Using tree based method to predict potential defaulters

Group K

June 3, 2024

Abstract

- P2P lending is one of the most emerging disruptors in the financial sector. Lending Club is a P2P platform based in America. Despite its flexibility in providing instant lending, this industry carries high risks for investors lending money. In this project, We referenced the paper by Mauritsius et al (2019), our results show that both Random forest and LightGBM have good performance in detecting potential defaulters.

Data Understanding

- The data downloaded on the lending club's website where the date range from Q1 2007 - Q4 2015
- Including the current loan status (Current, Late, Fully Paid, etc.) and latest payment information. Variables include credit scores, number of finance inquiries, address including zip codes and state, and collections among others.
- Dataset contains over 22 million observations and 145 variables.

Motivation

- Before lending money, the lender wants to ensure they can get it back.
  - Rejecting a legitimate customer may result in limited missed benefits, but inadvertently accepting a high-risk client could lead to substantial losses.
- In other words, label defaulters as 1 and non-defaulters as 0. Then, our model will prioritize achieving a higher recall rate for the defaulters (1).

Our dependent variable

- In this dataset, the loan status (dependent variable) has 9 categories in order of severity.
    - We do not need to predict all categories, only to identify high-risk individuals.
- We categorize them into 2 levels of severity.

Our dependent variable

Status	Current	Fully Paid	In Grace Period	Late (16-30 days)	Fully Paid*
Amount	788,950	646,902	10,474	5,786	1,988

Table: Low Risk (assign 0)

Status	Late (31-120 days)	Charged Off	Charge off*	Default
Amount	23,763	168,084	761	70

Table: High Risk (assign 1)

- * means the individuals don't meet the credit policy.

Feature selection & creation

- Remove feature that is possibly not a causal factor

  e.g. member ID, issue date, etc.
  - Remove 7 features.
- Remove the reverse causality features, avoiding hindsight bias

  e.g. Principal received to date; we won't know the actual amount before the whole cycle is ended.
  - Remove 11 features.
- Mauritsius et al.(2019) did not include this step, resulting in the accuracy of their model being 99%.

- Define variables that could potentially determine whether a borrower is a potential defaulter:
  e.g. Installment-to-Income Ratio , Number of Satisfactory Bankcard Accounts ÷ Total Bankcard Accounts
- Some applicants apply for the loan jointly, so we combine the data of secondary applicants together.
  e.g. we define the debt-to-income ratio (DTI) as the average of individual and joint DTI: $dti = (dti + dti_{\text{joint}})/2$.
- 82 features remains in model.

Data cleaning

- Convert categorical data into numerical one with `LabelEncoder()`
- Fill `na` values with `mode()` or `median()`, depend on its data type.

```
# encoding
df['emp_length'] = label_encoder.fit_transform(df['emp_length'])

# fill na with mode
emp_length_mode = df['emp_length'].mode()[0]
df['emp_length'] = df['emp_length'].fillna(emp_length_mode)
```

Model selection

- Our dependent variable is binary, and there are many categorical features in this case.
- Mauritsius et al.(2019) used the Naive Bayes Classifier, but our features do not satisfy the Independence Assumption & Same Distribution Assumption.
- Other algorithms like KNN, logistic regression, and SVM rely on algorithms related to the Euclidean distance of features.
  $\Longrightarrow$ We may use a random forest model for prediction and gradient boosting machine for comparison.

Classification Tree

Breiman, Friedman, Olshen and Stone (1984)

- $\{(\mathbf{x}_i, y_i) : i = 1, \ldots, N\}\,, \quad \mathbf{x}_i = (x_{i1}, \ldots, x_{ip})$
- $Y = f(\mathbf{X})$

1. Start at the root node, i.e., with all of the data.
2. Seek the splitting variable j and split point s that minimize the the sum of entropy in each split:
3. $$\min_{j,s} \left[\frac{|R_1(j,s)|}{N} \left(1 - \sum_{k=1}^{K} p_{k,R_1}^2 \right) + \frac{|R_2(j,s)|}{N} \left(1 - \sum_{k=1}^{K} p_{k,R_2}^2 \right) \right]$$

Details of Model Training

- LC assigns credit ratings A to G to each borrower based on their FICO score, and we have built models for different ratings.
- Random Forest and LightGBM both use random search CV to tune parameters for the highest recall rate.

Model	Best Parameters
RF	{'n_estimators':125,'min_samples_split':8, 'min_samples_leaf':2,'max_features':70,'max_depth':30}
LightGBM	{'num_leaves':50,'n_estimators':150,'learning_rate':0.25}

- We use out-of-bag data to test the Random Forest ($\sim$ 37%), and reserve 20% of the data as the testing set for the LightGBM.

Evaluation

- With the previous preprocessing and consideration, we can start training our model and use it to evaluate our testing dataset:

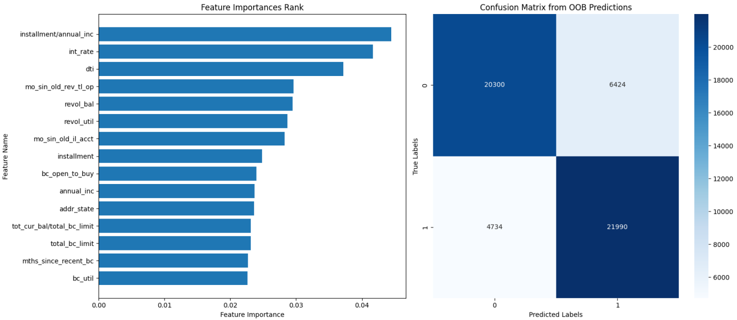

Figure: Risk prediction of F credit rating people with random forest model

Evaluation

- Since avoiding high risk people is our firs priority, recall rate should be the most suitable evaluation indicator for our prediction.
- The below table is the recall rate of random forest model and gradient boosting model:

Random Forest	Gradient Boosting
75.97%	74.70%

Table: Recall rate of two model

- The table shows that random forest model brings out a higher recall rate, so it should be a better algorithm in this case.

Demo — Application scenario assumption

- We are a loan company aiming to lend money to people with low credit ratings on the P2P platform.
- We want to use this model to help our employees identify potential defaulters.
- Whenever there's a new borrower, we only have to input their information into our model, and then we can get a prediction about this person.

- https://github.com/blossmuri/Lending-Club-Loan-Default-Prediction