Loan Default Prediction for Imbalanced Data

Ruby Li

Department of Applied Statistics, Social Science, and Humanities, New York University

APSAT-GE 2047: Messy Data and Machine Learning

Instructor: Ravi Shroff

May 2nd, 2024

Table of Contents

1. Motivation

Financial risk is a ubiquitous phenomenon encompassing various types of financing-related risks. Among these, credit risk, emanating from the Probability of Default (PD) in loan repayments, is a primary focus of this practical project. In real-world scenarios, Credit Risk Management (CRM) constitutes a fundamental process within banking institutions, aimed at evaluating, quantifying, and mitigating risks. This process significantly influences their strategic and operational approaches, including portfolio diversification and pricing strategies, optimal capital allocation, assessment of capital and liquidity adequacy, commercial lending policies, and more.

2. Research Questions

In the context of this project, all data sources are publicly available online. We utilize credit history data sourced from Alibaba Tianchi to perform an analysis of loan default prediction. Employing a data science approach, our objective is to forecast the likelihood of loan default by comprehensively evaluating applicant profiles and identifying creditworthy customers, thus mitigating potential risks for financial institutions. Put simply, we aim to determine whether to extend a loan to a customer based on their borrowing history. Addressing the challenge of imbalanced data, we employ undersampling techniques to rectify the disparity between positive and negative samples. Our methodology encompasses the utilization of three classification algorithms, namely logistic regression, KNN, and random forest, for data training, culminating in the development of a robust model for loan default prediction.

3. Literature Review

Credit risk assessment stands as a pivotal concern for financial institutions, attracting extensive scholarly exploration both domestically and internationally. Existing credit risk evaluation models primarily fall into three categories. The first category is anchored in mathematical and statistical principles, exemplified by methods such as discriminant analysis and logistic regression. For instance, Unver et al. (2018) conducted an analysis of over one hundred credit records from 2016 utilizing a logistic regression model. The second category comprises credit risk evaluation approaches grounded in machine learning models, including support vector machines and decision trees. For instance, Cao et al. (2013) proposed the utilization of a particle swarm optimization algorithm to enhance cost-sensitive support vector machines for predicting loan default risk. In essence, credit risk assessment remains a focal point of scholarly inquiry, with research methodologies predominantly categorized into mathematical and statistical techniques, traditional machine learning approaches, and deep learning methodologies.

4. Description of Data

In the realm of data analysis, pre-processing steps form the foundation for robust predictive modeling. Our initial step in the data pre-processing pipeline is to take a glimpse at the overall datasets.

The dataset comprises two distinct subsets: the training data (train.csv) with 800,000 entries and 47 features, and the testing data (testA.csv) with 200,000 entries and 46 features. Then, looking into the details of some of the descriptions of the dataframe: the average loan amount is 14416, with a standard deviation of 8716 and the mean default rate is 0.199513. Moreover, There are no duplicate rows in the datasets, however, there are 22 columns that have missing values. Missing values pose a significant challenge in data analysis, potentially leading to biased or inaccurate outcomes if not addressed properly. Thus, in order to better identify missing values across the dataset. A bar plot visualization is used here (Figure 1), offering a clear depiction of missing data proportion by feature, from which it shows that some columns have a higher proportion of missing data than others. And the (n11) column requires special attention because a significant portion of its data is missing, which may affect any analysis or models that rely on this feature.

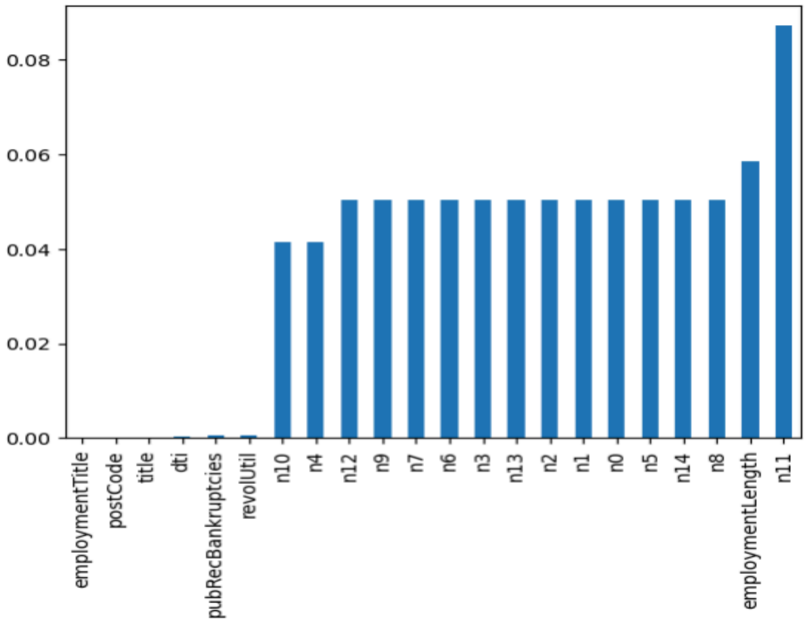


Figure 1: Visualizing the percentage of the missing values of each variable

In the next step, we plot out histograms to compare the distribution of numerical variables across the training and testing datasets (Figure 2, in Appendix). Such visual comparisons are vital for assessing the consistency between these datasets, a crucial factor in model training. If the distributions vary significantly, it might suggest sample bias or other issues that could jeopardize the model's performance on unseen data. Ensuring similar distributions helps in validating the generalizability of the predictive models developed. However, within these two data frames, most features seem to have overlapping distributions, indicating that the test set indeed captures the characteristics of the train set which is a positive sign for model generalization.

At last, we create a heatmap (Figure 3, in Appendix) of correlations to reveal which variables are strongly associated with each other as it can inform feature selection, and help detect potential multicollinearity issues that could impact the performance of certain machine learning models. From the heatmap of the training data frame, we find the n-ith variables have a high degree of correlation with each other that might need further investigation as a high correlation can suggest redundancy.

5. Data Cleaning

In our study, we analyze a comprehensive dataset comprising 1 billion observations and encompassing 47 variables about bank loan data. Initially, we conducted an examination of the variable classes, identifying that "grade" and "subGrade" were recorded as character variables rather than factors. "Grade" denotes the loan grade, categorized into seven classes denoted by letters from "A" to "G". Similarly, "subGrade" represents the loan subgrade, with each "grade" category further divided into five subgrades, resulting in a total of 35 subcategories within "subGrade". To facilitate our analysis, we employed the as.factor() function to convert these character variables into factors.

Subsequently, we addressed the variable "employmentLength", which contained descriptors such as "1 year", "2 years", "<1 year", and "> 10+ years". We sought to extract only the numerical values from these descriptors, thereby utilizing the gub("\D+", "", employmentLength) function. Here, "\D+" serves as a pattern for identifying one or more non-digit characters. By applying this function, we retained solely the numerical values, subsequently converting them into numeric variables for analysis. Furthermore, we processed the "issueDate" variable, initially recorded as a character variable. Recognizing its significance, we converted it into a date variable to facilitate temporal analysis. Also, for “earliestCreditLine”, it has values like “May-97”. The character here means month and the last two numbers mean the last two numbers of the year. If the number is larger than 24, it means it is in the 20th century. If the number is less than 24, it means it is in the 21st century. Therefore, we add “-01” to each value and then use as.Date to convert it into a date variable.

Then, an examination of missing values was conducted for each column. It was noted that the "employmentLength" column exhibited an excess of 40,000 missing values. To address this, a decision tree algorithm was employed for imputation purposes. Specifically, all non-missing values were utilized to train the model, which was subsequently leveraged to predict and fill in the missing values within the "employmentLength" column. Additionally, to handle missing values in the "dti" (debt-to-income ratio) and "revolUtil" (revolving line utilization rate) variables, a strategy involving replacement with the median was employed. This approach ensures robustness in addressing missing data while preserving the integrity of the dataset for subsequent analyses.

Ultimately, feature selection was performed to mitigate the overfitting issue and reduce computational costs. The presence of an extensive array of features in a dataset can engender the curse of dimensionality, thereby amplifying computational intricacies and prolonging model training durations. Moreover, redundant or noisy features within the dataset can exacerbate overfitting, wherein the model discerns patterns specific to the training data but falters when applied to test data. In instances where a dataset harbors an abundance of features, only a subset may substantially influence model performance. Feature selection serves to conserve computational resources and storage space, thereby augmenting model efficiency.

To execute feature selection, a random forest model was employed to compute the importance of each variable. Subsequently, a bar plot was generated to visualize variable importance, with variables arranged in descending order of importance. Notably, variables such as "pubRec", "initialListStatus", "pubRecBankruptcies", "applicationType", and "policyCode" were found to possess importance values below 1000, significantly lower than others. Consequently, these variables will be excluded from subsequent analyses.

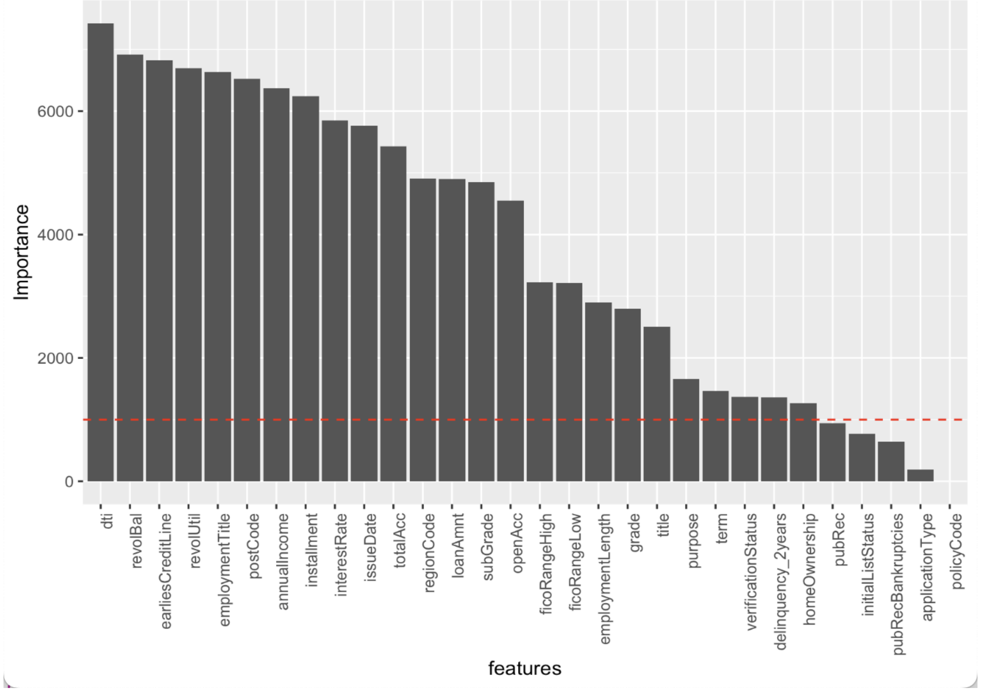


Figure 4: Visualizing the importance of each feature

6. Model Construction

6.1 Construct Random Forest Model

To fit a random forest model, first, we split 0.25 percent of the cleaned dataset randomly as the testing data, and the rest we set as my training data. Since the totally cleaned dataset contains 798823 rows, now our training data set has 599117 rows of 24 variables while the testing data set contains 199706 rows of 24 variables. However, the training dataset is not a balanced design, because within the 599117 observations, 479669 observations of our goal variable "isDefault" have the value 0 while there are only 119448 observations that have "isDefault" value 1. The ratio of 0 and 1 is about 0.8 to 0.2 within the training set, rather than being an equally designed dataset of 0.5 to 0.5, thus, directly using this data frame could affect the performance of the random forest model, potentially leading it to be biased towards predicting the majority class which is no default.

To solve this problem, we use the undersampling method. We randomly selected 159,375 instances of class '0' to match the 159,375 instances of class '1' and then "rbind" the two data frames together to form a new training set.

We first chose the random forest model because we did not know where the impact of the predictors on the dependent variable was linear. Also, random forest models can provide rankings of feature importance, helping to understand which features have the most influence on the target variable's prediction. Moreover, to mitigate the randomness in the sampling process and provide a more generalized understanding of our random forest model's performance, we sampled three times to create three balanced datasets and used them to fit three random forest models. The reason why we sampled three times instead of five times as in the other models is that it takes a long time to run the model.

	Sample 1	Sample 2	Sample 3
Accuracy	64.23%	64.21%	64.14%

Precision	31.56%	31.50%	31.46%
Recall	67.54%	67.26%	67.34%
F1 score	0.43	0.43	0.43
AUC	0.71	0.71	0.71

Table 1: Accuracy, Precision, Recall, F1 score, AUC of three randomly sampled random forest models

Table 1 shows the overall performance of three random forest models, from which we can tell that all models have similar accuracy levels around 64%, suggesting that the random forest models are moderately effective at identifying the correct labels but still have space for improvement. Meanwhile, the precision values are around 31.5% across all models, indicating that a significant proportion of the positive predictions made by the models are false positives, which is not ideal, especially in a domain like lending where the cost of false positives can be high. The recall values are reasonably high, around 67%, telling that the models are quite good at identifying most of the actual defaults. This is useful in risk aversion scenarios where it is critical to capture as many defaults as possible, applying to our background condition that the bank needs to choose their borrowers cautiously as there is some probability that they may not pay it back.

The AUC values of all of the three random forest models are all above 0.70, indicating a good level of predictive accuracy for each model and suggesting that the models have a reasonable ability to discriminate between the classes (default vs. non-default). Moreover, the close range of these values (0.709 to 0.711) also indicates that the model performance is stable and reliable across different iterations, which ensures the generalizability of the models.

The Precision-at-k% plot for the random forest model in Figure 5 (in Appendix) shows a very high precision at the start level, approximately around 0.6, making the model relatively effective for identifying the most probable defaults among the top few cases, which is advantageous in scenarios where only a limited number of cases can be managed due to resource constraints. However, the precision shows a steep decrease from about 0.6 to roughly 0.4 as K increases to around 5,0000 to 10,0000 cases. This sharp drop implies that as we expand the number of top cases considered, the accuracy of the model in identifying true defaults rapidly diminishes. Thus, when we use this random forest model to predict a new dataset that is roughly the same size as our original dataset which is around 8,0000 observations, it will have limited precision.

6.2 Construct Logistic Regression Model

Then, we tried another classification method to see whether logistic regression would have a better performance in the default prediction scenario. For the logistic regression model, first, we also dealt with the imbalance data structure problem by repeating the same step shown in the previous model. Then, we run the logistic regression model by using 5 different sampled train datasets. After that, we use the test datasets to check the accuracy and precision of the model to analyze the model's performance. On average the accuracy is 65.30%, and the precision is 64.59%, and the AUC values are around 0.71. Therefore, we can see that the logistic regression model is in general better than the random forest model because the accuracy is two times higher.

	Sample 1	Sample 2	Sample 3	Sample 4	Sample 5
Precision	64.20%	64.50%	64.47%	64.97%	64.82%
Accuracy	65.28%	65.30%	65.17%	65.44%	65.35%
AUC	0.71	0.71	0.71	0.71	0.71

Table 2: Precision, Accuracy, and AUC values for five randomly sampled logistic regression models

In the end, we also constructed a precision-at-k% plot (Figure 6, in Appendix). Similar to random forest, the plot shows a very high precision at the start level, approximately around 0.82, making the model highly effective for identifying the most probable defaults among the top few cases, which is advantageous in scenarios where only a limited number of cases can be managed due to resource constraints. However, the precision decreases generally as K increases.

6.3 Construct KNN model

Unlike parametric models such as logistic regression, KNN does not make assumptions about the underlying distribution of the data, making it particularly useful for nonlinear relationships or when the data distribution is unknown. Similar to previous models, we randomly sampled non-default observations five times and combined the selected sample with default observations at the ratio of 1:1, thus balancing the proportion of negative and positive samples. We input the undersampling training data and test data into the model to start the model training. As undersampling data are 5-fold, we implemented the model training and testing part for five datasets orderly. Combined with all the prediction results derived from five data sets, the loan default prediction has the AUC around 0.582.

	Sample 1	Sample 2	Sample 3	Sample 4	Sample 5
Recall	54.1%	53.0%	52.8%	55.2%	53.9%
F1-score	0.318	0.309	0.320	0.301	0.316
AUC	0.58	0.58	0.60	0.56	0.59

Table 3: Recall, F1-score, and AUC for the KNN model

	Predicted value = 0	Predicted value = 1
Actual Value = 0	85474	74290
Actual Value = 1	18372	21568

Table 4: Confusion matrix for the KNN model

In our project, particular attention needs to be paid to the true negative rate. In real-world scenarios, misclassifying customers who actually default as non-default can have detrimental financial consequences for banks. Therefore, minimizing false negatives is crucial to avoid such risks. According to the table, the true negative (TN) value is approximately 0.428, indicating a relatively higher rate. This suggests that the KNN model we applied may not be suitable for real-world applications and could result in a relatively high error rate in decision-making processes.

Predicted probabilities have a balanced distribution across classes, resulting in a consistent precision regardless of the number of neighbors considered. When classes are evenly distributed, the relative proportions of neighbors from each class remain constant as k increases, leading to stable precision values.

7. Evaluate and Interpret Models

7.1 Compare Random Forest, Logistic Regression, and KNN

In evaluating the performance of logistic regression, KNN, and random forest models applied to a dataset which aims to predict the default rate, it's evident that each model has its strengths and is suited to different operational needs. The logistic regression model stands out for its high precision of 0.6459 and accuracy of 0.6531, making it highly effective at correctly identifying defaults among its positive predictions. Particularly notable is its Precision at K curve, which starts near 0.8 for the top cases, highlighting its ability to accurately target the highest-risk individuals. This feature diminishes as k increases but maintains a practical level even at broader thresholds.

In comparison, the random forest models demonstrated superior AUC values around 0.71, suggesting better overall ranking capabilities across all thresholds but with lower precision in immediate top cases. This makes them suitable for applications requiring consistent performance across a wide range of scenarios. On the other hand, the KNN models showed lower performance in terms of both AUC and precision, making them less ideal unless further optimized.

Considering these insights, the logistic regression model is recommended for scenarios where high accuracy in top-ranked predictions is crucial, due to its superior initial precision and ease of implementation.

7.2 Interpret the Logistic Regression Model

In the evaluation of the coefficients from the logistic regression model used to predict loan defaults, Figure 8 (in Appendix) shows that certain variables significantly influence the likelihood of a borrower defaulting. Among these, the loan grade stands out as the most critical factor, with higher grades (e.g., grades F and G) having considerably higher coefficients. This suggests that loans classified under these grades are much more likely to default. Specifically, the coefficients for grade F and grade G are 2.864 and 3.159, respectively, indicating a strong positive relationship with the probability of default. This trend across the grades suggests that as the perceived risk inherent in the grade increases, so does the probability of default, highlighting the grading system's effectiveness in risk stratification.

Another significant variable impacting default likelihood is the loan term, denoted by a positive coefficient of 0.3206. This suggests that loans with longer terms are associated with a higher default probability, potentially due to the increased uncertainty and financial strain over longer periods. Conversely, the interest rate, which one might intuitively expect to be positively correlated with default risk, actually shows a negative coefficient of -0.03198. This counterintuitive finding could imply that higher interest rates might be associated with loans that have undergone more rigorous screening processes, or that higher rates discourage riskier borrowing behavior.

The coefficient for the installment amount is positive, albeit small (0.0009575), indicating that as the installment amount increases, so does the likelihood of default, albeit slightly. This could reflect the increased burden on the borrower as the payment amount rises, potentially impacting their ability to consistently meet payment obligations.

Interestingly, the loan amount itself shows a negative coefficient (-0.00002013), suggesting that higher loan amounts slightly decrease the likelihood of default. This might be reflective of a financial institution’s lending strategy, where larger loans are extended to more creditworthy individuals or under terms that are more manageable relative to the borrower's financial situation.

Additionally, certain subgrades such as subGradeB1 and subGradeB2 have negative coefficients, indicating a lower likelihood of default compared to the baseline subgrade. This differentiation within subgrades further emphasizes the nuanced approach of grading systems in assessing risk.

In conclusion, the logistic regression analysis highlights that loan grading, term, and installment amounts are pivotal in predicting loan defaults, with the grading system playing a particularly dominant role. Understanding these relationships helps in tailoring risk management strategies, ensuring that lending practices are both prudent and aligned with the borrower's ability to fulfill their obligations. Such insights are invaluable for financial institutions looking to optimize their loan portfolios, mitigate risks, and enhance decision-making processes in lending.

8. Implications and Limitation

In terms of the implication and limitation, this paper will illustrate this part in two aspects. The first aspect is model selection and the second part is sampling method.

The random forest model is known for its robust performance with large datasets, demonstrating efficient training capabilities. Despite encountering imbalanced target variables, such as our 'isDefault' variable, the random forest model maintains effectiveness due to its utilization of bagging (bootstrap aggregating). This technique involves training multiple decision trees on various bootstrap samples of the dataset, thereby mitigating model variance and averting overfitting. Such characteristics are particularly advantageous when handling imbalanced data distributions. In comparison to the k-nearest neighbors (KNN) and logistic regression models, the random forest model offers insights into feature importance, elucidating the factors that exert the most significant influence on predicting loan default probability. This feature proves invaluable for comprehending the underlying drivers of default risk. Moreover, in real-world scenarios, where customer loan history features may number in the thousands, selecting an efficient feature selection method becomes significant.

While random forest models offer good predictive performance, they are often considered as "black box" models, meaning that it can be challenging to interpret how exactly the model arrives at its predictions. This lack of interpretability might be a concern in scenarios where understanding the reasons behind a prediction is important, such as regulatory compliance or explaining decisions to stakeholders. That’s why the model prediction results can only partially help us to make decisions about whether to give a loan to a customer. For example, it is very common in commercial banks to have a recorded credit evaluation score, which is very robust to assess the goodness of credit for customers. In the United States, a very popular way to assess credit is the FICO score (Ranging from 300~850). The five most important aspects are payment history, credit limit, credit utilization, hard pull, and soft pull (The last 2 terms are two types of credit reports that can be retrieved by banks).

Considering the undersampling method we used to deal with the imbalance of the dataset, there also exist some trade-offs. Considering the business reality of the financial industry, expanding negative samples when dealing with unbalanced dataset problems is difficult to achieve in real life because the information of default users is very precise, and arbitrarily expanding default users will make the results lack interpretation. Due to this reason, it is generally recommended to use the undersampling method. While undersampling seems to perfectly solve the imbalance issue, when we only extract about 20% of the training data in order to keep "isDefault" balanced, a large proportion of information was not taken into account. This may lead us to misunderstand the driving factor in data. In the fintech industry, besides sampling methods, cost-sensitive learning also works well with imbalanced classification by considering the cost of misclassification.

Reference

Maiorca, D., Mercaldo, F., Giacinto, G., Visaggio, C. A., and Martinelli, F. (2018). R-packdroid: Api package based characterization and detection of mobile ransomware.*In Proceedings of the Symposium on Applied Computing*, pages 1718–1723. ACM.

Yang L, Wang K-L, Wang J-M. (2008) A comparative study of major modeling approaches for credit scoring [J]. *Economic Management*(6).

Martinelli, F., Marulli, F., and Mercaldo, F. (2013). Evaluating convolutional neural networks for effective mobile malware detection. Procedia Computer Science,*Procedia Computer Science*, 112:2372–2381.

Chawla, N.V.(2005)Data mining for unbalanced data sets: An overview.*In Data mining and knowledge discovery handbook*,Springer:853-867.

Appendix

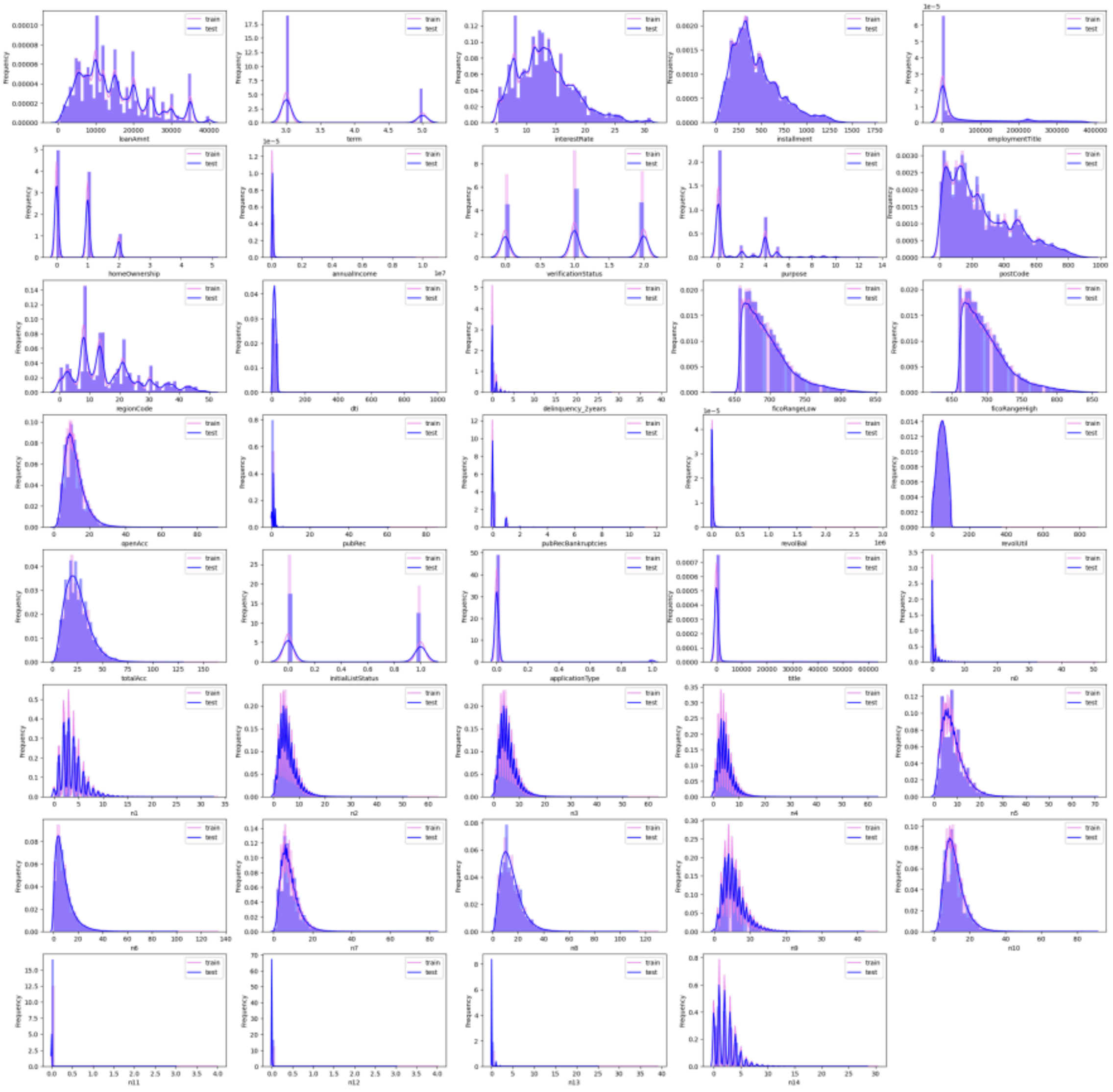


Figure 2: Comparing distribution of train data and test data

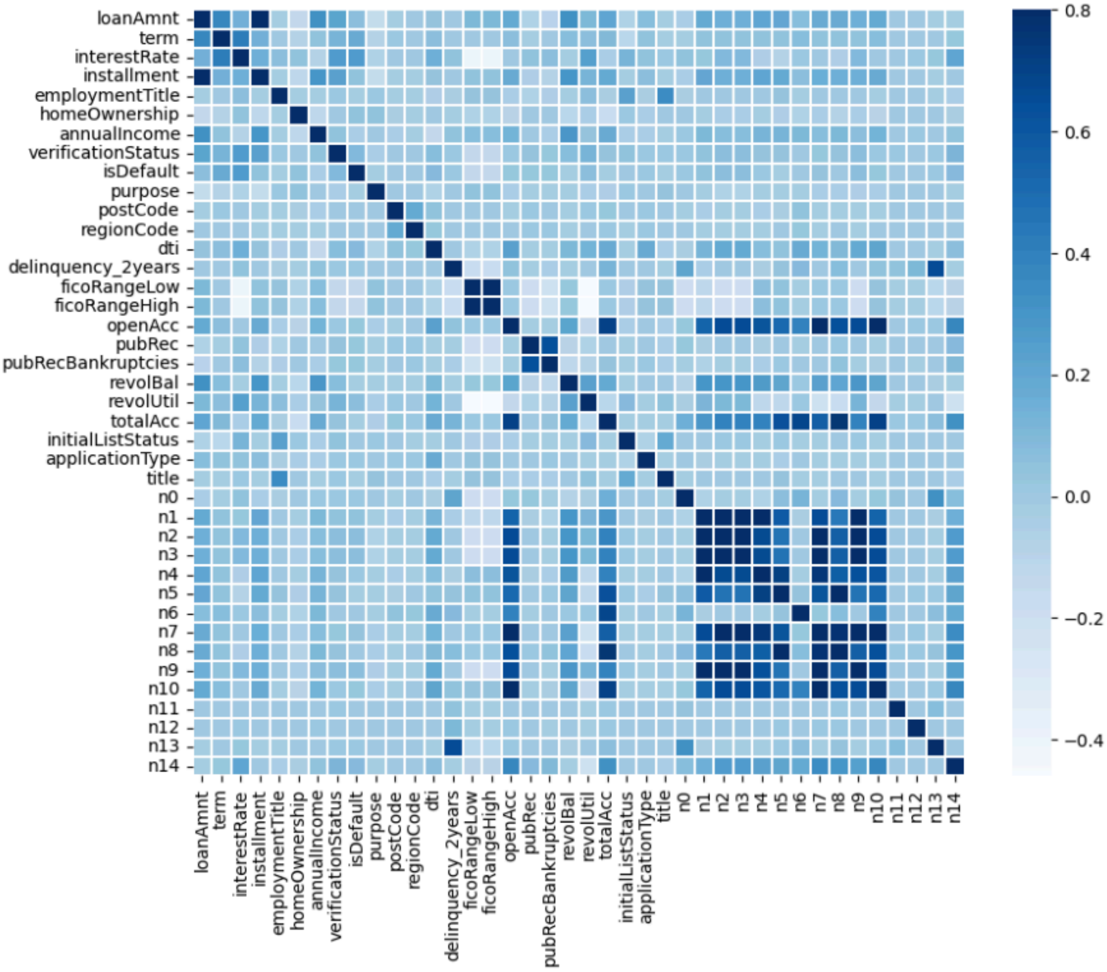


Figure 3: Correlation between two variables

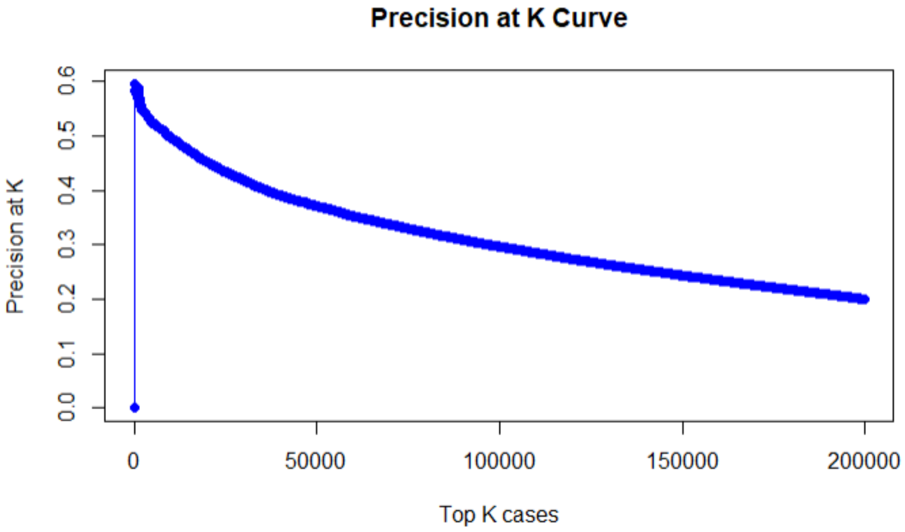


Figure 5: Precision-at-k% plot for the random forest model

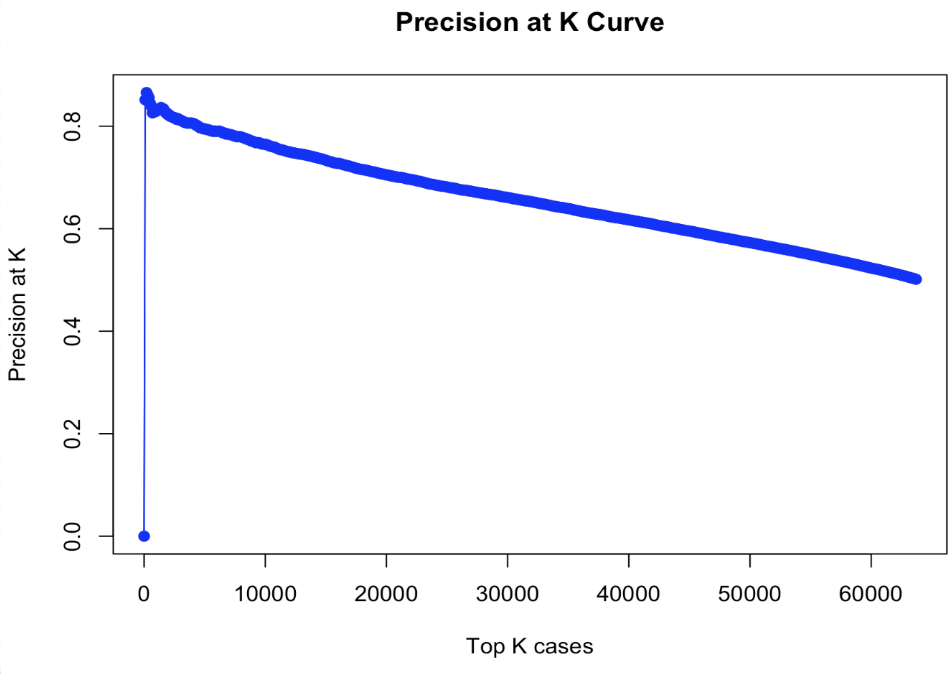


Figure 6: Precision-ar-k% plot for the logistic regression model

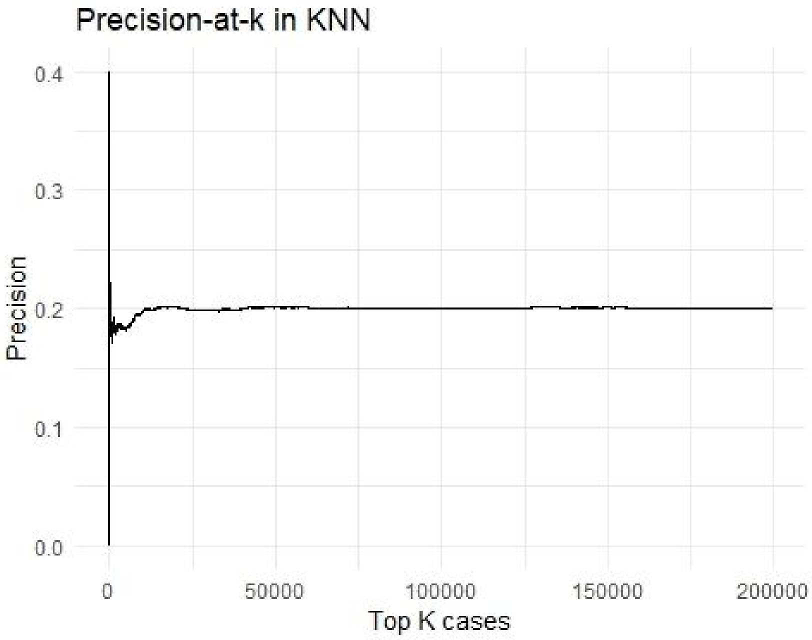
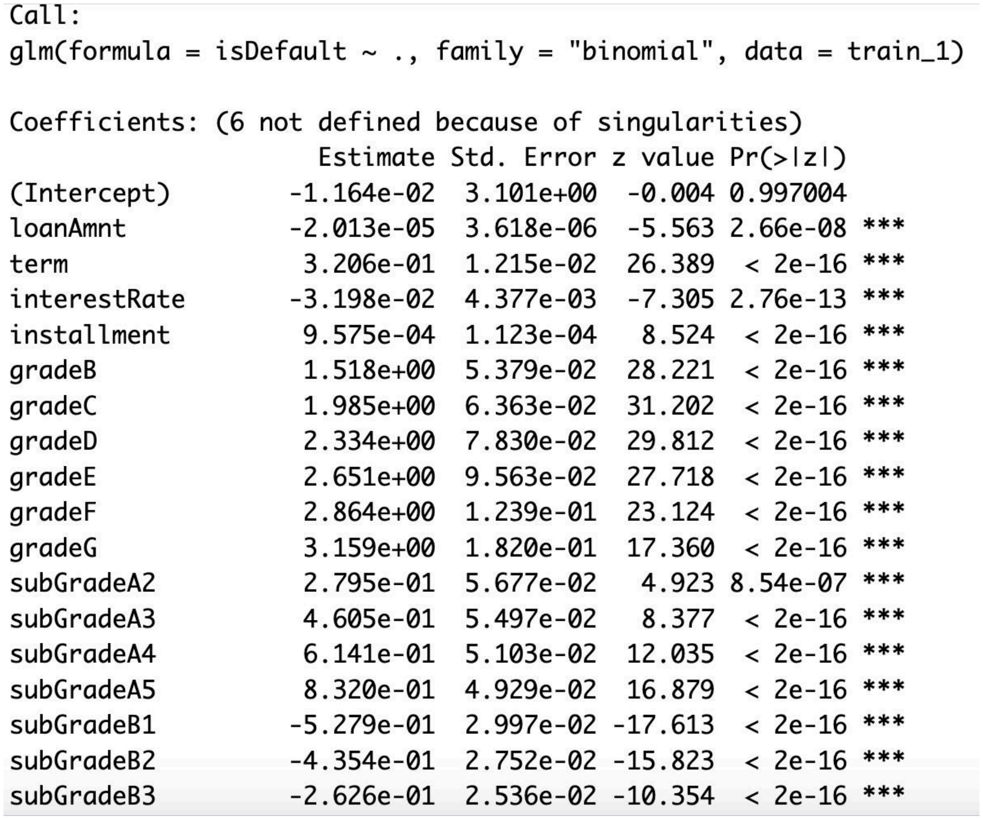


Figure 7: Precision-at-k% plot for the KNN model



```
Call:
glm(formula = isDefault ~ ., family = "binomial", data = train_1)

Coefficients: (6 not defined because of singularities)
                    Estimate Std. Error z value Pr(>|z|)
(Intercept)       -1.164e-02  3.101e+00  -0.004 0.997004
loanAmnt          -2.013e-05  3.618e-06  -5.563 2.66e-08 ***
term               3.206e-01  1.215e-02  26.389  < 2e-16 ***
interestRate      -3.198e-02  4.377e-03  -7.305 2.76e-13 ***
installment        9.575e-04  1.123e-04   8.524  < 2e-16 ***
gradeB             1.518e+00  5.379e-02  28.221  < 2e-16 ***
gradeC             1.985e+00  6.363e-02  31.202  < 2e-16 ***
gradeD             2.334e+00  7.830e-02  29.812  < 2e-16 ***
gradeE             2.651e+00  9.563e-02  27.718  < 2e-16 ***
gradeF             2.864e+00  1.239e-01  23.124  < 2e-16 ***
gradeG             3.159e+00  1.820e-01  17.360  < 2e-16 ***
subGradeA2         2.795e-01  5.677e-02   4.923 8.54e-07 ***
subGradeA3         4.605e-01  5.497e-02   8.377  < 2e-16 ***
subGradeA4         6.141e-01  5.103e-02  12.035  < 2e-16 ***
subGradeA5         8.320e-01  4.929e-02  16.879  < 2e-16 ***
subGradeB1        -5.279e-01  2.997e-02 -17.613  < 2e-16 ***
subGradeB2        -4.354e-01  2.752e-02 -15.823  < 2e-16 ***
subGradeB3        -2.626e-01  2.536e-02 -10.354  < 2e-16 ***
```

Figure 8: Summary of Logistic Regression Model