VRIJE UNIVERSITEIT AMSTERDAM

Group Assignment 4: Credit Scoring

Quantitative Financial Risk Management

Authors: A 15

Jip de Boer j.a4.de.boer@student.vu.nl (2710065)

Yunji Eo y.eo@student.vu.nl (2735445)

Lars de Graaff l.de.graaff@student.vu.nl (2790679)

Xuan Zhou x.zhou9@student.vu.nl (2653511)

Supervisor:

Dr. C.S. Bos

May 26, 2025

1 Introduction

This report develops and evaluates multiple default-classification models that leverage individual borrower profiles to accurately predict whether an applicant is expected to default or not. Such default and non-default classifier models are crucial for credit providers to assess borrower risk, inform lending decisions and manage portfolio performance effectively. The classifiers investigated in this study are a logistic regression, a Decision Tree and a Random Forrest. To train these models a credit default dataset from Kaggle [Fusion, 2011] is used which contains various characteristics of borrowers. Although the classifier models form the core methodology of this study, they are preceded by a crucial phase of extensive data cleaning and preparation.

This report proceeds as follows, Section 2 outlines the characteristics of the raw data, addresses issues such as the handling of missing values and prepares all predictor variables to establish a robust set of predictor variables for the modeling phase. Following this data preparation, Section 3 describes the application of logistic regression. To provide a comparative perspective on predictive performance, Section 4 explores alternative machine learning approaches. This includes Decision Tree and Random Forest classifiers. Finally, Section 5 concludes the report as it proposes recommendations for further actions.

2 Data

The dataset that is investigated in this report is a credit default dataset obtained from Kaggle [Fusion, 2011] with a total sample size of 149,563 observations. To give an overview of what data is available in this dataset a data dictionary is provided in Table 1. This section covers three data-cleaning steps, namely the removal of implausible observations, the treatment of missing values and the log transformation of heavily right-skewed predictor variables.

Table 1: Data dictionary.

Variable Name	**Description**	**Type**
Defaulted	Person experienced 90 days past due delinquency or worse	Binary (1=Yes/0=No)
Revolving Credit Utilization	Total balance on credit cards and personal lines of credit except real estate and no installment debt like car loans divided by the sum of credit limits	Percentage
Age	Age of borrower in years	Integer
30–59 Days PDC	Number of times borrower has been 30–59 days past due but no worse in the last 2 years	Integer
Debt Ratio	Monthly debt payments, alimony, living costs divided by monthly gross income	Percentage
Monthly Income	Monthly income	Integer
Open Credit Lines Count	Number of Open loans (installment like car loan or mortgage) and Lines of credit (e.g. credit cards)	Integer
90+ Days Late Count	Number of times borrower has been 90 days or more past due	Integer
Real Estate Loans Or Lines	Number of mortgage and real estate loans including home equity lines of credit	Integer
60–89 Days PDC	Number of times borrower has been 60–89 days past due but no worse in the last 2 years	Integer
Number Of Dependents	Number of dependents in family excluding themselves (spouse, children etc.)	Integer

Data cleaning - realistic conditions

As a first step for the data-cleaning phase, we define the "unrealistic" value conditions based on the variable features and drop the corresponding flagged rows. The conditions applied are as follows.

- `age` ≤ 10: Drop customer records where the age is less than or equal to 10, as individuals that young are unlikely to legally hold independent credit accounts.
- `Revolving Credit Utilization` > 1000: Remove records where the revolving credit utilization ratio exceeds 1000%, which would indicate the customer is using more than 10 times their available credit limit. This is an implausible and generally institutionally invalid scenario.
- `30-59 Days PDC` = 98: Drop records where the number of times a customer was 30 to 59 days past due is equal to 98. This value is used as a placeholder for missing data and is also unrealistically high given typical reporting ranges.
- `60-89 Days PDC` = 98: Similarly, remove observations where this variable equals 98, as it indicates missing or corrupted values rather than a meaningful delinquency count.
- `90+ Days Late Count` = 98: Exclude records where this value equals 98, for the same reasons as given for the 30–59 and 60–89 days late counts.

After the identification of these unrealistic rows, 437 observations are dropped because of the conditions described above.

Data cleaning - missing value handling

Next, missing values are addressed to circumvent distortions in the model training. First, we summarize the missing value statistics for the dataset to get an understanding of where values are missing. To do so, an overview is presented in Table 2. Only two of the eleven predictor variables in the dataset have missing values where `monthly income` has the highest percentage of missing values, with a missing value rate of 19.77% and `number of dependents` has a missing value rate of 2.60%, both rounded to 2 decimal places.

Table 2: Missing value summary.

String Name	Missing Count	Missing Percentage
Monthly Income	29565	19.77%
Number of Dependents	3882	2.60%

Before deciding how to handle missing values, we first explore the available options. One approach is to remove all incomplete observations, another is to impute missing values using the median (for continuous variables) and the mode (for discrete variables). The risk of imputing missing values is creating synthetic profiles that can be unrealistic and hard to detect given the high prevalence of missing entries, while deleting those observations risks distorting predictor distributions and biasing classifiers trained on the reduced dataset. To assess which strategy best supports our goal of accurately predicting defaults and non-defaults, we begin by visualizing the distribution of the predictor variables before and after deleting missing values to identify if this significantly changes the predictor variable distributions. If not, deleting observations with missing values is preferable, as it prevents the creation of synthetic profiles and poses a negligible risk of classifier bias.

Histograms of the predictor variables based on the original dataset in which only the nonsensical observations are deleted are shown in Figure 7 in Appendix A and a version with custom bin widths for improved readability is presented in Figure 1.

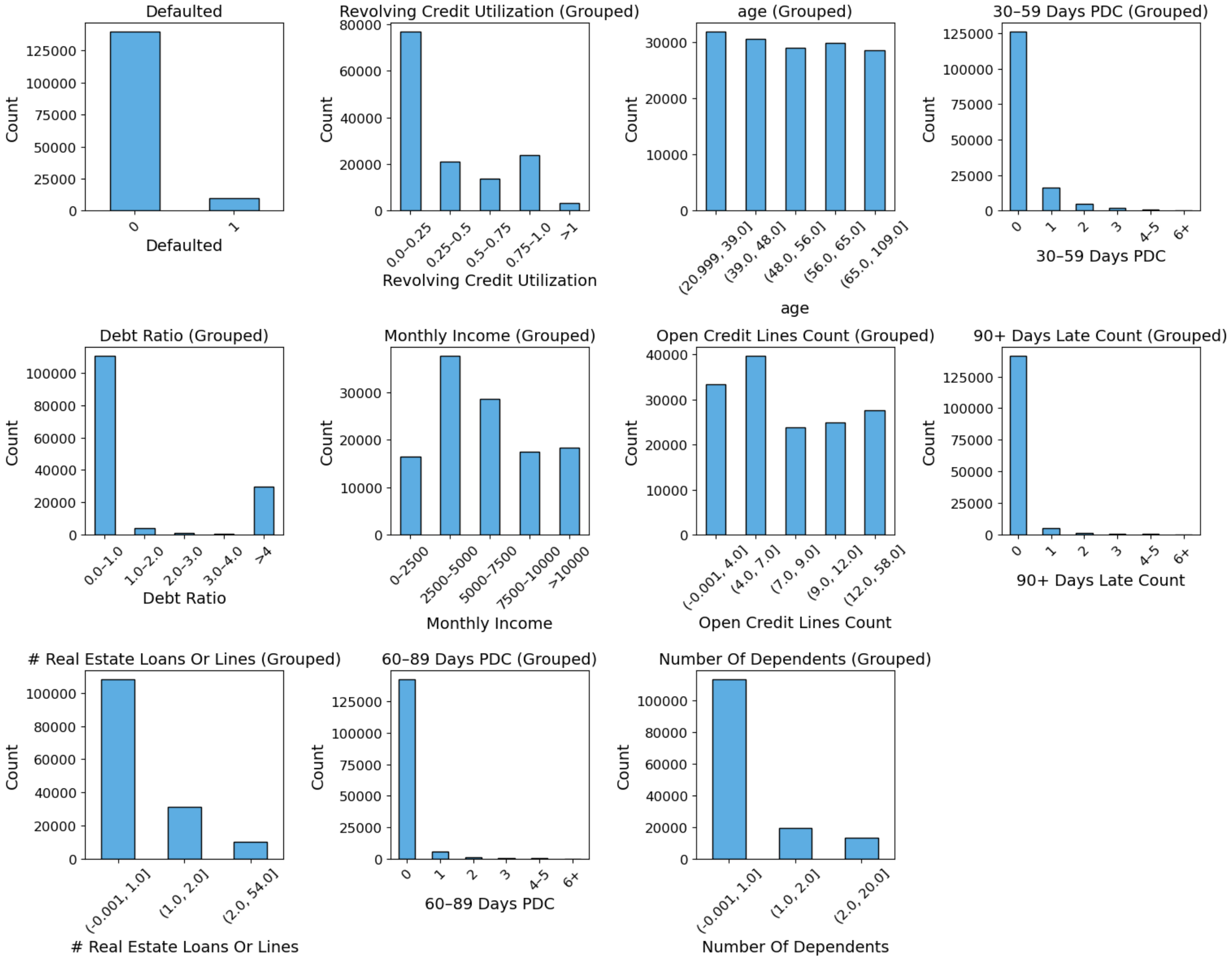


Figure 1: Histograms of all predictor variables based on the raw dataset after dropping the nonsensical observations. Bin widths vary to improve visibility. For example, rightmost bars often span wider ranges than those on the left.

To better understand the structure of the missing data, we also plot histograms for observations that contain at least one missing value, which accounts for roughly 20% of the full sample. This plot of histograms is shown in Figure 8 in Appendix A. One notable finding is that the predictor variable histograms for records with missing values closely match those obtained after only removing nonsensical observations (see Figure 7 in Appendix A). This appears to be strengthening the argument that dropping the observations with missing values might not affect the sample distributions of the predictor variables. To further investigate this, we drop all rows with missing values and visualize the resulting predictor variable distributions in Figure 2.

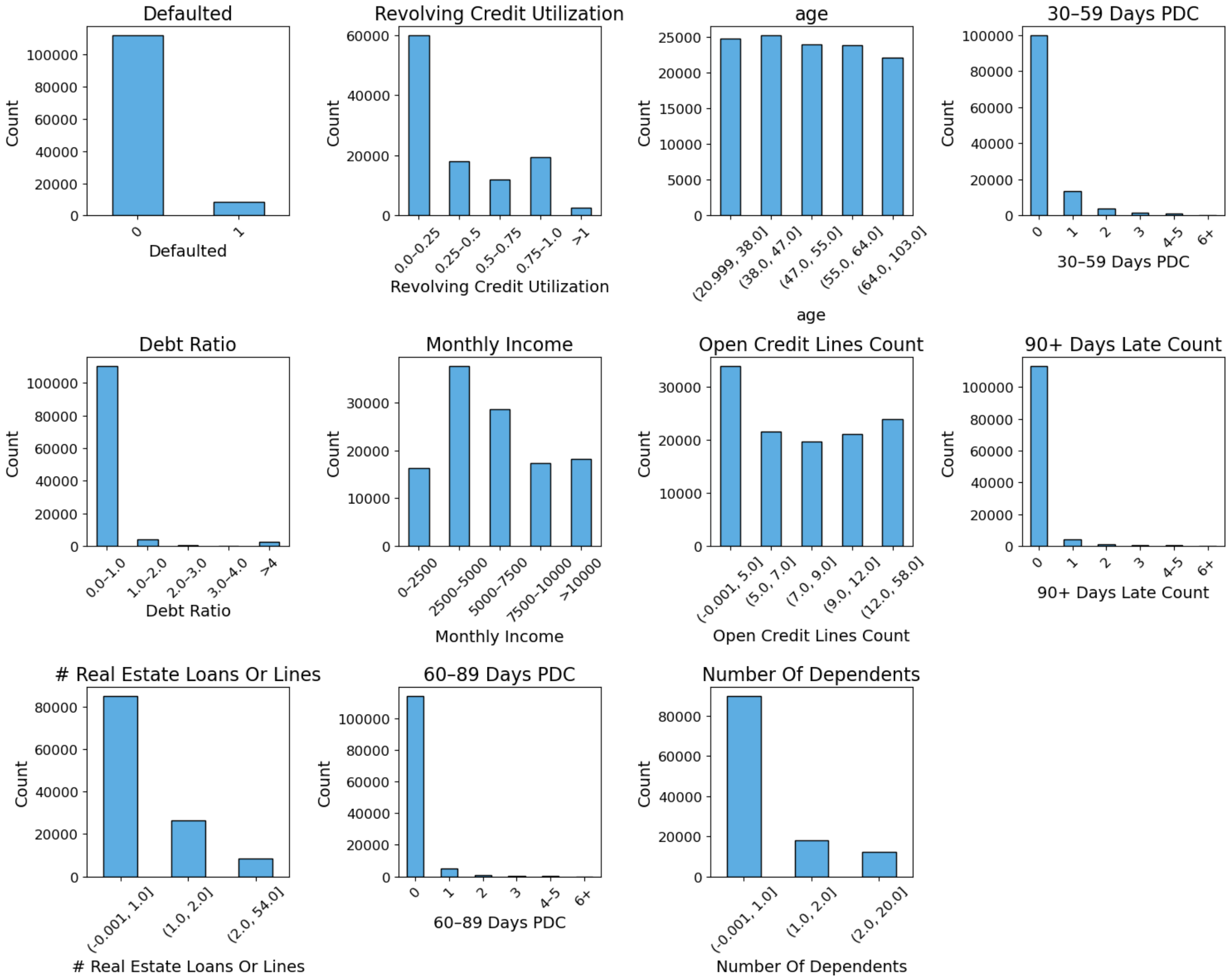


Figure 2: Histograms of all predictor variables based on the dataset after dropping missing values and deleting the nonsensical observations. Bin widths vary to improve visibility. For example, rightmost bars often span wider ranges than those on the left.

Comparing these predictor variable histograms in Figure 1 and Figure 2 reveals several shifts in the data distribution. For instance, the `age` variable shows a higher concentration in the middle-age bins, while extreme values in `Debt Ratio` (greater than 4) are significantly reduced. Similarly, `Open Credit Lines Count` shows an increased frequency in the lower range (e.g., from 0 to 5). These changes suggest that removing missing values does slightly affect the predictor variables distributions. However, the specific changes only slightly lowers the influence of extreme values. This is where a trade-off needs to be made. Although deleting real data could be seen as not optimal, this route on the other hand mitigates the risk of creating unrealistic profiles by imputing missing values as we can not manually check all new 29,565 profiles that then would be created. Therefore, the decision is made to delete the observations with missing values. Also, one other benefit of this route is that the data becomes slightly more balanced as after this adjustment to the dataset the default percentage in the dataset becomes slightly higher as indicated in Table 3 below.

Table 3: Summary of Default and Non-default Observations Before and After Removing Missing Values.

	Class	Count	Percentage
Original	Non-defaults (0)	139,690	93.40%
	Defaults (1)	9,873	6.60%
After Dropping Missing Values	Non-defaults (0)	111,730	93.11%
	Defaults (1)	8,268	6.89%

After removing all observations with missing values, the histograms of the final predictor variable set, prior to rebalancing, are shown in Figure 2. It can be observed that the predictor variables `Revolving Credit Utilization`, `Debt Ratio`, `Real Estate Loans or Lines` and `Number of Dependents` are mostly right-skewed. This is evident from the concentration of observations in the lower value ranges. Note again that the bars do not represent equal-width intervals as explained above. Also, the credit history related variables `30-59 Days PDC`, `60-89 Days PDC`, and `90+ Days Late Count` have similar data distribution patterns and align with the reality that short-term overdue can be mostly recovered while when it enters the stage of longer overdue payments, the conversion rates become higher as indicated by the 60-89 days and 90+ days past due rates are relatively higher. Overall, removing both nonsensical entries and those with missing values provides the most reliable foundation for accurate default versus non-default prediction

Data cleaning - log transformations

After treating missing values and noting that most predictors are right-skewed, we apply log transformations to selected variables to improve distributional symmetry and reduce the risk of heteroskedasticity disturbing the predictor performance of the used classifiers in this study. We leave the Y variable `Defaulted` and the explanatory variable age unchanged, the other variables are log transformed by taking the log of their observed values. The variable histogram with the log transformed predictor variables is shown in Figure 3 and can be compared to the histograms of the predictor variables without the log transforms in Figure 9 in Appendix A. A review of the variable histograms reveals that `Revolving Credit Utilization`, `30-59 Days PDC`, `60-89 Days PDC`, and `90+ Days Late Count` retain distributions similar to their pre-transformation shapes. In contrast, `Monthly Income` and `Open Credit Lines Count` exhibit markedly more symmetric, near-normal distributions. Finally, log-transforming `Debt Ratio`, `Real Estate Loans or Lines`, and `Number of Dependents` yields finer grained binning and substantially diminishes the impact of extreme values.

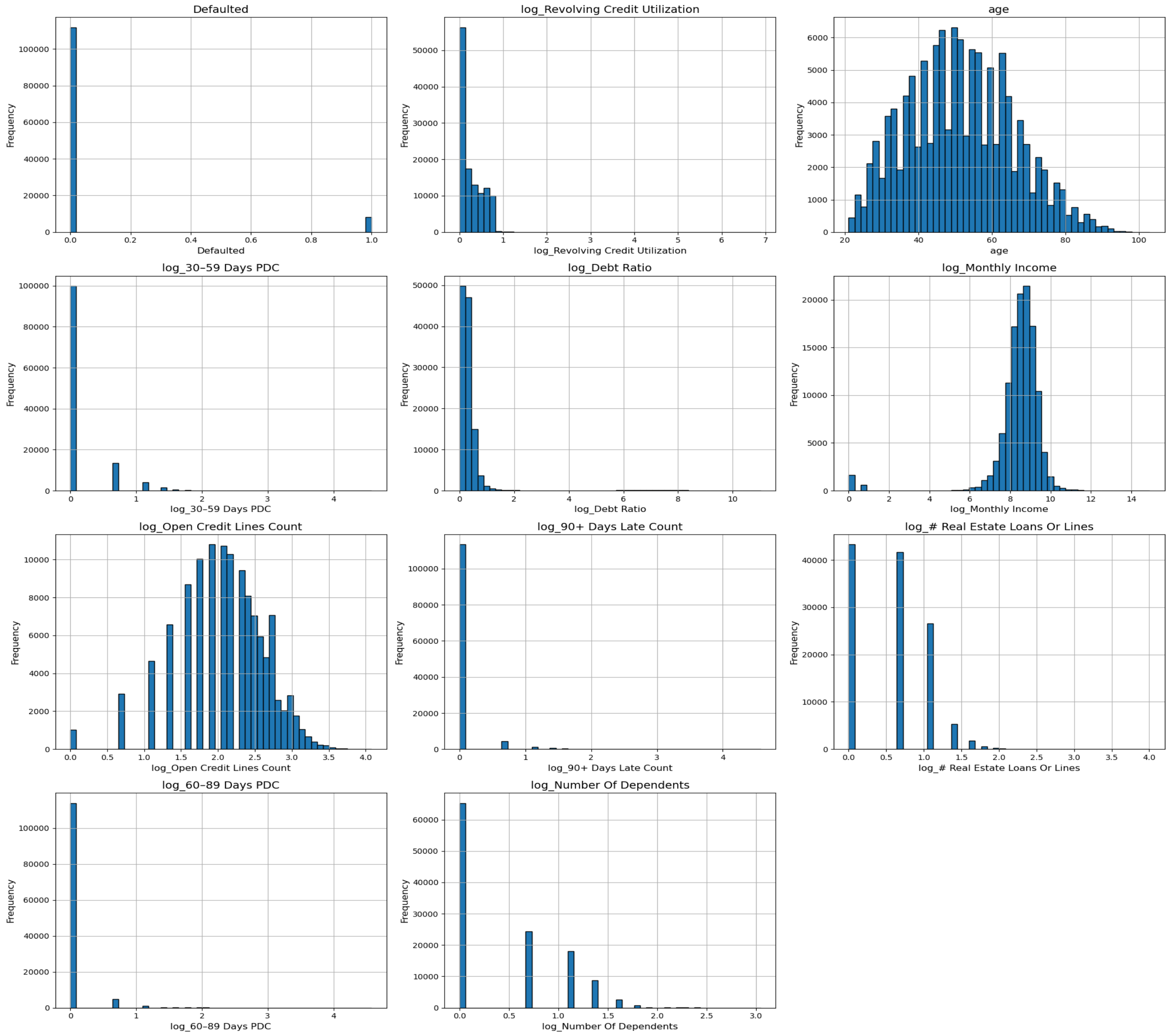


Figure 3: Variable histogram with cleaned and log transformed data.

In addition, we also plot the scatter chart for each predictor variable to identify potential outliers hidden in the data, the scatter plot is shown in the appendix as Figure 10. In line with the observations above, we still notice the outliers for the variables of `30-59 Days PDC`, `60-90 Days PDC`, and `90+ Days Late Count`. By the logic of if the log-delinquency values are greater than 3, we remove these four rows of observations and move to the further analysis and modeling part.

Data cleaning - downsampling and upsampling

Considering the fact that the default customer counts are far smaller than the ones of non-default customers, the predicted results can be biased on predicting non-defaults which may harm the usefulness of the default and non-default prediction models trained in this study. For this report we set the aim to predict both defaults and non-defaults equally well, although the argument that predicting defaults correctly yields a higher benefit for the credit providers can be seen as valid. However, since we do not have information of the costs of predicting false defaults and false non-defaults, we move forward with the aim of predicting both equally well. To deal with this unbalanced dataset, this study applies undersampling (use fewer non-default data) and oversampling (generate synthetic observations from the defaulted group), specifically using SMOTE (Synthetic Minority Over-sampling TEchniques, seek for the closest ones in the predictor space and take the sample as the linear combinations). Hence, the data is both undersampled to a 50/50 ratio of both defaults and non-defaults with the number of non-defaults downsampled to equal the number of defaults and oversampled to a 50/50 ratio using SMOTE such that the number of defaults equals the number of non-defaults.

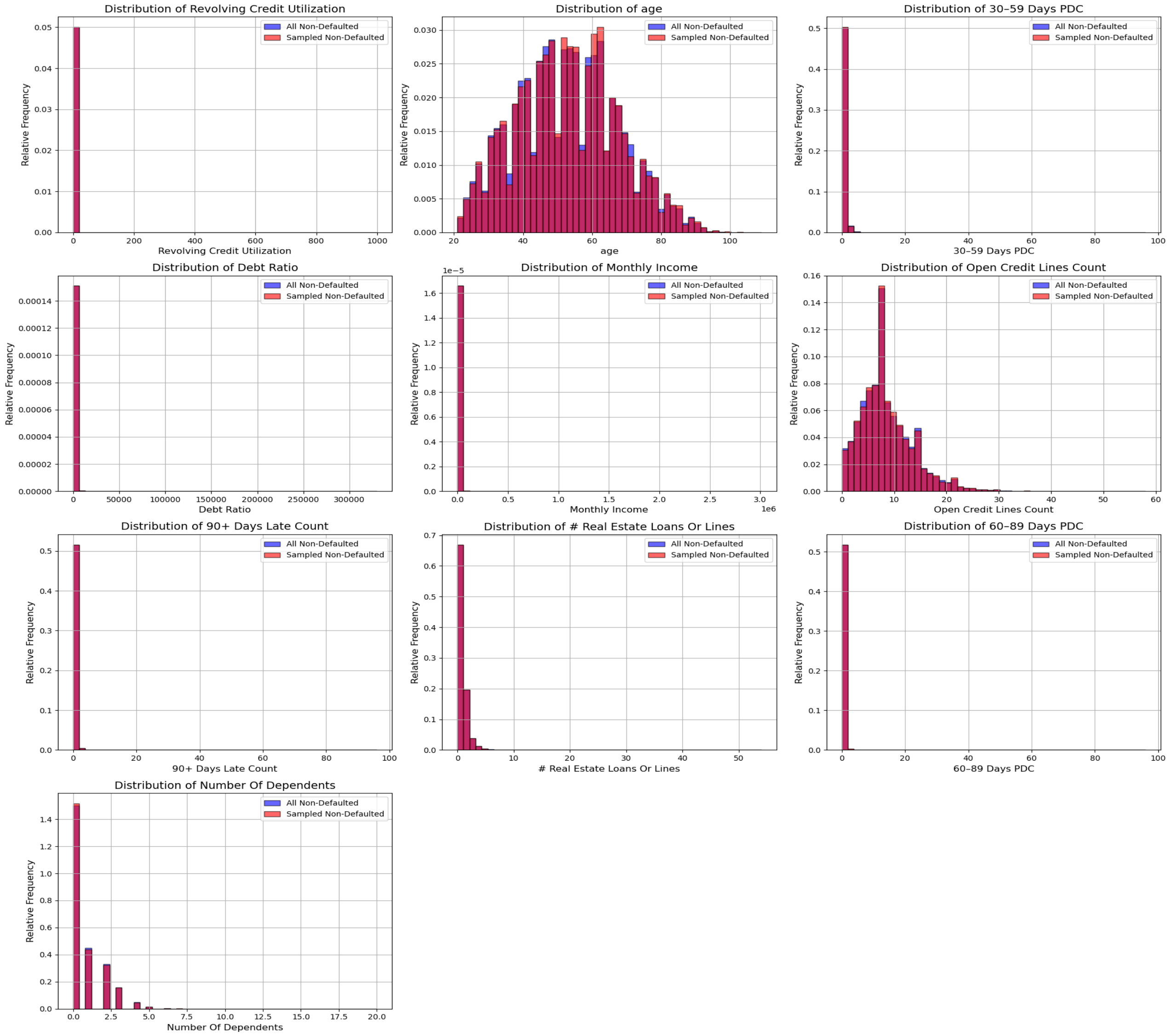


Figure 4: Histograms, one for each predictor variable, to compare the distribution of each for the non-default group both of the preprocessed group and the undersampled group for rebalancing.

We first conduct the graphic comparison analysis between the set of downsampled non-defaulted observations and all non-defaulted observations. The plot to illustrate that both yield approximately the same distributions for each predictor variable is shown in Figure 4. In this Figure the blue bar represents all non-defaulted observations and the red one represents the downsampled non-defaulted set of observations. We observe that for most predictive variables, the bar distributions overlap. This indicates that the downsampling process does not influence the distributions of the predictor variables which makes it robust to the risk of biased predictions due to the downsampling. The same result yields for the oversampling technique using SMOTE.

To summarize, to address the potential class imbalance issue, two resampling strategies were applied to create balanced datasets,

- Downsampling: the logic is that it randomly selects the non-defaulted records to match the number of defaulted ones, and shuffles all together to create the new dataframe, resulting in a balanced dataset as illustrated in Table 4.
- SMOTE (Synthetic Minority Oversampling Technique): it creates the "artificial" population by linear interpolating to find the closest neighbors, producing a balanced dataset as displayed in Table 4.

Table 4: Downsampling and Upsampling comparison of default and non-default.

	Default or not	Count	Percentage
Downsampling	Non-defaults (0)	8265	50.0%
	Default (1)	8265	50.0%
(SMOTE) Upsampling	Non-defaults (0)	111729	50.0%
	Default (1)	111729	50.0%

To illutrate that the finalized datasets for subsequent analysis are balanced Table 4 can be investigated. Both downsampling and upsampling strategies maintain a 50%/50% ratio of defaulted to non-defaulted borrowers, yielding two balanced, cleaned datasets to train and validate the default and non-default classifier models on.

3 Regression Analysis

This section displays the results from applying Logistic Regression analysis to predict whether an applicant is expected to default or not using the prepared dataset in the previous section. Logistic regression models were applied separately on two balanced datasets, processed through downsampling and SMOTE, which are described previously in the Data section.

Table 5: Model Performance Comparison: Downsampling vs. SMOTE (Logistic Regression Only).

Dataset	**Model**	**Accuracy**	**AUC**	**Prec (D)**	**Rec (D)**	**F1 (D)**	**Prec (ND)**	**Rec (ND)**	**F1 (ND)**
Downsampling	Logistic Regression	0.7621	0.8437	0.7957	0.7054	0.7478	0.7354	0.8189	0.7749
SMOTE	Logistic Regression	0.7589	0.8473	0.7736	0.7321	0.7523	0.7457	0.7857	0.7652

Prec (D): Precision (Default), Rec (D): Recall (Default), F1 (D): F1 score (Default), Prec (ND): Precision (Non-Default), Rec (ND): Recall (Non-Default), F1 (ND): F1 (Non-Default).

Model performance metrics, including Accuracy, Precision, Recall, and F1 Score, were compared between the downsampling and SMOTE datasets. The comparative results are presented in Table 5. The results show similar accuracy and AUC values across both datasets, with slightly different precision and recall scores for the default and non-default classes.

Table 6: Confusion Matrices for Logistic Regression.

Dataset	Model	Actual	Predicted: 0	Predicted: 1
Downsampling	Logistic Regression	0	6768	1497
		1	2435	5830
SMOTE	Logistic Regression	0	87789	23940
		1	29931	81798

Label 0 = Non-Default, Label 1 = Default. Actual: 0 and Predicted: 0 is TN; Actual: 1 and Predicted: 0 is FN; Actual: 0 and Predicted: 1 is FP; and Actual: 1 and Predicted: 1 is TP.

Table 6 presents the confusion matrices for logistic regression on both the downsampling and SMOTE datasets. For the downsampling dataset, the model correctly classified 6,768 non-default cases and 5,830 default cases. However, it also misclassified 1,497 non-default as defaults and 2,435 defaults as non-defaults. For the SMOTE dataset, the model correctly predicted 87,789 non-defaults and 81,798 defaults. It misclassified 23,940 non-defaults as defaults, and 29,931 defaults as non-defaults.

Overall, logistic regression showed reasonable performance in predicting defaults, serving as a basedline for further comparision with machine learning models in the next section.

4 Extra Analysis

Table 7: Model Performance Comparison: Downsampling vs. SMOTE.

Dataset	Model	Accuracy	AUC	Prec (D)	Rec (D)	F1 (D)	Prec (ND)	Rec (ND)	F1 (ND)
Downsampling	Logistic Regression	0.7621	0.8437	0.7957	0.7054	0.7478	0.7354	0.8189	0.7749
Downsampling	Decision Tree	0.6817	0.6817	0.6805	0.6849	0.6827	0.6829	0.6784	0.6806
Downsampling	Random Forest	0.7697	0.8415	0.7792	0.7527	0.7657	0.7608	0.7867	0.7735
SMOTE	Logistic Regression	0.7589	0.8473	0.7736	0.7321	0.7523	0.7457	0.7857	0.7652
SMOTE	Decision Tree	0.9158	0.9158	0.9098	0.9232	0.9164	0.9221	0.9084	0.9152
SMOTE	Random Forest	0.9509	0.9895	0.9598	0.9413	0.9505	0.9424	0.9606	0.9514

Prec (D) stands for the Precision (Default), Rec (D) stands for Recall (Default), F1 (D) stands for F1 score (Default), Prec (ND) stands for Precision (Non-Default), Rec (ND) stands for Recall (Non-Default), F1 (ND) stands for F1 (Non-Default).

Extra Analysis is conducted by using Decision Tree and Random Forest to assess whether these machine learning methods would have a higher precision than the standard Logistic Regression above. Again, the analysis was applied to both the downsampling and SMOTE datasets. Performance metrics are reported in

Table 7 including the Logistic Regression result to compare. The results show that while Random Forest achieves the highest accuracy, especially on the SMOTE dataset, Decision Tree performs less effectively than both Random Forest and Logistic Regression in the Downsampling dataset. Additionally, Random Forest yields the highest F1 scores, highlighting the benefits of this method.

For each model, confusion matrices were generated in table 8 to provide a detailed view of performance. For both datasets, Random Forest outperforms Logistic Regression and Decision Tree in terms of the number of correctly classified cases. Specifically, Random Forest achieves the highest true positives and true negatives with the fewest misclassifications in the SMOTE dataset. Decision Tree also improves over Logistic Regression in the SMOTE dataset. Overall, Logistic Regression shows more misclassifications compared to the machine learning models.

Table 8: Confusion Matrices for Different Models.

Dataset	**Model**	**Actual**	**Predicted: 0**	**Predicted: 1**
Downsampling	Logistic Regression	0	6768	1497
		1	2435	5830
Downsampling	Decision Tree	0	5607	2658
		1	2604	5661
Downsampling	Random Forest	0	6502	1763
		1	2044	6221
SMOTE	Logistic Regression	0	87789	23940
		1	29931	81798
SMOTE	Decision Tree	0	101500	10229
		1	8579	103150
SMOTE	Random Forest	0	107324	4405
		1	6560	105169

Label 0 = Non-Default, Label 1 = Default. Actual: 0 and Predicted: 0 is TN; Actual: 1 and Predicted: 0 is FN; Actual: 0 and Predicted: 1 is FP; and Actual: 1 and Predicted: 1 is TP.

In addition to model comparison, the Receiver Operating Characteristic (ROC), which is a graphical representation that illustrates the diagnostic ability, curve is presented. The ROC curve plots the true positive rate against the false positive rate at various threshold settings. A key summary measure of the ROC curve is the AUC. An AUC value close to 1.0 indicates superior classification performance. When comparing the different classifier models, Random Forest achieved the highest AUC values, particularly with SMOTE, followed by Decision Tree and Logistic Regression. This shows that ML methods can capture complex relationships in the data more effectively than the standard Logistic Regression model.

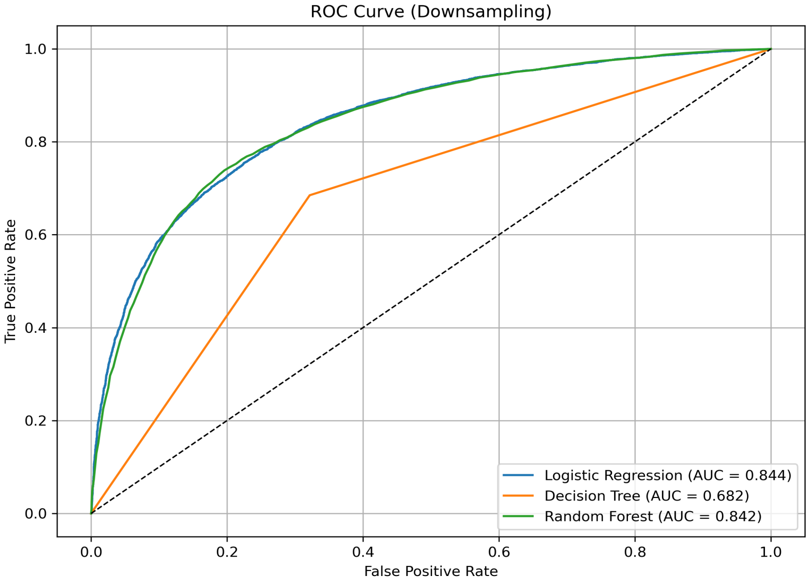


Figure 5: ROC curve of down-sampling case results of logistic regression, decision tree, random forest.

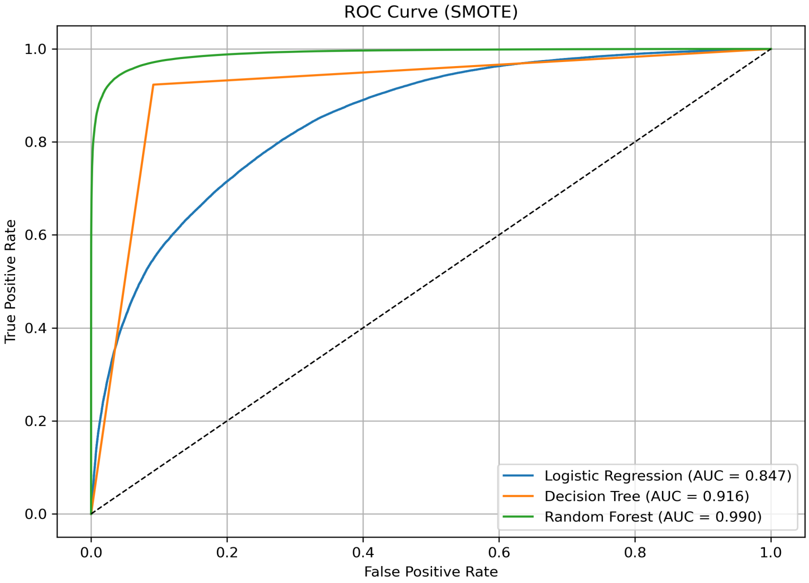


Figure 6: ROC curve of SMOTE case results of logistic regression, decision tree, random forest.

5 Recommendations for Further Actions

Before modeling and in-depth analysis, a detailed data preparation and cleaning procedure is performed to ensure the robustness of modeling observations. First, we define the 'unrealistic' conditions to filter the observations. Then, we drop the missing values after considering the two methods of handling missing values: simply dropping them or replacing them with the median. We also check the distributions of the missing values. Third, we conduct a logarithmic transformation to decrease the impact of unstable variances and achieve a normal distribution. We then identify and remove outliers graphically. Finally, we generate two final datasets using the downsampling and upsampling approaches with the SMOTE strategy to prepare balanced datasets for the subsequent modeling procedure.

However, there are several possible improvements that could be made. Firstly, in the 'unrealistic' conditions section, due to the limited information about the target credit products, for example, whether it is a mortgage or a credit card, we have set a fairly conservative minimum age limit rather than a 'minimum plus maximum'. We have also observed that the older age group, 64–103, still accounts for a significant proportion of the overall data set. Secondly, regarding the missing values, the majority of missing values come from the 'Month Income' field, accounting for 19.77%. We have to drop these values due to the lack of alternative data that can represent customers' income levels in our current dataset. This also suggests that we better include more variables that represent income levels. Lastly, the default customer ratio in our original data set is 6.6%, which increases to 6.89% after the missing values are removed. When we create balanced upsampling and downsampling datasets, the density of defaults increases to 50%. The decision is based on the goal of achieving a higher level of precision in identifying defaults, though, due to missing information about the company, its products, target customers, and risk appetite, this choice may also lead to inaccurate results in further analyses instead.

Regarding the regression and extra analyses, it is notable that the Random Forest model significantly outperformed logistic regression in this study. This finding is particularly interesting when compared to the general view that advanced machine learning models do not always offer superior performance over logit regression. Additionally, considering alternative resampling techniques like ADASYN could provide further insights or enhance model robustness.

References

Credit Fusion. Give me some credit. `https://www.kaggle.com/c/GiveMeSomeCredit`, 2011. Accessed: 2025-05-26.

A Appendix

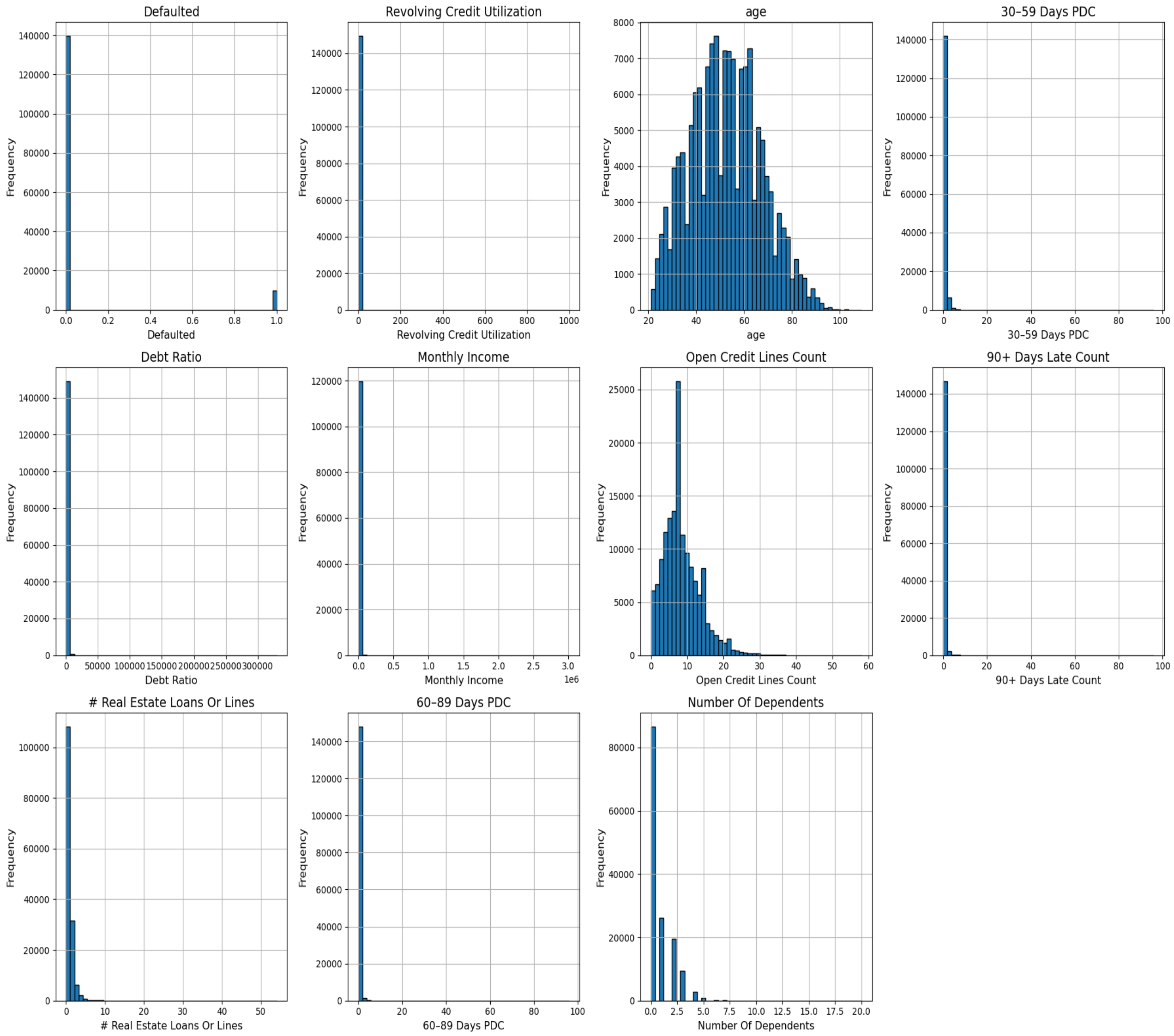


Figure 7: Histograms, one for each predictor variable, with all data from the non preprocessed dataset included, only nonsensical observations are deleted.

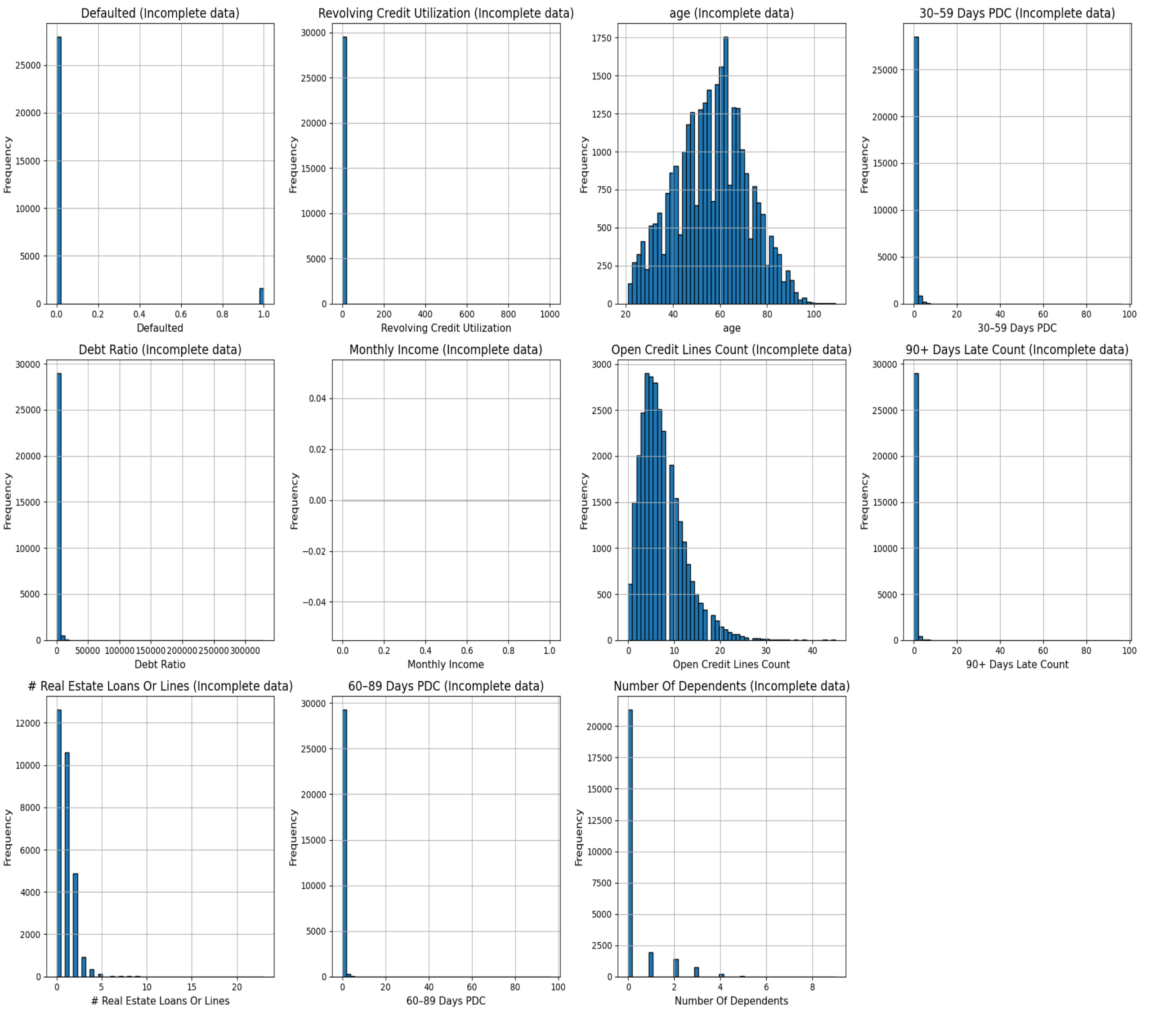
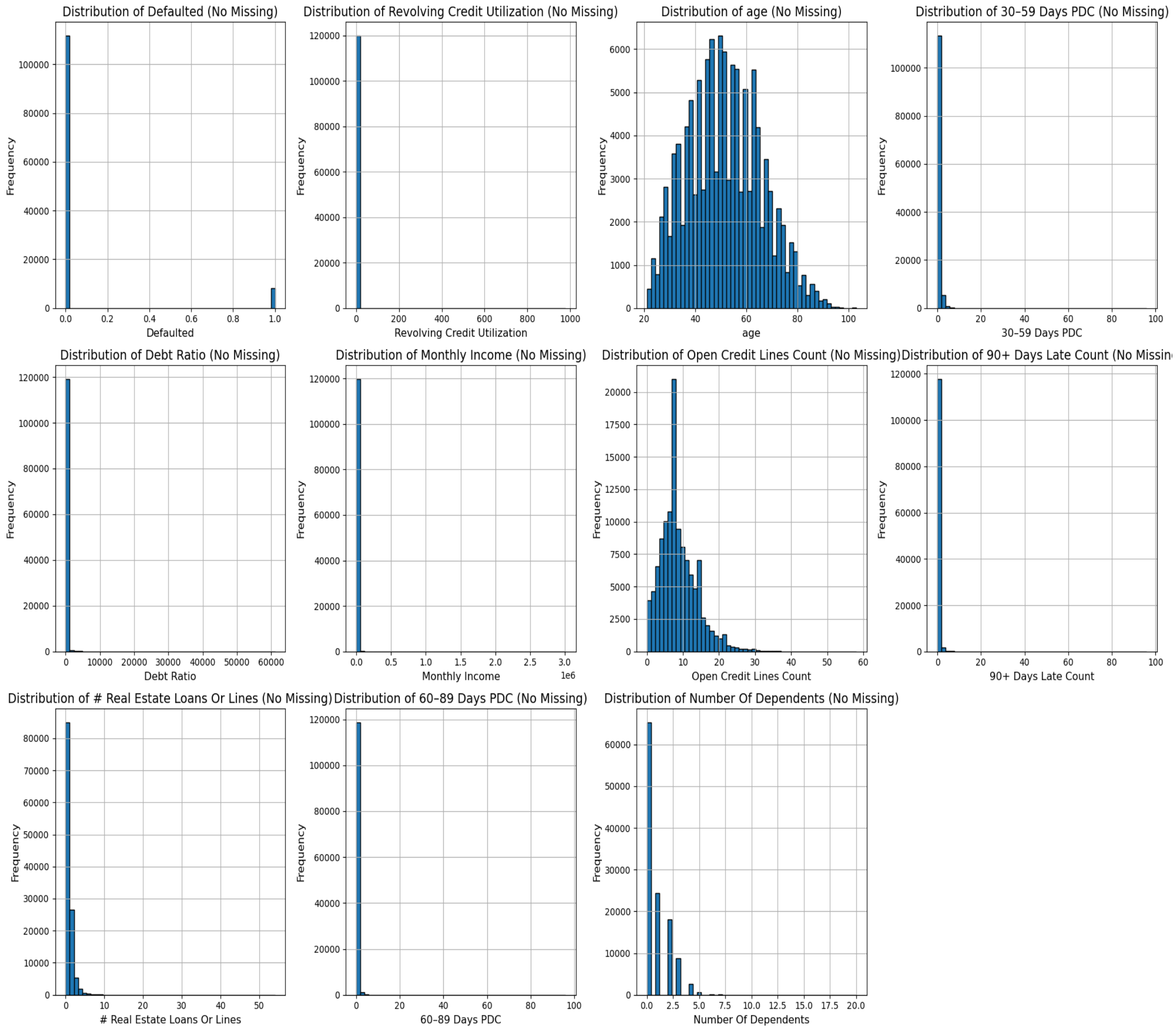


Figure 8: Histograms, one for each predictor variable, of the observations with at least one missing value.



Distribution of Defaulted (No Missing)

Distribution of Revolving Credit Utilization (No Missing)

Distribution of age (No Missing)

Distribution of 30–59 Days PDC (No Missing)

Distribution of Debt Ratio (No Missing)

Distribution of Monthly Income (No Missing)

Distribution of Open Credit Lines Count (No Missing)

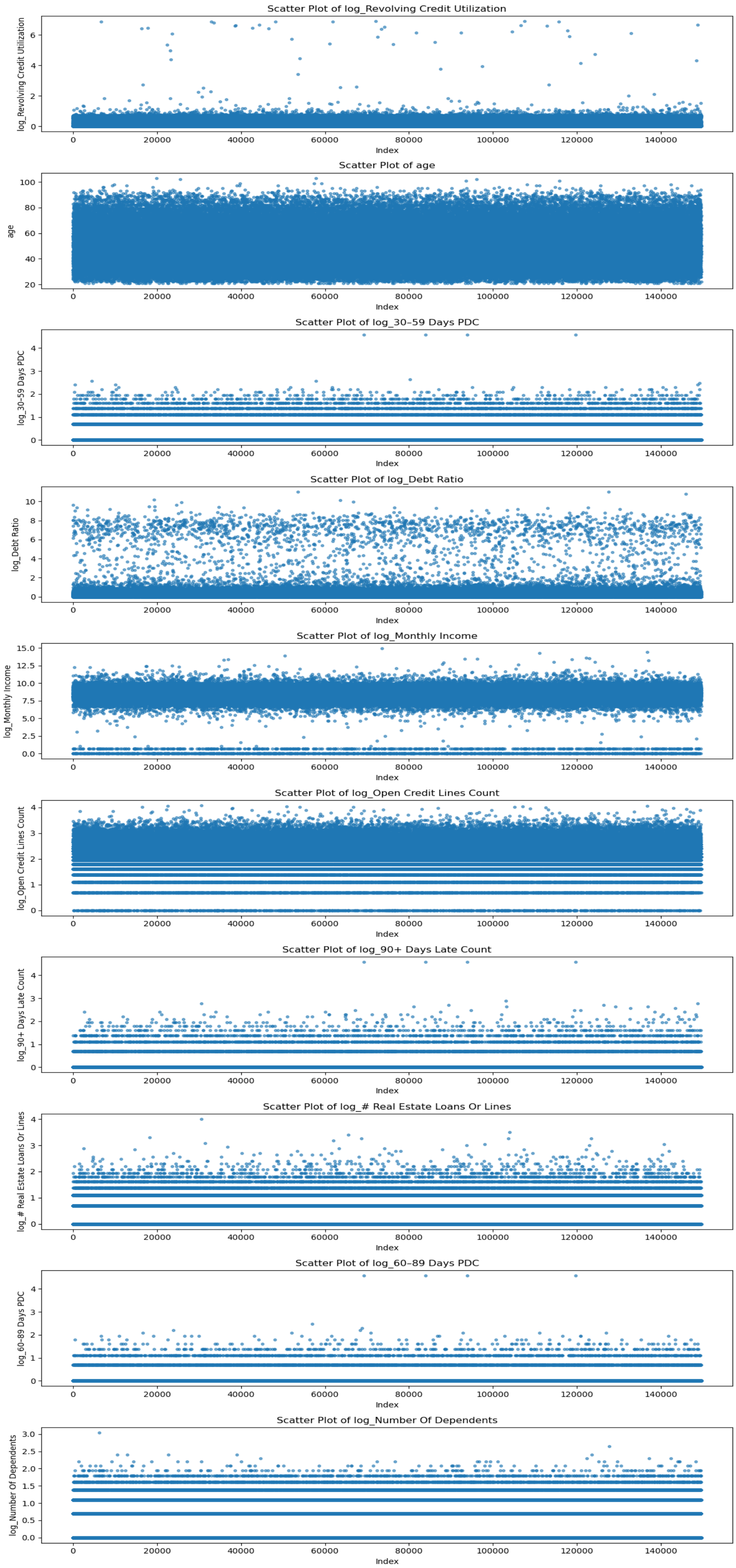
Distribution of 90+ Days Late Count (No Missin

Distribution of # Real Estate Loans Or Lines (No Missing)

Distribution of 60–89 Days PDC (No Missing)

Distribution of Number Of Dependents (No Missing)

Figure 9: Histograms, one for each predictor variable, with the nonsensical observations as well as the missing values deleted.



Scatter Plot of log_Revolving Credit Utilization

Scatter Plot of age

Scatter Plot of log_30–59 Days PDC

Scatter Plot of log_Debt Ratio

Scatter Plot of log_Monthly Income

Scatter Plot of log_Open Credit Lines Count

Scatter Plot of log_90+ Days Late Count

Scatter Plot of log_# Real Estate Loans Or Lines

Scatter Plot of log_60–89 Days PDC

Scatter Plot of log_Number Of Dependents

Figure 10: Predictor variables scatter plots with cleaned and log transformed data.