An Optimal Investment Strategy of Lending Club

IBM Practicum Project Spring 2020

Group Members:

Zihan (Zilean) Chen, Yulong (Michael) Wang,

Chenyi (Leo) Yang, Jie (Jay) Qian,

Jiaqi (Ivan) Su, Heze Liu

Sponsor:

Alan King

Supervisors:

Morton Lane, Matthew D Murphy

Contents

Abstract

The main idea of this project is that we use data mining techniques to extract effective information from real-life databases and utilize this information to form investment strategies of a synthetic retail bank. By analyzing the loan information of Lending Club database, we developed several machine learning classifiers and implement them with various portfolio management techniques in our simulation of 40 investment cycles. Built upon the previous work of the IBM team, we have improved and redesigned the data preprocessing process, default detection models, investment mechanism, portfolio management strategies and simulation parameters. In addition, we have experimented adding more features and factors to our algorithm, such as prepayment detection models. After comparing multiple sets of parameters, we conclude that XGBoost model using under sampling technique generates the best performance. Even though actively portfolio management brings little benefits in our simulation, it will require further study on tuning the parameters to confirm its ineffectiveness.

Background

Lending Club is an American peer-to-peer lending company. It enables borrowers to create unsecured personal loans between $1,000 and $40,000. The standard loan period is three years and can be extend to five years. Investors can search and browse the loan listings on Lending Club website and select loans that they want to invest in based on the information supplied about the borrower, amount of loan, loan grade, and loan purpose. Investors make money from interest. Lending Club makes money by charging borrowers an origination fee and investors a service fee.

Lending Club database contains millions of loans, in this project, we use these data to dig up and analyze the behavior characters of borrowers, train different machine learning models, such as Random Forest, Logistic Regression, neural network, to estimate the prepay and default probability of each loans.

With these models, we simulate a retail bank, experiment with optimization approaches, implement different dynamic investment strategies and analyze outcomes, to guide the bank's lending and borrowing strategies over time, and to report the risk and rewards of our experiments.

Data Preprocessing

1. Database and Its Distribution

The raw Lending Club (LC) database details 2,260,668 loans issued by Lending Club from June 2007 to December 2018. The loan status is shown in Table 1-1.

Table 1-1 Lending Club Loans Description

Loan Status	Fully paid	Charged off	Late (16-30 days)	Late (31-120 days)	Current
Num of Loans	1,041,952	261,655	3,737	21,897	919,695
Loan Status	In Grace Period	Default	Violation of Credit Policies. Status: Charged Off	Violation of Credit Policies. Status: Fully Paid	
Number	8,952	31	761	1,988	

Obviously, 'Current', 'Fully paid' and 'Charged off' loans account for the most important part of the data set.

Their yearly distributions are shown in Figure 1-1, 1-2, and 1-3.

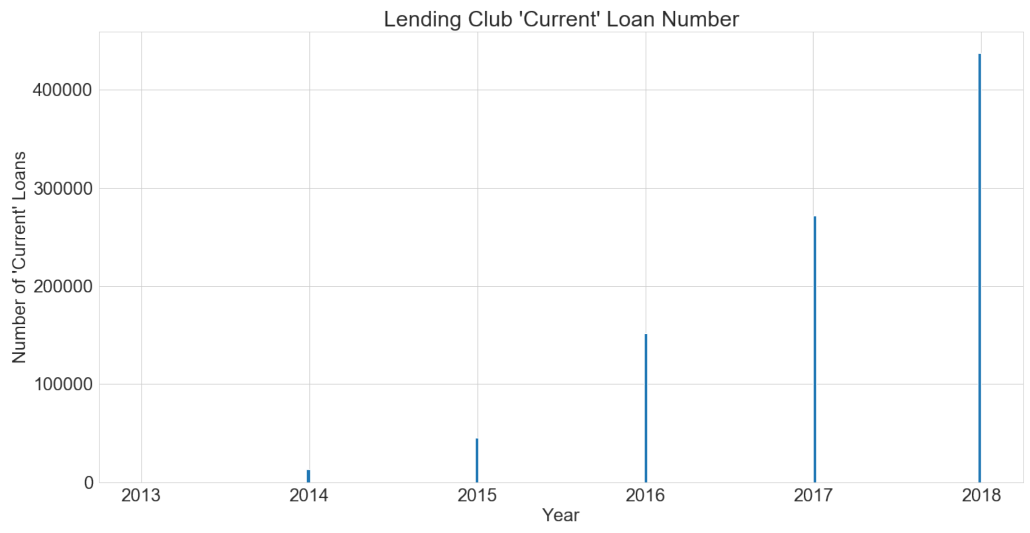


Figure 1-1 The Distribution of "Current" Loans in LC dataset

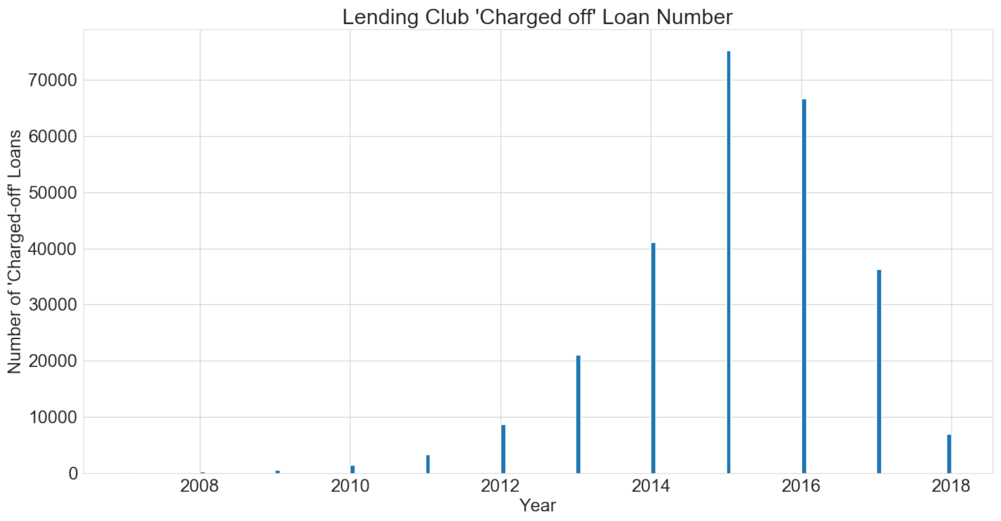


Figure 1-2 The Distribution of “Charged-off” Loans in LC dataset

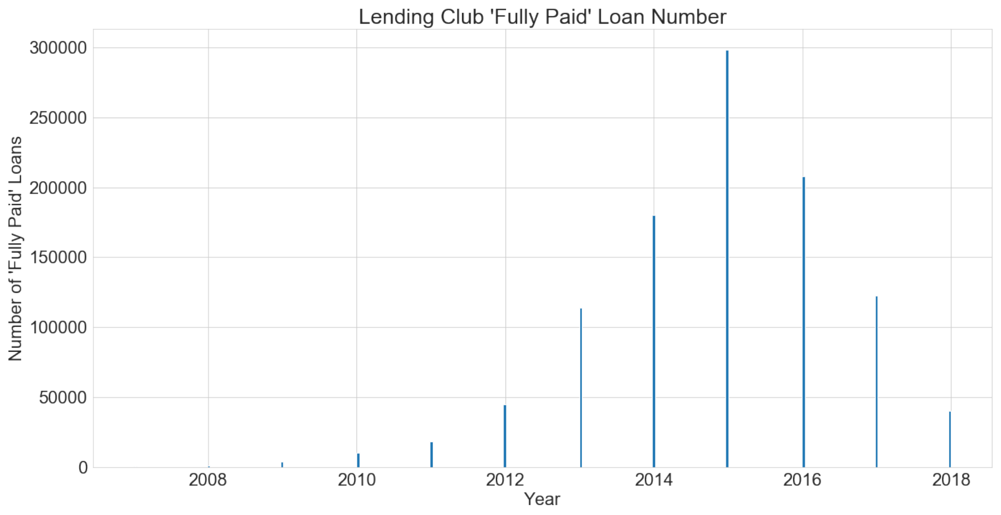


Figure 1-3 The Distribution of “Fully-paid” Loans in LC dataset

2. Data Description

From the Lending Club database, we can further extract the characteristics of loans with different maturities. In this project, we only took “Charged off” loans and “Fully Paid” loans into account. The number of issued loans and the default rate can be found in Figure 2-1, 2-2 and 2-3. As we have seen from the previous chart, we have a large proportion of loans in “Current” state after 2015, so the downward trend in Figure 2-3 is not surprising because we have many loans in pending state. Overall, the default rate on loans issued each month fluctuates around 0.2. It means that there is no significant correlation between time and the choice to default. Particularly, the characteristics that we derived (Loss Given Default) are shown in Table 2-1.

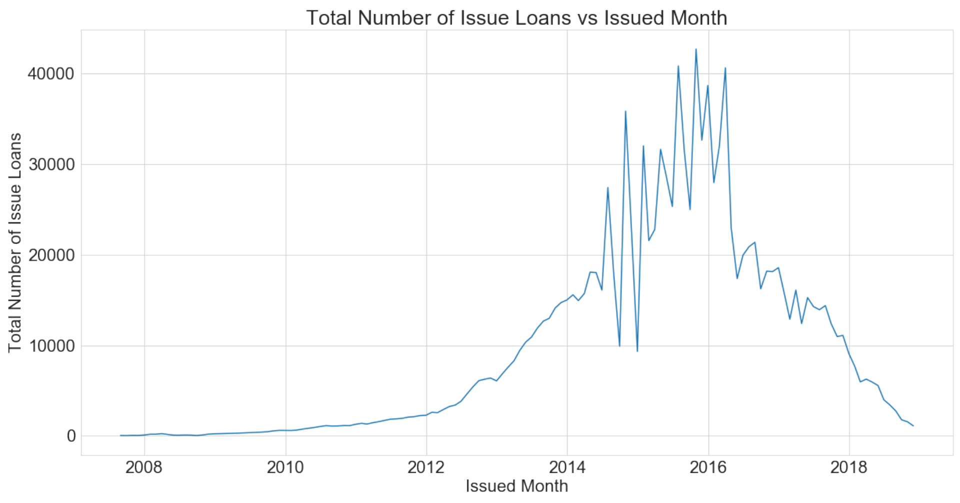


Figure 2-1 Issued Loan Number Each Month

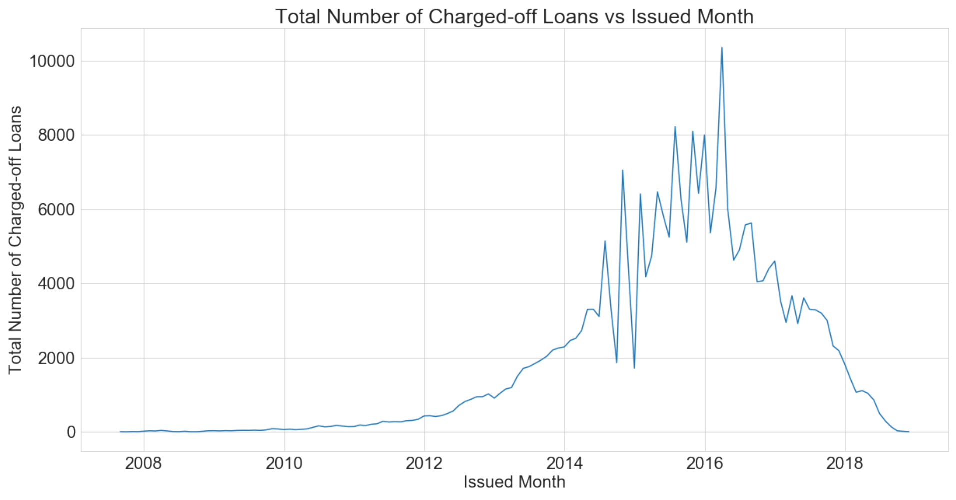


Figure 2-2 Number of Charged off Loans Each Month

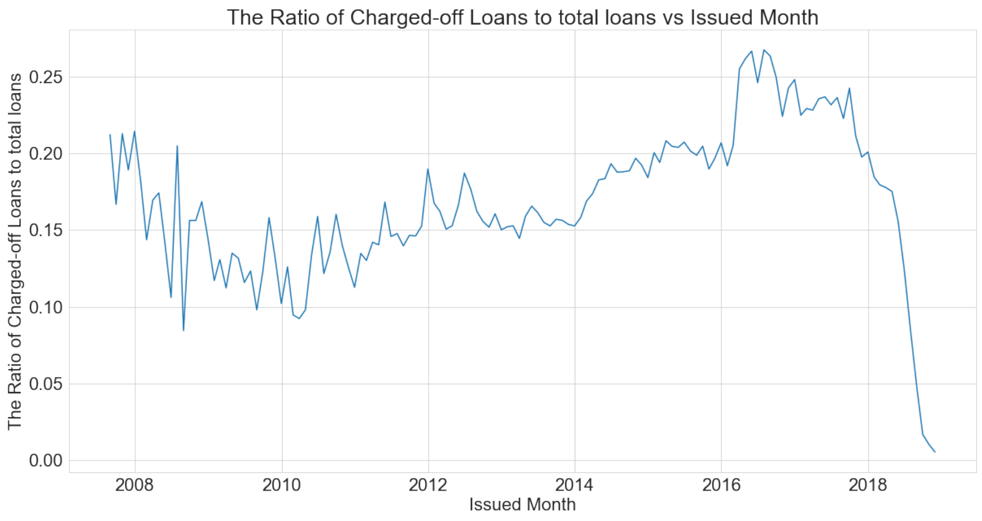


Figure 2-3 Default Ratio Each Month

Table 2-1 Loss Given Default of Lending Club Dataset

	LC Dataset	Charged-off Loans	Charged-off within 12m	Charged-off within 12m (36m loans)	Charged-off over 12m (60m loans)
Num of Loans	1,301,293	261,655	111,333	70,942	40,391
Total Loan Amnt	18,759,209,700	4,034,413,275	1,732,306,900	905,925,975	826,380,925
Average Loan Amnt	14,415.823	15418.8	15559.7	12769.95	20459.53
Avg Default Rate	0.1993	1	1	1 (0.1595 given 36m loans)	1 (0.3242 given 60m loans)
Total Loss	/	-1,934,44,3652	-1,229,525,826	-621,161,336	-608,364,489
Avg Loss	/	-7457.779	-11043.678	-8755.9	-15061.88

We found that for defaulted loans, the average loan amount is slightly higher than the average for the entire database, which means as the loan amount increases, the average default risk page increases as well. The average of default loans within 12 months is not much different from that of total default loans, indicating that the default situation is not affected by the life period. If we look at loans with terms of

36 months and 60 months separately, we will find that for loans with terms of 60 months, both the default probability and the average loan amount are much higher than those with terms of 36 months. In future simulation, are we willing to take a 60-month loan into our portfolio and take a higher risk, or a more robust 36-month loan? We will show the results in the simulation section.

3. Data Cleaning

This database shows that each loan contains up to 145 characteristic values such as “loan amount” and “issued month”. To facilitate the subsequent machine learning model, we did the following transformation:

3.1 Drop Blank Records

In order to avoid the negative impact of excessive zeroing on subsequent machine learning model training, we will delete a total of 58 features that exceed 30% of the total data amount, such as “member id” and “url”.

To ensure the integrity of the subsequent model training, we deleted 0.18% of the missing date-related data from the total data.

3.2 Date Encoding

To ensure the convenience of subsequent model training, we took the month as the minimum unit and converted date-type data into numeric types. Therefore, June 2007 corresponds to date variable 0, February 2019 corresponds to date variable 140.

3.3 Important Features Extraction

We created 4 new variables, named “TotalGain”, “PvGain”, “LoanStatus” and “LifeOfLoan”, where “TotalGain” represents the total profit/loss we sustained from investing the loan at the end of the period. Its calculation formula is as follows:

$$\begin{aligned} Total\ Gain = {} & Total\ Received\ Amount + Total\ Received\ Interest + Recoveries \\ & - Funded\ Amount - Recoveries\ Collection\ Fee \end{aligned}$$

The variable “LoanStatus” was set to indicate whether the loan is fully paid or charged off, where ‘0’ represents ‘Fully paid’ and ‘1’ represents ‘Charged off’.

“LifeOfLoan” is the continuous life cycle of the loan, which is calculated as follows:

$$Life\ Of\ Loan = \ Last\ Payment\ Date - Issued\ Date$$

“PvGain” means that we discount the final yield of the loan to the value of the month of issue at a given discount rate.

3.4 Object-typed Features Encoding

We converted the following Features “term”, “initial_list_status”, “application_type”, and “disbursement_method” to ‘0’ and ‘1’ since these features have only two status.

Especially, we made a special transformation of the Features “grade” and “home_ownership” since they are intuitively important but not considered in the previous classifier. The transformation methods are shown in Table 3-1.

Table 3-1 Encoded Dictionary for ‘grade’ and ‘home_ownership’

Feature ‘grade’	Encoded Number	Feature ‘home_ownership’	Encoded Number
A	7	"MORTGAGE"	6
B	6	"RENT"	5
C	5	"OWN"	4
D	4	"OTHER"	3
E	3	"NONE"	2
F	2	"ANY"	1
G	1		

3.5 Imputer

For encoded datasets, we set simple imputer to fill up those N/A blanks. By using the package ‘SimpleImputer’ from “sklearn.imputer”, we filled the blanks by the mean value.

Finally, we got the filtered dataset for prepayment prediction, default prediction and simulation. The filtered dataset has 1,301,293 rows and 84 features.

Prepayment Prediction Model

Prepayment stands for the settlement of a debt or installment loan before its official due date. The purpose of our project is to simulate a retail bank based on the Lending Club database, so as to verify whether our investment strategy is reasonable. Therefore, the situation of prepayment of the loan will have a certain impact on the amount of cash flow we simulate. In this project, we added the prediction of loan prepayment in advance and applied it to the simulation process to detect whether loan prepayment has an impact on our equity, income and loss.

1. Data Visualization

As can be seen from Figure 4-1, prepay is a common phenomenon among loans issued by lending club -- almost 80% of loans issued each month since 2010 will be prepay. In particular, because for the current database, the data runs through February 2019, this means that a lot of data has been in current state since at least 2016. Therefore, we will find that from March 2016 onwards, the prepayment ratio is 100%. After that, as the data is updated, we believe that the prepayment ratio will still fluctuate around 80%.

By analyzing figure 4-2, 4-3, 4-4 and 4-5, we can get that the prepayment is more obvious in the 60-month loans -- 75.8% of the loan will be repaid within 30 months. Through further data analysis, we found that for a 60-month loan, more than 50% of the loan will choose to pay off within 12 months; but for a 36-month loan, it is possible to pay in advance for each installment. We think it depends on two things: the average loan amount and the risk aversion of the borrower. As we can see from the second part “Data Preprocessing”, the average loan amount of 36 months is significantly less than that of 60 months, which means it is easier for the borrower to pay off the debt. Moreover, because Lending Club's main customer base wants a small sum of money in the short term to balance its books, 60 months is too long for lenders -- when they are able to pay off their debts, the obvious manifestation is prepayment.

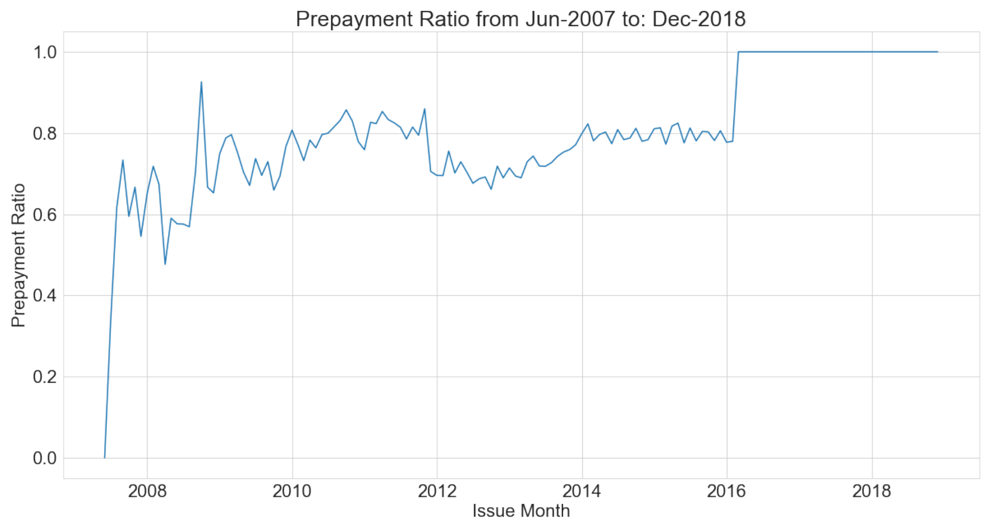


Figure 4-1 Prepayment Ratio Each Month

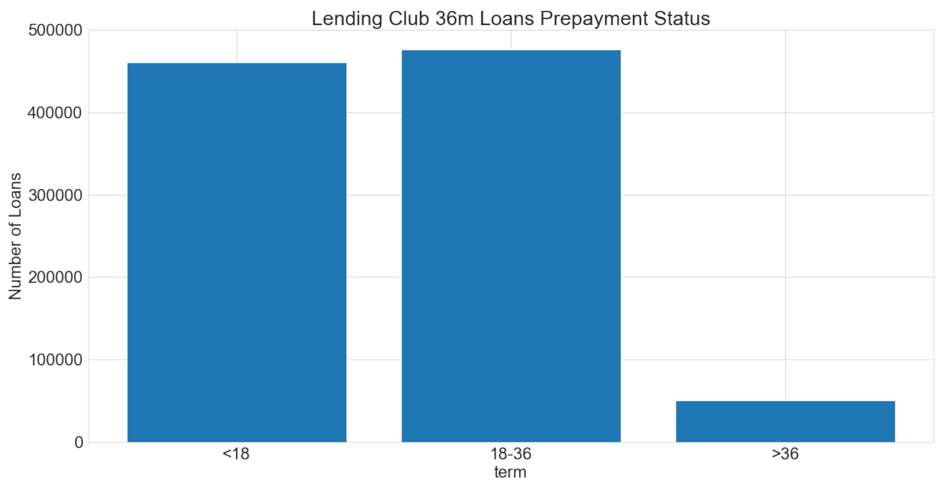


Figure 4-2 Histogram of 36m Loans

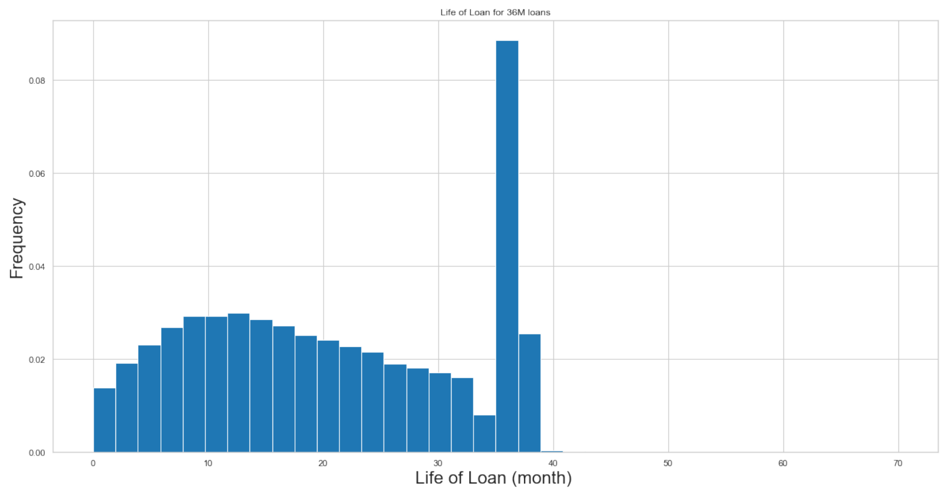


Figure 4-3 Histogram of 36m Loans

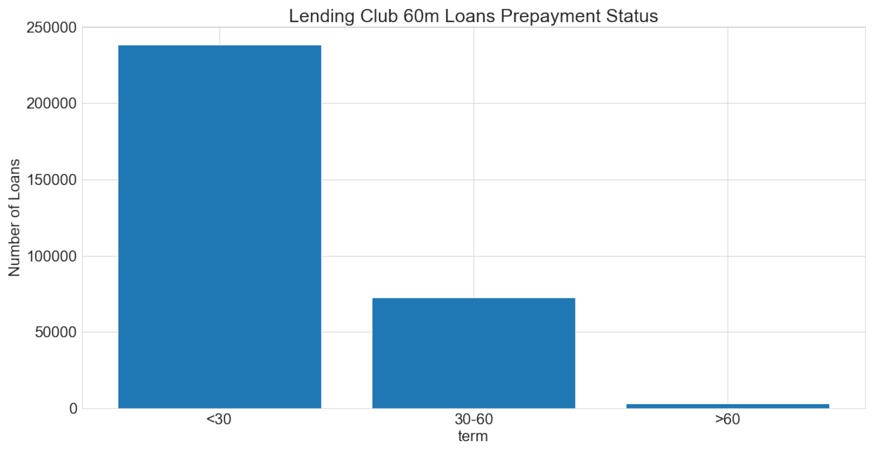


Figure 4-4 Histogram of 60m Loans

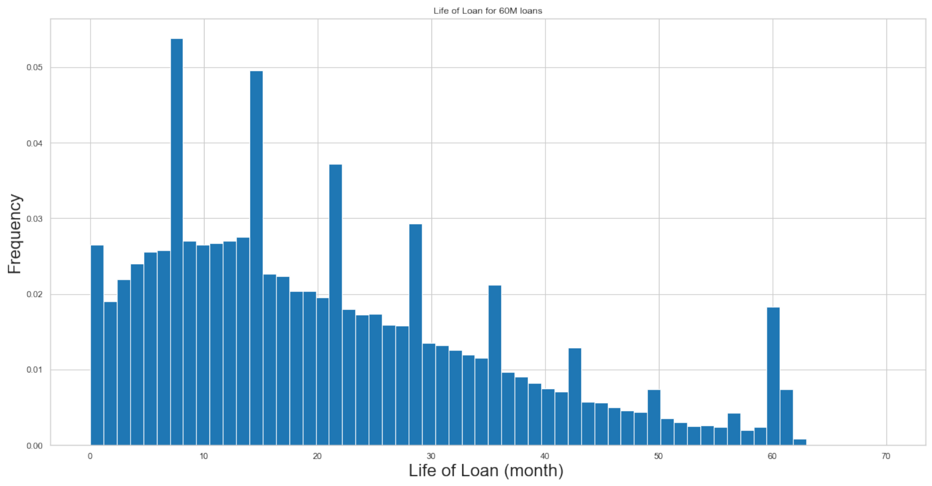
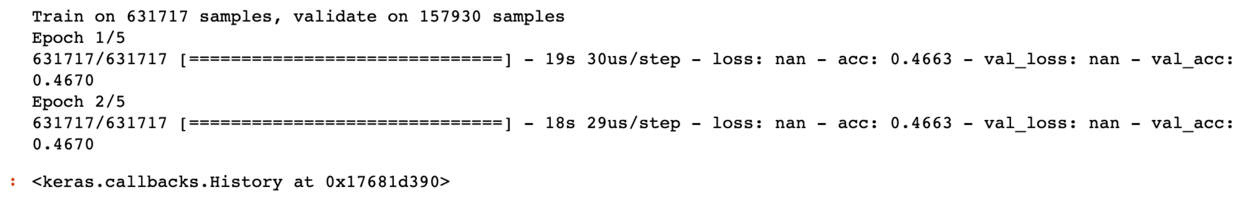


Figure 4-5 Histogram of 60m Loans

2. Neural Network

After discussion, we thought the change in the risk-free interest rate may affect the decision of the prepay, so want to simulation process into a chronological simulation, so that we can according to the time to add in the loan life cycle interest rate changes. However, our experiments showed that people's decision to make prepayments is not directly related to the current interest rate.

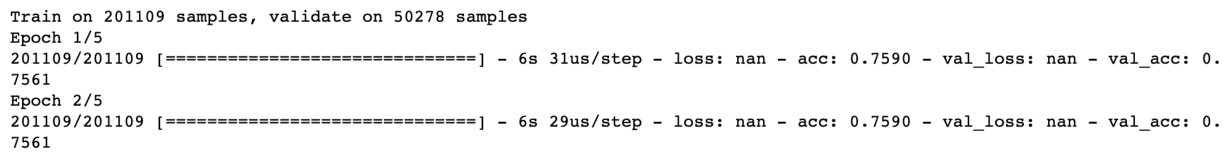
Based on the previous work done by the IBM team, we selected the 29 features with the highest contribution after PCA (Principal Component Analysis) to make the prepayment prediction. Since the 36-month loan and the 60-month loan have different performances in the prepayment situation, we decided to separate them during the training. According to the outcomes in Figure 5-1, 5-2, we ended up with an accuracy of 46.63% for the 36-month prepayment and 75.9% for the 60-month loan.



```
Train on 631717 samples, validate on 157930 samples
Epoch 1/5
631717/631717 [==============================] - 19s 30us/step - loss: nan - acc: 0.4663 - val_loss: nan - val_acc:
0.4670
Epoch 2/5
631717/631717 [==============================] - 18s 29us/step - loss: nan - acc: 0.4663 - val_loss: nan - val_acc:
0.4670

: <keras.callbacks.History at 0x17681d390>
```

Figure 5-1 36m Loans Prepayment Prediction Results (Early Stop = 2)



```
Train on 201109 samples, validate on 50278 samples
Epoch 1/5
201109/201109 [==============================] - 6s 31us/step - loss: nan - acc: 0.7590 - val_loss: nan - val_acc: 0.
7561
Epoch 2/5
201109/201109 [==============================] - 6s 29us/step - loss: nan - acc: 0.7590 - val_loss: nan - val_acc: 0.
7561
```

Figure 5-2 60m Loans Prepayment Prediction Results (Early Stop = 2)

3. Conclusion

We applied this process to the final policy section, changing the process of choosing a loan so that we were more likely to choose a loan that would not default. It turned out that our gains didn't increase, and they also fell sharply after we turned down many loans. We attribute it to the following reasons:

1) There are not many features in the Lending Club database that describe prepayment behavior. Based on the features with very low correlation, we tend to predict the same result, resulting in the accuracy of the prediction is similar to the original proportion of the data.
2) Prepayments occur for most loans in the Lending Club database. In the simulation process, if we choose the loans that do not prepay, it means that we greatly reduce the range of our choices. As a result, we don't have a lot of options. The interest we had to pay was close to the interest and principal we received, so the yield didn't match our expectations.
3) Reviewing our strategy, we made a new round of investments based on the balance sheet each period. Prepayments affect the cash flow of interest by shifting the time series only. Hence, it has little impact on the all-in strategy of each phase.

In a conclusion, prepayment behavior did not have the dramatic impact on the simulation process as we expected so that we choose not to apply its outcome in our investment strategy.

Default Prediction Model

The key point of this project is how the results of historical data analysis can be applied to the prediction of future gains and losses, and as close as possible to real life situations through our established strategies. In this chapter, we will discuss the use of machine learning to analyze historical data, so as to select loans in the simulation section.

The previous work of the IBM team was to convert all the features into integer or float forms and predicted the loan status using the Random Forest Classification. The model brought 79.91% accuracy to the forecast. When the model is applied to the subsequent simulation section, we find that income does not rise steadily. We believe that this is due to the following reasons:

1) Missing the balancing part to the database. In the original Random Forest Classification, we input two loan states with the proportion close to 4:1. This leads to a bias in the prediction.
2) Redundant dummies. The previous classification converted all the “string” type features into dummies and only deleted those features that could not be detected when issuing loans. We need to do the PCA and eliminate highly related features.
3) The Confusion Matrix was not ideal enough. Although we got nearly 80% prediction accuracy, our model was more inclined to predict the loan status as fully paid. Therefore, it was easier for us to wrongly select the loan charged off in the subsequent simulation process, which brought us greater losses.

1. Exploratory Data Analysis

When we did the PCA for the previous core feature set, we found it strange to have an equally weighed PCA (as it shown in Figure 6-1). The reasons of this result are as follows:

1) Redundant Features. As features increase, it was inevitable for machine learning to assign weight to each feature. As a result, those important features do not contribute significantly to the classification.
2) High Correlation between features. If the characteristics we input are highly correlated, the accuracy will largely decrease.

At the same time, too much data input will increase the computation time in the simulation process. Therefore, we came up with the idea to revise the process of data preprocessing, reduce the number of features as much as possible, and ensure that each feature has enough contribution to the accuracy of the model.

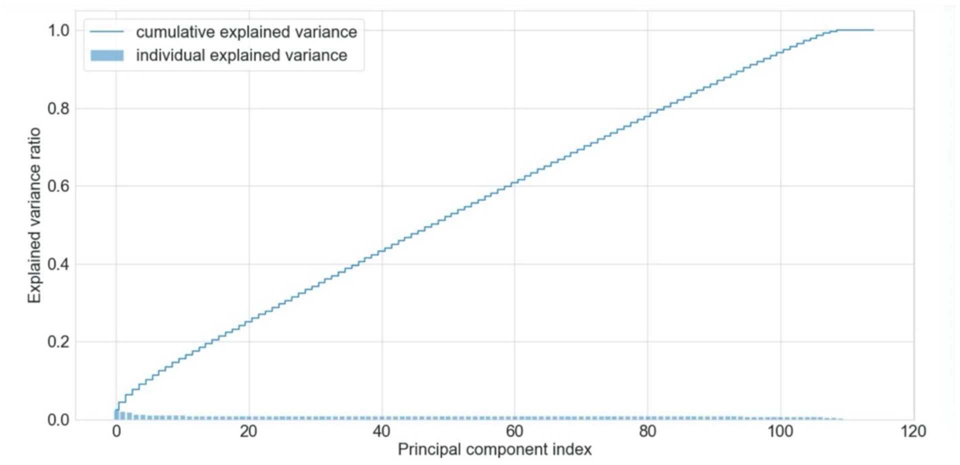


Figure 6-1 PCA for The Core Feature Set

We first removed the feature "states" and its dummies. In the original version done by IBM, feature “states” were separate to 51 sub-features (more than 1/3 of the entire feature set).

As shown in Figure 6-2, 6-3, we have classified the loans issued to different regions. The results show that although the number of loans issued varies from state to state, there is no significant difference in default rates (close to 0.2). Hence, we deleted all the state-related features in the data preprocessing part.

We did the similar transformation for feature “grade” and “homeownership” (please refer to Table 2-2). Finally, we selected 16 features (not 17 features because we will not know features “last payment amount” when we issued loans).

After we did the previous preprocessing, we will get the most important features for the default prediction. The correlation heatmap and the PCA results are shown in Figure 6-4, 6-5.

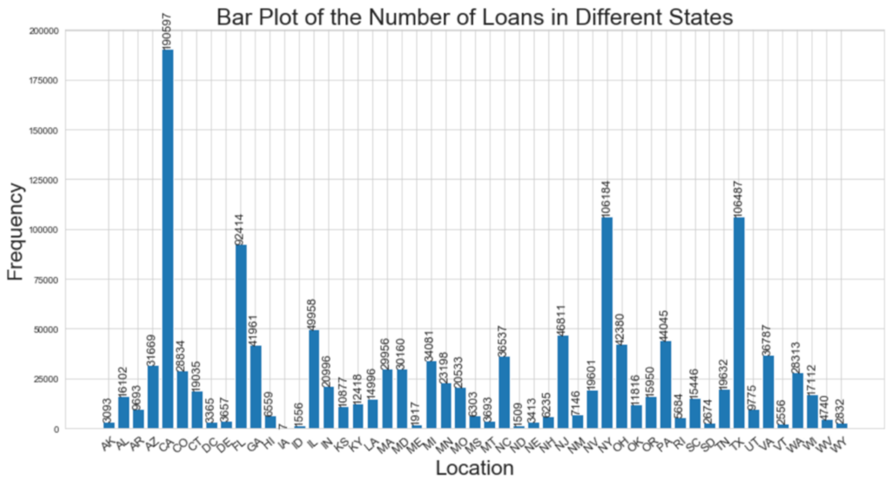


Figure 6-2 Number of Loans Issued in Different States

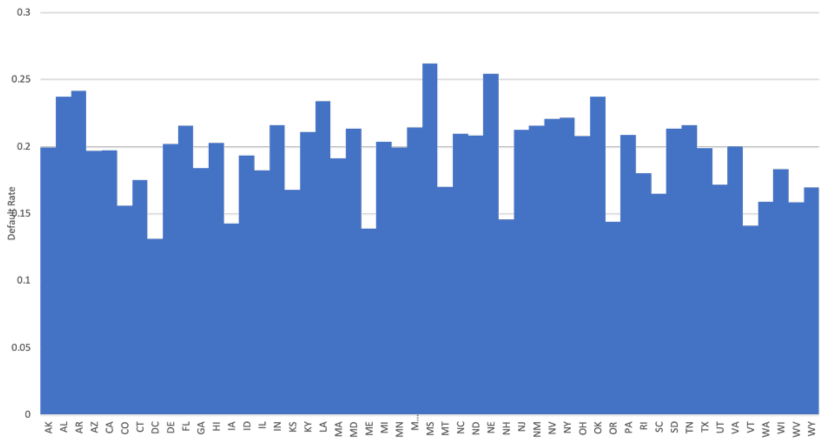


Figure 6-3 Default Rate of Loans in Different States

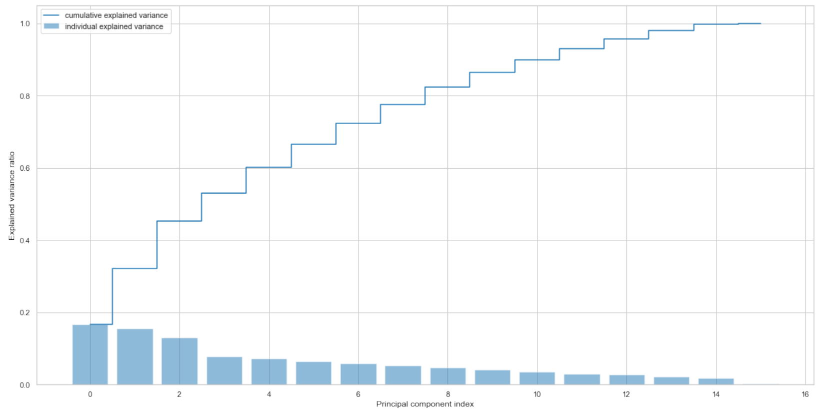


Figure 6-4 PCA for The Minimum Feature Set

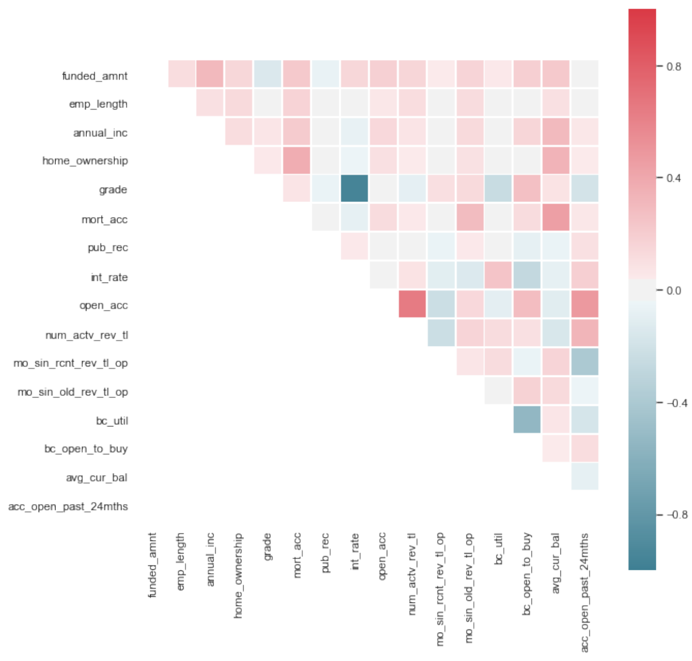


Figure 6-5 Heatmap of The Minimum Features Set

2. Different Probability of Default Models

To visualize our training process, we compared four models we trained with the models made by the IBM team. As it shown in Table 7-1, when we reduce the number of features, the accuracy of the model's prediction is improved. Also, the simulation part costs us much less time than before and we can get our graphs, Balance sheet and Income Statement within 3 minutes. Especially, even though the under sampling XGBoost model have a lower accuracy, it largely improves the recall score to 0.68. What we need in this model is accurate detection of defaulted loans so that our losses can be reduced. Hence, we chose the under-sampling XGBoost model for the final simulation part.

For clarification, even though the feature “last_payment_amount” has a huge contribution to our classification accuracy (XGBoost: 0.85 with 0.9 Recall Score), we finally deleted this feature since it violates our simulation part. It is a feature that we cannot detect when we issued a loan.

Table 7-1 Machine Learning Result Comparison

Model	IBM Random Forest	Random Forest	Logistic Regression	XGBoost	Undersampling XGBoost
Training set size (features)	(553049, 115)	(553049, 16)	(553049, 16)	(553049, 16)	(200000, 16)
Training set size (label)	(553049,1)	(553049,1)	(553049,1)	(553049, 1)	(200000, 1)
Testing set size (features)	(97597, 115)	(97597, 16)	(97597, 16)	(97597, 16)	(97597, 16)
Testing set size (label)	(97597,)	(97597, 1)	(97597, 1)	(97597, 1)	(97597, 1)
Accuracy	0.7991	0.7993	0.8002	0.8025	0.6376
Confusion Matrix	[[76486,1654] [17952,1505]]	[[76503,1624] [17960,1510]]	[[77985, 142] [19362, 108]]	[[76860,1267] [18008,1462]]	[[**49129**,28998] [6369,**13101**]]

Dynamic Investment Strategy

1. Bank Simulator

When we first approached this project, we have determined that the simulation process would play a unique role that separated us from many similar projects based on Lending Club data. Most of other Kaggle projects focused on the default detection model training, which prioritized improving model accuracy. However, with a practical perspective, we were focusing on forming a practical investment strategy. Thus, we had kept in mind some drawbacks of other projects while designing our simulation process.

First of all, we have determined that many Kaggle projects exposed themselves to the risk of overfitting by using features that were not available for new loans, such as ‘last_pymnt_amnt’(last payment amount). Since this feature was considered one of the leading indicators of delinquency, including such feature would significantly inflate our model accuracy. Given the fact that we would not have such information on hand when a new loan coming in, the models trained under such approach were not be useful in practice.

Second, many “accuracy-driven” approach ignore the imbalance characteristic of the lending club dataset that most of loans were labeled as 0 (non-default). Given that, a classifier merely predicted all samples as 0 would have an unusual high accuracy rate. Such classifiers were meaningless and had no practical meaning.

Last but not least, focusing on the prediction accuracy did not take in consideration of magnitude of the losses. By merely predicting the probability of default, we were not able to grasp the big picture of our risk exposure. For example, despite the model may catch 99% of the defaults within our sample, the left over 1% may consist of extreme value that would greatly impact our performance. Due to this reason, we were not able to draw conclusion on which model was better with only accuracy on hand. In other words, without a validation simulation, different models were not comparable.

Based on above reasons, a validation simulation was deemed to be essential to evaluate our models as one investment strategy.

Serving as our base line for our simulation, a random forest classifier was used. Some other simulation parameters including:

1) the total investment period: 120 months,

2) investment cycle: 3 months,

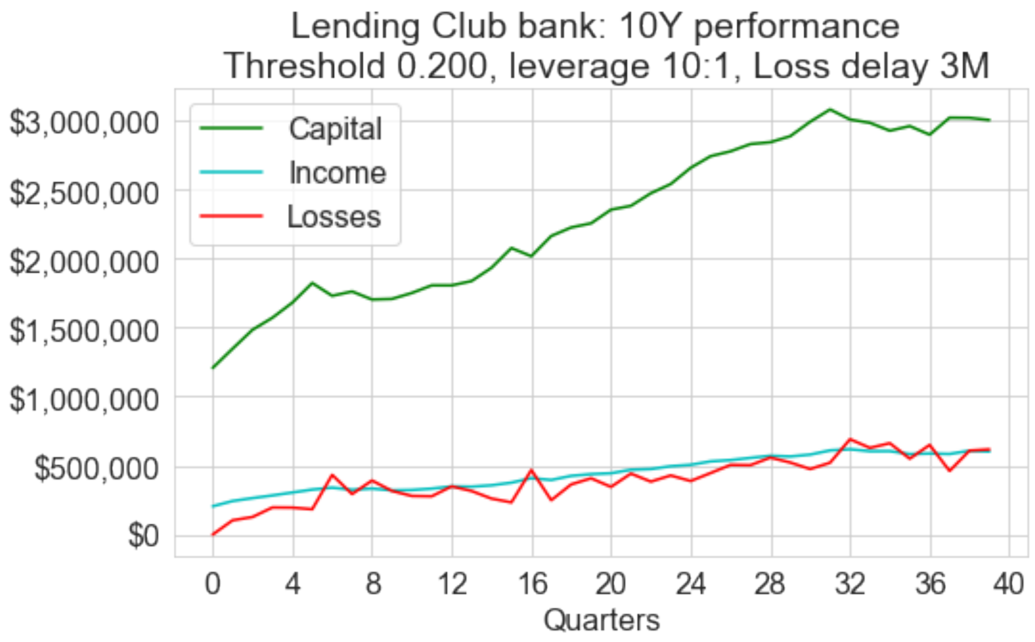
3) starting capital: $1,000,000,

4) leverage ratio: 10:1 on capital,

5) cost of financing: 3%,

6) default score threshold: 0.2.

Our simulation algorithm followed the strategy that we would borrow 10x current equity at the beginning of each investment cycle. Then we would evaluate 2000 randomly generated loans using our model and filtered loans based on our default score threshold. Keeping track with our interest’s income and loss along the way, we would pay off all debt on hand as well as borrow new loans given equity at that time. To have a simulation closer to reality, we also set a loss delay of 3 months, meaning that we would allow 3 months of recovery period to record default loss on our book. The performance was attached below. As we can see, using this model, we reach $3,000,000 capital after 10 years, which was approximately 11.6% annually.

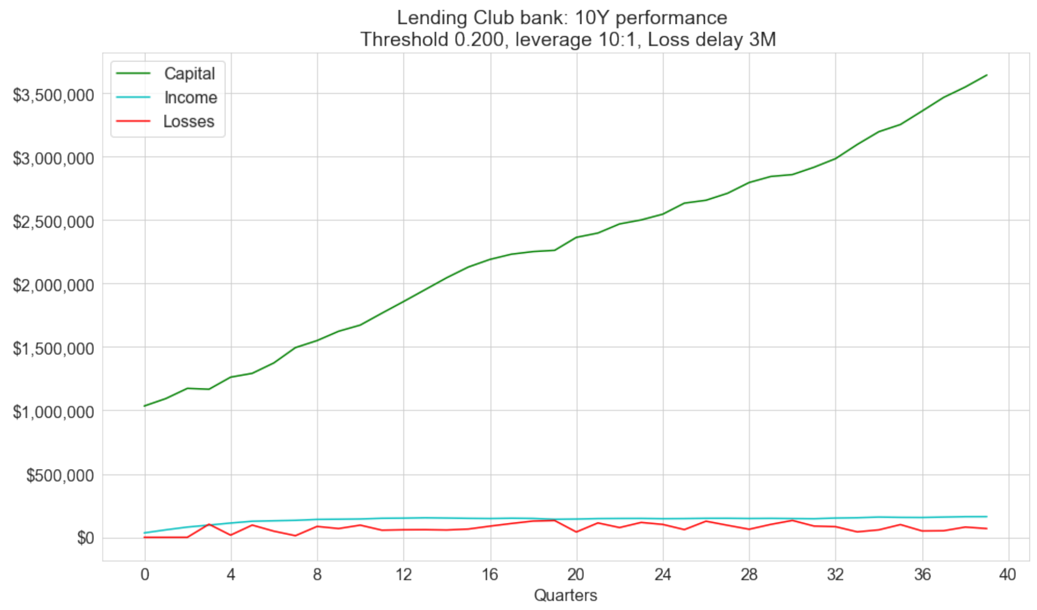


2. Dynamic Leverage

First of all, we found out that, in previous simulation, there were extra cash on hand because we fixed the new loan we borrowed in every cycle. When there was not enough investment opportunity, those capital were wasted while we still needed to pay interests on it. As a result, we changed the financing mechanism that we would evaluate the potential capital needed in that investment cycle before we borrowed, and we would make sure all cash were fully invested in every investment cycle.

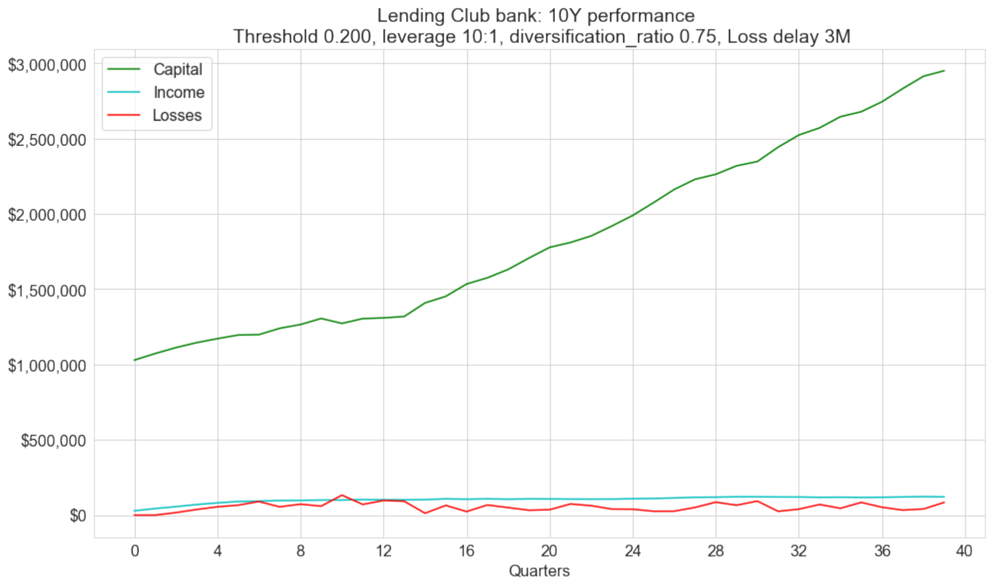
3. XGBoost Model

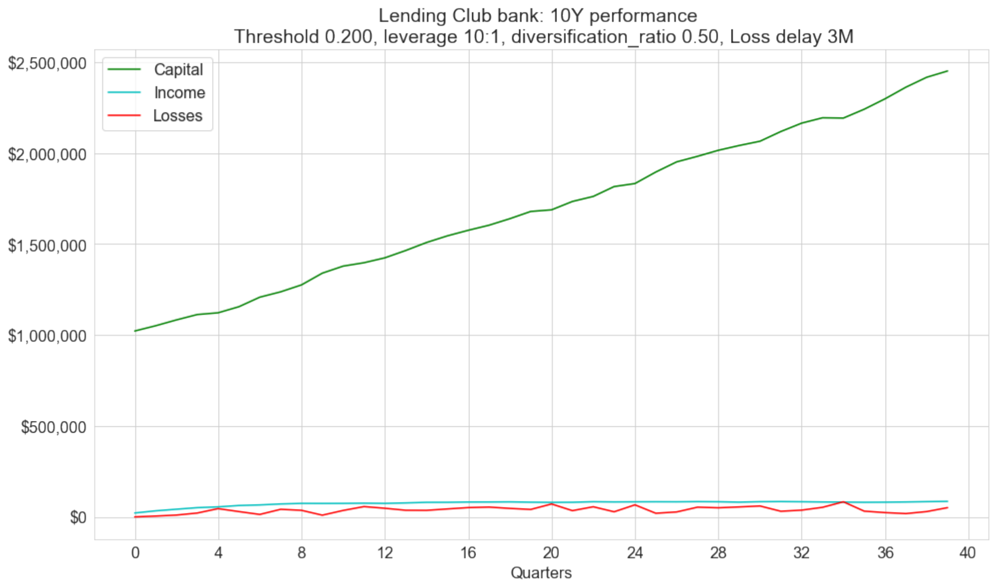
The second modification we made is to change our default prediction model. We have tried different models with hyperparameter tuning including random forest, logistic regression, neural network and XGBoost. We also used under-sample method in response to the imbalance nature of our data. Among all models, we found XGBoost provide best performance curve that the net incomes (income – losses) were all positive in 40 investment cycles and the curve was more stable and smoother. The annual return is approximately 13.5%.



4. Diversification

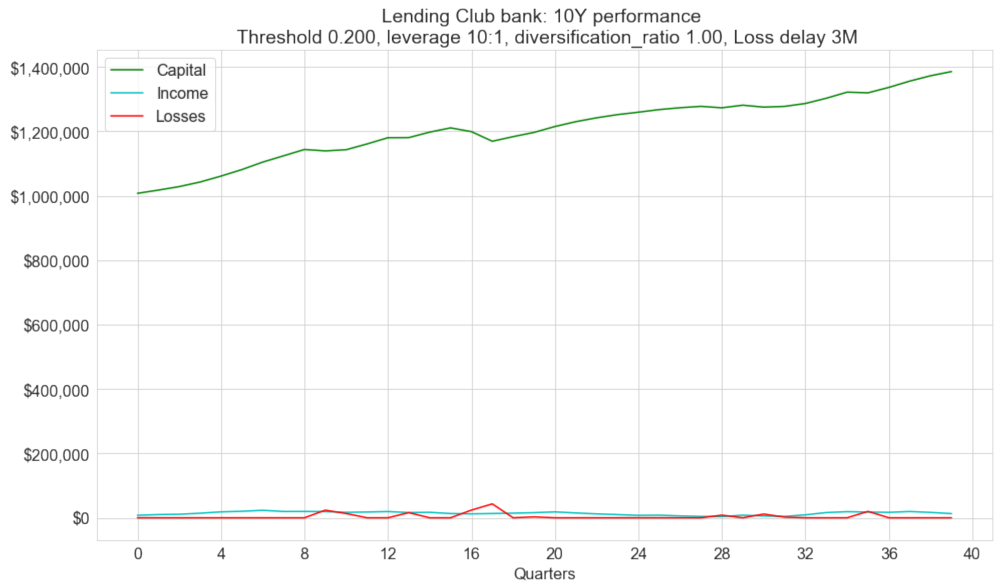
Next, we tried to approach the simulation on a portfolio management perspective. Could we further control our losses by diversification? Given our current algorithm, we were selecting loans randomly so that it was possible we ran out of capital before we could invest all loans below our default threshold. Also, theoretically, we could reduce our loss by diversification if there were a few very big losses. In order to answer this question, we created one more parameter: diversification ratio, the percentage of total amount of each loan we would invest. A 0.5 diversification ratio meant that we would only invest 50% of the total amount of each loans we selected, and we would use those capital to invest loans that were not selected before. However, the graph shown that diversification did not improve our performance. We thought this may due to the fact that the benefit of diversification did not outweigh the losses of income by partial investment. We had also tried several different numbers, but the improvement was limited, and, in some cases, it even reduced our net income to negative. The graphs of diversification of 0.75 and 0.5 were attached below.





5. Dynamic Portfolio Management

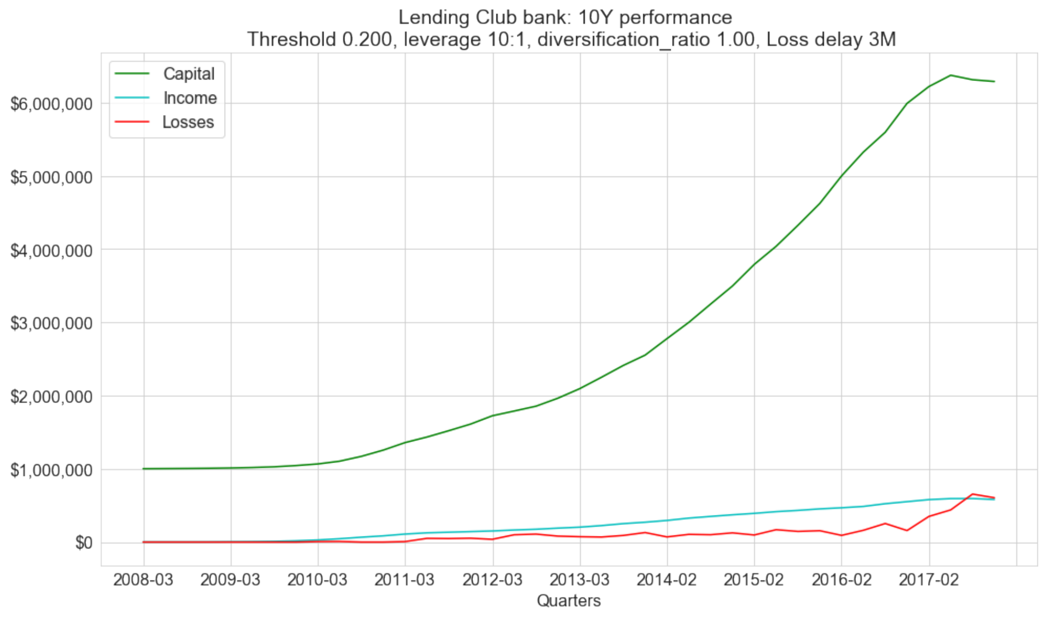
When we were conducting simulation, one question had raised, which is whether we should closely monitor our portfolio and actively rebalancing it. As we know, in reality, portfolio managers would make investment decision based on their current positions and risk exposure. In addition, they would not have all the opportunities listed in the beginning of the investment period. By considering loans one by one instead of all loans under the threshold, we had better control over our risk exposure and term structure (the ratio of short-term loans to long term loans) of our portfolio. As a result, we implemented a dynamic portfolio management approach which took consideration of current term structure and average default score level during selection of new loans. On the other hand, this approach may negatively impact the portfolio performance due to the opportunity cost. For example, if we wanted to reduce the average default score of our current portfolio, we may reject some relatively high-risk loans even if they were below our threshold. Thus, the order of loans would also impact our performance. Surprisingly, we found that the level of capital reduced significantly under this approach. With a pre-set short term to long term ratio 10:1, our capital at end of the investment period reduced to approximately $1,400,000. There were also several negative incomes. The graph was attached below.



As a summary of our attempt to add portfolio management strategy into our simulation, despite both diversification and dynamic portfolio management provide worse performance compared with original simulation, we considered both as a good foundation for further study and believed they added value to the simulation to better mimic the reality.

6. Chronological Sample Selection

Given the fact that the issue date of the loans was available in our dataset, we had also experimented our strategy with an actual back test that dated back from 2018-1-1. With previous parameters, this simulation reached over $6,000,000 at the end of the investment period. See attached graph



However, there was one thing caught us attention that it seems in the last few quarters, the losses had raised and crossed income meaning that in the last few quarters, the net incomes were negative. After digging into the data, we found out that in the last few quarters, the number of samples decreased because there are many loans were still ongoing. In other words, only prepaid and default loans were included in our dataset, which may contribute to the trend in the end.

Summary

In this project, we successfully optimized the default loan classification and simulation process and constructed the prepayment prediction model. The details are as follows.

First, we revisited Lending Club's database. Through our exploration, we found that most loans since 2016 are in the "Current" state. Therefore, the main research subject of this project is loans from 2011 to 2015, corresponding to the rapid growth of Lending Club. In the data we analyzed, 20.07% of the loans were in the state of "charged off", while the rest were fully paid. Our chart shows that the default rate of Lending Club has not changed significantly over time and scale but fluctuates around 20%. Further, we studied the Loss Given Default of the loan with the status of "Charged off". We found that there was a significant difference in the average loan line, default rate, and average loss between the 36-month and 60-month loans. The average loan amount for the 36-month loan was $12,769, with an average loss of $8,755.9. In comparison, the average loan amount for a 60-month loan is $20,459.5, with an average loss of $15,061.8. This result provided useful information for our subsequent simulation of the dynamic process: we would prioritize the 36-month loan in the dynamic strategy.

Second, we tried to construct the prepayment prediction model and applied it to simulation. We also found that almost 80% of the loans will be prepaid, and prepayment of the 60-month loans happens more frequently than that of 36-month loans. When we applied this model to simulation, we found that it did not improve our performance but reduced the output. The reason might be: (1) there were not enough prepayment related features, so the model has difficulties to find out the connection between label and features (2) about 80% of the loans will prepay, which made the model was more inclined to forecast results to prepay (3) we weren't able to choose enough loans to maintain our strategy in the simulation part, so that the increase of income was not obvious. Conclusively, we did not use the prepayment prediction in the final simulation.

Third, we further improved the accuracy of the default model and the speed of the simulation process. Compared with the previous 115 features, we achieved higher prediction accuracy with only 16 features. More importantly, we greatly improved the recall score (from 0.08 to 0.68) through the under-sampling method. This means that we can detect those loans that are likely to default more accurately, so as to avoid losses.

Last but not least, we have designed a simulation process used to compare different default detection models. In addition, we have integrated active portfolio management strategies into our simulation. By comparing performances of different models under different settings, we conclude that strategy using XGBoost model generate the best performance, which is approximately 13.5% annual return across 10 years investing period. In addition, we find our current diversification method and dynamic portfolio technique added little benefit to the strategy. Future studies can further tune those parameters to verify their impacts on our simulation process

Appendix

[1] Data Source: https://www.kaggle.com/wendykan/lending-club-loan-data

[2] Original Strategy: https://github.com/kingaj12/bank-lc-sim

[3] Optimal Strategy: https://github.com/jieqian2/bank-lc-sim/tree/master/notebook