Yi Ding

dingy21@wfu.edu

12/9/2022

# Executive Summary

## Problem

We are to examine a database with over 29,000 loan records to build and find a model that best identify default loans. By building, tuning and evaluating several models, we will identify the best model, and most significant factors that influence the status of a loan. Through this process, we aim to describe what a defaulted loan looks at, as well as false predictions and accurate predictions.

## Key Findings

* As interest rate, monthly payment owed by borrowers and amount of loan increase, they exhibit larger impacts on loan default predictions. For example, the largest impact is exhibited when interest rate exceeds 20%, and when loan amount reaches around 25000. On the other hand, although monthly payment owed by borrowers show similar trend, its impact grows at the fastest rate between 0 to 500
* Funded-amnt-inv exhibits completely different trend as previous variables. Its impact reaches the highest when it is close to 0, while its impact drops to close to 0 when it exceeds 22000. Other variables with similar impacts on loan default predictions are last payment amount and borrower’s annual income
* When interpreting top true positive predictions, last payment amount at 0 is the top contributor to a defaulted loan, followed by when home ownership is rent and interest rate over 10%. This means that a default loan is more likely with a last payment of 0, home ownership of rent and interest rate over 10%

## Model Performance Summary & Interpretation

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Model | Partition | AUC | Accuracy | Precision | Recall |
| Logistic Model | train | 0.8747990 | 0.8607686 | 0.5401181 | 0.4963294 |
| test | 0.8716126 | 0.8551601 | 0.5192012 | 0.5029762 |
|  |  |  |  |  |  |
| XGBoost Model | train | 0.9353594 | 0.8983352 | 0.6599369 | 0.6677306 |
| test | 0.9047125 | 0.8750839 | 0.5838235 | 0.5907738 |
|  |  |  |  |  |  |
| Random Forest | train | 0.9926225 | 0.9658878 | 0.8591340 | 0.9246728 |
| test | 0.8759145 | 0.8693754 | 0.5865103 | 0.4464286 |

The XGBoost Model is the best performing model out of the three models built. It contains the highest accuracy, at 0.8750839, meaning its over 87% accurate in making predictions for defaulted loans for the test dataset. This is verified with its test dataset AUC-ROC of 0.9047125. A model with higher AUC-ROC means the model is better at distinguishing the positive class from the negative. As an AUC-ROC approaches 1, it is better at classifying a true default as default, while predicting non-default as a non-default. With an AUC-ROC at over 90%, the model can accurately make correct predictions 90% of the times.

## Recommendations

* As any loan default will cause a loss for the financial institute, we should try to avoid loans whose borrower did not pay their last payment, did not own a house, and borrowed for debt consolidation, at interest rate higher than 10%
* Although, in some cases, a low amount of last payment and high interest rate may cause a false prediction that classifies a non-defaulted loan as default, a true loan default causes much more harm to the institution. Thus, it is recommended that we watch for loans whose borrowers pay less than 100 in their last payments with interest rates higher than 10%
* As mentioned above, it is reasonable for the institution to decrease its threshold in terms of determining default, to maximize its chance to identify a true loan default. This is because a true default causes much greater loss than a falsely predicted non-default loan. We recommend that the institution to operate the model at a threshold at 0.406, as the marginal increase in true positive rate reaches its maximum at this level and starts to decline beyond this threshold

# MODEL REPORT

# Detailed Analysis & Steps

### File(s) Summary

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| File Name | Record Count | Column Count | Numeric Columns | Character Columns |
| loan\_train.csv | 29777 | 53 | 31 | 22 |
| loan\_holdout.csv | 12761 | 52 | 31 | 21 |

For categorical/character data in loan\_train.csv

Table

Description automatically generated

Table

Description automatically generated



For numeric data in loan\_train.csv

Table

Description automatically generated

Table

Description automatically generated

Table

Description automatically generated



## Target Summary

|  |  |  |
| --- | --- | --- |
| Loan Status | n | pct |
| current | 25300 | 0.8496491 |
| default | 4477 | 0.1503509 |

Chart, bar chart

Description automatically generated

The chart and graph above show the frequency and percentage of different loan status. The default case only accounts for 15.04% of total records, meaning the default accuracy for this project is 84.96%.

## Exploratory Data Analysis & Screening

### Descriptive Statistics

Chart, bar chart, histogram

Description automatically generatedChart, histogram

Description automatically generatedChart, histogram

Description automatically generatedChart, histogram

Description automatically generatedChart, histogram

Description automatically generatedChart

Description automatically generated

Chart, histogram

Description automatically generatedChart

Description automatically generatedChart, histogram

Description automatically generatedChart, histogram

Description automatically generatedChart, histogram

Description automatically generatedChart, histogram

Description automatically generated

Chart

Description automatically generatedChart, histogram

Description automatically generatedChart

Description automatically generatedChart, histogram

Description automatically generatedChart

Description automatically generatedChart, treemap chart

Description automatically generatedChart

Description automatically generatedChart

Description automatically generatedChart

Description automatically generatedChart

Description automatically generatedChart

Description automatically generated with low confidenceChart

Description automatically generatedChart

Description automatically generated

### Correlation Analysis

Scatter chart

Description automatically generated with medium confidence

### Frequency Analysis

Chart, bar chart

Description automatically generatedChart, bar chart

Description automatically generatedChart, bar chart

Description automatically generated

Chart, bar chart

Description automatically generatedChart, bar chart

Description automatically generatedChart, bar chart

Description automatically generatedChart, bar chart, histogram

Description automatically generatedChart, bar chart

Description automatically generatedChart, bar chart

Description automatically generatedChart, bar chart, histogram

Description automatically generatedChart

Description automatically generated with medium confidenceChart

Description automatically generated with medium confidenceGraphical user interface, chart

Description automatically generatedChart, bar chart

Description automatically generatedChart

Description automatically generated with medium confidenceChart

Description automatically generatedChart

Description automatically generated

## Initial Screening & Exploration

The x-intercept line is used to show the threshold percentage of loan defaults in the target variable. A variable is more likely to have larger impact on predicting loan defaults when one category exceeds the default percentage of target. The following graphs indicate categorical variables with potentials to be predictive determinants.

A picture containing table

Description automatically generatedChart

Description automatically generatedChart, bar chart

Description automatically generated

Chart

Description automatically generatedTable

Description automatically generatedChart

Description automatically generated with low confidence

## Data Preparation & Transformation

Data Cleaning & Transformation

* Deleted the “%” in int\_rate and revol\_util and transformed them to numeric data
* Transformed empl\_year into character (word)
* Converted characters into factors, i.e., mutated loan\_status into factor

For numeric data:

* Removed unique identifiers such as id and member\_id as they are not useful for model generalization
* Removed variables with low complete rate such as mths\_since\_last\_delinq (36.5%), mths\_since\_last\_record (8.63%)
* Imputed missing values with median

For categorical data:

* Removed highly missing values such as next\_pymnt\_d (complete rate of 7.9%)
* Removed variables with high cardinality such as emp\_title, url, desc, title, zip\_code, addr\_state, earliest\_cr\_line, last\_pymnt\_d, last\_credit\_pull\_d
* Imputed missing values with unknown
* Dummy encoded
* Removed variables that are highly sparse and unbalanced

For recipe:

* Sample down the majority case (loan\_status = current) with under\_ratio = 3

According to the machine learning algorithm used (logistic, XGBoost, random forest), there is no need to scale or center data.

### Derive new variables

As mentioned above, in\_rate and revol\_util as numeric percentage are stored as character, we removed the “%” sign and converted them to numbers. Besides, issue\_d, last\_pymnt\_d, next\_pymnt\_d and last\_credit\_pull\_d are date records stored as character, and should be mutated to timestamp, but since we are not using them in the predictive model, we didn’t do the further transformation.

## Model Building

Data partitioning - split the data into 70/30 train/test split using random sampling

Graphical user interface, text, application

Description automatically generated

Text

Description automatically generated

3 models were built for predicting loan status – logistic regression, XGBoost, and random forest

## Model Training

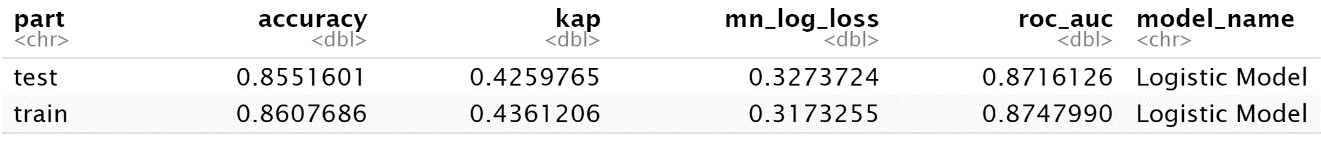
All models use the sample recipe

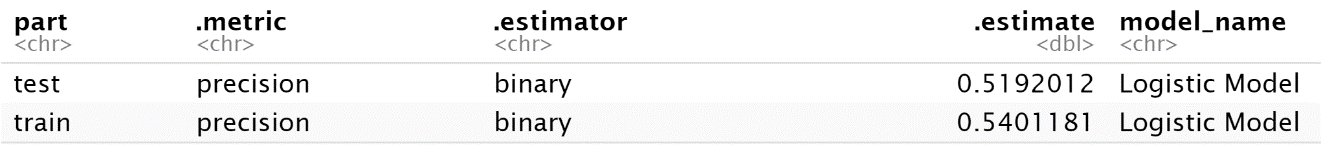
* Loan\_status ~ term + grade + sub\_grade + emp\_length + home\_ownership + verification\_status + pymnt\_plan + purpose + out\_prncp + out\_prncp\_inv + total\_rec\_late\_fee + last\_pymnt\_amnt + collections\_12\_mths\_ex\_med + policy\_code + acc\_now\_delinq + chargeoff\_within\_12\_mths + delinq\_amnt + pub\_rec\_bankruptcies + fico\_range\_low + fico\_range\_high + inq\_last\_6mths + mths\_since\_last\_delinq + mths\_since\_last\_record + open\_acc + pub\_rec + revol\_bal + revol\_util + total\_acc + loan\_amnt + funded\_amnt + funded\_amnt\_inv + int\_rate + installment + annual\_inc + dti + delinq\_2yrs + application\_type + empl\_year + tax\_liens

Logistic Regression Model

Text

Description automatically generated







Chart

Description automatically generatedChart

Description automatically generated

Chart, histogram

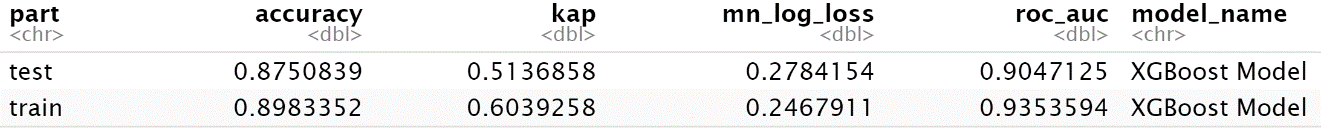
Description automatically generatedTable

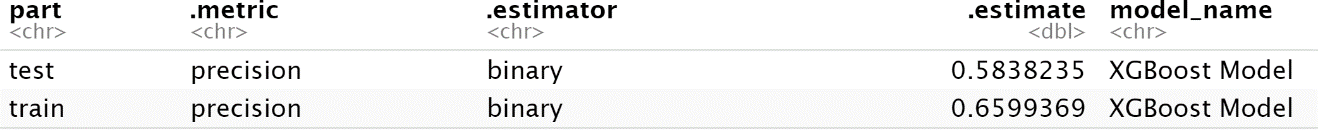
Description automatically generated

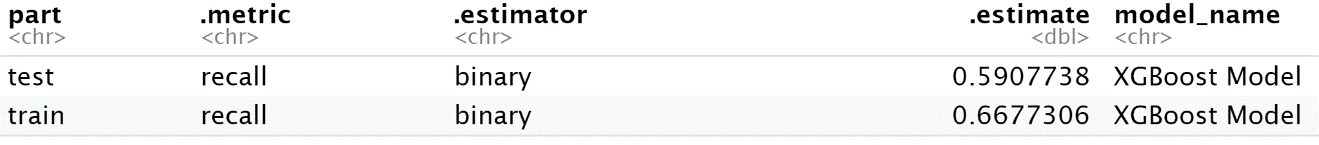
XGBoost Model

Text

Description automatically generated







Chart

Description automatically generatedChart

Description automatically generated with medium confidenceChart, histogram

Description automatically generatedTable

Description automatically generated

Chart, treemap chart

Description automatically generatedChart, treemap chart

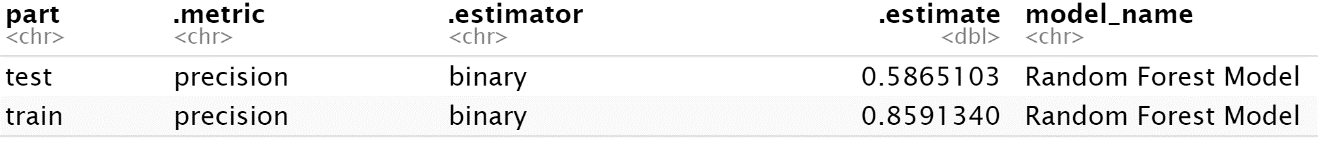
Description automatically generated

Random Forest Model

Graphical user interface, application

Description automatically generated

## 





Chart

Description automatically generatedChart

Description automatically generated

Chart, histogram

Description automatically generatedTable

Description automatically generated

## Model Comparison

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Model | Partition | AUC[[1]](#footnote-1) | Accuracy[[2]](#footnote-2) | Precision[[3]](#footnote-3) | Recall[[4]](#footnote-4) |
| Logistic Model | train | 0.8747990 | 0.8607686 | 0.5401181 | 0.4963294 |
| test | 0.8716126 | 0.8551601 | 0.5192012 | 0.5029762 |
|  |  |  |  |  |  |
| XGBoost Model | train | 0.9353594 | 0.8983352 | 0.6599369 | 0.6677306 |
| test | 0.9047125 | 0.8750839 | 0.5838235 | 0.5907738 |
|  |  |  |  |  |  |
| Random Forest | train | 0.9926225 | 0.9658878 | 0.8591340 | 0.9246728 |
| test | 0.8759145 | 0.8693754 | 0.5865103 | 0.4464286 |

According to chart above, the logistic model is least overfitting compared with other models, as the difference between its training and testing dataset is relatively small. The random forest model is the most overfitting one and its accuracy, precision and recall differentiate significantly between training and testing part. That’s probably due to its hyperparameters and the complexity of model recipe. To improve its performance, we should consider further tuning of its hyperparameters.

Overall, the XGBoost model performs the best with a highest testing auc\_roc of 0.9047, which means that the model is 94.16% accurate in distinguishing the default and current cases. A precision of 0.5838 tells that for all default cases we predicted, 58.38% of them are correctly identified; a recall of 0.5908 indicates that we can correctly identify 59.08% of all default cases that actually happened. With an accuracy of 0.875 in testing, indicates that we predict correctly for 87.5% of loan status.

As our best model only performs slightly better than the default accuracy, to improve its efficiency, we will continue to tune the hyperparameters of model algorithms and refine our feature selections. Besides that, we could also enhance model performance by leveraging threshold. According to the operating range above, FPR increases as the threshold decreases, while precision decreases and recall increases. FPR is the percentage of predicting an actual negative case as positive in terms of total actual negative cases. In this project, FPR means, in total actual current loans, the percentage of incorrectly identified a current loan as a default. As FPR increases, the possibility of accurately predicting default (TPR) increases, but the precision would be lower since we allow more inaccurate predictions. This is a tradeoff between precision and recall, if we want higher precision, we should lower the FPR (raise the threshold), the model will allow for more false negatives to slip by; if we want higher recall, then we should increase the FPR (lower the threshold), the model will allow for more false positives to slip by. (see graph below as reference)

A picture containing diagram

Description automatically generated

Table

Description automatically generated 🡨 example of XGBoost threshold range

More specifically, as FPR increases, recall increases more less. Therefore, when setting the threshold, a balanced selection should be made to maximize the return of both precision and recall.

To address the overfitting issues, we will consider using cross validation, adding regulations, and modifying model recipe for model maintenance and refinement.

## Partial Dependence Interpretation

Partial Dependence Plot (PDP) is a visualization of the features' impact on the predicted outcome, which shows the dependence between features and target response. Based on the model variable importance chart, we selected the most significant determinants and created PDP as shown below.

Chart, line chart

Description automatically generatedChart, line chart

Description automatically generatedChart, line chart

Description automatically generated

As we can see above, the predicted average impact grows as interest rate on the loan/monthly payment owed by borrowers/amount of loan increases. For interest rate, its average impact will reach the highest as it exceeds 20%. When monthly payment owed by borrowers is between 0 to approximately 500, its average impact increases the fastest. The impact of loan amount steps up the most as it reaches approximately 25000.

Chart

Description automatically generatedChart, line chart

Description automatically generatedChart, line chart

Description automatically generated

Application

Description automatically generated with low confidence

last\_pymnt\_amnt as the most important variable for all three models, its average impact is only significant when the payment amount is small, once last payment is over around 3000, its impact bounce back to almost 0 – meaning that a last payment amount over 3000 is not a determinant for detecting loan default.

The fluctuation of the average impact of total\_rec\_late\_fee is caused by outliers or anomalies between 10 to 25 (approximately). It seems to be an important predictor as its average impact tends to be stable around 0.40 to 0.42 with a late fee received to date over 45.

The average impact of prediction drops down as funded\_amnt\_inv increases and close to 0 when funded\_amnt\_inv is more than 22,000, which means for records with a total amount of committed by investors higher than 22,000, funded\_amnt\_inv is no longer a decisive predictor for identifying loan default.

Borrower’s annual income should be an effective differentiator in predictive models; however, the steep drop down is due to the extreme outliers. To gain a more comprehensive understanding of its impact, we should remove anomalies that with an extremely high annual income.

## Top Predictions

### Top 10 True Positives

True positive prediction means we accurately predict a loan default as default. The rank order is the descending predicted possibility of being default (from highest to lowest).

Table

Description automatically generated

Chart, timeline

Description automatically generatedChart

Description automatically generated

Graphs above are the breakdown plot of top 1 and top 2 true positive predictions, visualizing the contribution details of useful features. As we can see, last\_pymnt\_amnt = 0 is a significant contributor to the correct default prediction. Besides that, interest rate is also important for accurately identifying default case. Compared with other top true positive predictions, purpose of debt\_consolidation and interest rate over 10% seem to be a sign of loan default.

### Top 10 False Positives

Table

Description automatically generated

Chart, waterfall chart

Description automatically generatedChart

Description automatically generated

False positive prediction means we predict an actual current case as default. As we can see above, last\_pymnt\_amnt contributes the most, followed by total\_rec\_late\_fee, int\_rate and term. last\_pymnt\_amnt makes less contribution to total prediction as it increases, when it reaches 68.01 for the 1st false positive case, last payment amount only accounts for 0.261 of a total predicted probability of 0.944. In comparison with a contribution of 0.589 in the 2nd case, its impact decreased by 35%. Based on the top 10 false positive cases, we found that the XGBoost model may not be efficient for predicting cases with high last payment amount, home ownership of mortgage and loan term of 60 months.

### Top 10 True Negatives

Table

Description automatically generated

Chart, timeline

Description automatically generatedChart

Description automatically generated

### Bottom 10 False Negatives

Table

Description automatically generated

Chart, timeline

Description automatically generated with medium confidenceTimeline

Description automatically generated

False negative prediction means we predict a default case as current. Above are the cases with high probability of current but occur to be default. As we can see, high last\_pymnt\_amnt starts to have negative impact to score predictions while other variables contribute very few. This tells that even though a borrower pays a large amount of money for last payment, they could also be default. Our model tends to conclude low last payment amount as more likely to default, therefore, for borrowers with high last payment amount, it will decrease the probability of default, result in inaccurate predictions.

## Anomaly Detection

Set up an isolation recipe and removed unique identifiers and highly missing values, impute missing numeric with median, impute missing nominal with unknown, dummy encode nominal variables and assign previously unseen factor level to a new value.

Graphical user interface, text

Description automatically generated

Graph below shows the hyperparameter of isolation forest

Graphical user interface, text, application

Description automatically generated

Chart below indicates min, max and average depth of tree in isolation and min, max and average anomaly score



### Global Anomaly Detection

Chart, histogram

Description automatically generatedChart, histogram

Description automatically generated

Based on the distributions of average tree depth and anomaly score above, most trees have an average depth between 7.5 and 7.9 with an average anomaly score between 0.59 to 0.60. Thus, we select average\_depth ≤ 7.4 or anomaly score ≥ 0.607 as the anomaly flag then apply to the recipe.

By setting the flag of anomaly (average\_depth ≤ 7.4), we detected 236 cases of anomaly with 29541 cases identified as normal. The anomaly rate is 0.793%.

Background pattern

Description automatically generated

Top 10 Anomalies

Table

Description automatically generated

Then we fit a tree to separate anomalies and normal cases.

Graphical user interface, text, application, email

Description automatically generated

Text

Description automatically generatedDiagram

Description automatically generated

In this tree, node 2 indicates that for addr\_state\_AZ < 0.5, 99.38% of them are normal, but it also contains 180 cases of anomaly (76.27% of total anomaly cases), which means it is not a good indicator for differentiating anomaly and normal. If int\_rate ≥ 18.63 then last\_pymnt\_amnt ≥ 692.905 could be important filter for detecting anomaly as 73.08% of its components tend to be anomalies. Rules below are derived from this tree.

A picture containing chart

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Here coverage of 0% means anomalies are very few cases.

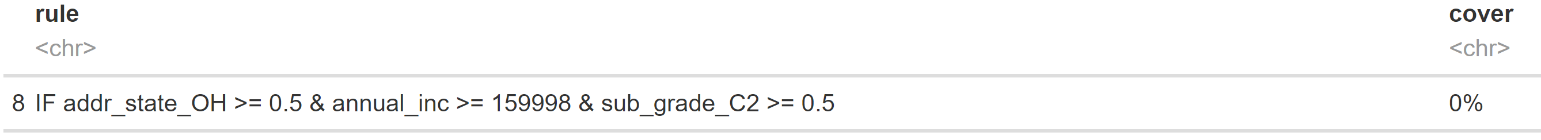
### Local Anomaly Detection

We identify a record with average depth ≤ 7.4 and anomaly score ≥ 0.607 (id = 28010) as an anomaly then use its features as an anomaly rule to fit a tree

Text

Description automatically generatedTimeline

Description automatically generated



As we can see above, following the anomaly rule, only one record was detected as anomaly, which means the feature of the record used as the anomaly rule is not general enough to include more cases. But from this tree, we could tell that in the sample of 913 (addr\_state\_OH ≥ 0.5), only 2.41% of them have an annual income higher than 159,998, the annual income of the rest 891 people (accounts for 97.6%) is lower than 159,998. Therefore, annual income ≥ 159,998 could be an effective filter for detecting anomalies and a signal of extreme outliers which should be removed in the further refinement of the model.

1. AUC\_ROC entails how capable the model is at distinguishing between results, being true negative, false negative, true positive, or false positive. As ROC\_AUC approaches 1, the model is more likely to predict a positive as positive, and a negative as negative. [↑](#footnote-ref-1)
2. Accuracy is one metric for evaluating models, it measures what percentage of the total sample did we predict accurately. [↑](#footnote-ref-2)
3. Precision is a percentage of true positive among all predicted positive cases. [↑](#footnote-ref-3)
4. Recall is a percentage of true positive among all actual positive cases. [↑](#footnote-ref-4)