

MICRO LOAN FINANCE

Submitted by:

MUKUL DHAMIJA

**ACKNOWLEDGMENT**

This includes mentioning of all the references, research papers, data sources, professionals and other resources that helped you and guided you in completion of the project.

· [1. Introduction](https://towardsdatascience.com/loan-default-prediction-for-profit-maximization-45fcd461582b#7e86)  
· [2. Data Cleaning and Exploratory Analysis](https://towardsdatascience.com/loan-default-prediction-for-profit-maximization-45fcd461582b#30be)  
· [3. Modeling](https://towardsdatascience.com/loan-default-prediction-for-profit-maximization-45fcd461582b#2a76)  
∘ [3.1 Preprocessing](https://towardsdatascience.com/loan-default-prediction-for-profit-maximization-45fcd461582b#7fe4)  
∘ [3.2 Model Selection](https://towardsdatascience.com/loan-default-prediction-for-profit-maximization-45fcd461582b#a303)  
∘ [3.3 Model Optimization](https://towardsdatascience.com/loan-default-prediction-for-profit-maximization-45fcd461582b#33ab)  
· [4. Conclusions](https://towardsdatascience.com/loan-default-prediction-for-profit-maximization-45fcd461582b#015a)  
  [References](https://towardsdatascience.com/loan-default-prediction-for-profit-maximization-45fcd461582b#5616)

1. Introduction

This project is part of my freelance data science work for a client. There is no non-disclosure agreement required and the project does not contain any sensitive information. So, I decided to showcase the data analysis and modeling sections of the project as part of my personal data science portfolio. The client’s information has been anonymized.

The goal of this project is to build a machine learning model that can predict if a person will default on the loan based on the loan and personal information provided. The model is intended to be used as a reference tool for the client and his financial institution to help make decisions on issuing loans, so that the risk can be lowered, and the profit can be maximized.

2. Data Cleaning and Exploratory Analysis

The dataset provided by the client consists of 2,981 loan records with 33 columns including loan amount, interest rate, tenor, date of birth, gender, credit card information, credit score, loan purpose, marital status, family information, income, job information, and so on. The status column shows the current state of each loan record, and there are 3 distinct values: Running, Settled, and Past Due. The count plot is shown below in Figure 1, where 1,210 of the loans are currently running, and no conclusions can be drawn from these records, so they are removed from the dataset. On the other hand, there are 1,124 settled loans and 647 past-due loans, or defaults.

The dataset comes as an Excel file and is nicely formatted in tabular forms. However, a variety of problems do exist in the dataset, so it would still need extensive data cleaning before any analysis can be made. Different types of cleaning methods are exemplified below:

(1) Drop features: Some columns are duplicated (e.g., “status id” and “status”). Some columns may cause information leakage (e.g., “amount due” with 0 or negative number infers the loan is settled) In both cases, the features need to be dropped.

(2) Unit Conversion: Units are used inconsistently in columns such as “Tenor” and “proposed payday”, so conversions are applied within the features.

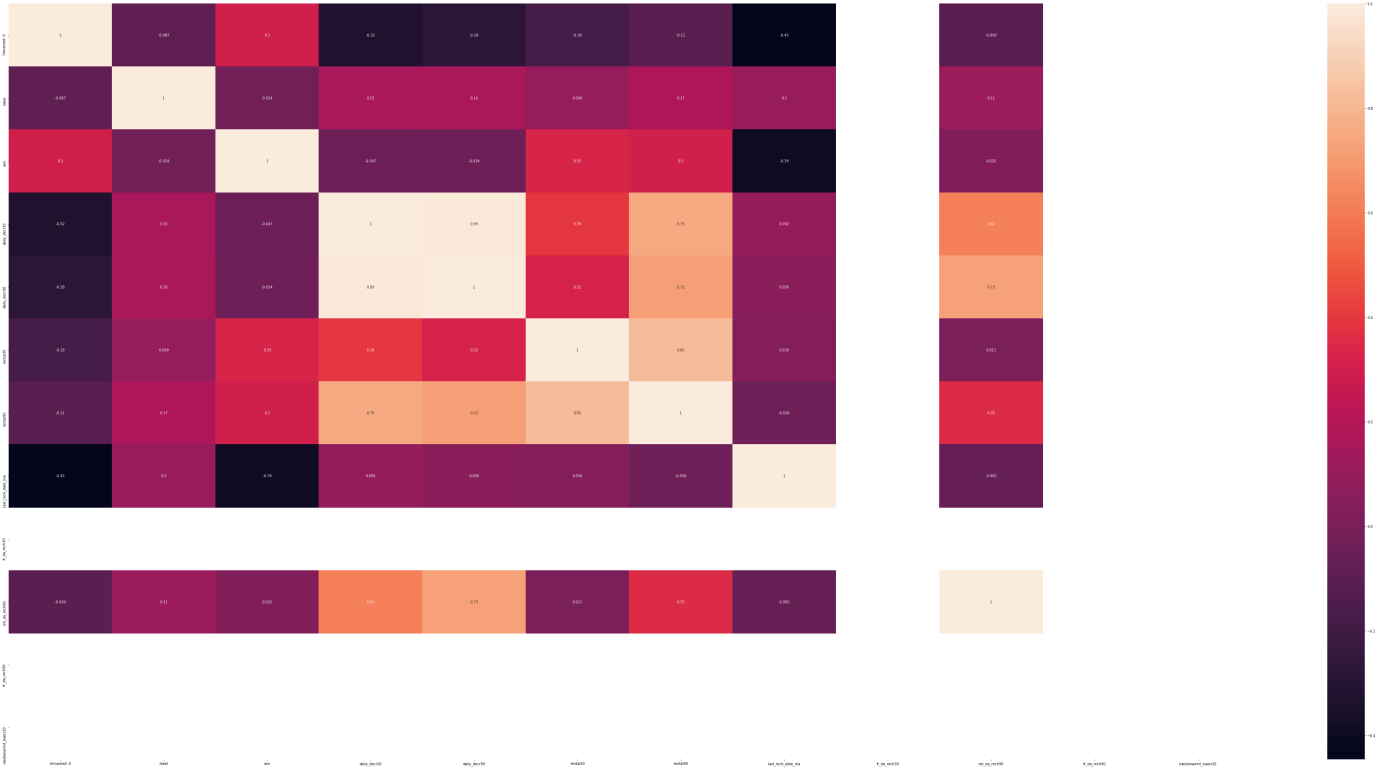
(3) Resolve Overlaps: Descriptive columns contain overlapped values. E.g., the income of “50,000–99,999” and “50,000–100,000” are essentially the same, so they need to be combined for consistency.

(4) Generate Features: Features like “date of birth” are too specific for visualization and modeling, so it is used to generate a new “age” feature that is more generalized. This step can also be seen as part of the feature engineering work.

(5) Labeling Missing Values: Some categorical features have missing values. Different from those in numeric variables, these missing values may not need to be imputed. Many of these are left for reasons and could affect the model performance, so here they are treated as a special category.

After data cleaning, a variety of plots are made to examine each feature and to study the relationship between each of them. The goal is to get familiar with the dataset and discover any obvious patterns before modelin

Figure 2: Heatmap of Pearson’s Correlation Coefficients between numerical variables



For numerical and label encoded variables, correlation analysis is performed. Correlation is a technique for investigating the relationship between two quantitative, continuous variables in order to represent their inter-dependencies. Among different correlation techniques, Pearson’s correlation is the most common one, which measures the strength of association between the two variables. Its correlation coefficient scales from -1 to 1, where 1 represents the strongest positive correlation, -1 represents the strongest negative correlation and 0 represents no correlation. The correlation coefficients between each pair of the dataset are calculated and plotted as a heatmap in Figure 2.

From the heatmap, it is easy to locate the highly correlated features with the help of color coding: positively correlated relationships are in red and negative ones are in red. The status variable is label encoded (0 = settled, 1 = past due), so that it can be treated as numerical. It can be easily found that there is one outstanding coefficient with status (first row or first column): -0.31 with “tier”. Tier is a variable in the dataset that defines the level of Know Your Customer (KYC). A higher number means more knowledge of the customer, which infers that the customer is more reliable. Therefore, it makes sense that with a higher tier, it is less likely for the customer to default on the loan. The same conclusion can be drawn from the count plot shown in Figure 3, where the number of customers with tier 2 or tier 3 is significantly lower in “Past Due” than in “Settled”.

Besides the status column, some other variables are correlated as well. Tier is correlated with loan amount, interest due, tenor, and interest rate. Customers with a higher tier tend to get higher loan amount and longer time of repayment (tenor) while paying less interest. Interest due is highly correlated with interest rate and loan amount, same as expected. A higher interest rate usually comes with a lower loan amount and tenor. Proposed payday is highly correlated with tenor. On the other side of the heatmap, the credit score is positively correlated with monthly net income, age, and work seniority. The number of dependents is correlated with age and work seniority as well. These listed relationships among variables may not be directly related to the status, the label that we want the model to predict, but they are still good practice to get familiar with the features, and they could also be useful for guiding the model regularizations.

The categorical variables are not as convenient to investigate as the numerical features because not all categorical variables are ordinal: Tier (Figure 3) is ordinal, but Self ID Check (Figure 4) is not. So, a pair of count plots are made for each categorical variable, to study their relationships with the loan status. Some of the relationships are very obvious: customers with tier 2 or tier 3, or who have their selfie and ID successfully checked are more likely to pay back the loans. However, there are many other categorical features that are not as obvious, so it would be a great opportunity to use machine learning models to excavate the intrinsic patterns and help us make predictions.

3. Modeling

Since the goal of the model is to make binary classification (0 for settled, 1 for past due), and the dataset is labeled, it is clear that a binary classifier is needed. However, before the data are fed into machine learning models, some preprocessing work (beyond the data cleaning work mentioned in section 2) needs to be done to generalize the data format and be recognizable by the algorithms.

3.1 Preprocessing

Feature scaling is an important step to rescale the numeric features so that their values can fall in the same range. It is a common requirement by machine learning algorithms for speed and accuracy. On the other hand, categorical features usually cannot be recognized, so they have to be encoded. Label encodings are used to encode the ordinal variable into numerical ranks and one-hot encodings are used to encode the nominal variables into a series of binary flags, each represents whether the value exists.

After the features are scaled and encoded, the total number of features is expanded to 165, and there are 1,735 records that include both settled and past-due loans. The dataset is then split into training (70%) and test (30%) sets. Due to its imbalance, Adaptive Synthetic Sampling (ADASYN) is applied to oversample the minority class (past due) in the training class to reach the same number as the majority class (settled) in order to remove the bias during training.

3.2 Model Selection

There are 6 classification algorithms selected as the candidate for the model. K-nearest Neighbors (KNN) is a non-parametric algorithm that makes predictions based on the labels of the closest training instances. Naïve Bayes is a probabilistic classifier that applies Bayes Theorem with strong independence assumptions between features. Both Logistic Regression and Linear Support Vector Machine (SVM) are parametric algorithms, where the former models the probability of falling into either one of the binary classes and the latter finds the boundary between classes. Both Random Forest and XGBoost are tree-based ensemble algorithms, where the former applies bootstrap aggregating (bagging) on both records and variables to build multiple decision trees that vote for predictions, and the latter uses boosting to continuously strengthen itself by correcting mistakes with efficient, parallelized algorithms.

All of the 6 algorithms are commonly used in any classification problem and they are good representatives to cover a variety of classifier families. The training set is then fed into each of the models with 5-fold cross-validation, a technique that estimates the model performance in an unbiased way, with a limited sample size. The mean accuracy of each model is shown below in Table 1:

https://miro.medium.com/max/60/1*qoeTe1pBN4dIzQ476Lin3A.png?q=20

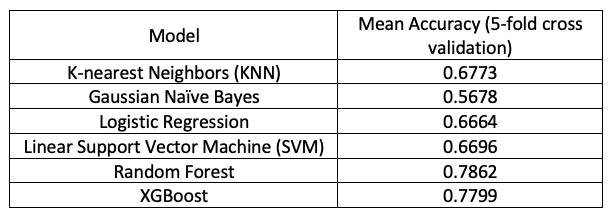


Table 1: Model Performance of the classifier candidates

It is clear that all 6 models are effective in predicting defaulted loans: they are all above 0.5, the baseline set based on a random guess. Among them, Random Forest and XGBoost have the most outstanding accuracy scores. This result is well expected, given the fact that Random Forest and XGBoost has been the most popular and powerful machine learning algorithms for a while in the data science community. Therefore, the other 4 candidates are discarded, and only Random Forest and XGBoost are then fine-tuned using the grid-search method to find the best performing hyperparameters. After fine-tuning, both models are tested with the test set. The accuracies are 0.7486 and 0.7313, respectively. The values are a little bit lower because the models have never seen the test set before, and the fact that the accuracies are close to those given by cross-validations infers that both models are well fit.

3.3 Model Optimization

Even though the models with the best accuracies are found, more work still needs to be done to optimize the model for our application. The goal of the model is to help make decisions on issuing loans to maximize the profit, so how is the profit related to the model performance? In order to answer the question, two confusion matrices are plotted in Figure 5 below

Confusion matrix is a tool that visualizes the classification results. In binary classification problems, it is a 2 by 2 matrix where the columns represent predicted labels given by the model and the rows represent the true labels. For example, in Figure 5 (left), the Random Forest model correctly predicts 268 settled loans and 122 defaulted loans. There are 71 defaults missed (Type I Error) and 60 good loans missed (Type II Error). In our application, the number of missed defaults (bottom left) needs to be minimized to save loss, and the number of correctly predicted settled loans (top left) needs to be maximized in order to maximize the earned interest.

Some machine learning models, such as Random Forest and XGBoost, classify instances based on the calculated probabilities of falling into classes. In binary classifications problems, if the probability is higher than a certain threshold (0.5 by default), then a class label will be placed on the instance. The threshold is adjustable, and it represents a level of strictness in making the prediction. The higher the threshold is set, the more conservative the model is to classify instances. As seen in Figure 6, when the threshold is increased from 0.5 to 0.6, the total number of past-dues predict by the model increases from 182 to 293, so the model allows fewer loans to be issued. This is effective in lowering the risk and saves the cost because it greatly decreased the number of missed defaults from 71 to 27, but on the other hand, it also excludes more good loans from 60 to 127, so we lose opportunities to earn interest.

In order to balance the trade-off between the decrease in revenue and a decrease in cost, an optimization problem has to be solved by adjusting the threshold and seeking the optimum. If “Settled” is defined as positive and “Past Due” is defined as negative, then by using the layout of the confusion matrix plotted in Figure 6, the four regions are divided as True Positive (TN), False Positive (FP), False Negative (FN) and True Negative (TN). Aligned with the confusion matrices plotted in Figure 5, TP is the good loans hit, and FP is the defaults missed. We are more interested in these two regions. To normalize the values, two commonly used mathematical terms are defined: True Positive Rate (TPR) and False Positive Rate (FPR). Their equations are shown below:

When the threshold is at 0, the model reaches the mostaggressive setting, where all loans are expected to be settled. It is essentially how the client’s business performs without the model: the dataset only consists of the loans that have been issued. It is clear that the profit is below -1,200, meaning the business loses money by over 1,200 dollars per loan.

If the threshold is set to 0, the model becomes the most conservative, where all loans are expected to default. In this case, no loans will be issued. There will be neither money lost, nor any earnings, which leads to a profit of 0.

To find the optimized threshold for the model, the maximum profit needs to be located. In both models, the sweet spots can be found: The Random Forest model reaches the max profit of 154.86 at a threshold of 0.71 and the XGBoost model reaches the max profit of 158.95 at a threshold of 0.95. Both models are able to turn losses into profit with increases of almost 1,400 dollars per person. Even though the XGBoost model boosts the profit by about 4 dollars more than the Random Forest model does, its shape of the profit curve is steeper around the peak. In the Random Forest model, the threshold can be adjusted between 0.55 to 1 to ensure a profit, but the XGBoost model only has a range between 0.8 and 1. In addition, the flattened shape in the Random Forest model provides robustness to any fluctuations in data and will elongate the expected lifetime of the model before any model update is required. Therefore, the Random Forest model is suggested to be deployed at the threshold of 0.71 to maximize the profit with a relatively stable performance.

4. Conclusions

This project is a typical binary classification problem, which leverages the loan and personal information to predict whether the customer will default the loan. The goal is to use the model as a tool to help make decisions on issuing the loans. Two classifiers are built using Random Forest and XGBoost. Both models have the capability of turning the loss to profit by over 1,400 dollars per loan. The Random Forest model is preferred to be deployed due to its stable performance and robustness to errors.

The relationships between features have been studied for better feature engineering. Features such as Tier and Selfie ID Check are found to be possible predictors that determine the status of the loan, and both of them have been confirmed later in the classification models because they both appear in the top list of feature importance. Many other features are not as obvious on the roles they play that affect the loan status, so machine learning models are built in order to discover such intrinsic patterns.

There are 6 common classification models used as candidates, including KNN, Gaussian Naïve Bayes, Logistic Regression, Linear SVM, Random Forest, and XGBoost. They cover a wide variety of algorithm families, from non-parametric to probabilistic, to parametric, to tree-based ensemble methods. Among them, the Random Forest model and the XGBoost model give the best performance: the former has an accuracy of 0.7486 on the test set and the latter has an accuracy of 0.7313 after fine-tuning.

The most important part of the project is to optimize the trained models to maximize the profit. Classification thresholds are adjustable to change the “strictness” of the prediction results: With lower thresholds, the model is more aggressive that allows more loans to be issued; with higher thresholds, it becomes more conservative and will not issue the loans unless there is a high probability that the loans can be paid back. By using the profit formula as the loss function, the relationship between the profit and the threshold level has been determined. For both models, there exist sweet spots that can help the business turn from loss to profit. Without the model, there is a loss of more than 1,200 dollars per loan, but after implementing the classification models, the business is able to yield a profit of 154.86 and 158.95 per customer with the Random Forest and XGBoost model, respectively. Even though it reaches a higher profit using the XGBoost model, the Random Forest model is still recommended to be deployed for production because the profit curve is flatter around the peak, which brings robustness to errors and steadiness for fluctuations. Due to this reason, less maintenance and updates would be expected if the Random Forest model is chosen.

The next steps in the project are to deploy the model and monitor its performance when newer records are observed. Adjustments will be needed either seasonally or anytime the performance drops below the baseline standards to accommodate for the changes brought by the external factors. The frequency of model maintenance for this application does not to be high given the amount of transactions intake, but if the model needs to be used in an accurate and timely fashion, it is not difficult to transform this project into an online learning pipeline that can ensure the model to be always up to date.

References

[1]: Decoding the Confusion Matrix, Towards Data Science, 2019: <https://towardsdatascience.com/decoding-the-confusion-matrix-bb4801decbb>