accounts opened in past 24 months
open_act_il	Number of currently active installment trades
open_rv_12m	Number of revolving trades opened in past 12 months
open_rv_24m	Number of revolving trades opened in past 24 months
out_prncp	Remaining outstanding principal for total amount funded
out_prncp_inv	Remaining outstanding principal for portion of total amount funded by investors
pct_tl_nvr_dlq	Percent of trades never delinquent
percent_bc_gt_75	Percentage of all bankcard accounts > 75% of limit.
policy_code	publicly available policy_code=1 new products not publicly available policy_code=2
pub_rec	Number of derogatory public records
pub_rec_bankruptcies	Number of public record bankruptcies
purpose	A category provided by the borrower for the loan request.
pymnt_plan	Indicates if a payment plan has been put in place for the loan
recoveries	post charge off gross recovery
revol_bal	Total credit revolving balance

Figure A5.1: LC Data Dictionary (Part 1)

LoanStatNew	Description
revol_util	Revolving line utilization rate, or the amount of credit the borrower is using relative to all available revolving credit.
sub_grade	LC assigned loan subgrade
tax_liens	Number of tax liens
term	The number of payments on the loan. Values are in months and can be either 36 or 60.
title	The loan title provided by the borrower
tot_coll_amt	Total collection amounts ever owed
tot_cur_bal	Total current balance of all accounts
tot_hi_cred_lim	Total high credit/credit limit
total_acc	The total number of credit lines currently in the borrower's credit file
total_bal_ex_mort	Total credit balance excluding mortgage
total_bal_il	Total current balance of all installment accounts
total_bc_limit	Total bankcard high credit/credit limit
total_cu_tl	Number of finance trades
total_il_high_credit_limit	Total installment high credit/credit limit
total_pymnt	Payments received to date for total amount funded
total_pymnt_inv	Payments received to date for portion of total amount funded by investors
total_rec_int	Interest received to date
total_rec_late_fee	Late fees received to date
total_rec_prncp	Principal received to date
total_rev_hi_lim	Total revolving high credit/credit limit
url	URL for the LC page with listing data.
verification_status	Indicates if income was verified by LC, not verified, or if the income source was verified
verified_status_joint	Indicates if the co-borrowers' joint income was verified by LC, not verified, or if the income source was verified
zip_code	The first 3 numbers of the zip code provided by the borrower in the loan application.
revol_bal_joint	Sum of revolving credit balance of the co-borrowers, net of duplicate balances
sec_app_fico_range_low	FICO range (high) for the secondary applicant
sec_app_fico_range_high	FICO range (low) for the secondary applicant
sec_app_earliest_cr_line	Earliest credit line at time of application for the secondary applicant
sec_app_inq_last_6mths	Credit inquiries in the last 6 months at time of application for the secondary applicant
sec_app_mort_acc	Number of mortgage accounts at time of application for the secondary applicant
sec_app_open_acc	Number of open trades at time of application for the secondary applicant
sec_app_revol_util	Ratio of total current balance to high credit/credit limit for all revolving accounts
sec_app_open_act_il	Number of currently active installment trades at time of application for the secondary applicant
sec_app_num_rev_accts	Number of revolving accounts at time of application for the secondary applicant
sec_app_chargeoff_within_12_mths	Number of charge-offs within last 12 months at time of application for the secondary applicant
sec_app_collections_12_mths_ex_med	Number of collections within last 12 months excluding medical collections at time of application for the secondary applicant
sec_app_mths_since_last_major_derog	Months since most recent 90-day or worse rating at time of application for the secondary applicant
hardship_flag	Flags whether or not the borrower is on a hardship plan
hardship_type	Describes the hardship plan offering
hardship_reason	Describes the reason the hardship plan was offered
hardship_status	Describes if the hardship plan is active, pending, canceled, completed, or broken
deferral_term	Amount of months that the borrower is expected to pay less than the contractual monthly payment amount due to a hardship plan
hardship_amount	The interest payment that the borrower has committed to make each month while they are on a hardship plan
hardship_start_date	The start date of the hardship plan period
hardship_end_date	The end date of the hardship plan period
payment_plan_start_date	The day the first hardship plan payment is due. For example, if a borrower has a hardship plan period of 3 months, the start date is the start of the three-month period in which the borrower is allowed to make interest-only payments.
hardship_length	The number of months the borrower will make smaller payments than normally obligated due to a hardship plan

LoanStatNew	Description
hardship_dpd	Account days past due as of the hardship plan start date
hardship_loan_status	Loan Status as of the hardship plan start date
orig_projected_additional_accrued_int	The original projected additional interest amount that will accrue for the given hardship payment plan as of the Hardship Start Date. This field will be null if the borrower has broken their hardship payment plan.
hardship_payoff_balance_amount	The payoff balance amount as of the hardship plan start date
hardship_last_payment_amount	The last payment amount as of the hardship plan start date
disbursement_method	The method by which the borrower receives their loan. Possible values are: CASH, DIRECT_PAY
debt_settlement_flag	Flags whether or not the borrower, who has charged-off, is working with a debt-settlement company.
debt_settlement_flag_date	The most recent date that the Debt_Settlement_Flag has been set
settlement_status	The status of the borrower’s settlement plan. Possible values are: COMPLETE, ACTIVE, BROKEN, CANCELLED, DENIED, DRAFT
settlement_date	The date that the borrower agrees to the settlement plan
settlement_amount	The loan amount that the borrower has agreed to settle for
settlement_percentage	The settlement amount as a percentage of the payoff balance amount on the loan
settlement_term	The number of months that the borrower will be on the settlement plan

Figure A5.2: LC Data Dictionary (Part 2)

Final Features to be used in models

Feature Name	Rationale
home_ownership	Borrower's housing situation affects loan repayment ability
verification_status	Income verification indicates creditworthiness
term	Loan duration impacts repayment and risk
grade	Assigned credit grade summarizing borrower's risk level
hardship_flag	Flags borrowers under hardship programs
hardship_reason	Provides reason behind hardship flag
purpose	Loan purpose impacts borrower's repayment motivation
mths_since_last_delinq	Captures borrower's delinquency recency
emp_length	Measures employment stability
acc_now_delinq	Number of currently delinquent accounts
acc_open_past_24mths	Number of accounts opened in recent 2 years
annual_inc	Borrower's income to assess ability to repay
chargeoff_within_12_mths	Borrower's recent severe delinquencies
delinq_2yrs	Delinquency history in past 2 years
delinq_amnt	Outstanding delinquent amount
dti	Debt burden relative to income
fico_range_high	Borrower's highest reported FICO score range
il_util	Utilization ratio for installment loans
inq_fi	Finance-related credit inquiries
inq_last_12m	Credit inquiries in the past 12 months
int_rate	Interest rate assigned based on borrower's risk
loan_amnt	Principal amount of the loan
mort_acc	Number of mortgage accounts
num_accts_ever_120_pd	Accounts with very late payments
num_actv_rev_tl	Active revolving credit accounts
num_bc_tl	Number of bankcard accounts
num_il_tl	Number of installment accounts
open_acc	Total open accounts indicating financial activity
pct_tl_nvr_dlq	Percentage of tradelines never delinquent
percent_bc_gt_75	Proportion of high-utilization bankcards
pub_rec	Public derogatory records
pub_rec_bankruptcies	History of bankruptcies
revol_bal	Total revolving balance
revol_util	Revolving utilization rate
tot_coll_amt	Total collections recorded
tot_cur_bal	Total balance across all accounts
total_acc	Total number of tradelines
total_cu_tl	Number of finance company trades
emp_length_flag	Flag for unemployed vs employed
had_delinquency_flag	Flag for historical delinquency
credit_util_ratio	Ratio of used credit to available credit
income_to_loan_ratio	Income relative to requested loan size
installment_to_income_ratio	Monthly repayment burden
delinq_flag	Flag for any delinquency presence

Figure A6: Features used for model building

Appendix B: EDA Graphs

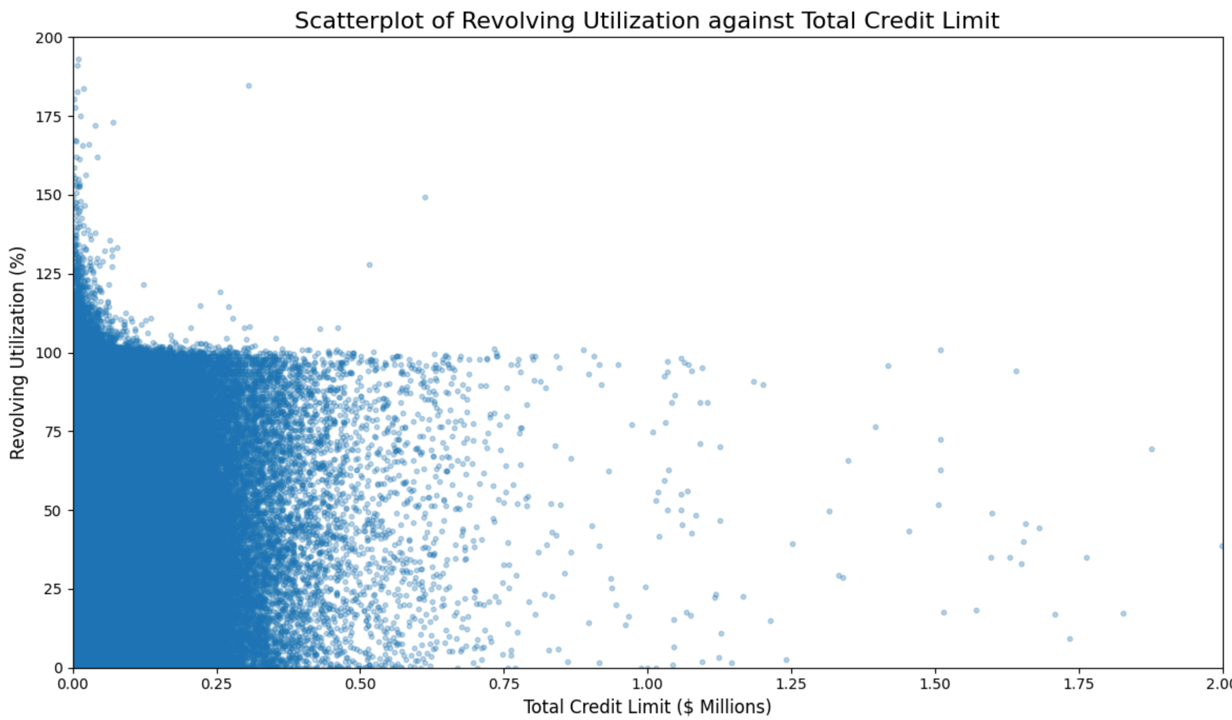

Figure B1: Revolving Credit Utilisation vs Total Credit Limit

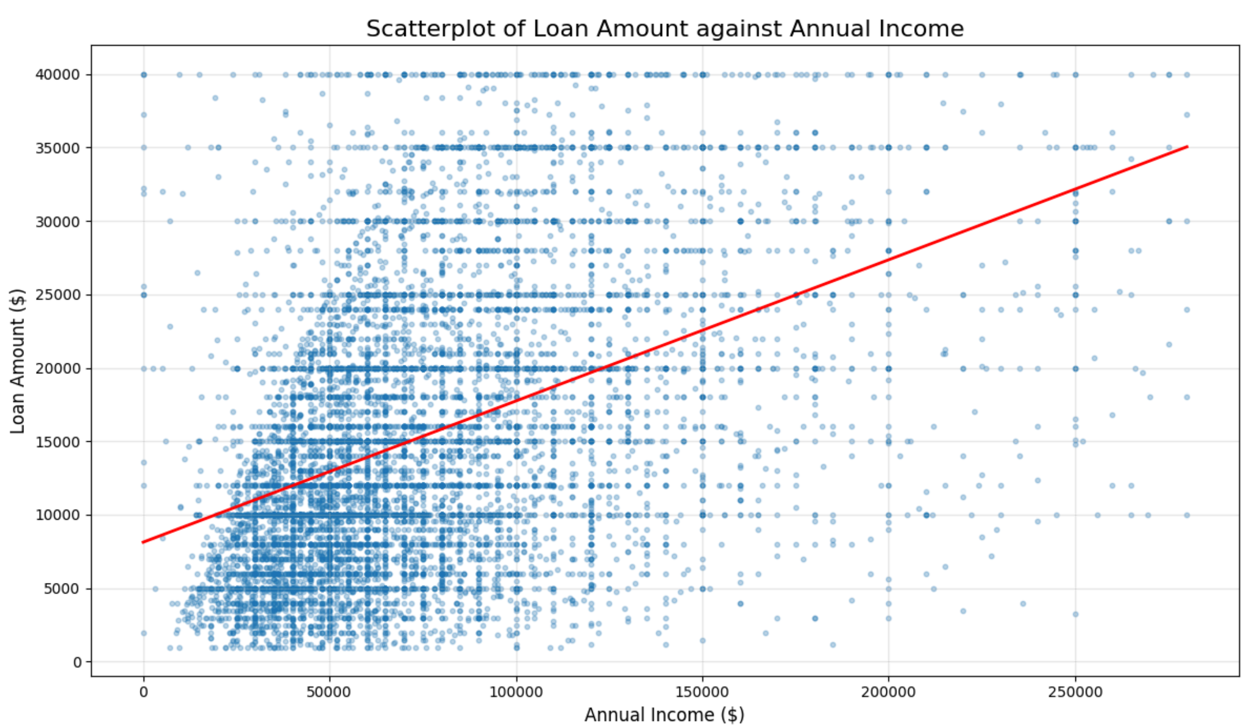

Figure B2: Loan Amount vs Annual Income

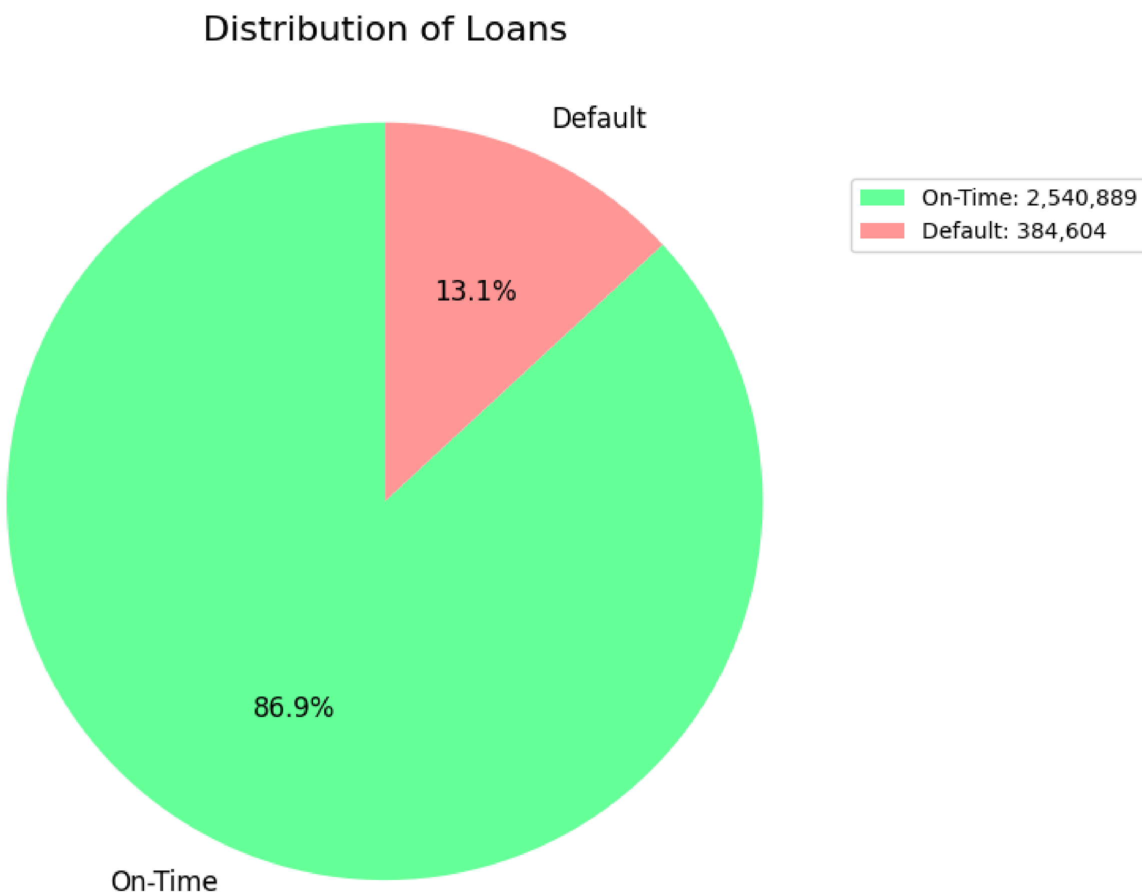

Figure B3: Distribution of Loans

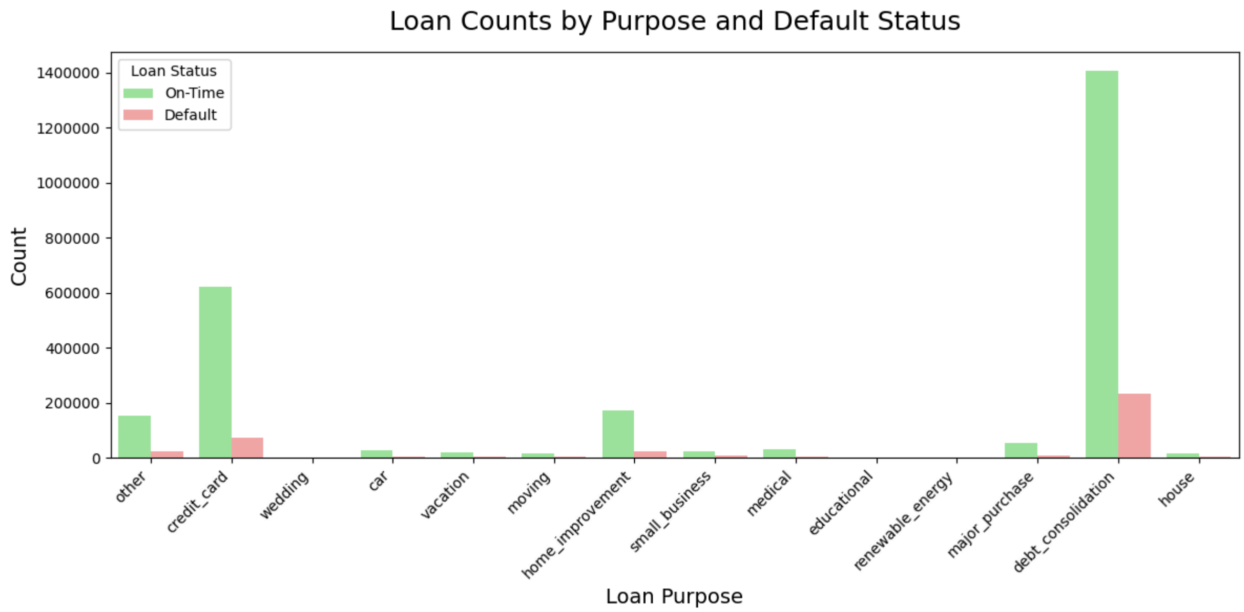

Figure B4.1: Loan Counts grouped by Purpose and Default Status

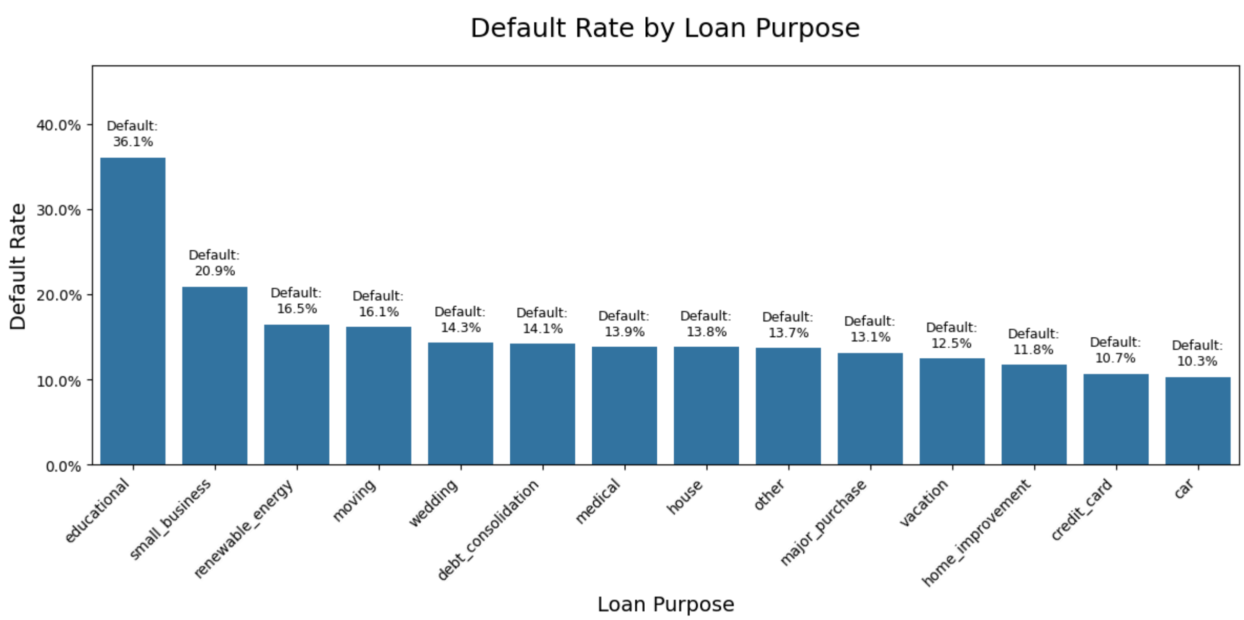

Figure B4.2: Default Rate vs Loan Purpose

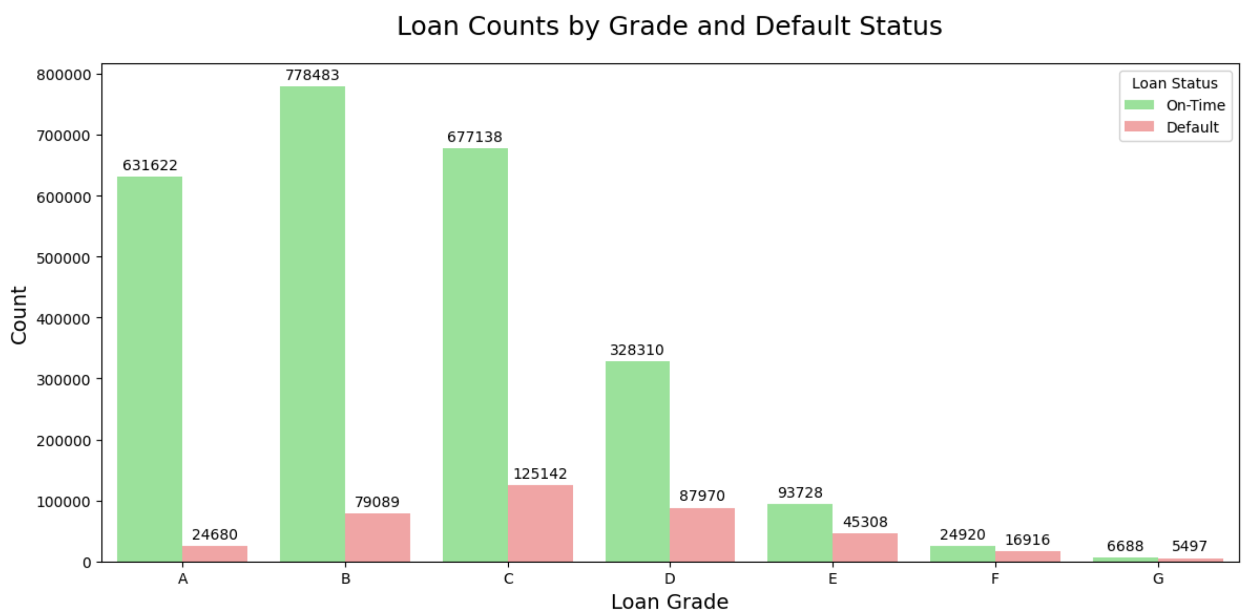

Figure B5.1: Loan Counts grouped by Grade and Default Status

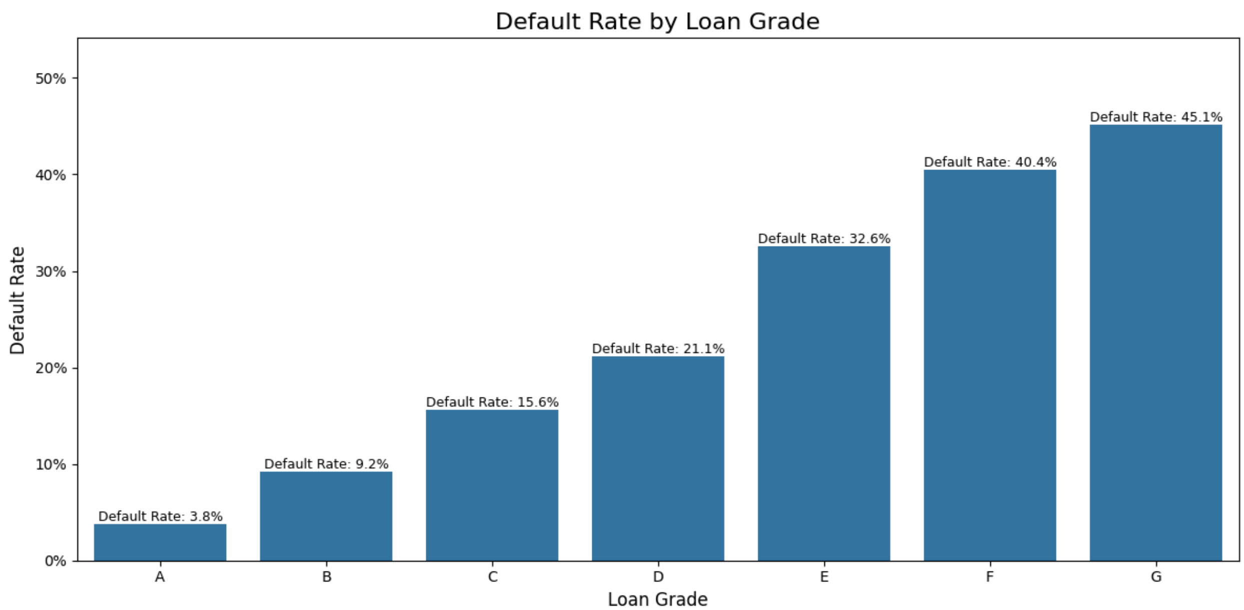

Figure B5.2: Default Rate vs Loan Grade

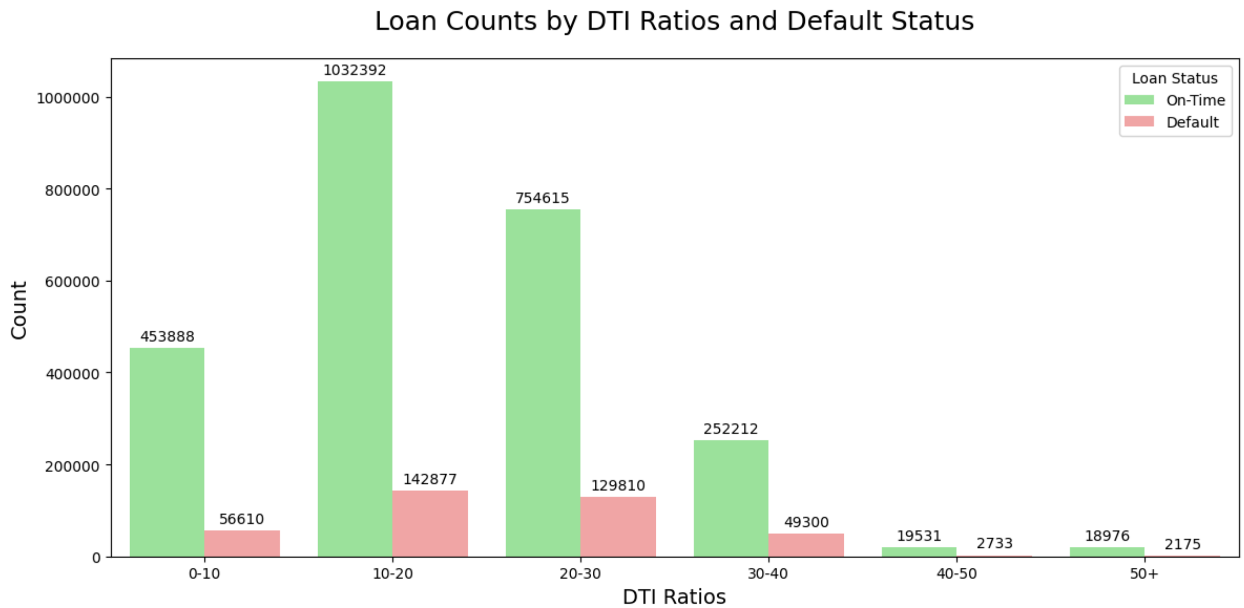

Figure B6.1: Loan Counts grouped by DTI Ratios and Default Status

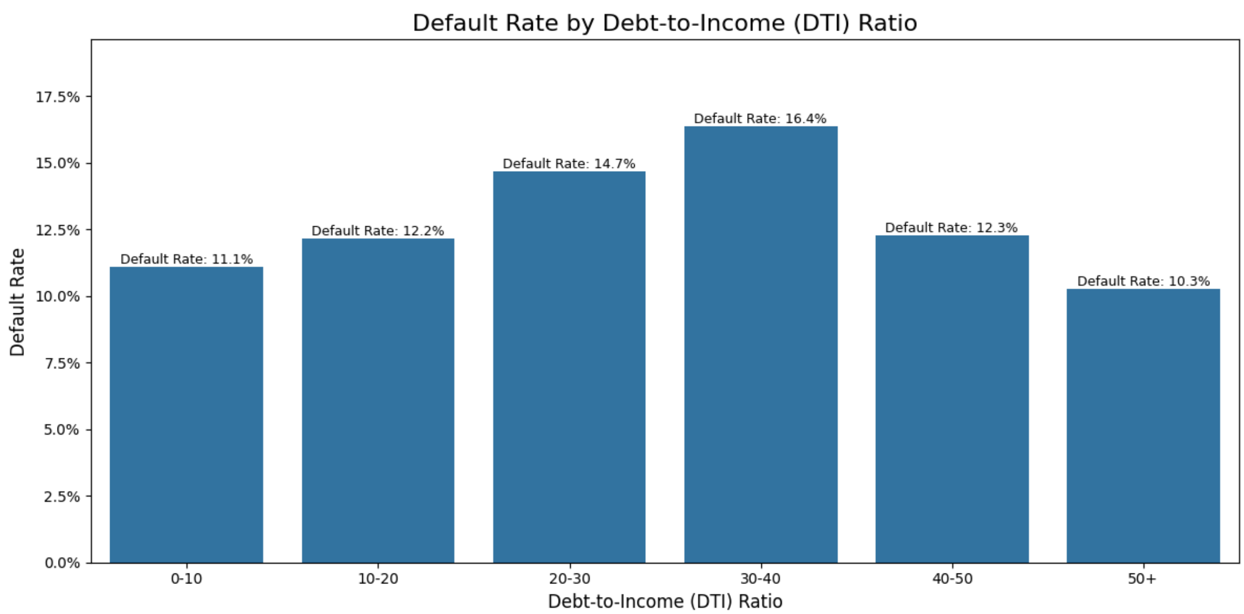

Figure B6.2: Default Rate vs Debt-To-Income (DTI) Ratio

Appendix C: Data Cleaning & Preparation

feature	cardinality
policy_code	1
term	2
pymnt_plan	2
hardship_flag	2
debt_settlement_flag	2
verification_status_joint	3
application_type	3
verification_status	4
deferral_term	4
hardship_length	4
hardship_status	4
initial_list_status	4
num_tl_30dpd	5
sec_app_inq_last_6mths	7
home_ownership	7
grade	7
num_tl_120dpd_2m	8
acc_now_delinq	9
loan_status	11
hardship_type	11
chargeoff_within_12_mths	12
emp_length	12
pub_rec_bankruptcies	13
hardship_loan_status	14
purpose	15
hardship_reason	17
collections_12_mths_ex_med	18
open_il_12m	19
open_acc_6m	20
sec_app_collections_12_mths_ex_med	21
sec_app_chargeoff_within_12_mths	22
mths_since_recent_inq	27
sec_app_mort_acc	28
open_rv_12m	30
inq_last_6mths	30
open_il_24m	33
num_tl_op_past_12m	33
num_tl_90g_dpd_24m	35
sub_grade	36
delinq_2yrs	37
inq_fi	37
sec_app_open_act_il	42
num_actv_bc_tl	43
tax_liens	44
pub_rec	44
hardship_dpd	45
payment_plan_start_date	46
fico_range_low	47
hardship_start_date	47
fico_range_high	48
hardship_end_date	48
mort_acc	48
inq_last_12m	49
num_accts_ever_120_pd	49
open_rv_24m	50
num_rev_tl_bal_gt_0	54
addr_state	56
acc_open_past_24mths	57
sec_app_fico_range_high	58
num_actv_rev_tl	59
open_act_il	59
sec_app_fico_range_low	63
num_bc_sats	65
total_cu_tl	66
sec_app_open_acc	71
last_fico_range_low	73
last_fico_range_high	74
num_bc_tl	79
num_op_rev_tl	85
num_sats	92
sec_app_num_rev_accts	94

Figure C1.1: Feature Cardinality

open_acc	94
next_pymnt_d	113
num_rev_accts	123
num_il_tl	124
mths_since_last_record	126
last_pymnt_d	151
issue_d	157
last_credit_pull_d	157
total_acc	164
mths_since_last_delinq	180
mths_since_recent_bc_dlq	182
mths_since_recent_revol_delinq	185
mths_since_last_major_derog	190
all_util	195
mo_sin_rcnt_tl	242
percent_bc_gt_75	298
il_util	307
mo_sin_rcnt_rev_tl_op	353
mths_since_rcnt_il	426
mths_since_recent_bc	580
mo_sin_old_il_acct	586
sec_app_earliest_cr_line	646
pct_tl_nvr_dlq	683
int_rate	686
earliest_cr_line	739
mo_sin_old_rev_tl_op	792
zip_code	990
sec_app_revol_util	1234
revol_util	1376
bc_util	1455
loan_amnt	1648
funded_amnt	1648
delinq_amnt	2803
dti_joint	4225
funded_amnt_inv	9923
dti	13275
tot_coll_amt	16724

tot_coll_amt	16724
hardship_amount	20520
total_bc_limit	21529
total_rec_late_fee	22516
annual_inc_joint	26965
total_rev_hi_lim	33024
max_bal_bc	35117
hardship_last_payment_amount	53255
title	63892
revol_bal_joint	71226
avg_cur_bal	90557
orig_projected_additional_accrued_interest	91252
bc_open_to_buy	99954
installment	105888
annual_inc	107248
revol_bal	108747
hardship_payoff_balance_amount	164577
recoveries	171728
total_bal_il	181833
collection_recovery_fee	196028
total_il_high_credit_limit	214449
total_bal_ex_mort	240519
_c0	435108
out_prncp	520618
tot_cur_bal	530806
out_prncp_inv	565905
tot_hi_cred_lim	571247
emp_title	609444
total_rec_prncp	649018
total_rec_int	741607
last_pymnt_amnt	851352
total_pymnt_inv	1545590
total_pymnt	2046528
id	2808211
url	3009655

Figure C1.2: Feature Cardinality (Continued)

Appendix D: Modelling

Sampling Method	Accuracy	Precision	Recall	F1_Score	ROC_AUC	PR_AUC
Base	0.9128	0.4000	0.0007	0.0014	0.7303	0.2043
SMOTE	0.6370	0.1561	0.7187	0.2565	0.7372	0.2154
ADASYN	0.6157	0.1519	0.7441	0.2523	0.7367	0.2122
TomekLinks	0.9128	0.4000	0.0012	0.0024	0.7308	0.2054
ENN	0.9115	0.4140	0.03759	0.0689	0.7331	0.2095
SMOTE & ENN	0.4874	0.1308	**0.8655**	0.2273	0.7370	0.2116
SMOTE & TomekLinks	0.6370	0.1561	0.7188	0.2565	0.7372	0.2154

Table D1: Logistic Regression (Sampling Techniques)

Grid Search:

Sampling Method	Accuracy	Precision	Recall	F1_Score	ROC_AUC	PR_AUC
SMOTE & ENN	0.3494	0.1127	0.9404	0.2012	0.7366	0.2148

Table D2: Logistic Regression - Best Model Grid Searched

Sampling Method	Accuracy	Precision	Recall	F1_Score	ROC_AUC	PR_AUC
Base Model	0.8770	0.1550	0.0929	0.1160	0.5010	0.08840
SMOTE	0.7130	0.1560	0.5210	0.2400	0.5006	0.08840
ADASYN	0.6180	0.1390	0.6570	0.2300	0.5000	0.08830
TomekLinks	0.8760	0.1540	0.0942	0.1169	0.5006	0.08840
ENN	0.8710	0.1560	0.1070	0.1270	0.5006	0.08840
SMOTE & ENN	0.5110	0.1260	**0.7770**	0.2170	0.5006	0.08840
SMOTE & TomekLinks	0.7130	0.1560	0.5210	0.2400	0.5006	0.08840

Table D3: Naïve Bayes (Sampling Techniques)

Grid Search:

Sampling Method	Accuracy	Precision	Recall	F1_Score	ROC_AUC	PR_AUC
SMOTE & ENN	0.5110	0.1260	0.7770	0.2170	0.5006	0.08840

Table D4: Naïve Bayes Result - Best Model Grid Searched

Sampling Method	Accuracy	Precision	Recall	F1_Score	ROC_AUC	PR_AUC
Base Model	0.9128	0.4755	0.0055	0.0108	0.4582	0.0806
SMOTE	0.5786	0.1348	0.7085	0.2266	0.6314	0.1365
ADASYN	0.5977	0.1347	0.6671	0.2242	0.5914	0.1256
TomekLinks	0.9121	0.4016	0.0176	0.0337	0.4560	0.0803
ENN	0.9092	0.3713	0.0611	0.1050	0.4564	0.0803
SMOTE & ENN	0.4448	0.1209	0.8565	0.2119	0.6482	0.1299
SMOTE & TomekLinks	0.5580	0.1334	0.7409	0.2261	0.6155	0.1237

Table D5: Decision Tree (Sampling Techniques)

Grid Search:

Sampling Method	Accuracy	Precision	Recall	F1_Score	ROC_AUC	PR_AUC
SMOTE & ENN (maxDepth = 8)	0.5026	0.1264	0.7966	0.2182	0.6295	0.1471
SMOTE & ENN (maxDepth = 3)	0.4351	0.1177	0.8437	0.2065	0.6402	0.1285

Table D6: Decision Tree - Best Model Grid Searched

Sampling Method	Accuracy	Precision	Recall	F1_Score	ROC_AUC	PR_AUC
Base Model	0.9131	0.7153	0.0042	0.0085	0.7281	0.2103
SMOTE	0.6631	0.1524	0.6283	0.2452	0.7092	0.1873
ADASYN	0.6811	0.1588	0.6187	0.2527	0.7165	0.1865
TomekLinks	0.9131	0.5877	0.0079	0.0155	0.7271	0.2084
ENN	0.9080	0.3599	0.0713	0.1192	0.7146	0.1991
SMOTE & ENN	0.5205	0.1340	0.8242	0.2305	0.7212	0.1883
SMOTE & TomekLinks	0.6949	0.1617	0.5981	0.2546	0.7202	0.1904

Table D7: Random Tree (Sampling Techniques)

Grid Search:

Sampling Method	Accuracy	Precision	Recall	F1_Score	ROC_AUC	PR_AUC
SMOTE & ENN	0.5184	0.1337	0.8261	0.2301	0.7206	0.1890

Table D8: RandomTree - Best Model Grid Searched

Feature Importance:

Feature Selection	Accuracy	Precision	Recall	F1_Score	ROC_AUC	PR_AUC
SMOTE & ENN	0.5369	0.1367	0.8117	0.2340	0.7206	0.1902

Table D9: Random Forest - Post Feature Importance

Sampling Method	Accuracy	Precision	Recall	F1_Score	ROC_AUC	PR_AUC
Base Model	0.9128	0.4673	0.0062	0.0123	0.7306	0.2106
SMOTE	0.6590	0.1512	0.6314	0.2440	0.7046	0.1814
ADASYN	0.6496	0.1467	0.6271	0.2378	0.6953	0.1681
TomekLinks	0.9135	0.6119	0.0188	0.0365	0.7505	0.2381
ENN	0.9069	0.3728	0.0999	0.1575	0.7491	0.2321
SMOTE & ENN	0.5242	0.1327	**0.8060**	0.2280	0.7131	0.1871
SMOTE & TomekLinks	0.6572	0.1514	0.6373	0.2447	0.7070	0.1838

Table D10: GBTClasssifier (Sampling Techniques)

Grid Search:

Sampling Method	Accuracy	Precision	Recall	F1_Score	ROC_AUC	PR_AUC
SMOTE & ENN	0.5711	0.1376	0.7451	0.2324	0.7050	0.1841

Table D11: GBTClassifier - Best Model Grid Searched

Sampling Method (50,0.1)	**Accuracy**	**Precision**	**Recall**	**F1_Score**	**ROC_AUC**	**PR_AUC**
Base	0.9131	0.6356	0.0062	0.0123	0.6350	0.1543
SMOTE	0.6352	0.1526	0.6999	0.2506	0.7230	**0.1874**
ADASYN	0.6326	0.1526	0.7065	0.2510	0.7239	0.1867
TomekLinks	0.9131	0.6298	0.0057	0.0113	0.6399	0.1634
ENN	0.9117	0.4133	0.0328	0.0607	0.6193	0.1588
SMOTE & ENN	0.4029	0.1182	**0.9068**	0.2092	0.7122	0.1706
SMOTE & TomekLinks	0.6352	0.1526	0.6999	0.2506	0.7230	0.1874

Table D12: Support Vector Machine (Sampling Techniques)

Grid Search:

Sampling Method	**Accuracy**	**Precision**	**Recall**	**F1_Score**	**ROC_AUC**	**PR_AUC**
SMOTE & ENN (100, 0.5)	0.3397	0.1115	0.9441	0.1995	0.7262	0.1870
SMOTE (100, 0.5)	0.6332	0.1537	0.7122	0.2528	0.7295	**0.2064**

Table D13: Support Vector Machine - Best Model Grid Searched

Best Model	Accuracy	Precision	Recall	F1_Score	ROC_AUC	PR_AUC
Logreg	0.3494	0.1127	0.9404	0.2012	0.7366	0.2148
Naïve	0.5110	0.1260	0.7770	0.2170	0.5006	0.0884
DT	0.4448	0.1209	0.8565	0.2119	0.6482	0.1299
RF	0.5184	0.1337	0.8261	0.2301	0.7206	0.1890
GBT	0.5711	0.1376	0.7451	0.2324	0.7050	0.1841
SVM	0.6332	0.1537	0.7122	0.2528	0.7295	0.2064

Table D14: Best Model Overview

	PCA Component	Importance
4	PC5	0.158994
2	PC3	0.101674
1	PC2	0.068223
13	PC14	0.053679
3	PC4	0.051580
20	PC21	0.041296
24	PC25	0.030978
21	PC22	0.027443
12	PC13	0.020673
60	PC61	0.020324
6	PC7	0.018630
11	PC12	0.018607
10	PC11	0.018360
23	PC24	0.017292
48	PC49	0.016552
51	PC52	0.015024
25	PC26	0.014800
28	PC29	0.013481
9	PC10	0.011015
26	PC27	0.010784

Figure D15: Feature Importances

NUS National University of Singapore
School of Computing

	PC1	PC2	PC3	PC4	PC5	PC6	PC7	PC8	PC9	PC10
acc_now_delinq	0.022842	0.023270	0.021970	0.000151	0.012776	0.030288	0.081831	0.070879	0.034562	0.000952
acc_open_past_24mths	0.182188	0.134799	0.039183	0.213033	0.273052	0.070651	0.009309	0.007849	0.036000	0.020576
annual_inc	0.119237	0.094016	0.028135	0.057864	0.089717	0.018535	0.143230	0.169618	0.030190	0.448314
chargeoff_within_12_mths	0.045751	0.053770	0.040298	0.017187	0.016258	0.039773	0.087502	0.079090	0.054653	0.026498
credit_util_ratio	0.202339	0.013976	0.133155	0.240971	0.057221	0.254753	0.113620	0.011644	0.115743	0.092824
delinq_2yrs	0.172505	0.205601	0.143513	0.020681	0.096372	0.126230	0.248863	0.224006	0.124959	0.050031
delinq_amnt	0.008035	0.007828	0.005800	0.002071	0.007247	0.011050	0.033276	0.029764	0.017632	0.002122
delinq_flag	0.192023	0.226733	0.158642	0.019465	0.111540	0.140050	0.269134	0.237853	0.112287	0.047323
dti	0.031785	0.038069	0.099172	0.019748	0.039362	0.085877	0.063467	0.091008	0.026475	0.324311
emp_length	0.083366	0.053604	0.015576	0.064614	0.046705	0.019064	0.020826	0.029644	0.093226	0.218918
emp_length_flag	0.042563	0.014085	0.024182	0.014613	0.052996	0.011774	0.058135	0.017166	0.040289	0.241179
fico_range_high	0.147465	0.252045	0.223562	0.058057	0.121314	0.058968	0.146618	0.141195	0.127024	0.025887
grade	0.261922	0.464613	0.812255	0.128039	0.492665	0.043840	0.318566	0.095780	0.440787	0.191899
had_delinquency_flag	0.257348	0.277764	0.171747	0.079476	0.037599	0.004975	0.114723	0.162483	0.161873	0.000812
hardship_flag	0.021747	0.002749	0.252154	0.354124	0.355218	0.098341	0.022245	0.039874	0.003614	0.003036
hardship_reason	0.099302	0.051538	0.921393	1.221283	1.191518	0.352970	0.082281	0.142180	0.046055	0.046424
home_ownership	0.395545	0.310928	0.119114	0.387771	0.293901	0.458248	0.227939	0.035125	0.211662	0.163946
il_util	0.076869	0.000104	0.062086	0.063726	0.159063	0.228298	0.071543	0.049237	0.230961	0.079973
income_to_loan_ratio	0.035219	0.005098	0.085714	0.010659	0.018037	0.077892	0.242631	0.279145	0.126364	0.413802
inq_fi	0.122804	0.040695	0.029178	0.117408	0.188819	0.250732	0.026825	0.030426	0.129997	0.035708
inq_last_12m	0.150862	0.053878	0.029920	0.129411	0.202869	0.234134	0.040113	0.041877	0.108021	0.021507
installment_to_income_ratio	0.004204	0.001684	0.030501	0.005919	0.005010	0.032693	0.056424	0.067222	0.001729	0.216249
int_rate	0.107737	0.185738	0.333833	0.044726	0.194181	0.013399	0.130063	0.036881	0.167990	0.063366
loan_amnt	0.106597	0.140688	0.117812	0.076797	0.149356	0.136870	0.182536	0.194230	0.105125	0.018718

	PC11	PC12	PC13	PC14	PC15	PC16	PC17	PC18	PC19	PC20	PC21	PC2
acc_now_delinq	0.001957	0.014380	0.009861	0.130861	0.519368	0.054743	0.057279	0.249851	0.225176	0.036583	0.023134	0.00663
acc_open_past_24mths	0.066431	0.006815	0.008386	0.069034	0.002347	0.097259	0.086545	0.038111	0.038541	0.024255	0.025475	0.00387
annual_inc	0.020139	0.041693	0.079790	0.071850	0.079651	0.353598	0.133840	0.141179	0.121623	0.020538	0.069414	0.02100
chargeoff_within_12_mths	0.036310	0.000732	0.002642	0.033692	0.152100	0.006926	0.040759	0.172345	0.364113	0.025873	0.024333	0.02704
credit_util_ratio	0.206371	0.003247	0.017918	0.023527	0.017392	0.015639	0.110868	0.005248	0.005367	0.014472	0.074857	0.03143
delinq_2yrs	0.084552	0.012722	0.010775	0.007869	0.032354	0.005813	0.017899	0.015158	0.047557	0.018936	0.008596	0.00019
delinq_amnt	0.002007	0.008074	0.003256	0.094163	0.357076	0.052404	0.044115	0.194159	0.154711	0.046184	0.026233	0.00322
delinq_flag	0.058297	0.010797	0.012551	0.010948	0.078133	0.004801	0.013920	0.031328	0.009855	0.022459	0.003091	0.00943
dti	0.024793	0.231246	0.189346	0.057997	0.002332	0.074983	0.359536	0.258698	0.147894	0.031460	0.001300	0.00182
emp_length	0.041605	0.196842	0.216612	0.273451	0.111090	0.446002	0.216353	0.036426	0.022608	0.014748	0.018012	0.02704
emp_length_flag	0.111212	0.311780	0.309213	0.253274	0.087261	0.352663	0.135485	0.144868	0.076819	0.030858	0.015421	0.01954
fico_range_high	0.061364	0.011519	0.007785	0.057810	0.011778	0.083099	0.113908	0.039163	0.028641	0.011852	0.004647	0.00303
grade	0.655734	1.166674	1.422402	1.635891	0.195353	0.545526	0.741746	0.343072	0.202726	0.073218	1.508362	0.84837
had_delinquency_flag	0.068409	0.008651	0.009291	0.035691	0.021097	0.052404	0.038478	0.036901	0.020900	0.102308	0.003315	0.01091
hardship_flag	0.011177	0.032834	0.021825	0.000327	0.011505	0.021665	0.006477	0.015451	0.009556	0.001326	0.006109	0.00073
hardship_reason	0.078529	0.151624	0.098091	0.123194	0.078304	0.260056	0.100528	0.129774	0.166957	0.112880	0.510022	0.40238
home_ownership	0.334329	0.191537	0.084421	0.053846	0.014531	0.297137	0.110180	0.142513	0.078165	0.013281	0.046216	0.01044
il_util	0.239519	0.085166	0.058833	0.044419	0.015771	0.039487	0.121765	0.000649	0.003437	0.018787	0.087681	0.05255
income_to_loan_ratio	0.013747	0.125030	0.213248	0.066424	0.016720	0.112141	0.195929	0.028447	0.037074	0.001881	0.041080	0.05234
inq_fi	0.068169	0.025175	0.009018	0.092868	0.108528	0.091792	0.094612	0.273308	0.174713	0.071544	0.058239	0.04016
inq_last_12m	0.100425	0.053170	0.029286	0.122573	0.123612	0.076470	0.135962	0.266389	0.156107	0.055103	0.013085	0.02581
installment_to_income_ratio	0.030045	0.161100	0.145414	0.065564	0.010050	0.018651	0.431261	0.397706	0.252682	0.084152	0.026608	0.01277
int_rate	0.240482	0.022141	0.019518	0.120310	0.025713	0.068428	0.020268	0.098486	0.060191	0.000612	0.022894	0.07309
loan_amnt	0.045307	0.220134	0.150692	0.026101	0.063598	0.300022	0.029289	0.102889	0.077968	0.025380	0.033606	0.05203

Figure D16 : PCA Loadings

	Overall_Weighted_Importance
hardship_reason	0.917467
purpose	0.709867
grade	0.551343
mths_since_last_delinq	0.335412
home_ownership	0.161594
int_rate	0.111067
hardship_flag	0.107039
acc_open_past_24mths	0.094730
loan_amnt	0.092761
revol_util	0.089750
revol_bal	0.087836
percent_bc_gt_75	0.085105
verification_status	0.083220
fico_range_high	0.082407
il_util	0.082323
inq_last_12m	0.079960
inq_fi	0.079933
term	0.078859
mort_acc	0.077052
tot_cur_bal	0.073352
num_actv_rev_tl	0.072251
total_acc	0.070986
total_acc	0.070986
delinq_flag	0.069674
pub_rec_bankruptcies	0.069209
emp_length	0.069194
num_il_tl	0.069131
annual_inc	0.068644
emp_length_flag	0.068194
pub_rec	0.067466
had_delinquency_flag	0.066396
credit_util_ratio	0.065850
num_bc_tl	0.065614
open_acc	0.065504
delinq_2yrs	0.064716
pct_tl_nvr_dlq	0.064542
total_cu_tl	0.064228
num_accts_ever_120_pd	0.060654
income_to_loan_ratio	0.058937
dti	0.056568
delinq_amnt	0.046926
installment_to_income_ratio	0.045748
chargeoff_within_12_mths	0.043118
acc_now_delinq	0.034149
tot_coll_amt	0.032646

Figure D17: Weighted Importance of features

Best Model	Accuracy	Precision	Recall	F1_Score	ROC_AUC	PR_AUC
Old “Test” Set	0.5184	0.1337	0.8261	0.2301	0.7206	0.1890
Final Test Set	0.5357	0.1342	0.8099	0.2302	0.7228	0.1893

Table 18: Best Model Test Data Evaluation

Appendix E: Insights

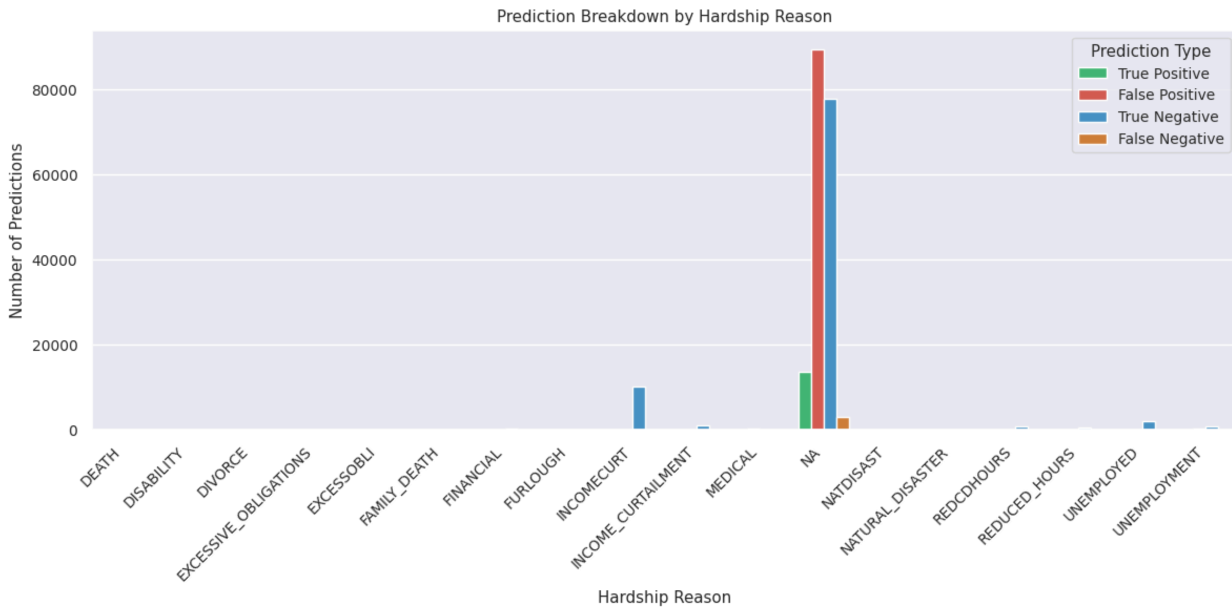

Figure E1: Hardship Reason Prediction

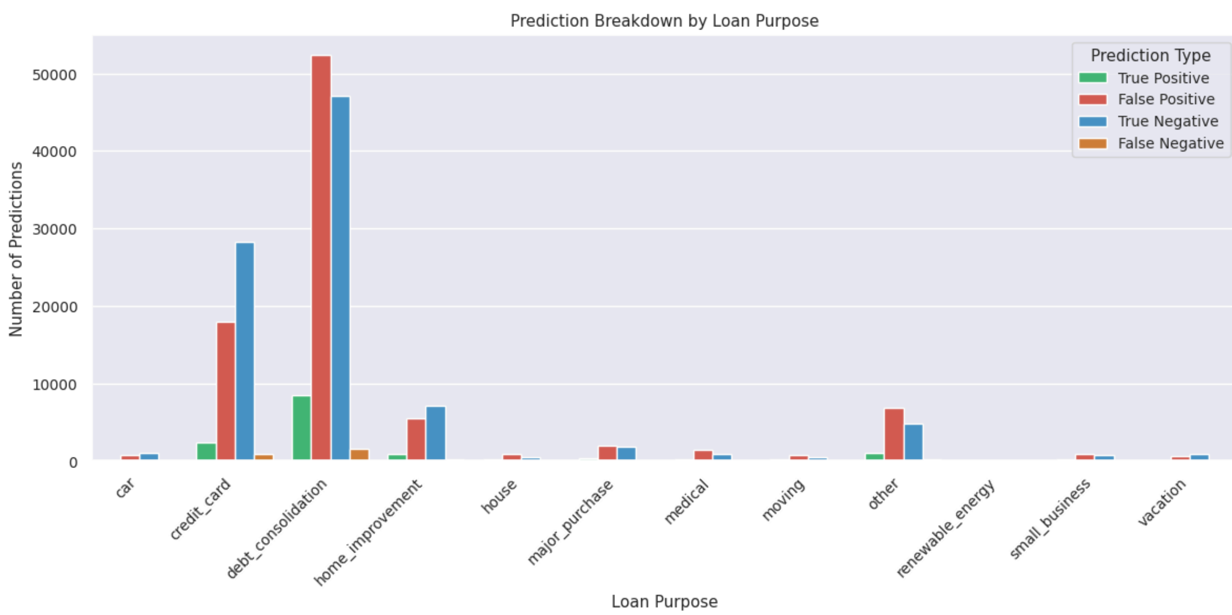

Figure E2: Loan Purpose Prediction

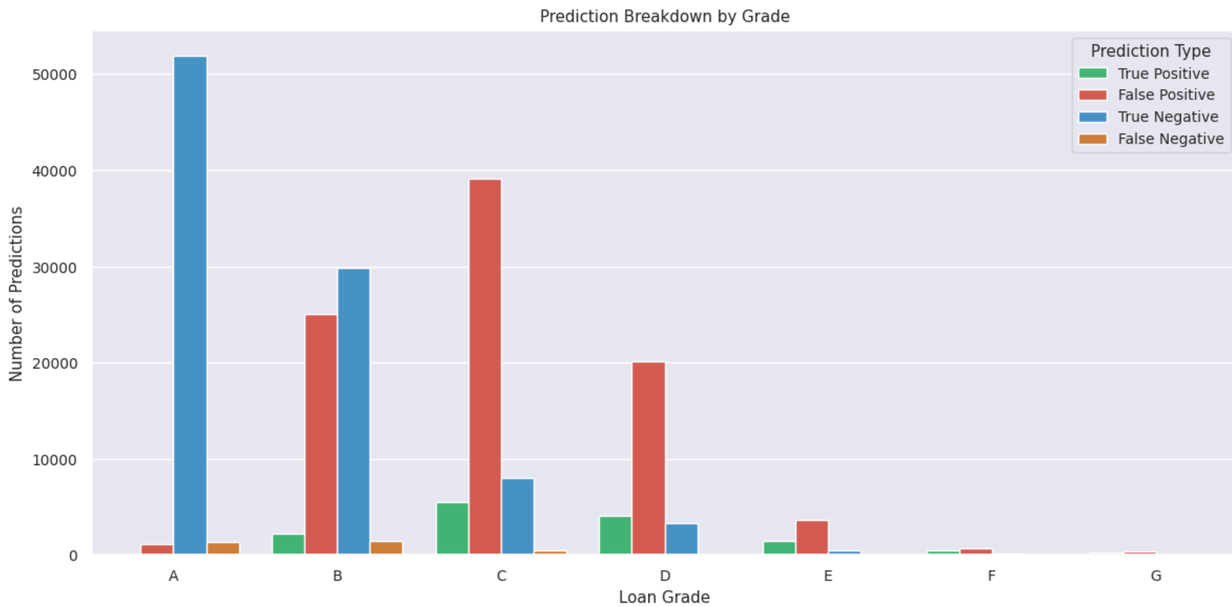

Figure E3: Grade Prediction

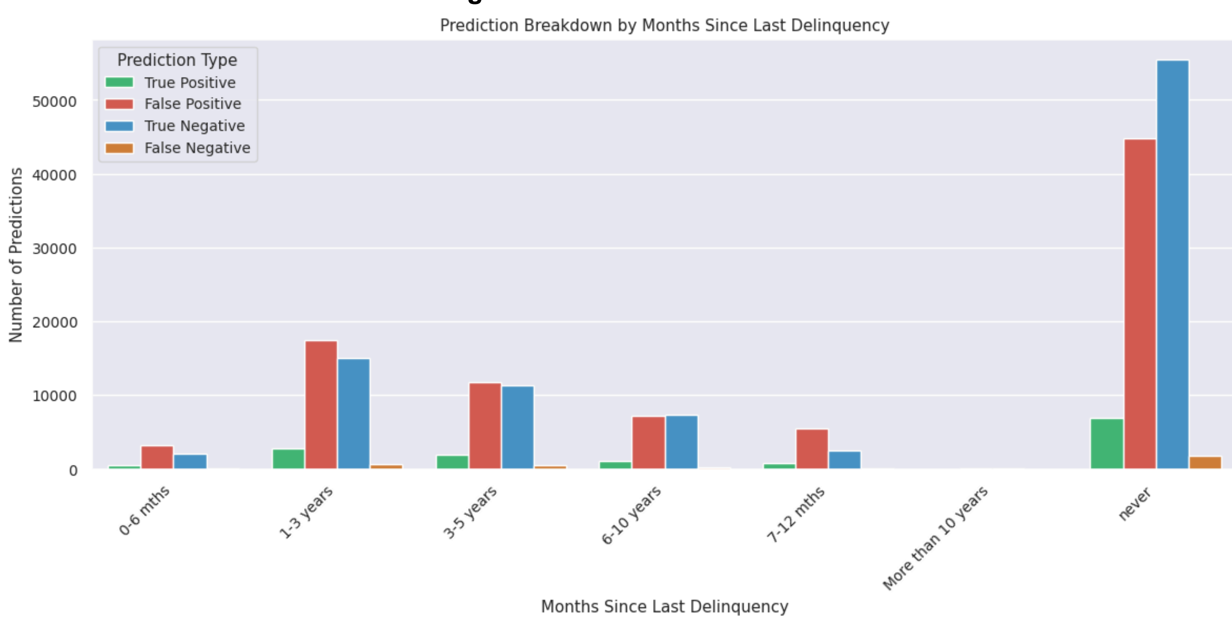

Figure E4: Mths_since_last_delinq Prediction

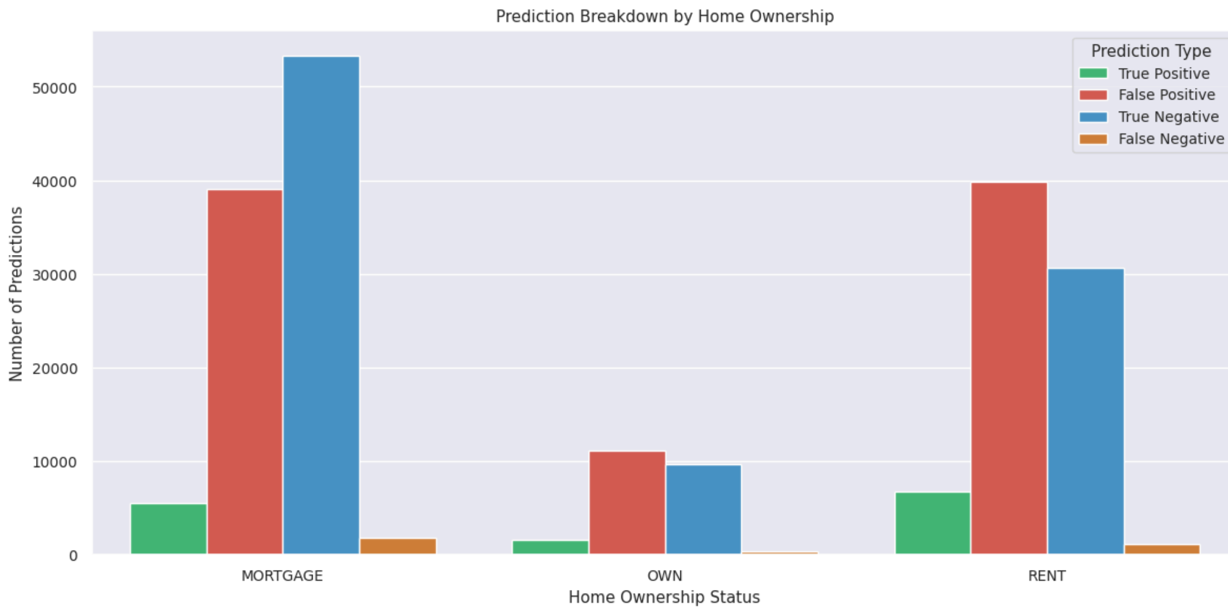

Figure E5: home_ownership Predictions