Final Report_Predicting Default Rate for Lending Club

Background

Lending Club, a marketplace for personal loans that matches borrowers who are seeking a loan with investors looking to lend money and make a return. Each borrower fills out a comprehensive application, providing their past financial history, the reason for the loan, and more. Lending Club evaluates each borrower's credit score using past historical data (and their own data science process!) and assigns an interest rate to the borrower.

Approved loans are listed on the Lending Club website, where qualified investors can browse recently approved loans, the borrower's credit score, the purpose for the loan, and other information from the application.

Once an investor decides to fund a loan, the borrower then makes monthly payments back to Lending Club. Lending Club redistributes these payments to investors. This means that investors don't have to wait until the full amount is paid off to start to see returns. If a loan is fully paid off on time, the investors make a return which corresponds to the interest rate the borrower had to pay in addition to the requested amount.

Many loans aren't completely paid off on time, however, and some borrowers default on the loan.

That's the problem we'll be trying to address as we clean some data from Lending Club for machine learning. Let's imagine we've been
tasked with building a model to predict whether borrowers are likely to pay or default on their loans.

Problem Statement

The purpose of this data science project is to address the default loans and come up with a delinquency prediction model for the Lending Club. These models could help LendingClub investors make better-informed investment decisions.

Data Collection

The dataset is provided by Lending Club and only subseted on 2018Q1, which is with 107,864 loans and 100 variables for each loan.

There is a document-LCDataDictionary document released by Lending Club, which is useful for understanding what each column represents in the data set. there are total eight categorical variables in the dataset:

- User feature (general)
- User feature (financial specific): income, credit scores, credit lines
- Loan general feature
- Loan payment feature
- Current loan payment feature
- Secondary application info
- Hardship
- Settlement

Target variables: loan_status

Data Cleaning

Firstly, we should import the dataset by pandas and conduct some basic clearing tasks to remove some information that is not relevant for prediction. Therefore, this stage focuses on collecting the data, organizing it and making sure it's well defined.
The columns that falled on the following categories:Loan general feature, Loan payment feature, Current loan payment feature, Hardship,Settlement, should be excluded from the dataset, because they are related to the target variable and have risks for leaking information from the future(after the loan has already been funded) . Then the features categories are included User feature (general),User feature (financial specific), Secondary application info.

Secondly, we're going to:

- Drop the column of ID, which is a unique LC assigned ID for the loan listing, so it does not provide any meaningful information and doesn't affected the borrower's ability to pay back the loan.
- Drop the column of issue_d, which means the month which the loan was funded.
- Drop the column of url.

After completing the preliminary processing, the shape of the dataset is (107866, 112).

Then, due to the large dataset with 112 columns, it'd be better to have a big picture that missing values consist in the dataset. Below are the missing value table and the result shows that the variables are filled with missing value over 80%.

In [11]:
```
df_m = missing_table[missing_table['% of Total Values']>=80]
df_m
```

Out[11]:

	Missing Values	% of Total Values
member_id	107866	100.0
desc	107866	100.0
sec_app_mths_since_last_major_derog	102439	95.0
mths_since_last_record	92597	85.8
verification_status_joint	91849	85.2
sec_app_revol_util	91845	85.1
sec_app_mort_acc	91535	84.9
dti_joint	91535	84.9
annual_inc_joint	91535	84.9
revol_bal_joint	91535	84.9
sec_app_fico_range_low	91535	84.9
sec_app_earliest_cr_line	91535	84.9
sec_app_inq_last_6mths	91535	84.9
sec_app_fico_range_high	91535	84.9
sec_app_open_acc	91535	84.9
sec_app_open_act_il	91535	84.9
sec_app_num_rev_accts	91535	84.9
sec_app_chargeoff_within_12_mths	91535	84.9
sec_app_collections_12_mths_ex_med	91535	84.9
mths_since_recent_bc_dlq	86568	80.3

Before understanding the deep meaning of all columns, the columns with all missing values could be dropped at this stage, then the shape of the dataset is (107864, 110).

Nextly, we should remove the columns containing only the distinct value and the duplicated rows in the dataframe.

Lastly, exploring whether there are some columns which are highly multicollinearity,if yes, it could be dropped one of them due to the lots of columns existing. High Correlation Coefficients Pairwise correlations among independent variables might be high (in absolute value). Rule of thumb: If the correlation > 0.8 then severe multicollinearity may be present.

```
high_cor = sol_df[sol_df['cor_coef']>= 0.8]
high_cor
```

	var_1	var_2	cor_coef
0	loan_amnt	funded_amnt	1.000000
1	sec_app_fico_range_low	sec_app_fico_range_high	1.000000
2	fico_range_low	fico_range_high	1.000000
3	loan_amnt	funded_amnt_inv	0.999996
4	funded_amnt	funded_amnt_inv	0.999996
5	open_acc	num_sats	0.999464
6	num_actv_rev_tl	num_rev_tl_bal_gt_0	0.985972
7	tot_cur_bal	tot_hi_cred_lim	0.981144
8	total_bal_il	total_il_high_credit_limit	0.949234
9	loan_amnt	installment	0.944942
10	funded_amnt	installment	0.944942
11	funded_amnt_inv	installment	0.944839
12	acc_now_delinq	num_tl_30dpd	0.940527
13	total_bal_il	total_bal_ex_mort	0.907411
14	mths_since_recent_bc_dlq	mths_since_recent_revol_delinq	0.893902
15	total_bal_ex_mort	total_il_high_credit_limit	0.890592
16	bc_open_to_buy	total_bc_limit	0.867218

17	mths_since_last_delinq	mths_since_recent_revol_delinq	0.860612
18	open_acc	num_op_rev_tl	0.854261
19	num_op_rev_tl	num_sats	0.853703
20	last_fico_range_high	last_fico_range_low	0.852071
21	num_actv_bc_tl	num_actv_rev_tl	0.849102
22	open_rv_24m	acc_open_past_24mths	0.848753
23	num_bc_tl	num_rev_accts	0.844582
24	num_actv_bc_tl	num_rev_tl_bal_gt_0	0.843279
25	bc_util	percent_bc_gt_75	0.841719
26	open_rv_12m	num_tl_op_past_12m	0.840773
27	num_op_rev_tl	num_rev_accts	0.829068
28	num_actv_bc_tl	num_bc_sats	0.827612
29	total_rev_hi_lim	total_bc_limit	0.826346
30	tot_cur_bal	avg_cur_bal	0.817899
31	num_bc_sats	num_bc_tl	0.807642

Exploratory Data Analysis

Different parameters will serve different purposes and it is worth spending more time to dig into each of them and understand what features each column represents.

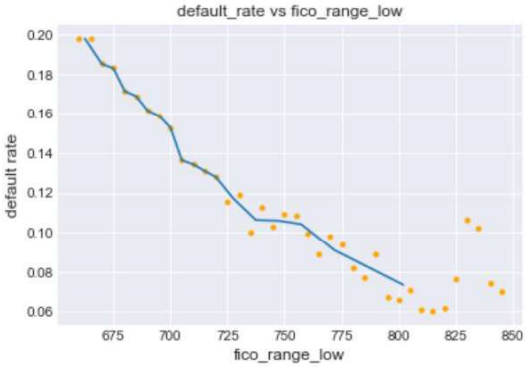
1. Univariate Analysis
   - Numerical Variables
     Subset the numerical features from the dataset and there are 64 columns. After dividing them into 5 groups to check their distribution of each variable, we just explore more if some abnormal trend shows on the plot.
     Two variables include ‘annual_inc’ and ‘dti’ has lots of extreme outliers and it should be preprocessed before the modelling.

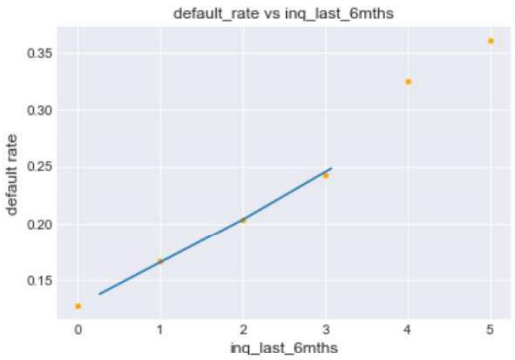
- Categorical Variable
  Below is the categorical columns and their associated classes.

	VarName	LevelsCount
4	emp_title	37288
14	revol_util	1075
11	zip_code	878
13	earliest_cr_line	641
19	sec_app_earliest_cr_line	560
1	int_rate	61
12	addr_state	50
16	last_credit_pull_d	36
3	sub_grade	35
9	purpose	13
10	title	12
5	emp_length	11
8	loan_status	7
2	grade	7
6	home_ownership	4
7	verification_status	3
18	verification_status_joint	3
15	initial_list_status	2
17	application_type	2
0	term	2

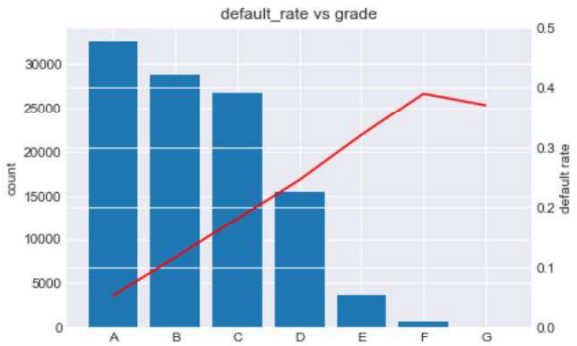
2.Bivariate analysis

- Exploring the relationship between numerical variables and default rate. Below are plots are showing that these two variables have some correlation with default rate.





- Studying the relationship of default rate with categorical variables and below two plots represent the two categorical variables having a positive relationship with default loan.



Preprocessing the Lending Club Dataset

Before applying the model, we should preprocess the dataset by handle the missing value and variable transformation.

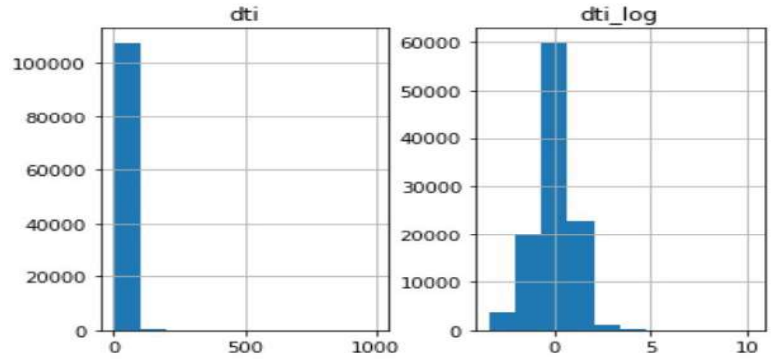
1. Deal with the missing values

The dataset has 79 columns and there are 29 columns that have missing values.We notice that most of the variables required providing second applicant's information filled

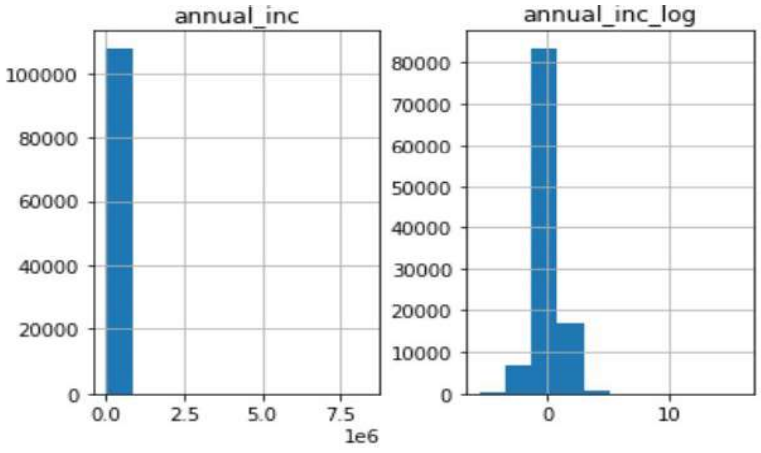
with lots of the missing value. From industry knowledge judgement, most of these variables did not provide too much information for the model, therefore, all those variables are dropped. After drop, there are totally 70 columns in the dataset still 20 variables with missing values. Top several variables filled with missing variables are about the join variable. Generally speaking, those variables are also not very important for the model, so all of them are dropped. At last, the left numerical variables with missing value are filled with median and the missing values in the categorical variables are filled with the category appearing most frequently.

2. Numerical variables transformation

According to the analysis result in phase two, we know that the distribution of these two numerical variables, dti and annual_inc,are rather right skew. Conducting log transformation for 'dti' and below are two plots before transformation and after.



Do the same thing for 'annual_inc'



3. Standardize the continuous variables

Rescale the data using scikit-learn using the MinMaxScaler class for modeling.

4. Categorical variable transformation

The main task is to get dummy for the categorical variables in this unit.

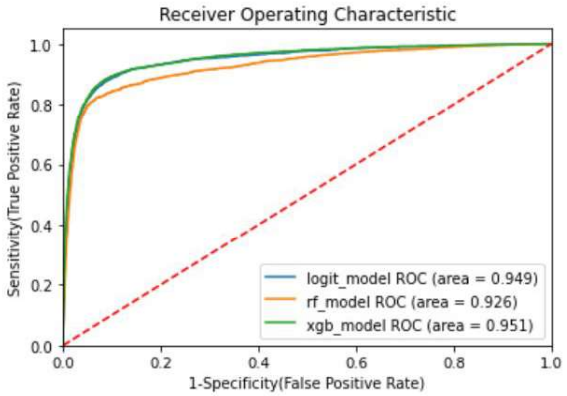
Machine learning Model Apply for predicting Default Rate

1. Apply for machine learning models

Applying the models for the preprocessed dataset and there are three models including Logistic Regression Model, Random Forest Model and Xgboosting Model.

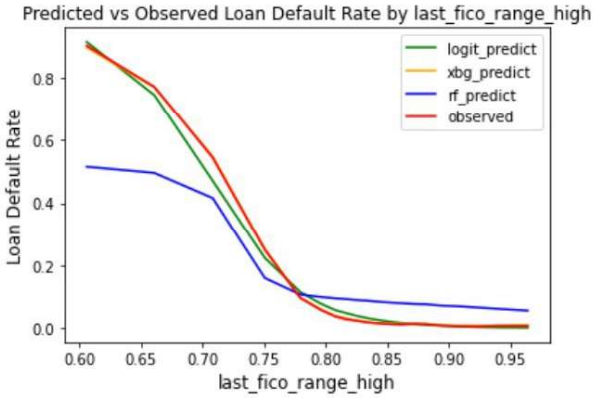
2. Model Comparison

The result of each model performance are great and all of them achieve great score from below plot.



3. Model Validation

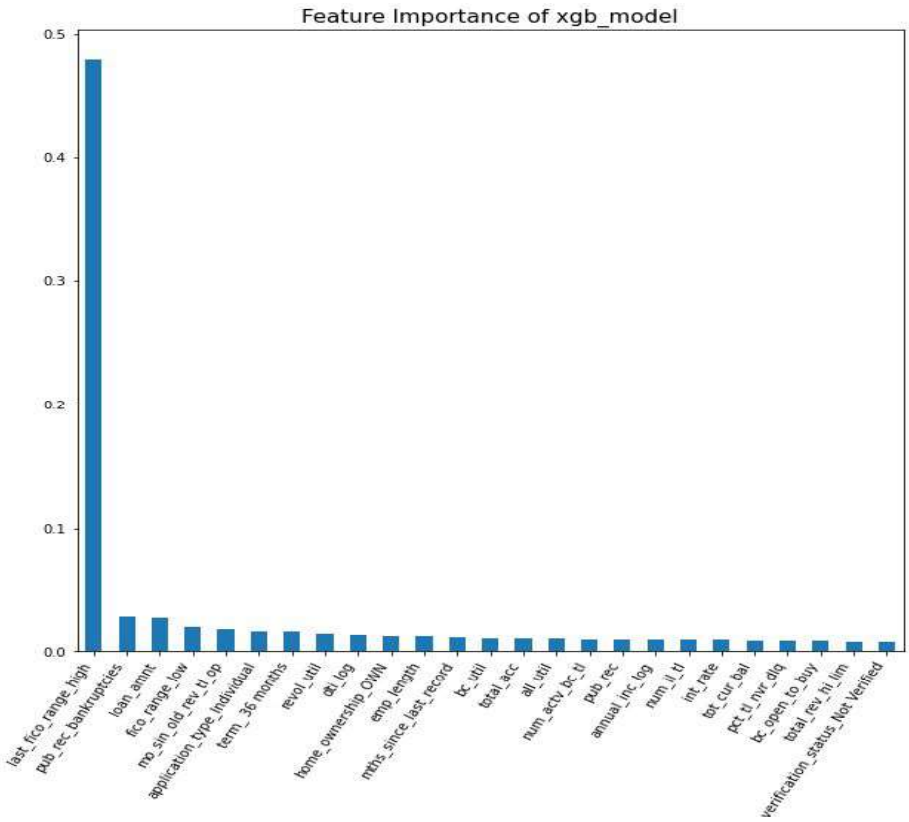
When validation the model among continuous variables. The variables of 'last_fico_range_high' show high correlation with default rate and even reach to 0.8 and other variables mainly surround 0.02~ 0.2. Therefore, it should be paid attention when on the feature selection stage.



Validation the model among categorical variables, but no any abnormal variables showing.

4. Feature Selection

In the feature selection, the plot shows the importance of 'last_fico_range_high' reaches above 0.7, which means that we should take a deep research for this variable again.

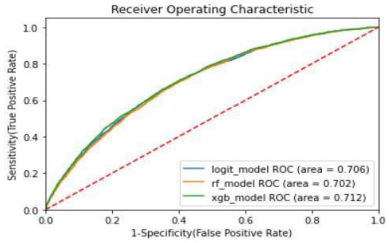


When a borrower applies for a loan, LendingClub gets the borrower's credit score from FICO — they are given a lower and upper limit of the range that the borrower's score belongs to, and they store those values as fico_range_low, fico_range_high. After that, any updates to the borrowers score are recorded as last_fico_range_low, and last_fico_range_high, which means that this variable is related to the target variable: loan_status.The solution is that this variable should be dropped and run the model again.

5. Fit the models and validate the models again

Below is the performance data for each model and it is normal that the performance has been decreased after removing the variable of last_fico_range_high.

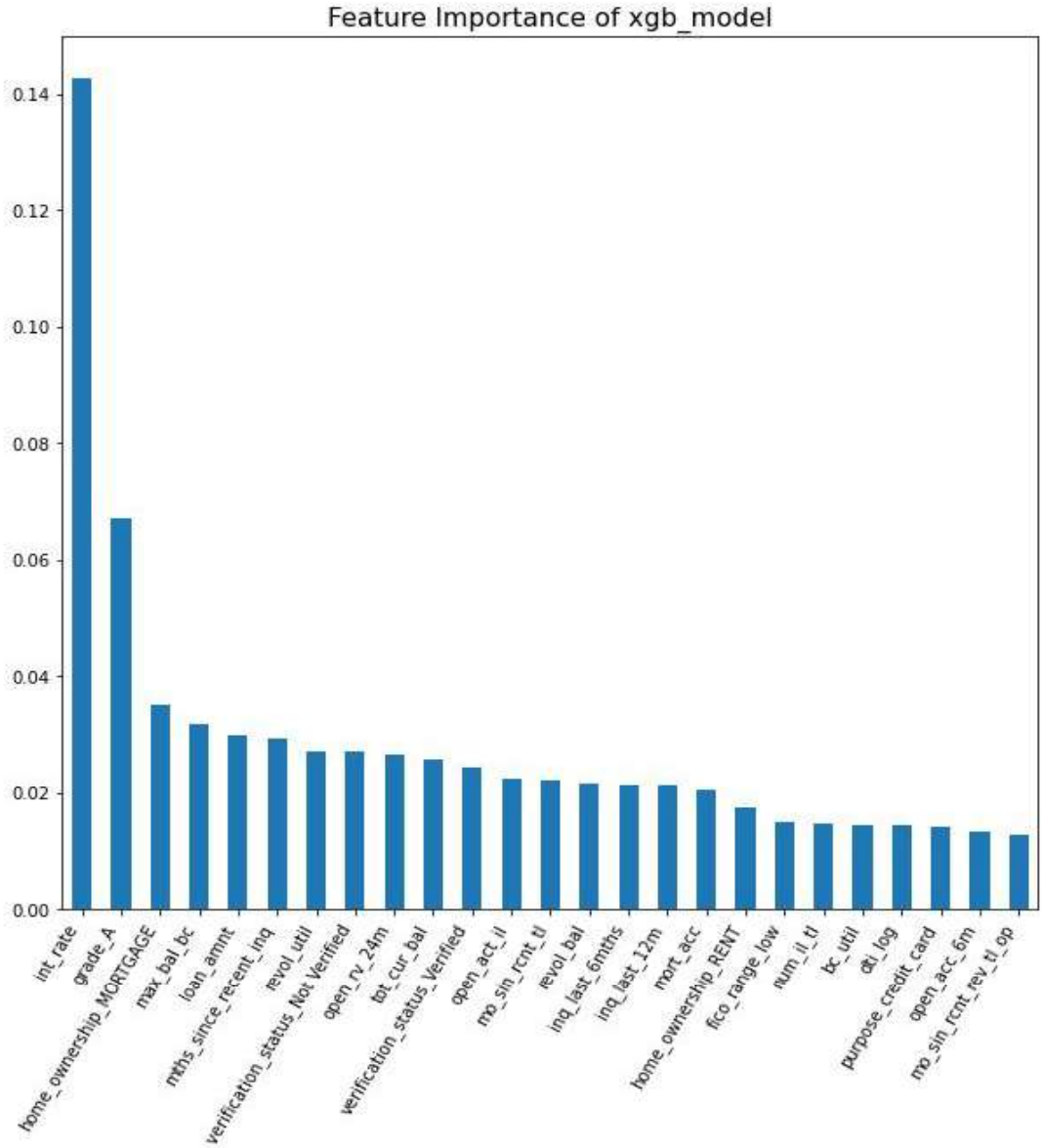
	t rain_auc	test_auc
logit_model	0.7012	0.7062
rf_model	0.7431	0.7024
xgb_model	0.7200	0.7116



Below is the table for classification_report of each model.

```
logit_model
              precision    recall  f1-score   support

           0       0.86      1.00      0.92     18536
           1       0.54      0.01      0.02      3037

    accuracy                           0.86     21573
   macro avg       0.70      0.50      0.47     21573
weighted avg       0.81      0.86      0.80     21573

rf_model
              precision    recall  f1-score   support

           0       0.86      1.00      0.92     18536
           1       0.00      0.00      0.00      3037

    accuracy                           0.86     21573
   macro avg       0.43      0.50      0.46     21573
weighted avg       0.74      0.86      0.79     21573

[15:29:13] WARNING: C:/Users/Administrator/workspace/xgboost-w
ault evaluation metric used with the objective 'binary:logisti
u'd like to restore the old behavior.
xgb_model
              precision    recall  f1-score   support

           0       0.86      1.00      0.92     18536
           1       0.75      0.00      0.01      3037

    accuracy                           0.86     21573
   macro avg       0.80      0.50      0.47     21573
weighted avg       0.84      0.86      0.80     21573
```

Feature Selection plot for Xgb_model



Conclusion

- From the data provided for the classification report, we decide to adopt the Xgboosting model as the final model.
- The mode feature selection, we know that the top 10 important features including int_rate, grade_A, home_ownership_mortgage, mac_bal_bc, loan_amnt, mths_since_reacent_inq, revol_util, verification_status_Not_Verified, open_rv_24m, tot_cur_bal.