Fintech: Predicting Repayment Status

July 29, 2024

Abstract

Use ML to predict the repayment status of borrowers – a classic topic of fintech. Also, I'll explore how to choose more efficient features.

1 Research Objectives

This article provides a detailed breakdown of a relatively traditional fintech project, covering various aspects such as data acquisition, data observation, data cleaning, data preprocessing, model selection and training, reflection and improvement (enhancing feature selection), and more.
This study focuses on the repayment status of borrowers from Lending Club Loans. Lending Club is the largest lending market in the United States, with over 4 million users obtaining more than $80 billion in personal loans through the platform. As the only large-scale digital marketplace bank, Lending Club offers a wide range of financial products and services to its users. Predicting a borrower's repayment status based on various factors is a crucial task for all lending companies (not limited to traditional banks).
Therefore, this article will primarily focus on predicting the repayment status of borrowers based on various characteristics from Lending Club Loans, aiming to achieve a selection objective.

2 Data Source

The data for this study primarily comes from the borrower and repayment status datasets available on the Lending Club official website.
After reading the data, we observe the general situation of the data table:
After gaining a general understanding of the data table (including the columns, their data types, the number of records, and the number of unique values), we can begin building the model.

3 Building the Model

3.1 Checking and Observing the Data

3.1.1 First, we need to check for missing values in the data.

We found that, aside from some missing values in the **emp_length** column, there are almost no missing values in other data columns. Since we are not using a time series model here, these missing values are not a major concern and can be directly handled by using *dropna* in subsequent modeling.

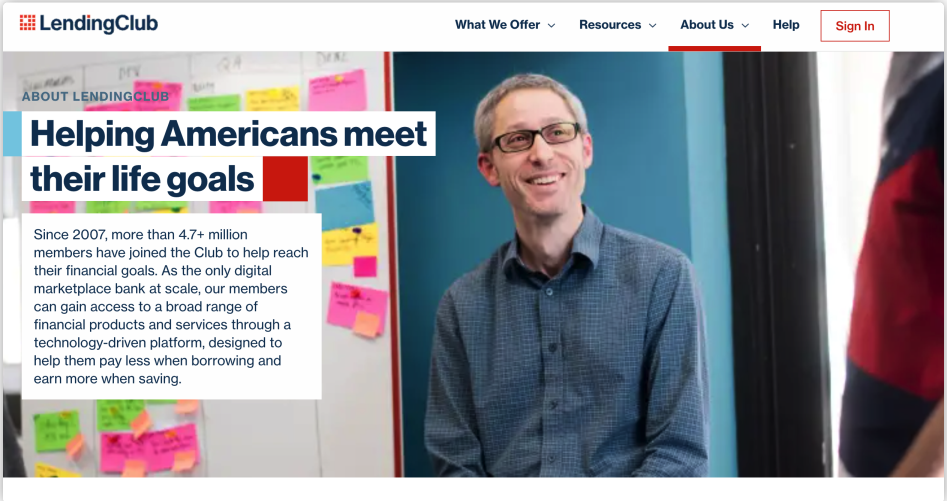

Figure 1: A screenshot of LendingClub: https://www.lendingclub.com

```
RangeIndex: 9004 entries, 0 to 9003
Data columns (total 29 columns):
 #   Column               Non-Null Count  Dtype
---  ------               --------------  -----
 0   id                   9004 non-null   int64
 1   loan_amnt            9004 non-null   int64
 2   funded_amnt          9004 non-null   int64
 3   funded_amnt_inv      9004 non-null   float64
 4   term                 9004 non-null   object
 5   int_rate             9004 non-null   float64
 6   installment          9004 non-null   float64
 7   grade                9004 non-null   object
 8   emp_length           8688 non-null   object
 9   home_ownership       9004 non-null   object
 10  annual_inc           9004 non-null   float64
 11  verification_status  9004 non-null   object
 12  purpose              9004 non-null   object
 13  addr_state           9004 non-null   object
 14  dti                  9004 non-null   float64
 15  earliest_cr_line     9004 non-null   int64
 16  inq_last_6mths       9004 non-null   int64
 17  open_acc             9004 non-null   int64
 18  pub_rec              9004 non-null   int64
 19  revol_bal            9004 non-null   int64
 20  revol_util           9001 non-null   float64
 21  total_acc            9004 non-null   int64
 22  out_prncp            9004 non-null   int64
 23  out_prncp_inv        9004 non-null   int64
 24  total_pymnt          9004 non-null   float64
 25  total_pymnt_inv      9004 non-null   float64
 26  total_rec_prncp      9004 non-null   float64
 27  total_rec_int        9004 non-null   float64
 28  loan_status          9004 non-null   object
dtypes: float64(10), int64(11), object(8)
```

Figure 2: Brief info of data

```
id                     9004
loan_amnt               604
funded_amnt             681
funded_amnt_inv        1234
term                      2
int_rate                 70
installment            3871
grade                     7
emp_length               11
home_ownership            3
annual_inc             1555
verification_status       3
purpose                  13
addr_state               45
dti                    2559
earliest_cr_line        458
inq_last_6mths            9
open_acc                 33
pub_rec                   3
revol_bal              7573
revol_util             1023
total_acc                63
out_prncp                 1
out_prncp_inv             1
total_pymnt            8962
total_pymnt_inv        8942
total_rec_prncp        2199
total_rec_int          8838
loan_status               2
dtype: int64
```

Figure 3: Unique numbers of each item

3.1.2 Next, we will check the correlation among the data.

It can be observed that there is a high correlation among **total_payment, total_payment_inv, total_rec_prncp, and total_rec_int**, indicating redundancy among these features.
In fact, in the "Reflection and Summary" section at the end of this article, we will also find that the importance of these features is relatively low. Therefore, when further optimizing the model, we can select only one representative feature from this group.

3.1.3 Then, we will observe the distribution characteristics of the data.

The above section discusses observing the distribution of features. Next, we will match the features with the repayment status to more intuitively examine the relationship between each feature and the repayment status. From the initial observations, it appears that the features **term, grade, emp_length, and purpose** all have some predictive power regarding the repayment status.
Next, we will visualize the regional distribution

3.2 Data Cleaning

First, convert categorical variables into numerical variables. The data table (showing a subset of the columns) will then be updated as follows.
Next, remove missing values and redundant features. Finally, perform data normalization, using z-score normalization.

3.3 Training the Model

Since the target variable y(repayment status) is a binary classification variable (0/1), we will not consider models that predict continuous outcomes (such as linear regression). Instead, we will consider four models: Logistic Regression, K-Nearest Neighbors (KNN), Random Forest, and Support Vector Machine (SVM). We will use Cross-Validation to select the best model among the four as our predictive model, employing an 80%-20% split for training and testing data.
The performance of the logistic regression model is the best; thus, we trained the data using logistic regression. On the test set, the model performs well, with high accuracy and recall rates, effectively predicting borrowers' repayment status.

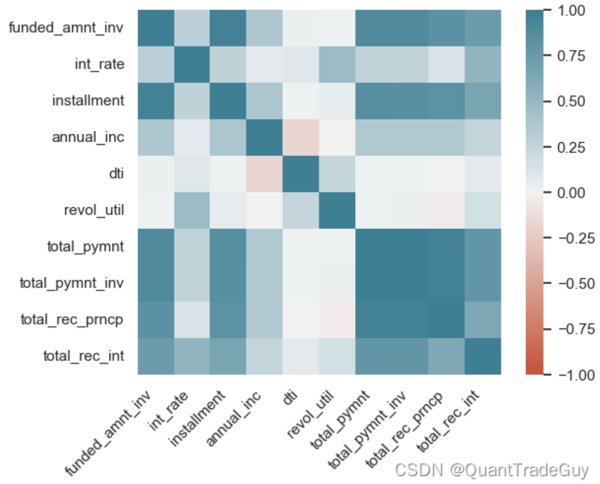

Figure 4: Correlation heatmap

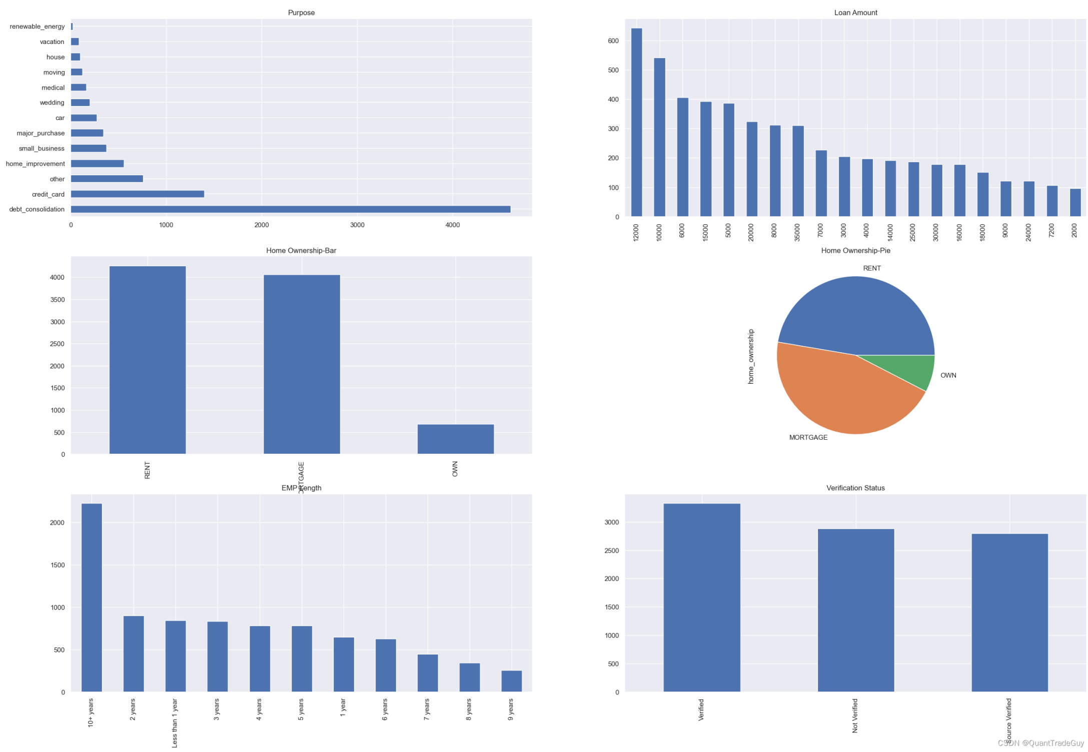

Figure 5: The distribution of features (before match)

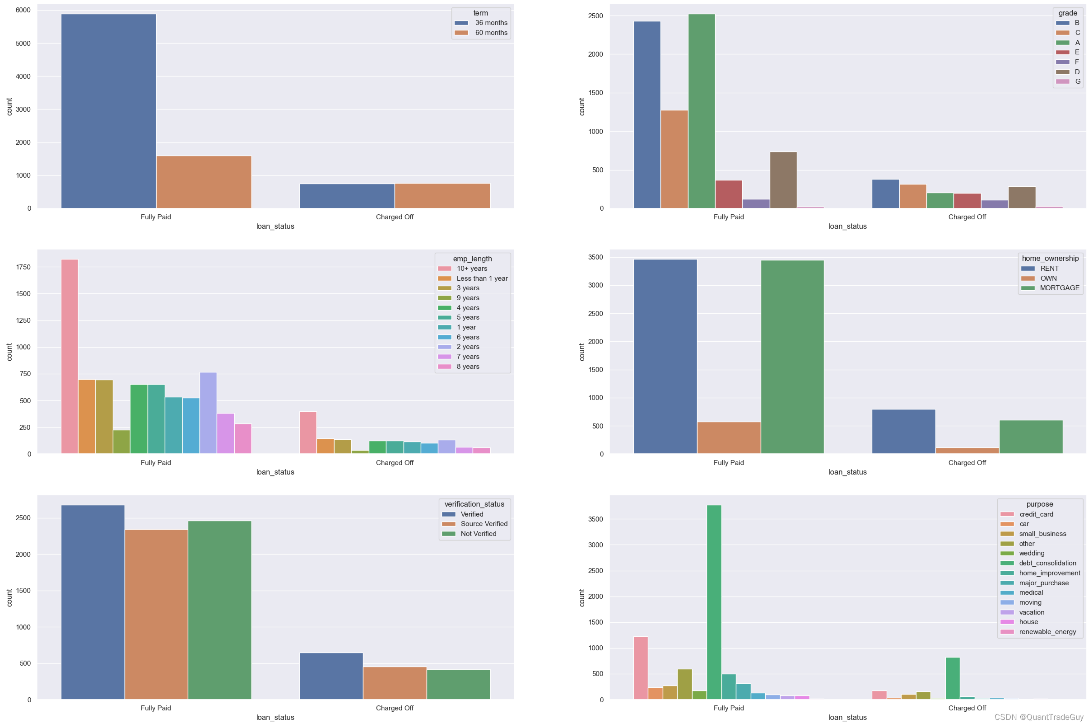

Figure 6: The distribution of features (after match)

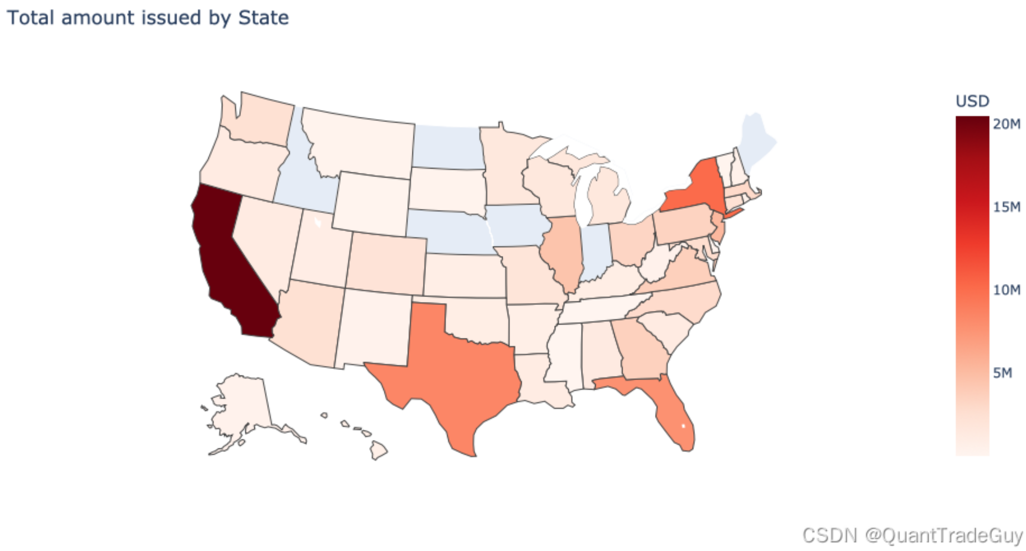

Figure 7: Visualize the regional distribution

	installment	grade	emp_length	home_ownership	...	revol_bal
0	162.87	1	10+ years	2	...	13648
1	59.83	2	Less than 1 year	2	...	1687
2	84.33	2	10+ years	2	...	2956
3	339.31	2	10+ years	2	...	5598
4	156.46	0	3 years	2	...	7963
5	109.43	4	9 years	2	...	8221
6	152.39	5	4 years	1	...	5210
7	121.45	1	Less than 1 year	2	...	9279
8	153.45	2	5 years	1	...	4032
9	402.54	1	10+ years	1	...	23336

Figure 8: Convert categorical variables into numerical variables

	precision	recall	f1-score	support
0	0.52	0.04	0.08	302
1	0.84	0.99	0.91	1499
accuracy			0.83	1801
macro avg	0.68	0.52	0.49	1801
weighted avg	0.78	0.83	0.77	1801

Figure 9: Performance on test set

4 Reflection and Improvement

We can further enhance the model. In addition to training with more models, one improvement approach is to perform feature selection, identifying important features and reducing multicollinearity to make the results more significant.

For feature refinement, we will employ two methods for cross-validation. The first method involves using the feature importance ranking from Random Forest.

The second method is to observe the coefficients of the logistic regression model to assess the extent (importance) of their impact on the predicted variables.

It can be seen that **annual_inc, int_rate, and revol_util** are likely the top three important features. Additionally, we find that **out_prncp and out_prncp_inv** have a minimal impact on the predicted variables. These variables can be removed or considered for feature merging to obtain higher-quality features. We can then continue to follow the steps outlined in the "Building the Model" section to train a more optimal model.

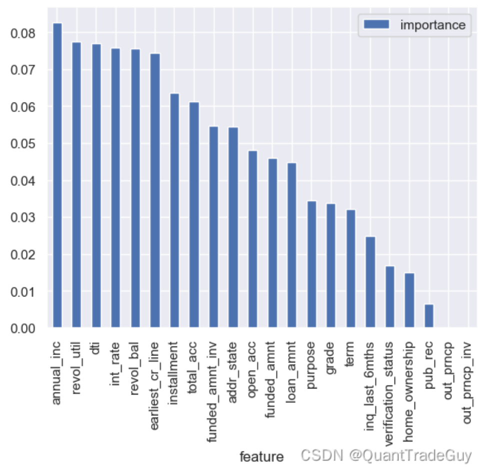

Figure 10: Random Forest Importance Ranking

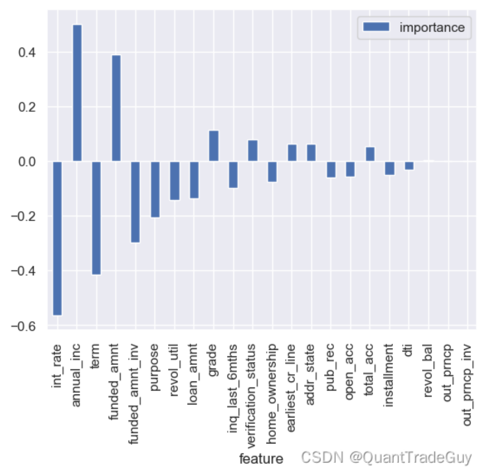

Figure 11: Linear Regression Importance Ranking