Predicting Loan Default for Czech Bank

Table of contents:

Introduction

Loan lending plays an important role in our everyday life and powerfully promotes the growth of consumption and the economy. Taking a loan has been inevitable for people since individuals around the world depend on loans to overcome financial constraints to achieve their personal goals, and organizations rely on loans to expand their production. In most cases, loan lending is beneficial to both the borrowers and the lenders. However, loan default is still unavoidable, which carries a great risk and may even end up in a financial crisis. Therefore, it is particularly important for a bank to identify whether a candidate is eligible to receive a loan.

In the past, the evaluation primarily depended on manual review, which was time-consuming and labor-intensive. Recently, banks have opted for machine learning approaches to automatically predict loan defaults based on certain features since it can highly enhance the accuracy and the efficiency of the prediction. On the one hand, banks can collect a massive amount of transaction data due to the prosperity of online shopping and mobile payments. On the other hand, machine learning models are rapidly evolving and have successful applications in various fields, motivating the bank industry to use them to predict loan default.

Dataset Description

For my analysis, the dataset is "1999 Czech Financial Dataset - Real Anonymized Transactions" which has been obtained from data.world. It contains real anonymized Czech Bank transactions, account information, and loan records released for PKDD'99 Discovery Challenge.

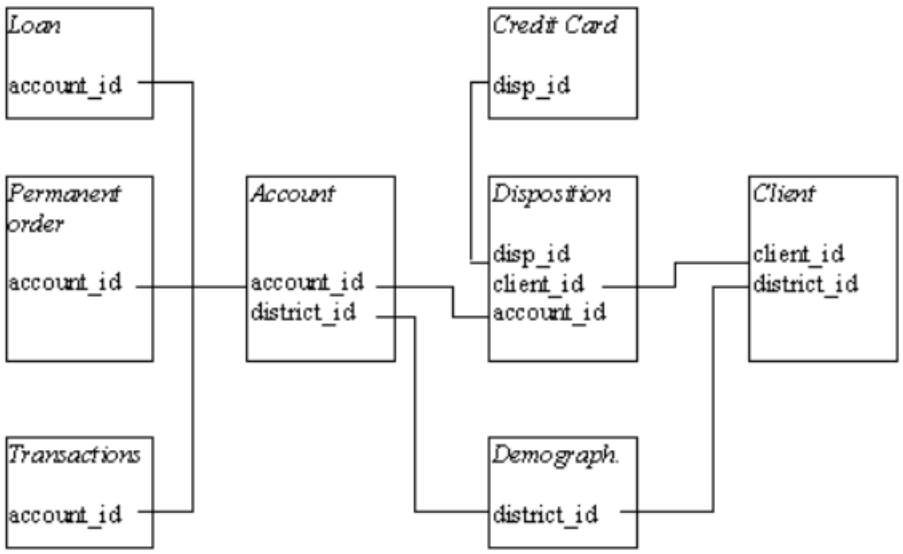
The data about the clients and their accounts consist of following relations:

- relation account (4500 objects in the file ACCOUNT.ASC) - each record describes static characteristics of an account,
- relation client (5369 objects in the file CLIENT.ASC) - each record describes characteristics of a client,

- relation disposition (5369 objects in the file DISP.ASC) - each record relates together a client with an account i.e. this relation describes the rights of clients to operate accounts,
- relation permanent order (6471 objects in the file ORDER.ASC) - each record describes characteristics of a payment order,
- relation transaction (1056320 objects in the file TRANS.ASC) - each record describes one transaction on an account,
- relation loan (682 objects in the file LOAN.ASC) - each record describes a loan granted for a given account,
- relation credit card (892 objects in the file CARD.ASC) - each record describes a credit card issued to an account,
- relation demographic data (77 objects in the file DISTRICT.ASC) - each record describes demographic characteristics of a district.

Each account has both static characteristics (e.g., date of creation, address of the branch) given in relation "account" and dynamic characteristics (e.g., payments debited or credited, balances) given in relations "permanent order" and "transaction". Relation "client" describes characteristics of persons who can manipulate with the accounts. One client can have more accounts, more clients can manipulate with single account; clients and accounts are related together in relation "disposition". Relations "loan" and "credit card" describe some services which the bank offers to its clients; more credit cards can be issued to an account, at most one loan can be granted for an account. Relation "demographic data" gives some publicly available information about the districts (e.g., the unemployment rate); additional information about the clients can be deduced from this.

The relation between the 8 tables is given as:



Methodology

1. Data Preprocessing

Loading libraries and Dataset: All the required libraries like tidyverse, caret, randomForest, rpart, glmnet, corrplot, readxl, tidyr, dplyr are loaded to leverage their functions for data manipulation, visualization, and machine learning. Data is loaded from individual excel files using `read_excel()` , creating separate dataframes for each table.

Datasets Merging: `left_join()` is used to merge files based on columns shared amongst the files. As combining all the 8 files increased the processing time, I selected only 5 files for analysis: loan, order, account, trans and district.

Column Renaming: The district dataframe is modified to rename the 'A1' column to 'district_id' for the purpose of `join()` operation.

Handling missing values: Columns with more than 50% missing values were identified and removed from the dataframe.

Handling missing values and duplicates: Duplicate rows are removed and any remaining rows with missing values are omitted to create a clean dataset.

Convert target variable to binary: The target variable ‘status’ is converted to binary format, where 1 represents defaulters (‘B’ and ‘D’) and 0 represents non – defaulters (‘A’ and ‘C’).

Convert categorical columns to numeric: One of the main reasons for converting categorical columns to numeric is so that they can be included in the correlation matrix.

Normalization: All the column values were normalized/standardized using `scale()` as few columns had only single digit values while others had more than 5 digit values.

Under – sampling: It is a crucial step when dealing with imbalanced datasets. In this dataset the number of non – defaulters are 275989 and defaulters are 26262. Under – sampling involves randomly removing instances from the majority class to create a more balanced distribution. This process ensures that the machine learning models are not biased towards predicting the majority class and can better identify patterns associated with the minority class. `ovun.sample` function from ‘ROSE’ library has been used here.

2. Data Splitting

This is a crucial part in machine learning workflow where its purpose is to divide the dataset into 2 subsets: one for training the model and other for evaluating its performance. In this project we created a 80-20 split that is 80% training data and 20% testing data.

3. Model Development and Evaluation

For my analysis I considered 3 cases for model development and evaluation for comparison of their performance.

Full Model

In the first case, I trained 2 models’ that are logistic regression and random forest using all the available features in the merged_data dataset. Then predict probabilities on test data and convert those probabilities into binary outcomes. Model performance is evaluated based on the confusion matrix generated using `confusionMatrix()` .

Reduced Model using Correlation Matrix

In the second case, first developed correlation matrix which showed how each variable is related to each other variable in the range -1 to 1. Negative values indicate they are negatively correlated, that is when one variable increases the other variable decreases. A positive value indicates they are positively correlated, that is when one variable increases another variable also increases. Here correlation matrix was used to pick relevant features that had good correlation with our target variable `status` which represented defaulters and non – defaulters. Same procedure as the first case is used for training, predicting probabilities and evaluating the model performance using `confusionMatrix()`.

LASSO – Reduced Model

In the third case, LASSO (L1 regularization) is applied to automatically select relevant features during the training. LASSO is an innovative variable selection method for regression which adds a penalty to the sum of coefficients in a regression while minimizing the residual sum of square (RSS). This allows LASSO to perform coefficient shrinkage and thus variable selection. Lambda is the tuning parameter in LASSO setting the shrinkage level of the coefficient. The higher the lambda, the higher the shrinkage level and a greater number of coefficients are shrunk to 0 and as a result fewer independent variables resulted from the regression process. LASSO is trained using the training data, and the optimal lambda (regularization parameter) is determined. R provides a function called `cv.glmnet()` to achieve this. The trained LASSO model makes predictions on the test set, and its performance is evaluated using a confusion matrix. Additionally, selected features and LASSO coefficients are visualized.

For all three cases including logistic regression and random forest I built a cost – based matrix to help business understand the dollar value. It is a variation of the traditional confusion matrix that incorporates the concept of costs associated with different types of classification errors. In real –

world, the consequences or costs associated with different types of errors are not the same. In case of predicting the loan defaulters,

TN (True Negatives): instances correctly predicted as non – defaulters.

TP (True Positives): instances correctly predicted as defaulters.

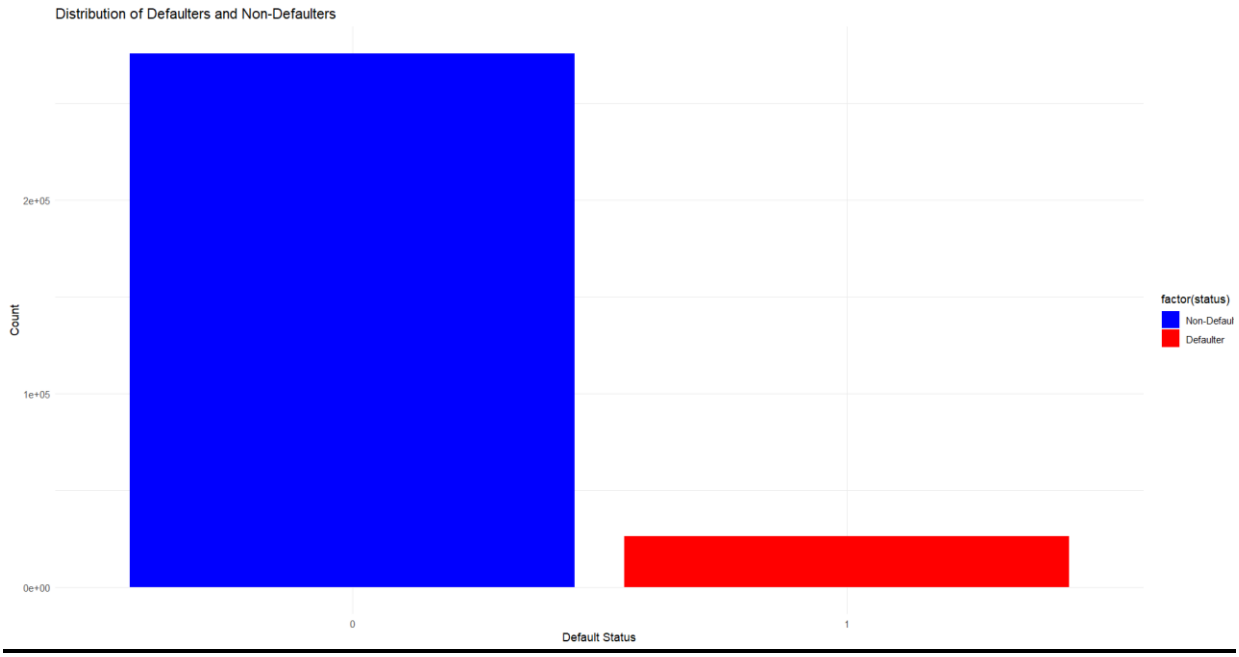
FP (False Positives): instances incorrectly predicted as defaulters.

FN (False Negatives): instances incorrectly predicted as non – defaulters.

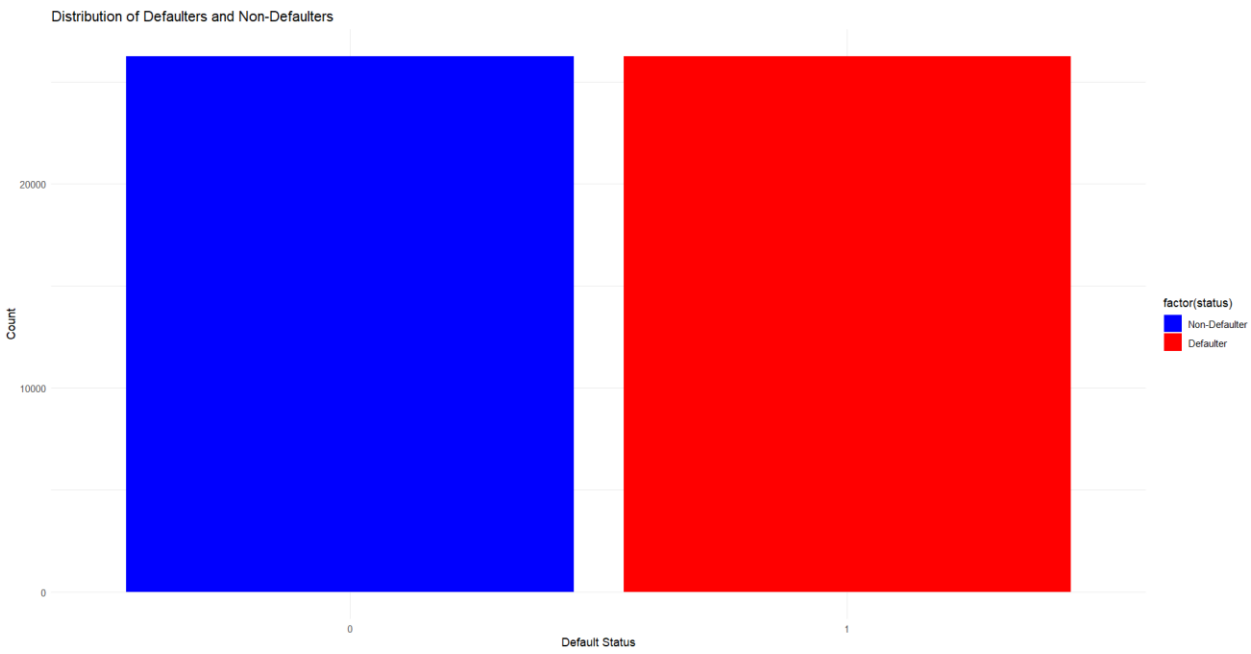
Here predicting the defaulters as non – defaulters is far too risky/costly than predicting non – defaulters as defaulters. So cost of FN > cost of FP.

Results

1. Distribution of defaulters and non – defaulters



2. Distribution of defaulters and non – defaulters after under – sampling



3. Confusion matrix and accuracy for logistic regression and random forest in case 1 where all available features are used for training.

logistic regression

```
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 2242 1871
         1 3010 3381

               Accuracy : 0.5353
```

Random forest

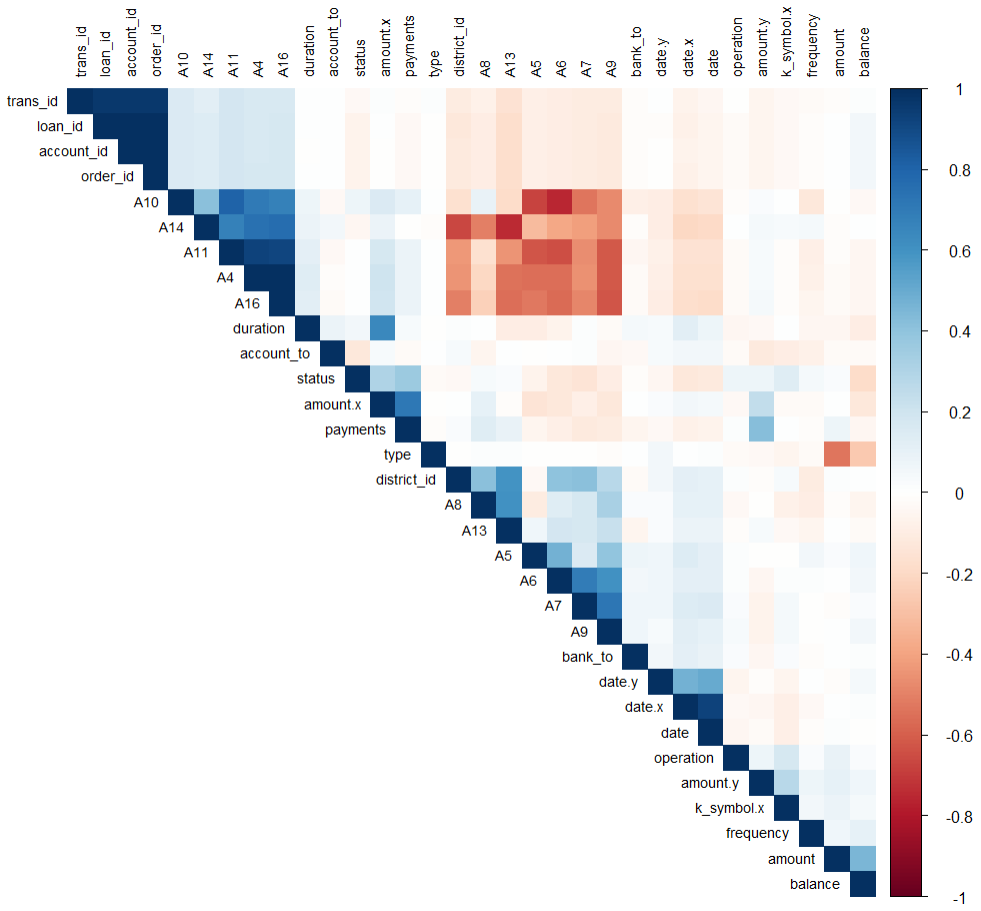
```
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 5252    0
         1    0 5252

               Accuracy : 1
```

Logistic regression gives an accuracy of 53.53% and Random Forest gives an accuracy of 100%.

4. Confusion matrix and accuracy for logistic regression and random forest in case 2 where model is trained based on features picked based on correlation matrix.



Here, in the correlation matrix we look for all other features that are related to the target variable ‘status’ and use them in training our models.

Logistic regression

```
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 3697 1820
         1 1555 3432

               Accuracy : 0.6787
```

Random Forest

```
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 5240    0
         1   12 5252

               Accuracy : 0.9989
```

From the results we can observe that the accuracy of logistic regression has improved from 53% to 67% with no change in random forest accuracy maintaining 100%.

5. Confusion matrix and accuracy for logistic regression in case 3 based on LASSO reduced regression.

```
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 3841 1201
         1 1411 4051

               Accuracy : 0.7513
```

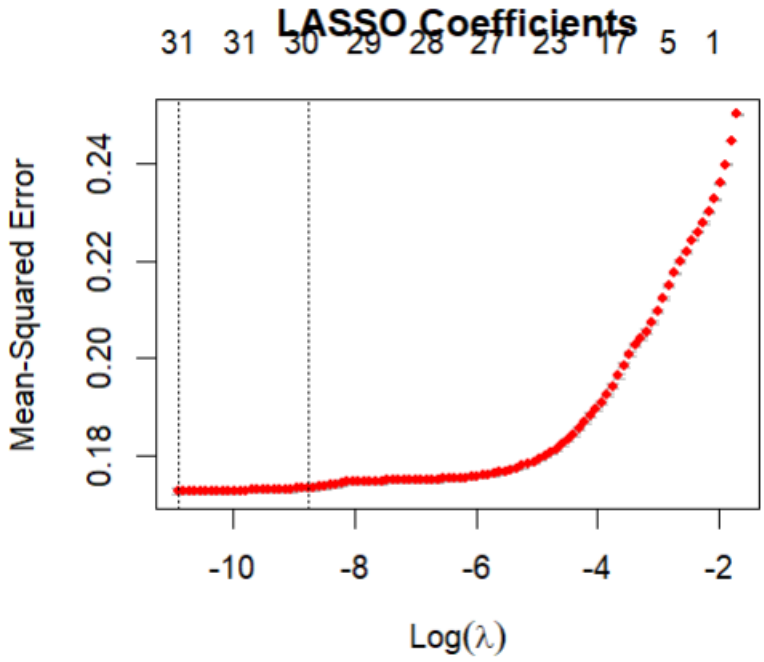
This case is an improvement over case 1(full – model) and case 2(correlation matrix based reduction).

The features that were selected to achieve this accuracy are;

```
Selected Features: account_id date.x amount.x duration payments status order_id bank_to account_to amount.y k_symbol.x trans_id dat
e.y type operation amount balance district_id frequency date A2 A5 A6 A7 A8 A9 A10 A11 A12 A14 A15 NA
```

Out of a total of 36 variables, 31 variables were selected by LASSO regression.

6. LASSO coefficients plot



This plot visualizes the coefficients of the LASSO model across different values of lambda.

Vertical lines: Represent the optimal lambda which is often chosen based on cross – validation.

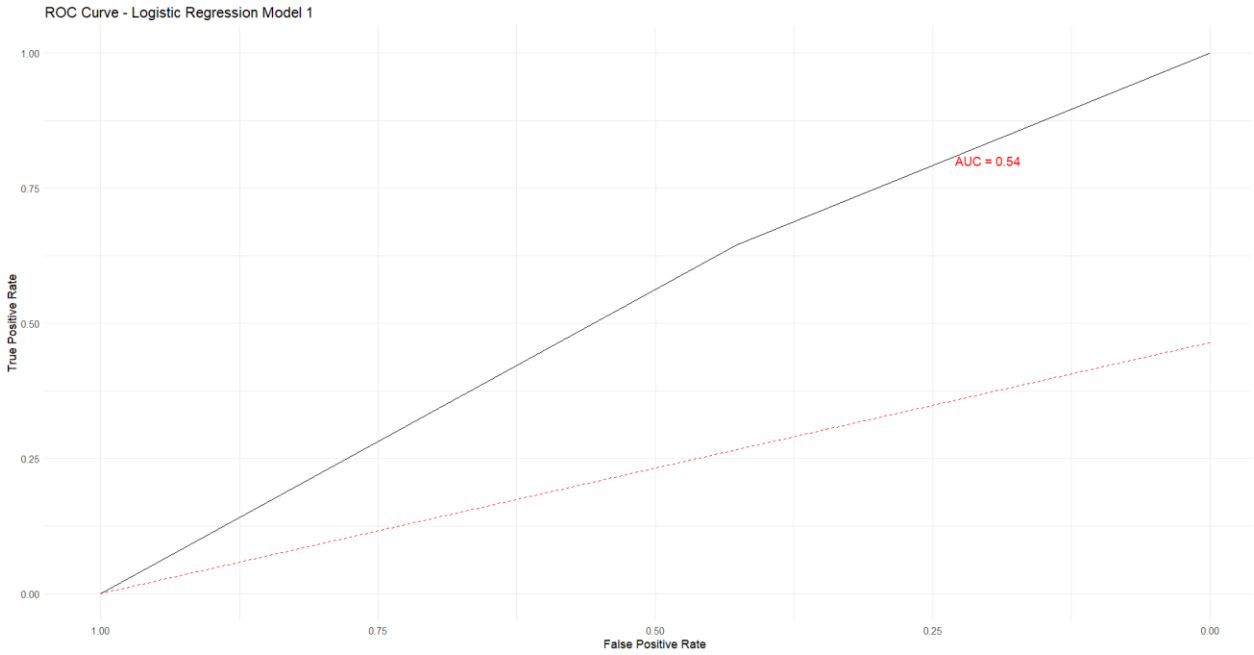
Direction of Shrinkage: The direction in which the coefficients move along the plot depends on the regularization strength. Stronger regularization forces more coefficients to zero.

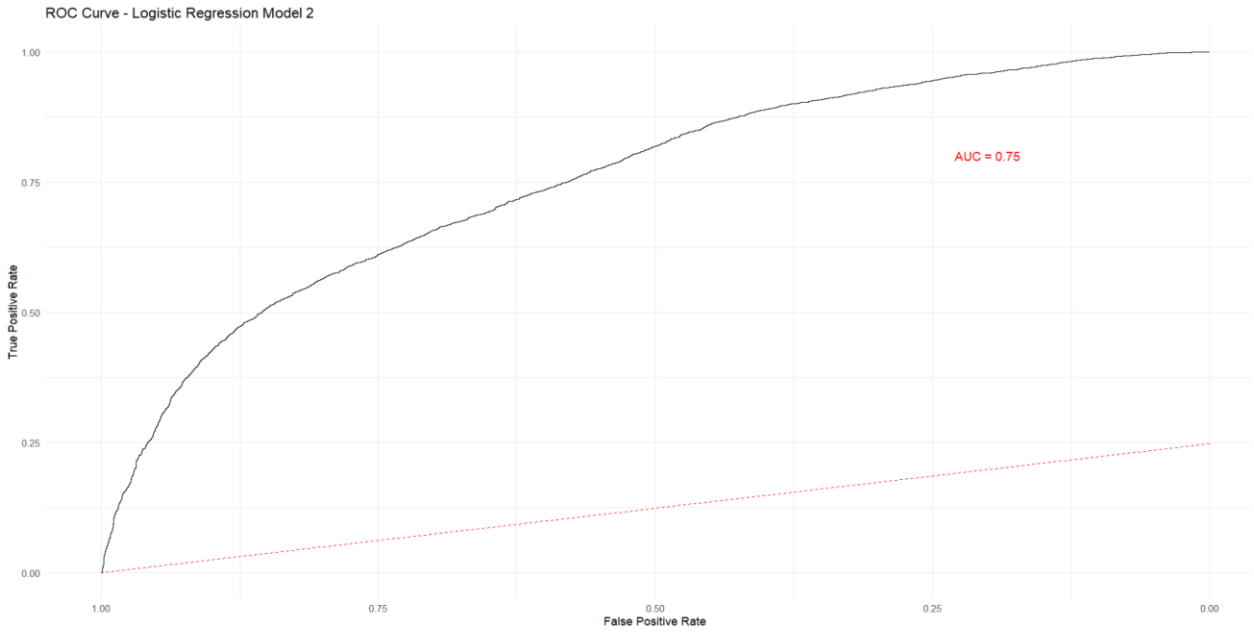
Variable Selection: Features with non – zero coefficients for a given lambda are selected by the model, and those with zero coefficients are excluded.

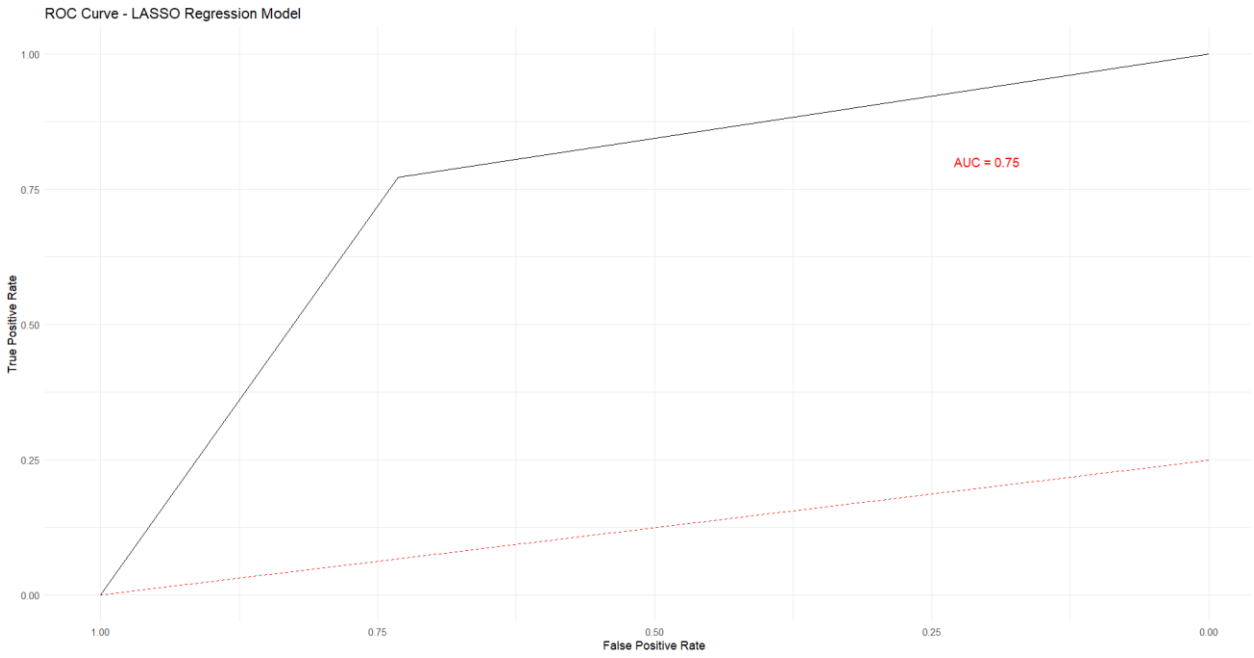
7. ROC Curves

Logistic Regression curves show an improvement from case 1 to case 2 to case 3 as shown below:

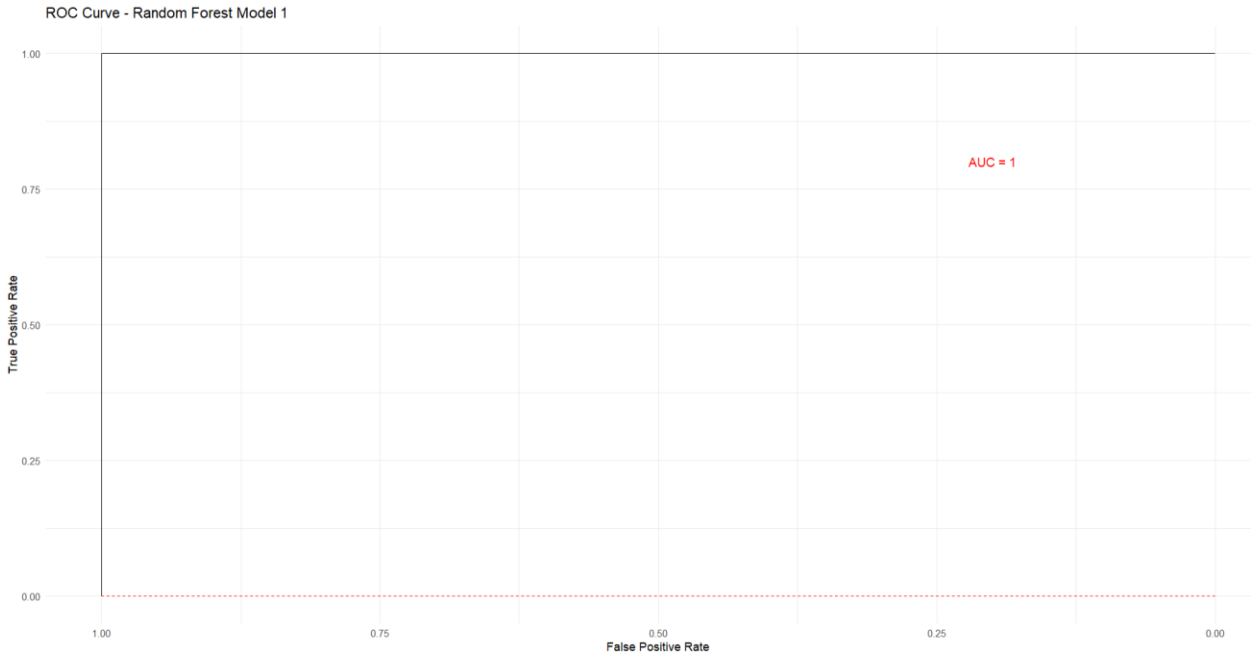
Logistic Regression

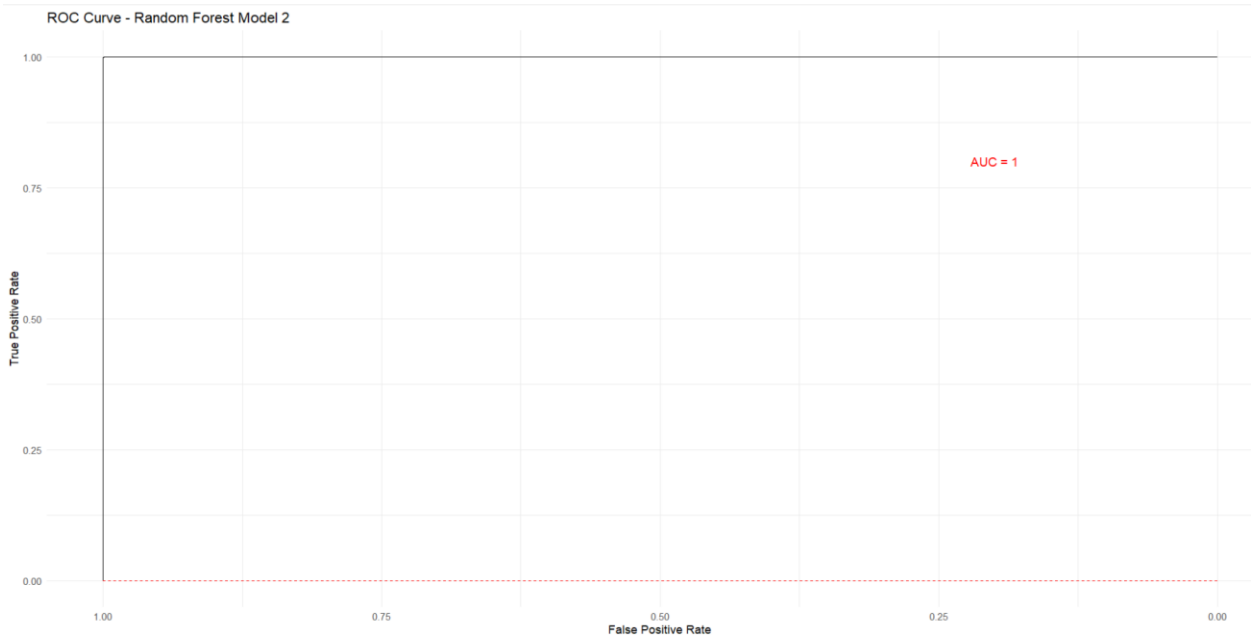


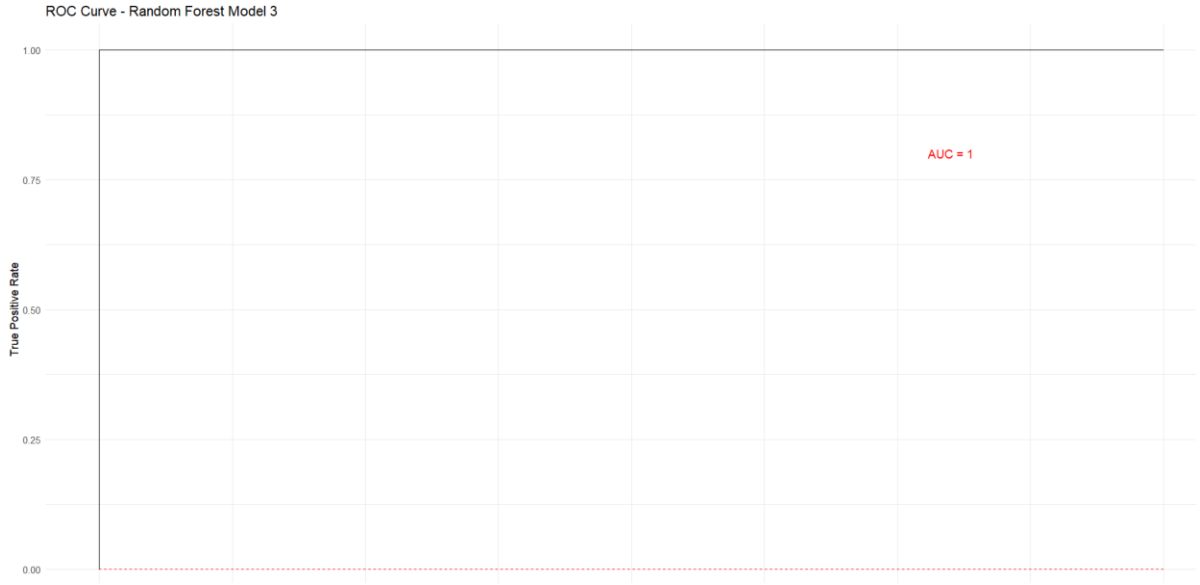




Random Forest performance has remained same over both cases.







8. Cost Based Matrix

Logistic Regression model 1

```
True Positives: 3381
True Negatives: 2242
False Positives: 1871
False Negatives: 3010
Total Cost: 39455
```

Logistic Regression model 2

```
True Positives: 3432
True Negatives: 3697
False Positives: 1820
False Negatives: 1555
Total Cost: 24650
```

Logistic Regression model 3

```
True Positives: 4051
True Negatives: 3841
False Positives: 1201
False Negatives: 1411
Total Cost: 20115
```

Among all three models, logistic regression model 3 is cost efficient.

Random Forest model 1

```
True Positives: 5252
True Negatives: 5252
False Positives: 0
False Negatives: 0
Total Cost: 0
```

Random Forest model 2

```
True Positives: 5252
True Negatives: 5240
False Positives: 0
False Negatives: 12
Total Cost: 120
```

Random Forest model 3

```
True Positives: 5252
True Negatives: 5252
False Positives: 0
False Negatives: 0
Total Cost: 0
```

Among the 3 models, random forest models 1 and 3 are cost efficient.

If given a choice between logistic regression and random forest model, random forest is the preferred model for this analysis.

Conclusion

In conclusion, this analysis provides a robust examination of loan default prediction using machine learning models. By navigating through data preprocessing, model development, training, and evaluation, we gained insights into factors influencing loan default and the capabilities of different algorithms. The presented results serve as a foundation for future improvements and refinements in predictive modeling for financial institutions.

From the obtained results we can conclude that Random Forest is the best model for predicting the loan default. But there is also a chance of improvement for logistic regression.