Peer-to-peer loan default prediction using machine learning methods

Yifeng Luo

Erasmus University Rotterdam

Peer-to-peer loan default prediction using machine learning methods

Master's Thesis

To fulfill the requirements for the degree of
Master of Science in Data Science and Marketing Analytics
at Erasmus University Rotterdam under the supervision of
prof. dr. ir. R. Dekker
and
prof.dr. ACD Donkers (Second Reader)

Yifeng Luo (431371)

Contents

Abstract

With the development of internet finance, an increasing number of online Peer-to-Peer lending platforms grow rapidly. Credit risk assessment and default prediction based on online loan users have become particularly important. In the online P2P lending business scenario, the loan amount is usually low and the loan volume is huge, and the traditional manual approval can no longer meet the needs of the online loan business scenario. Moreover, most of the online loan customer groups belong to the people without credit investigation, and the users are only based on basic information. The method of credit assessment is also difficult to effectively define the user's default risk.

This paper presents a loan default prediction model using real world user data from LendingClub. Multiple machine learning algorithms are employed to build up models. The SMOTE method is employed to mitigate the imbalance data problem. And a series of data cleaning methods are applied to prepare the data. The results show that deep neural network perform the best among all models.

1 Introduction

After the financial crisis in 2008, both the United States and Europe fell into a period of liquidity crisis, and there were more and more barriers to credit access (Calabrese, Osmetti, & Zanin, 2019). However, it also provides opportunities for financial innovation such as peer-to-peer(P2P) lending. In the past decade, P2P lending has grown popularly, which has implications for traditional lending markets. Rapid growing online lending platforms have created more flexible lending models. But just like the Sword of Damocles, online lending not only brings convenience, but also brings some risks (Emekter, Tu, Jirasakuldech, & Lu, 2015).

The fundamental premise behind P2P lending is that borrowers can apply online for loans and other lenders can fund these loans and receive interest payments, potentially making returns higher than bank interest. Compared to traditional commercial banks, the new form of online lending is more efficient in connecting borrowers and investors, which is one of the reasons that online P2P lending is booming where internet markets are mature.

However, P2P lending also meets a lot of challenges. E.g., compared with traditional bank loans, online lending lacks collateral, which increases lending risks. How to use limited user information to filter borrowers is particularly important. In this context, the key research question of this study arises:

How can we identify high-quality borrowers for online P2P lending platform using machine learning methods?

This question is particularly important, because low-quality borrowers might have higher chances of default, which will lead to a huge loss to investors. In particular, I address the question from two perspectives:

1. I apply machine learning algorithms to build up a predictive model for loan default.
2. I focus on identifying important drivers that explain default action, which has regulation implications.

All codes used in this paper are public on author's Github, which could be used to replicate the results.[1]

To answer the question, I obtain loan-level data from LendingClub during year 2016 to 2018 that covers 517,579 loans with 152 features. I apply three machine learning algorithms, viz. logistic regression, random forest and deep neural network to build up multiple predictive models and then obtain the feature importance to understand what factors are the main driving forces. In addition, multiple performance measures are applied to compare the results.

The findings of this thesis are mainly in four aspects. First, I find that deep neural network model yields the highest prediction accuracy on default when it comes to predicting the default action of borrowers. Second, the state-of-the-art technique SMOTE significantly improve out of sample performance when our target variable is very unbalanced. Third, I further find that debt to income(dti) ratio, interest rate and loan grade calculated by LendingClub shows significant importance to explain the default action. In the end, there are several researchers applying traditional classification performance metrics on loan default prediction using unbalanced LendingClub data, which potentially makes their performance looking good, but not meaningful in practice.

[1] see https://github.com/yifeng93/ESE_MSc_Thesis

The remainder of this paper is structured as following. In Chapter 2, a review of LendingClub and literature review on both P2P lending and P2P lending default prediction will be provided. Chapter 3 introduces how data is cleaned and exploratory data analysis in detail. In Chapter 4, all science methods applied in this study such as SMOTE, random forest and deep neural network, etc. will be introduced. Chapter 5 reports the research results. In the end, Chapter 6 presents the conclusion and future works.

2 Problem Description and Literature Review

In this chapter, a larger context of the thesis is provided. Since I mainly focus on the online P2P platform of LendingClub, I first give an overview of the platform itself, which is presented in Section 2.1. Then literature on online P2P lending is provided in Section 2.2. One of the main goals of the thesis is to use machine learning methods to predict loan default. So, existing literature on default prediction is provided, which is in Section 2.3. In the end, the summary and main contributions on literature of this paper is presented in 2.4.

2.1 A Review of LendingClub

LendingClub is an American P2P lending listed firm headquartered in San Francisco, California. It is the first P2P lender in America to register its products as a security with the SEC[2], and to offer loan transactions on the secondary market. LendingClub describes itself on the official website as following:

"We're the only full-spectrum fintech marketplace bank built on the belief that innovative, creative solutions deliver more value and a better experience. Since 2007, more than 4 million members have joined the Club to help reach their financial goals. As the only full-spectrum fintech marketplace bank at scale, our members can gain access to a broad range of financial products and services through a technology-driven platform, designed to help them pay less when borrowing and earn more when saving."[3]

In general, LendingClub has two main functions: (1) providing a new form of online platform for borrowers to apply for and access loans and (2) allowing investors to fund the loans. The general life of a loan in LendingClub is summarized as following:

1. Borrowers and investors create their accounts through the official website or official app of LendingClub.
2. Borrowers apply for loans by filling out several aspects information and wait for approval.
3. The credit check for borrowers.
4. Borrowers will receives the loan if they meet certain criteria such as:
    - The credit score that is higher than minimum credit score of 600.
    - The credit history has a minimum of 3 years.
    - The debt-to-income ratio of less than 40% for single applications, 35% for joint applicants.
5. Investors build up a portfolio that may consist of multiple notes.
6. Investors earn from the interest that borrowers pay.

[2]SEC is U.S. Securities and Exchange Commission, which is an independent agency of the federal government
[3]see https://www.lendingclub.com/company/about-us

One of the most important elements of applying or funding a loan successfully is the loan grades. Because the higher the loan grades are, the lower risk for the investors and also the lower interest rate for borrowers. LendingClub categorizes borrowers into seven loan grades: A through G. Within each loan grade there are five sub-grades. So, there are 35 loan grades in total. The loan grades can only be seen by investors. That is, borrowers do not know what grades their loan applications are. Appendix A.1 presents a screen shot of the web interface of the LendingClub seen by an investor. We can see that investors can decide whether to fund the loan based on the loan grades calculated by LendingClub, loan term, amount, purpose and the FICO[4] score range of the loan.

For borrowers, it is simple to pay back the loan after the loan is approved. They are given a payment schedule including the required monthly interest payment and the loan amount payment until the loan matures (either 36 or 60 months). LendingClub provides a full tutorial how to check and pay back the loan for borrowers.[5]

For investors, after checking available loans on marketplace as the screeshot shown in Appendix A.1, an investor can fund as little as $25 of the loan. For example, there is a loan application of $3,000 with the loan grade of A3. It is possible that investor A only funds 20% ($600) of the loan, investor B funds 30% ($900) and investor C funds the rest of 50% ($1500). Once the loan is fully funded, borrowers will receive the money in their bank account. Investors start to receive the interest payment monthly if the borrowers pay back the loan payment on time each month. In most cases, borrowers pay off the principal at the last repayment period. That is, investors can only receive interest payment each month. In general, there are four main risks an investors must confront.

1. Borrower defaults. Investors have little information about the borrowers especially after the loan issued. The limited information investors can have is the calculated loan grades by LendingClub for the loans. As the public data statistics(2007-2018) presented in Table 1, most loans are either fully paid or in the current status. The loan default rate (Consider charged off as default only) across all grades is about 20%. More detailed data analysis on default rates is done in Section 3.3.

2. Interest rate risk. The loan terms are 36 months or 60 months. The loan interest rate is fixed during the period. If other risk free investment opportunities have higher interest rate during the period, then LendingClub loan investment will not be the best investment.

3. Liquidity risk. There is a secondary market where investors can sell the loan on LendingClub Note Trading Platform. However, investors are likely to lose some principal in the process if an investor wants to liquidate the investment.

4. LendingClub bankruptcy. This risk is less likely to happen today since LendingClub went to IPO in 2014 and it has an influx of cash. But the risk still exists. E.g., LendingClub suddenly had a large number of borrowers defaulting, which damaged its reputation and lost customers. As a result, the LendingClub platform is no longer able to maintain operation. Then its collapse will make investors need to find the borrower to recover their investment, and the possibility of recovering the investment loss at this time is very small.

[4]FICO represents Fair Issac Co. and it is a scoring model meant to give lenders an idea of how customers handle money. Mostly it ranges from 300(poor) to 850(exceptional). For more information, see https://www.myfico.com/credit-education/fico-scores-vs-credit-scores

[5]see https://help.lendingclub.com/hc/en-us/articles/214519627-Making-loan-payments

Table 1: Statistics of Loan Status of LendingClub Public Data (2007-2018)

Status	**Definition**	**Number**
Current	The loan is currently being paid off on time	878,317
Fully Paid	The loan was fully paid off	1,076,751
Grace Period	Payments are late by 15 days or less on the loan	8,436
Late (16-30 days)	Payments are late by 16-30 days on the loan	4,349
Late (31-120 days)	Payments are late by 31-120 days on the loan	21,467
Default	Payments are late by 120-150 days on the loan.	820
Charged off	The platform believes that the further payments on this loan are unlikely. Payments are 150 days past due.	268,559

Source: Author

So far, we have discussed the basic information about LendingClub and the loan process for borrowers and investors. In short, LendingClub today owns more than 4 million users and it is representative in the field of online P2P lending that cannot be ignored.

2.2 Literature on P2P Lending

P2P lending is the loan origination between private individuals on online platforms, which also known as social lending or crowd lending (Bachmann et al., 2011). The new form of lending offers better return rates for investors and also better access to credit for borrowers who may not have access to banks (Milne & Parboteeah, 2016). It has attracted massive attention from both the industry and the academic since the first launch of Zopa in 2005 in United Kingdom. Dominated by U.S. based LendingClub[6] and Prosper [7] and U.K. based Zopa[8], these companies have succeeded providing alternative asset class for investors. However, some online platforms went out of business due to high default rates by offering interest rates to borrowers that were too high, e.g., China based Renrendai (Yao, Chen, Wei, Chen, & Yang, 2019).

(Basha, Elgammal, & Abuzayed, 2021) provides the most up to date literature overview on P2P lending. The study points out that researches at early stage mainly geographically skewed towards United States and China with focus on determinants of funding success and loan attributes. Because United States and China have the biggest crowed lending market. However, recent studies shift to examine funding success and default predictions, towards applying artificial intelligence. In fact, the study by (Wang, Greiner, & Aronson, 2009) in 2009 has provided the overview of concept of online peer lending and categorized several online platform such as LendingClub and Zopa into the model of profit-seeking, one of the four quadrants of a matrix separated by what they see as the two main

[6] see https://www.lendingclub.com/company/about-us
[7] see https://www.prosper.com/about
[8] see https://www.zopa.com/about

factors that differentiate lending models: level of separation (friends or strangers) and motivation of lending (economic or philanthropic).

(Wang, Chen, Zhu, & Song, 2015) model online P2P lending as a process, but there is no intrinsically distinction between the process of traditional bank lending and online P2P lending. They conclude that the information flow in P2P lending is more frequent and transparent. In addition, P2P lending adopts different credit audition method, which mostly relies on the support decision model in the P2P system. In the end, P2P online lending management may not be as advanced as traditional bank lending mainly due to lack of ability to track post-loan information (Wang et al., 2015).

Since P2P lending is relative new in financial industry and is developing rapidly, the level of supervision and regulation of platforms in various countries and regions is different. In the UK, online P2P is characterized to be self-regulated, but it is regulated by the Securities and Exchange Commission in the U.S.(Wardrop et al., 2016) China, one of the biggest crowed funding markets, issued a temporary management law in 2016 to supervise the fast growing online lending market, which is "Interim Measures for the Administration of Business Activities of Online Lending Information Intermediaries" [9]. However, this temporary management measure did not prevent the subsequent collapse of the Chinese P2P market in 2020. Renrendai[10], one of the biggest online P2P platform in China, was exposed that investors could not withdraw from their investments at the end of 2020. Then China Banking Regulatory Commission have further tightened the issuance of P2P platform licenses (Chorzempa & Huang, 2022).

2.3 Literature on Default Prediction

The supervision and regulation of online lending platforms in various countries and regions is still being improved. But indeed online P2P lending is playing a bigger role in the financial sector. In addition, benefit from the digitization of the process of online lending, an increasing number of institutions are using big data and applying machine learning methods to predict default action (H. Kim, Cho, & Ryu, 2020). It is common for some big financial institutions to adopt artificial intelligence automation system to control credit risk today (Dhaigude & Lawande, 2022).

The literature has been trying to understand what factors might or might not affect the default in P2P lending. One strand of the literature focuses on hard information. For example, (Kumar, Goel, Jain, Singhal, & Goel, 2018) finds that the relationship between borrowers and investors, debt-income ratio, borrowers' annual income, working duration, home ownership and occupation and whether or not a borrower possesses a checking account influence the loan default action. (Basha et al., 2021) finds that some demographic factors such as education level and gender have little influence on P2P lending. (Kelly, O'Toole, et al., 2016) finds that longer-term loans are less likely to become default because of the lower instalments. Another strand of the literature expands to soft information. For example, (Jiang, Wang, Wang, & Ding, 2018) uses soft information extracted from the texts associated with the loan applications as predictors, as well as (Xia, He, Li, Liu, & Ding, 2020), (Zhang, Wang, Zhang, & Wang, 2020).

Recent literature starts more discussions on using machine learning techniques for predictions. For example, (Ying, 2018) employs logistic regression, random forest and support vector machines models on bank credit data for a comparative study. The results show that random forest works better than the other two methods. (Kumar et al., 2018) work with LendingClub data to predict P2P

[9] see http://www.gov.cn/gongbao/content/2017/content$_5$181095.*htm*

[10] see https://www.crunchbase.com/organization/renrendai

loan default and infer that deep neural network model performs better than most of the traditional models. (Zhou, Li, Wang, Ding, & Xia, 2019) find that ensemble learning-based algorithms are more effective for imbalanced and high-dimensional credit data with missing values in P2P lending study. On this basis, (Xia, Liu, Da, & Xie, 2018) propose a heterogeneous ensemble learning-based loan rating method integrating multiple deep learning algorithms to process actual transaction data from a Chinese P2P lending firm. The results show that the combined algorithm is usually superior to others.

2.4 Contribution to Literature

My paper contributes to the literature in two ways. First, it contributes to a better understanding of the literature with an improved predicting method. Many papers in the literature use traditional classification measures to evaluate the performance of the predictive model. For example, (Zhu, Qiu, Ergu, Ying, & Liu, 2019), (X. Li et al., 2022) and (Xia et al., 2020) uses accuracy and AUC score, (J.-Y. Kim & Cho, 2019b) uses F1 score to measure performance. Table 2 provides a summary of criticism of some literature on LendingClub default prediction. Though using traditional performance measures may yield exceptionally high performance values, they lack realistic implications due to the unbalance of the target variable. These traditional classification performance metrics are unable to evaluate the model in a meaningful way. However, in my paper, I apply oversampling method to my data to obtain a more balanced data, at the same time, use the state-of-arts, i.e., Matthews Correlation Coefficient method, to compare performance across various machine learning algorithms, which provides more meaningful insights.

Second, it contributes to a better feature selection method, especially in the spirit of timing of the features. Most of the literature selects features that are associated with the timing right before the default (e.g., (J.-Y. Kim & Cho, 2019b), such as consumption records during the few months before the default date. However, realistically, the platform has very limited control over borrowers' behaviors after they have obtained the loan, thus, my thesis uses features that are pre-determined before the time when their loan application is approved. This way helps me alleviate any forward-looking biases in the model specifications.

Table 2: Criticism of Some Literature on LendingClub Default Prediction

Literature	**Criticism**
(Zhu et al., 2019)	1) A loose definition of default. I.e., (Zhu et al., 2019) consider *Current* status the same as $FullyPaid$, which is not correct. And this leads to a very low default rate(1.53%) in their paper. 2) They use ROC-AUC socre and accuracy as performance measures for their extremely unbalanced data, which is not meaningful in practice. 3) No confusion matrix reported.
(X. Li et al., 2022)	1) The same problems as (Zhu et al., 2019). That is, a loose definition of default, improper usage of performance measures and no confusion matrix provided. 2) This paper also has one of the same authors from (Zhu et al., 2019) Daji Ergu.

Literature	Criticism
(Xia et al., 2020)	1) Do not deal with unbalanced data, which leads to a high sensitivity but low specificity. 2) Improper usage of performance measures for unbalanced data(accuracy and ROC-AUC socre). 3) No confusion matrix reported
(J.-Y. Kim & Cho, 2019b)	1) Do not deal with unbalanced data. If MCC score is calculated based on confusion matrix they provided, their resulting model has only 0.17 MCC score. 2) (J.-Y. Kim & Cho, 2019b) apply a proper performance measure of F1 socre, but they use B as true positive(TP) instead of A, which is a misleading way. If A and B are reversed when calculating F1 score, their F1 score will be only 0.29 instead of 0.85. Note: the A and B in this criticism refer to the true positive and true negative position of confusion matrix from (J.-Y. Kim & Cho, 2019b). Figure 1 presents the original confusion matrix in their paper.
(Fu, 2017) & (Duan, 2019)	1) Improper usage of performance measures for unbalanced data(accuracy and ROC-AUC socre). 2) No confusion matrix provided.

Figure 1: Confusion Matrix from (J.-Y. Kim & Cho, 2019b)

Table 8. Confusion matrix.

Predict True	Charged Off	Fully Paid
Charged off	2155 (A)	7300 (C)
Fully paid	3115 (D)	30,575 (B)

Source: (J.-Y. Kim & Cho, 2019b)

3 Data

The data used in this study is LendingClub public data[11]. LendingClub public parts of the user data since 2015. According to U.S. federal law, users have the right to limit some information but not all.[12] The lending club public data does not include sensitive user information such as name or data of birth, but the features are enough for analysis.

In this chapter, the general information about the raw data will be introduced in Section 3.1. Then the data cleaning detail will be provided in Section 3.2. In the end, the exploratory data analysis is presented in Section 3.3.

3.1 LendingClub Data

The raw data contains statistics on 2,258,699 loans funded over the course of 12 year from January 2007 to December 2018. There is no unique value identification of borrowers or lenders. So, we do not know whether there are multiple loans from the same person or not. In this study, only the data from 2016 to 2018 are used for model implementations. There are mainly two reasons. First, there are too many missing values in some features before the year of 2016. That is, over 50 out of 150 features have the missing value rate over 95% before 2016. Second, the full data is too big to compute on a normal personal laptop. The hardware and software operation information for this study is provided in Table 3. To analyze the study using full data set, a GPU of RTX 3050 or higher and a RAM of 64GB or higher are recommended from the author[13].

Table 3: Laptop Specs and System Information

Item	**Value**
CPU	Intel i7–10510U @1.8Ghz
GPU	GeForce GTX 1650
RAM	16GB SODIMM DDR4L-2400
System	Windows 10 64bit Professional
Programming	Python 3.94

There are at most 152 features recorded for each loan, which are from different aspects such as basic loan information, user credit information and user demographic information. LendingClub provides a full description of all features for public data [14]. In this analysis, we focus on the loan default prediction, which means our response variable is the $loan\_status$. LendingClub categorize the loan status into 7 categories as Table 1 shown. I only use $FullyPaid$ as the classifier of not default and

[11] see https://www.kaggle.com/datasets/wordsforthewise/lending-club

[12] For more information, see https://www.lendingclub.com/legal/privacy-policy

[13] The computer specs of the library of Erasmus University is slightly worse than that of the author's laptop, and there is no graphics card. So, the author does not use the school computer for code testing. But the author briefly rented a cloud computer with a RAM of 64GB and a GPU of RTX 3050. It works much better than author's laptop.

[14] Download the data dictionary through this link: https://resources.lendingclub.com/LCDataDictionary.xlsx

$ChargedOff$ as the classifier of default because other loan status are still ambiguous for the default action.

3.2 Data Cleaning

As mentioned previously, the model implementations are only for data from 2016 to 2018. The data cleaning is applied to the subset of data from 2016 to 2018. In general, the data cleaning includes the following steps:

- Filter and remove useless features, which may includes features with too many missing values or features not relevant to this analysis.
- A series of feature engineering, which may includes imputing missing points, creating new features and feature transformation.
- Feature categorification for non-numerical features and standardizing numerical features.
- Split data into a training set(2016-2017) and a test set(2018).
- Using SMOTE technique to deal with unbalanced data problem in training set.

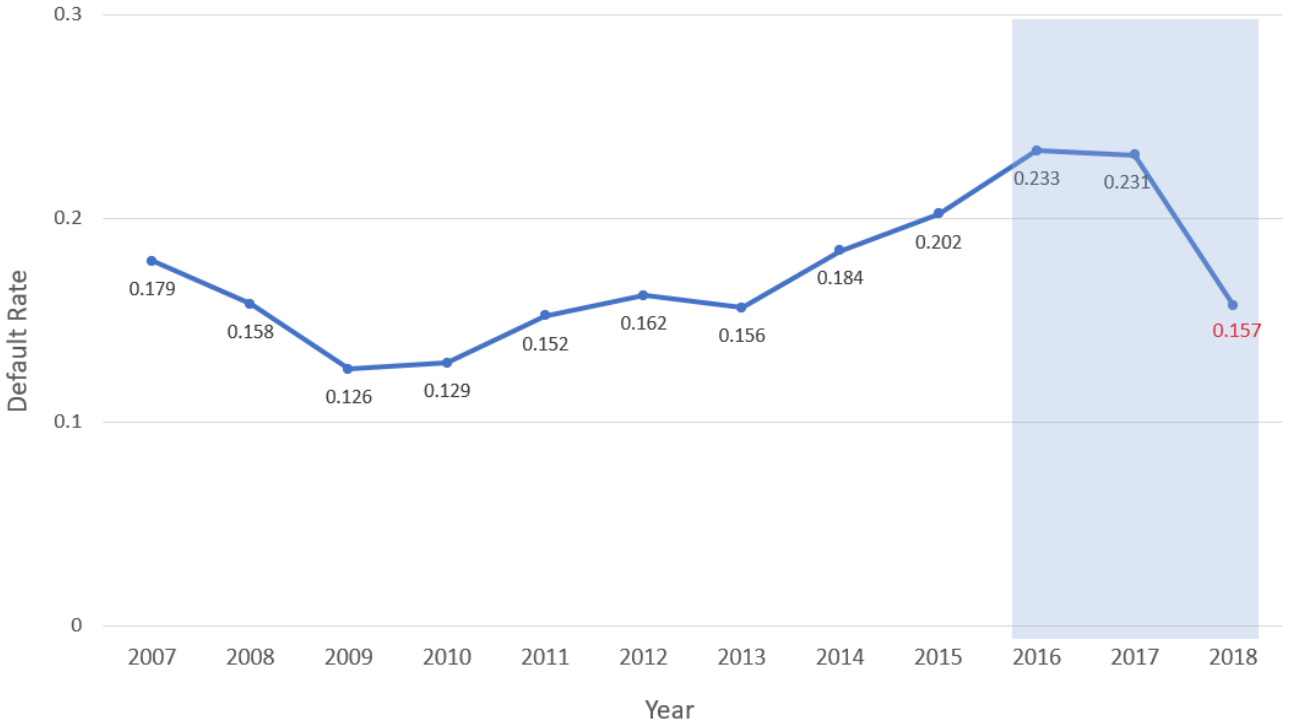
First, features with too many missing value are dropped. E.g., *desc*, which is the loan description provided by the borrower, has the missing value rate of 100%. The cut off point for dropping features with missing value I set is 50%. ((Z. Li, Li, Li, Hu, & Gao, 2021) choose 50% and (Tiwari, 2018) choose 30% in their study) On this basis, there are 44 features are dropped. Second, features would not have been available at the time of the loan application are also be dropped. Because one of the goals of the study is to use pre-loan information only to predict loan default. For example, variables $last\_pymnt\_amnt$ and $last\_pymnt\_d$, which are the last total payment amount received and last month payment was received. They are dropped because they are only available after the loan has been issued. On this basis, there are 59 features are dropped. Third, features that are independent from default action are dropped. For example, *id* is dropped because it is the assigned number of a loan by LendingClub, which is not relevant to default action. In the end, the remaining features are inspected one by one and different data engineering are applied if it is necessary. E.g., $fico\_range\_low$ and $fico\_range\_high$ are the lower and upper boundary range of the borrower's FICO score. A new feature called $fico\_score$ that is the average score of $fico\_range\_low$ and $fico\_range\_high$ is created since The $fico\_range\_low$ and $fico\_range\_high$ are highly correlated. Then I keep the feature $fico\_score$ only. Another example is that feature sub_grade has included information in feature $grade$. So, only sub_grade is kept. After the steps above, data is divided into a training set and a test set, which are data from the year 2017-2018 and the year 2018. That is, models will be trained using the data from the year 2016-2017 to predict the default action in the year of 2018. Imputation of missing values uses feature medians from training set only to avoid data leakage problem. Appendix A.2 presents the description of final selected or created features in the study.

3.3 Exploratory Data Analysis

In this sections, various exploratory data analysis are conducted. Some more feature engineering are also conducted based on the exploratory data analysis results.

First, the default rate of Lending Club through the years is checked. Figure 2 demonstrate the yearly default rate of LendingClub from 2007 to 2018. The light blue area is the data I use for the study. It is clear that the default rate keeps a relative low level before 2013. From 2014 to 2017 the default rate goes up sharply, which may due to the interest rate raised by the Federal Reserve of the United States. The target prediction year of this study is 2018, which has a relative low default rate of 15.7%. This creates a very changeable mission for this study that the target variable in the out of sample data is much more unbalance compared to the target variable in the training set.

Figure 2: Default Rate of LendingClub from 2007 to 2018



Then I apply exploratory data analysis to a more micro scope. However, we still have over 25 features after several data cleaning and feature selection processes. So, the histograms and box-plots of remaining features are all presented in Appendix A.3. Here I report some of noticeable findings from those plots in the Appendix A.3.

1. From the box-plots grouped by *loan_status* of multiple features, I find that some features show relative big different average values for different *loan_status*. E.g., *Charged_Off* group has lower FICO score but higher interest rate on average.

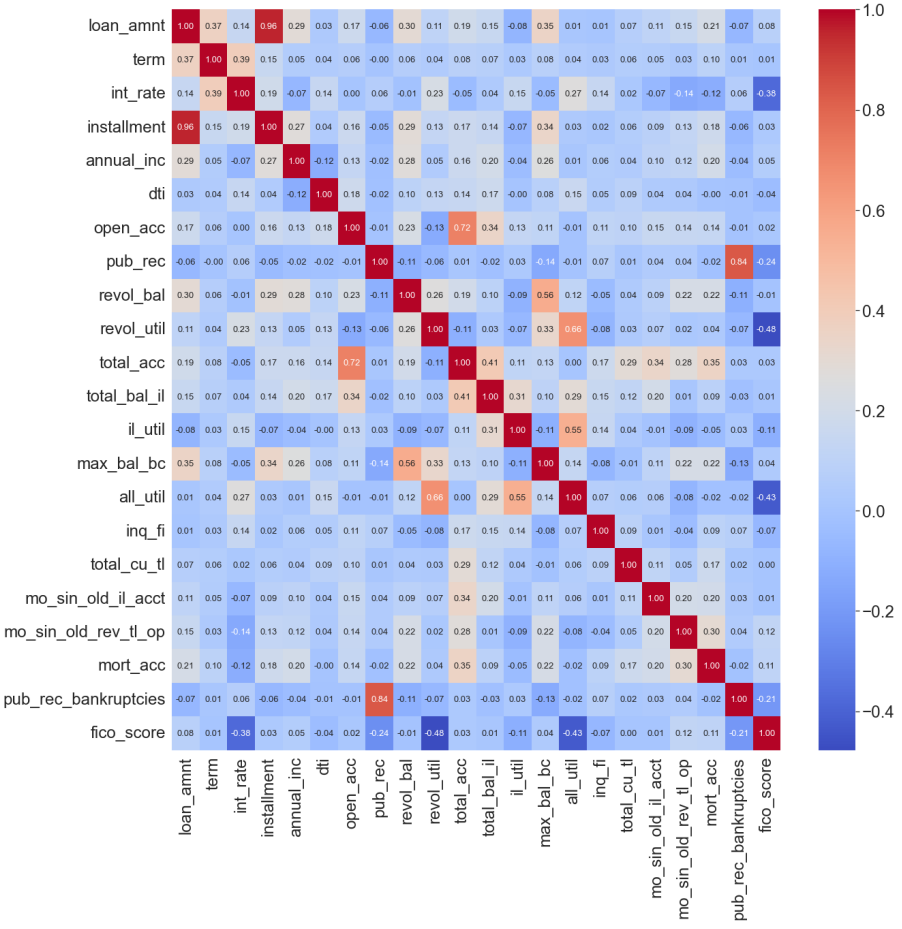
2. There are some numeric features having outliers. E.g., *int_rate*, *loan_amnt* and *dti*, etc. I use interquartile range (IQR) method following (Rousseeuw & Hubert, 2011) to identify outliers and create a subset data. In the end, 32,984 out of 517,579 outliers are detected. A subset data without outliers is created to compare with data with outliers. More detail is provided in Section 5.

3. There are 35 (sub)grades for loans ranging from A1(high grade) to G6(low grade). Most of loans are distributed at the (sub)grade of A5 to C4. Although there are not so many loans with

(sub)grade lower than F, the default rate goes up linearly as the loan (sub)grade goes down. Loans having a (sub)grade lower than F have default rate higher than 40%, which is extremely high.

4. There is no clear difference in default rate for different employment length(from 0 to 10+ years). However, it is surprising that verified income source group has higher default rate compared to the group which income source is not verified. Typically, we might expect a lower default rate for groups whose income is verified because information of income is more complete. The opposite is shown here. The possible reason is that some of the incomes of some people cannot be verified (such as cash income), and these unverifiable incomes are also real incomes.

Figure 3: Pearson Correlation Matrix

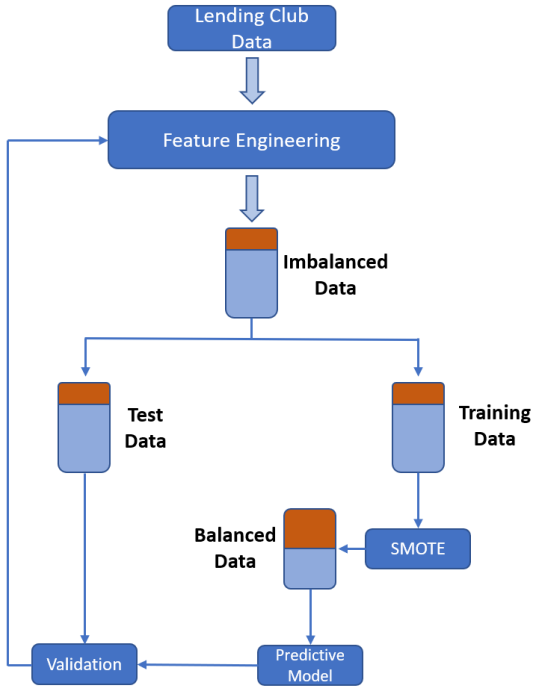


After checking features in a micro scope, analysis on features between each others is also conducted. Following (Benesty, Chen, Huang, & Cohen, 2009), a Pearson correlation check is conducted for numeric features. Figure 3 presents the Pearson correlation matrix where a Pearson correlation coefficient is from 0 to 1. A value of 1 means that the two features are perfectly correlated. Including both features that are strongly correlated may not affect the prediction performance, but it may not provide valid results about individual predictor, or about which predictors are redundant with respect to others (Midi, Sarkar, & Rana, 2010). As we see from Figure 3, *installment* and *loan_amnt* are highly correlated with a Pearson coefficient of 0.96. This makes sense because installment is calculated by using loan amount divided by term of loan. *pub_rec* and *pub_rec_bankruptcies* are number of derogatory public records and number of public record bankruptcies. They are also highly correlated with a Pearson coefficient of 0.84. In the end, I choose the threshold value of 0.7. On this basis, *installment*, *pub_rec* and *open_acc* are dropped. The rest of the numeric features are relative independent from each others.

4 Methods

In this chapter, I provide the main methods applied for the research. In general, I develop a decision support system to evaluate the risk of loan defaults. Figure 4 demonstrates the general model process. As we can see, the raw data first pass through feature engineering step to clean data and a number of qualified features are selected. Then, One of the challenges arises, which is the imbalanced data. This is a common problem in credit risk evaluation, and it can cause bias and misclassification. So, resampling approach is employed to mitigate imbalanced data problem. To be noticed, the resampling method only applies to the training set not to the test set, which is to avoid data leakage problem. Then, the balanced data will feed to the predictive model to train. In the end, the classification results are validated. Since we will evaluate multiple machine learning algorithms to evaluate their performance, the model implementation will be repeated for different machine learning algorithms and different parameters combinations. Three different algorithms are implemented in the end. Some methods need to tune multiple parameters such as deep neural network. All of their hyperparameters are reported in Table 4.

Figure 4: General Model Process



Source: Author

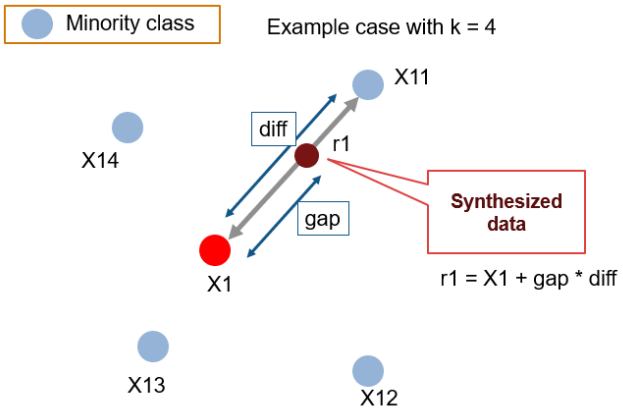
The remaining of this chapter is structured as following. Section 4.1 provides the method of SMOTE, which solves the imbalanced data problem. Then, three main machine learning algorithms are introduced in Section 4.2 (logistic regression), Section 4.3 (Random Forest) and Section 4.4 (deep neural network). In Section 4.5, a couple of programming libraries will be used for analysis such as Pandas and PyTorch will be introduced. Performance metrics are reported in Section 4.6. In the end, the implementation process is presented in Section 4.7.

4.1 SMOTE: Imbalanced Data Solution

Classification of data with imbalanced class distribution is problematic with most classifier machine learning algorithms, which assume a relative balanced class distribution. Class imbalance occurs when the number of observations in one class is very different from the observations in another class. For the scenario of multiple classification, it means the number of observation in each class is not balanced distributed (Namvar, Siami, Rabhi, & Naderpour, 2018). Classifiers may be biased towards to the majority class and the minority class may be ignored in such cases. For example, we want to predict whether a student can pass a math exam based on their other course grades. There are 50 students in a class and 49 of them are labeled as ”passed”. When we feed the data into the model, the machine will learn very “cleverly” that to predict a pass every time, and the accuracy rate will be

very high (98%). But this is not what we want. So, accuracy is a bad machine learning metric when working with imbalanced data. In our data set, the proportion of fully paid loan is about 80% and the rest of the loans are related to default issue. Obviously that we are confronting the imbalanced data problem.

Figure 5: SMOTE Working Mechanism



Source: https://github.com/minoue-xx/Oversampling-Imbalanced-Data

To mitigate the imbalanced data problem, synthetic minority oversampling technique (SMOTE) is applied, which is one of the resampling techniques. Unlike random oversampling techniques, SMOTE focuses on the minority class and creates new data points similar to it through interpolation between neighboring points using k-nearest neighbors (Chawla, Bowyer, Hall, & Kegelmeyer, 2002).

The general SMOTE working mechanism is presented in Figure 5. As illustrated in the graph, the total number of the oversampling observations is set up. For the binary class distribution, it is likely to be set to make the new distribution as 1:1. But also could be other distribution ratio based on need. Then the iteration starts by first selecting a positive class instance at random. Then the K-nearest neighbors for that instances are obtained. In the end, N of these K instances is chosen to interpolate new synthetic instances.SMOTE has the advantage of not creating duplicate data points, instead, it creates synthetic data points that differ slightly from the original data points.

4.2 Logistic Regression

Logistic regression is commonly used in classification problems due to its simplicity and interpretable ability. In this analysis, logistic regression is employed to better interpret the results. That is, the marginal effect on loan default will be calculated after having logistic regression results.

The general logistic regression formula is presented in Equation 1 where β_0 and β are parameters of a linear model with β_0 β denoting a vector of coefficients, $\beta = [\beta_1, \beta_2, \ldots, \beta p]^\top$. Equation 1 is derived from the relation between the log-odds of $P(Y_i = 1 \mid X_i = x_i)$ and a linear transformation of x_i, that is Equation 2.

$$P(Y_i = 1 \mid X_i = \mathbf{x}_i) = \frac{e^{\beta_0 + \beta^\top \mathbf{x}_i}}{1 + e^{\beta_0 + \beta^\top \mathbf{x}_i}} \tag{1}$$

$$\log \frac{P(Y_i = 1 \mid X_i = \mathbf{x}_i)}{1 - P(Y_i = 1 \mid X_i = \mathbf{x}_i)} = \beta_0 + \beta^\top \mathbf{x}_i \tag{2}$$

The class prediction can then be defined as Euqation 3 where c is a threshold parameter of the decision boundary cut off point, which is mostly 0.5 but it can be modified based on need. Further, in order to find the parameters β_0 and β , the maximization of the log-likelihood of Y_i is performed.

$$\hat{y}_i = \begin{cases} 1, & \text{if } P(Y_i = 1 \mid X_i = \mathrm{x}_i) \geq c \\ 0, & \text{if } P(Y_i = 1 \mid X_i = \mathrm{x}_i) < c \end{cases} \tag{3}$$

4.3 Random Forest

Random forest was first proposed by (Breiman, 2001) and can be seen as an extension of bagging ensemble learning. Decision tree is the baseline model for random forest algorithm. The general idea is to generate various differentiated decisions by constructing multiple feature tree models. The combination strategy mostly adopts voting or averaging to obtain a final decision.

Initially, random forest uses bootstrap resampling technique to select n samples randomly. In most cases, it select 2/3 of the samples from training set. Then it generate a new training sample set, each extracted training sample is applied to train a tree, and a forest is composed of n decision trees generated based on the bootstrap sample set. Each tree has the same distribution, and the classification error depends on the classification ability of each tree and the correlation between them. While The remaining data set that has not been extracted is called out-of-bag (OOB), and its error is an unbiased estimate that can be used to verify the performance of the model to prevent overfitting problem.

Decision tree algorithms are a huge family, and the random forest used in this article is constructed based on classification and regression trees. In the process of building each classification and regression tree, the splitting process of each node is completed by calculating the ”purity” of the split samples. The classification and regression tree uses the Gini coefficient to measure this so-called ”purity”, that is, random The forest uses the Gini index to split the tree to complete the decision. The smaller the Gini coefficient, the higher the purity of the sample and the better the effect of tree division. Assuming that the sample set T contains k categories, the Gini coefficient of the sample set can be expressed as Equation 4 where p_i is the probability that class i is contained in T.

$$\mathrm{gini}(T) = 1 - \sum_{i=1}^{k} p_i^2 \tag{4}$$

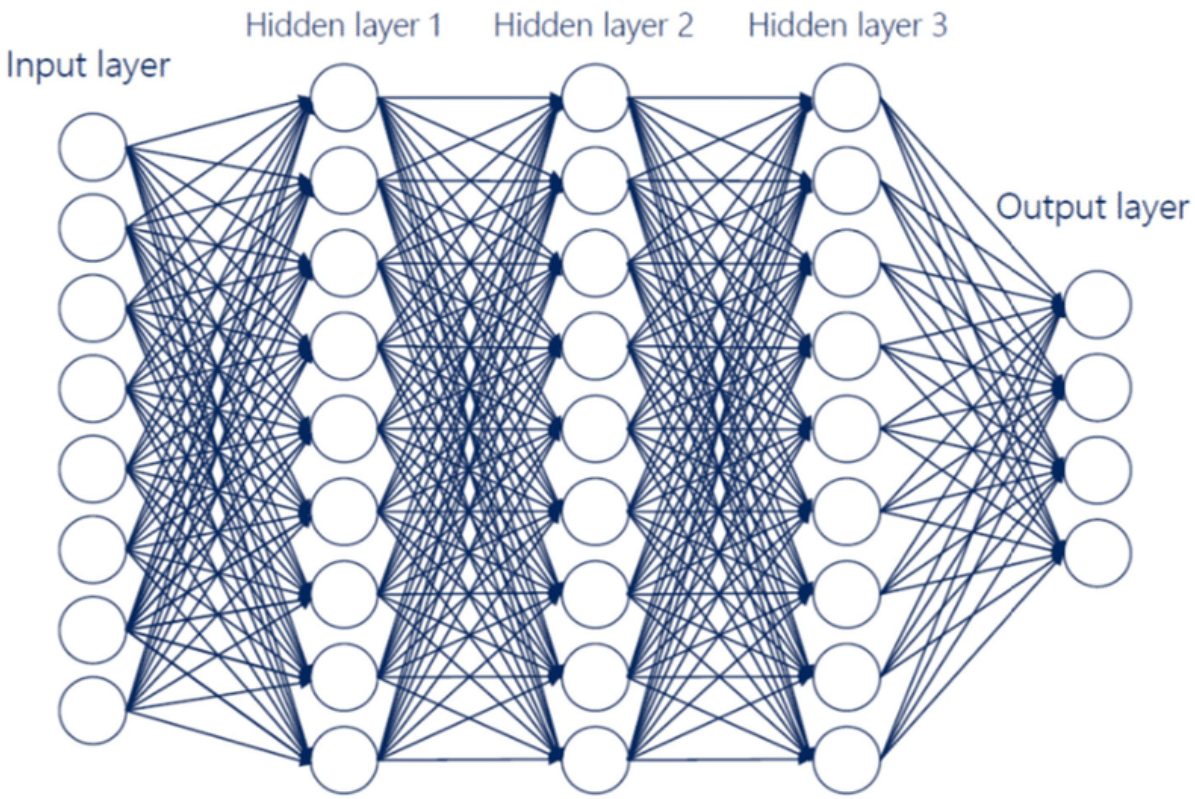
4.4 Deep Neural Network

Artificial neural network was first inspired by the concept of how human brain works and try to use algorithm to replicate the human learning process (Haykin, 2009). A neural network typically includes an input layer, an output layer and a hidden layer. When the number of hidden layer is more than or equal to two, the network is called a deep neural network (DNN). Figure 7 illustrates the architectures of a deep neural network. The element in each layer is called neuron, we can see that each neuron is connected with all the neurons in the previous and after layer.

At the beginning, the neural network is naive and does not know the function mapping the inputs and outputs. Therefore, we use cost function (also namely loss function) to measure the prediction error. E.g., Equation 5 is the mean squared loss function, which is commonly used in regression and binary classification tasks. Where n is the number of training instance, $\hat{Y}_i$ is the predicted value and Y_i is the true value.

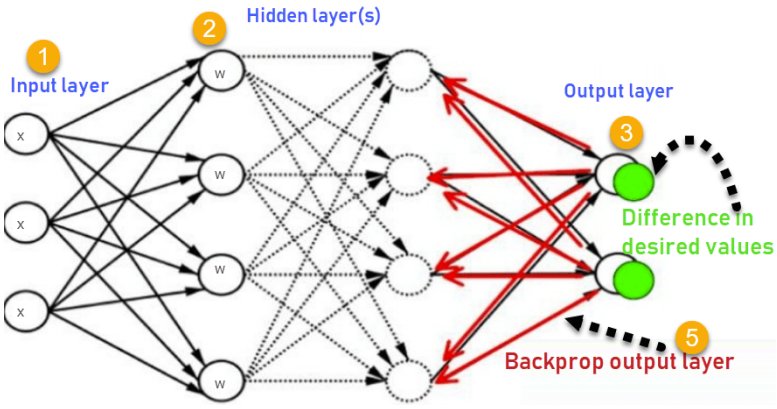
$$\mathrm{MeanSquareLoss} = \frac{1}{n} \sum_{i=1}^{n} \left(Y_i - \hat{Y}_i\right)^2 \tag{5}$$

Figure 7: Example of Deep Neural Network Structure



Source: (Merenda, Porcaro, & Iero, 2020)

Figure 6: Example of Back-propagation



Source: https://www.guru99.com/backpropogation-neural-network.html

By minimizing the loss with respect to the neural network parameters, we can optimize the model and improve the model accuracy. The process of minimize loss is called gradient descent, which finds a global minimum in training deep neural networks despite the objective function being non-convex (Du, Lee, Li, Wang, & Zhai, 2019). There two principals in the training process, which are feed forward and back-propagation. feed forward is the process that data goes through from input layer and it travels all hidden layers and to output layer. When data pass through each neuron, a transformation in that neuron with weight W and bias β will be done for the data. The back-propagation mostly includes the following step according to Figure 6. First, inputs X, arrive through the preconnected path. Second, input is modeled using real weights W The weights are usually randomly selected.

Third, calculate the output for every neuron from the input layer, to the hidden layers, to the output layer. Fourth, calculate the error in the outputs. In the end, travel back from the output layer to the hidden layer to adjust the weights such that the error is decreased. There are many parameters to be tuned for deep neural network such as number of layers, learning rate and dropout rate, etc, which are discussed in Table 4 from Section 4.7 in detail.

4.5 Programming Libraries

The programming language used in this study is Python and some Python based libraries will be applied. In this section, four main libraries that are employed in the analysis will be introduced.

4.5.1 PyTorch

PyTorch is an machine learning framework developed by Facebook's AI Research lab (FAIR). It has been widely used for applications such as computer vision and natural language processing. PyTorch is employed to build up the deep neural network due to its simplicity and efficiency.

PyTorch uses tensor as data format to process and adopts Kaiming/He initialization schemes to determine the neueral network weights. But some other machine learning libraries such as Keras employ Glorot/Xavier initialization schemes (Paszke et al., 2019). So, the results may be slightly different if non-PyTorch platform is used when to replicate.

4.5.2 Scikit-learn

According to (Pedregosa et al., 2011), Scikit-learn is a free machine learning library for Python programming language. It features various regression, classification and clustering machine learning algorithm such as K-means, random forest and gradient boosting, etc,. It is designed to interoperate with the Python numerical and scientific libraries NumPy and SciPy. Scikit-learn is the most commonly used Python library for machine learning. We employ Scikit-learn library for implementing logistic regression model and random forest model.

4.5.3 Imbalanced-learn

According to (Lemaître, Nogueira, & Aridas, 2017), Imbalanced-learn is an open source, MIT-licensed library relying on Scikit-learn and provides tools when dealing with classification with imbalanced classes. The project started in August 2014 by Fernando Nogueira and focused on SMOTE implementation. We Imbalanced-learn for implementing SMOTE to deal with the data imbalanced problem.

4.5.4 Pandas

According to (McKinney et al., 2010), Pandas is a software library written for the Python programming language for data manipulation and analysis. In particular, Pandas provide extremely streamlined forms of data representation. This helps to analyze and understand data better. And one of the best advantages of Pandas is that Pandas provides a huge set of important commands and features

which are used easily analyze data such as filtering and segmenting data according to the preference. We use Pandas library for the data cleaning.

4.6 Performance Measures

Appropriate evaluation metrics are important since all metrics have relevant assumptions on problems. Therefore, it is imperative to choose an evaluation metric that best captures what a specific project considers important about predictions. Since we have imbalanced data,which makes choosing model evaluation metrics challenging. The traditional performance metrics for classification problem do not work in this study. E.g., accuracy, error, sensitivity and specificity (He & Ma, 2013). That is, predictions always tend to be the majority class when we have imbalanced data set. Because this is the easiest way to have a relative higher accuracy. In this section, two performance metrics are introduced to specifically evaluate this study. Section 4.6.1 presents the Matthews correlation coefficient and Section 4.6.2 introduces the method of F-measure. The additional formulas used to derive Matthews correlation coefficient and F-measure such as confusion matrix are all provided in Appendix B.2.

4.6.1 Matthews Correlation Coefficient (MCC)

Matthews correlation coefficient was first introduced by biochemist Brian W. Matthews in 1975 (Matthews, 1975). For binary classification problem, the MCC formula is presented in Equation 6 where MCC is the Matthews correlation coefficient and TP,TN,FP and FN represent true positive, true negative, false positive and false negative. The interval for MCC value is between -1 and +1. perfect misclassification and perfect classification are reached for extreme MCC values -1 and +1. If MCC is 0, it is expected for a prediction no better than random. If MCC value is less than 1, it is expected for a prediction worse than random.

$$\text{MCC} = \frac{TP \times TN - FP \times FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}} \tag{6}$$

When the majority class and minority class are labeled as negative and positive in a imbalanced data set, it is expected to be a good model if the MCC score is between 0 and 1. Compared to receiver operating characteristic(ROC) and area under the receiver operating characteristic(ROC-AUC) measures, MCC score is a better metric in the classification problem when imbalanced problem is met, since it only generates a high score when the model can predict well in all of the four categories in the confusion matrix (true positives, true negatives, false positives and false negatives), in proportion to the the size of positives and negatives in the data. For example, (Chicco & Jurman, 2020) uses a data set that contain patients with with cancer traits to predict who will develop tumor or not. They explored various imbalanced and balanced data and find that MCC score is the only score that indicate a stable and reliable prediction in all situations, while F1 score and accuracy measures sometimes indicate overoptimistic predictions.

4.6.2 F-Measure

In binary classification, F-measure(or F-score) is a measure of a test's accuracy. There are two F-score which are F_1 score and F_β score. The F_1 score is the harmonic mean of the precision and

recall. The more generic F_β score applies additional weights, valuing one of precision or recall more than the other. F-measure is similar to MCC-score. That is, it is also a good performance metric when imbalanced problem met. It is recommended by (Cao, Chicco, & Hoffman, 2020) to use both MCC-score and F-measure in unbalanced classification problem.

I employ F_1 score in the study. The formula for F_1 measure is presented in Equation 7 where F_1 is the F_1 score and precision and recall are calculated by Equation 8 and 9. TP,TN,FP and FN represent true positive, true negative, false positive and false negative. F_1 score is ranging from 0 to 1 where 0 means perfect misclassification and 1 means perfect classification. Similarly, when we have imbalanced data set, if the minority class is labeled as positive and with more interest, the higher F_1 score represents the better prediction performance.

$$F_1 = \frac{2}{\text{recall}^{-1} + \text{precision}^{-1}} = 2\frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}} = \frac{TP}{TP + \frac{1}{2}(\text{FP} + \text{FN})} \tag{7}$$

$$Precision = \frac{TP}{TP + FP} \tag{8}$$

$$Precision = \frac{TP}{TP + FN} \tag{9}$$

4.7 Implementation

As described above, LendingClub's user data is divided into a training set(loans issued during 2016-2017) and a test set(loans issued in 2018). After a series of data cleaning and feature engineering(from Section 3.2), I built a logistic regression model((introduced in Section 4.2) and a random forest model(introduced in Section 4.3) using the Scikit-learn library(introduced in Section 4.5.2). Then use the Pytorch library(introduced in Section 4.5.1) to build a deep neural network model(introduced in Section 4.4). For each model, a series of parameter adjustments are carried out. Table 4 provides hyperparameter of multiple models in detail. As Table 4 presents, different algorithms have different parameter adjustments. This study uses three models, and the final resulting model parameters are chosen based on model performance.

For logistic regression, it is relatively easy to adjust parameters. We just need to give whether or not to penalize logistic regression. L1 and L2 denote LASSO and Ridge adjustments for logistic regression, respectively. Since the final model performances are not very different for hyperparameter, the resulting parameter is not to penalize logistic regression. In addition, the duration to run the code for logistic regression is not long. With the amount of data and computer configuration (introduced in Table 3) in this study, it can be completed in about one minute.

For the random forest model, the most important parameter is to control the number of decision trees and the deepest decision level. The trade off here is computation time and model accuracy. Here, the maximum depth level of the resulting model is none. That is, until you can no longer make decisions. However, this also results in a slightly longer computation time for our resulting model. The whole process takes about 10 minutes. To test all combinations of parameters of random forest, about 2 hours are needed.

For the deep neural network model, there are relatively many parameters that can be adjusted. The adjustment of parameters for different types of data affects the results (Smith, 2018). For example, for regression problems, a linear activation function may work better than a nonlinear one. A

relatively small learning rate has a normative effect, etc. In actual operation, the selection of epoch and the selection of the number of hidden layers and the number of neurons are more based on experience and a large number of trials. In this paper, the parameter tuning of the neural network takes the longest time. Because each set of parameters needs to be tested while controlling other parameters unchanged. The average duration of each test is about 15 to 20 minutes. To test all combinations of parameters of deep neural network, About 15 hours are needed. Finally, all optimal model results are reported in Chapter 5.

Table 4: Hyperparameter of Multiple Models

Model	**Hyperparameter**	**Parameter Space**	**Selected Hyperparameter**
Logistic regression	Penalization	L1,L2,none	none
Random Forest	Max depth	3,4,5,none	none
	Number of trees	10,20,30,auto	auto
	Max depth	3,4,5,none	none
	Bootstrap	True&False	True
Deep neural network	Batch size	256,512,1024,2048,5096	2048
	Learning rate	0.001-0.1	0.01
	Optimizer	SGD, Adam	Adam
	Epoch	50,100,200,300	100
	Activation function	ReLU,sigmoid,TanH	ReLU
	Dropout rate	0.05-0.4	0.1
	Layer&Neuron	(300,200,100),(200,100)	(200,100)

5 Results

In this chapter, the main analysis aims to predict default action with two parts. First, I compare the performance metrics across various algorithms in Section 5.1. Second, I analyze the feature importance of the model and present it in Section 5.2.

5.1 Model Performance

In this section, I use three algorithms (logistic regression, random forest and deep neural network) and then compare the performance metrics across these models. The detailed methodology descriptions for each model and each performance measure are provided in Section 4.

To have a comprehensive comparison, I generate four types of data which differ by whether it excludes outliers and/or conduct SMOTE techniques, with the detailed definitions in Section 3.2 and Section 4. I then apply the algorithms on these data sets and present various performance scores in Table 5.

In columns 1 and 2, I use "Y" and "N" to indicate whether the given sample applies SMOTE process or/and include outliers or not. The main performance metrics are Matthews Correlation Coefficient (MCC) and F_1 score, indicated in columns 3 and 4. For further comparison with the literature, I report the ROC-AUC score and ROC curves as well in column 5. As for confusion matrix of all models, they are reported in Appendix B.3.

A few conclusions can be drawn from the table. First, models applying to a data set with SMOTE processing yields better performance metrics compared to that without SMOTE process. All model performances (MCC score) are improved when SMOTE is applied. For example, when considering a data set with outliers (rows having column 2 with "Y") and logistic regression algorithm, MCC score is 0.22 (Logistic regression(1)) when applying SMOTE techniques vs 0.14 (Logistic regression(2)) without SMOTE techniques. Among them, the improvement of the deep neural network model is the most, increasing from 0.04 (Deep neural network(4)) to 0.23 (Deep neural network(3)), or from 0.1 (Deep neural network(2)) to 0.32 (Deep neural network(1)). Similar improvement can be found with F_1-score. This finding highlights the importance of using SMOTE techniques in the analysis.

Second, performance metrics are not sensitive to exclusion or inclusion of outliers. Across all models, the performance metrics are similar whether we include outliers or not. Given the nature of outliers, I decide to rely on models without outliers as baseline references, and conclude that models applying to a data set with have applied SMOTE processing yields the best performance metrics based on MCC score and F_1 score. Across all data sets that exclude outliers and are applied SMOTE techniques, the MCC score varies from 0.21 (Logistic regression (3)) to 0.23 (Deep neural network (3)), higher than performance metrics in other data types without outliers.

Third, ROC-AUC yields different conclusions compared to MCC score and F_1 score. For example, when examining deep neutral network algorithm(the confusion matrix for all models in Table 5 are reported in Appendix B.3), the model with highest performance would be the Deep neural network (2) with ROC-AUC of 0.9. However, this is different from the conclusion drawn from MCC score and F_1 score. While investigating the confusion matrix of the algorithm, it can be seen that the high ROC-AUC measure is due to a high specificity but a low sensitivity. Since the target variable of the sample is extremely unbalanced, while ROC-AUC score attaches the majority and the minority class the same importance, it might lead to improper conclusion that is not suitable to our analysis, in which, the minority class $Charged\_Off$ is more of interests. On this ground, F_1 score and MCC

score would better fit to evaluation the question compared to ROC-AUC score.

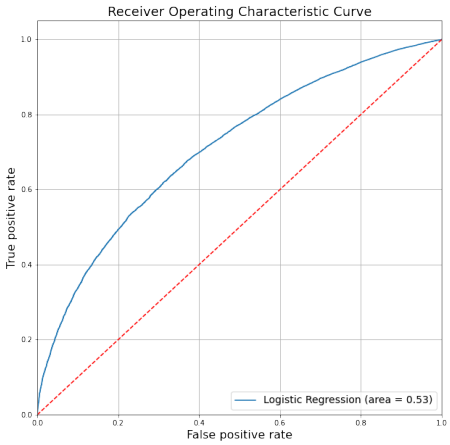
To further investigate the ROC-AUC score, I also report the ROC curve with ROC-AUC score for six models to further check the performance differences in Figure 8. It is clear that the ROC curve of the model deep neural network(2) towards to the up left the most, which also leads to the largest ROC-AUC score. We may conclude that deep neural network(2) is the best model among all. However, deep neural network(2) has a very low MCC-score and F_1-score, which means this model is not good at predicting the minority class($Charged\_Off$ in this case. Thus, relying on ROC-AUC might generate wrong conclusions.

To summarize, SMOTE mitigates the imbalanced data problem and improves the predictive power in loan default detection. Deep neural network algorithm with SMOTE performs the best in loan default prediction. It is hard to compare the model performance with other relevant literature since they refer to different periods. E,g., (Carmichael, 2014) select 2007-2013 and (J.-Y. Kim & Cho, 2019a) select 2007-2017. But in general, the performance of the optimal model in this study perform pretty good if we must compare with other relevant literature.

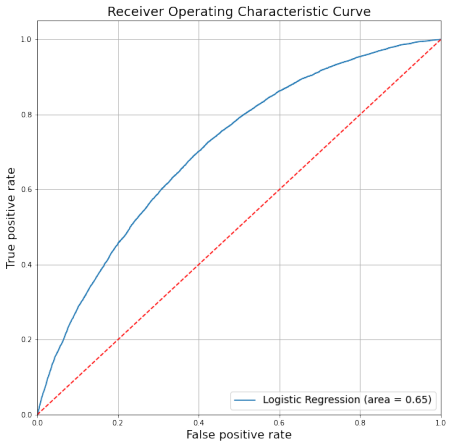
Table 5: Model Performance Table

Model	**SMOTE** (1)	**Outliers** (2)	**MCC score** (3)	**F1 score** (4)	**ROC-AUC** (5)
Logistic regression(1)	Y	Y	0.22	0.37	0.53
Logistic regression(2)	N	Y	0.14	0.15	0.65
Logistic regression(3)	Y	N	0.21	0.36	0.54
Logistic regression(4)	N	N	0.02	0.01	0.61
Random forest(1)	Y	Y	0.16	0.25	0.57
Random forest(2)	N	Y	0.14	0.15	0.54
Random forest(3)	Y	N	0,14	0.22	0.52
Random forest(4)	N	N	0.12	0.12	0.51
Deep neural network(1)	Y	Y	**0.32**	**0.44**	0.78
Deep neural network(2)	N	Y	0.1	0.07	0.90
Deep neural network(3)	Y	N	0.23	0.38	0.76
Deep neural network(4)	N	N	0.04	0.02	0.89

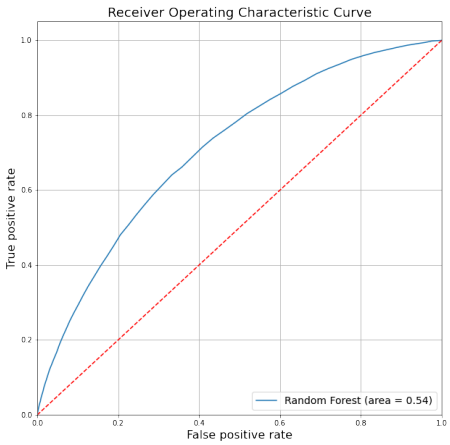
Figure 8: ROC-AUC of Various Models



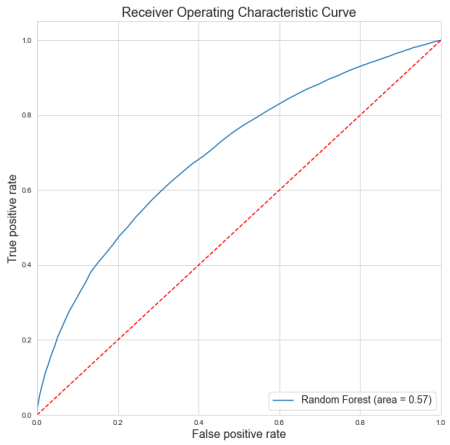
Logistic regression(2)



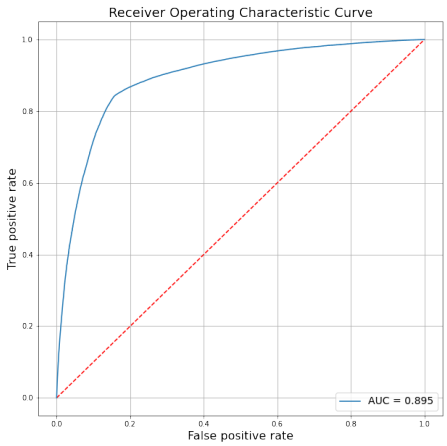
Logistic regression(1)



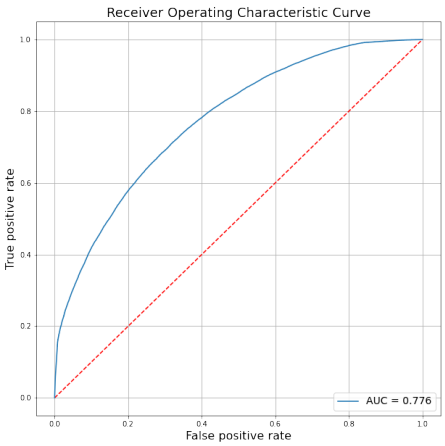
Random forest(2)



Random forest(1)



Deep neural network(2)



Deep neural network(1)

5.2 Feature Importance

After applying one-hot encoding for all categorical features, there are 118 features for model implementation in the end. It is not useful to interpret all 118 feature effects on default. So, only the top 10 important features for the specific model are reported based on the permutation feature importance test. In addition, outliers have little impact on final performance as we have discussed in Section 5.1. But applying SMOTE does improve model performance by solving imbalanced data problem. So, we select Logistic regression(1), Random forest(1) and Deep neural network(1) from Table 5 as baseline models to do permutation feature importance test. That is, three models are with outliers and applied SMOTE. The results for three models are reported in Figure 9, Figure 10 and Figure 11.

Features are descending sorted in the three plots based on permutation importance values. The permutation importance value on the horizontal axis represents how much accuracy will decrease compared to the baseline model if that specific feature is randomly removed from the baseline model. E.g., the permutation importance value of dti in Figure 9 is 0.029, which means that the logistic model accuracy will drop 2.9% on average if dti is removed from the model. Therefore, the higher the permutation importance value is, the more important the specific feature contribute to the model performance. According to the three permutation feature importance plots, there are some features overlap such as int_rate, $fico\_score$, $loan\_amnt$ and dti. But some features are unique in specific models. E.g., the categorical feature *home_ownership* with its category *RENT* ranks very high in random forest model but not appears in the other two models.

In general, the average magnitude of permutation importance value is the largest in deep neural network(1) model. The top 5 feature importance values in Deep neural network(1) are greater than 2%. Random forest(1) model have the smallest average permutation importance value where most of feature values are lower than 1%. This also explains that Deep neural network(1) model perform the best through all model implemented.

To conclude from the feature importance results, the loan grading system(*sub_grade*) and calculated fico score($fico\_score$) by LendingClub platform are relative useful in filtering risky borrowers. From the borrowers' side, some key financial indicator such as dti and $annual\_inc$ strongly affect default action. At last, int_rate is always one of the most important roles in lending and default action.

The feature importance results are align with the findings by (Serrano-Cinca, Gutiérrez-Nieto, & López-Palacios, 2015). They also find that loan grade calculated by LendingClub has clear relationship with probability of default. And interest rate assigned depends on the grade assigned, which is also a strong predictor. However, (Emekter et al., 2015) find that revolving credit utilization explains loan default, which is not shown in our results. It is difficult to say that the results are not align since they refer to different period of data(2007-2012).

Figure 9: Permutation Feature Importance of Logistic Regression(1)

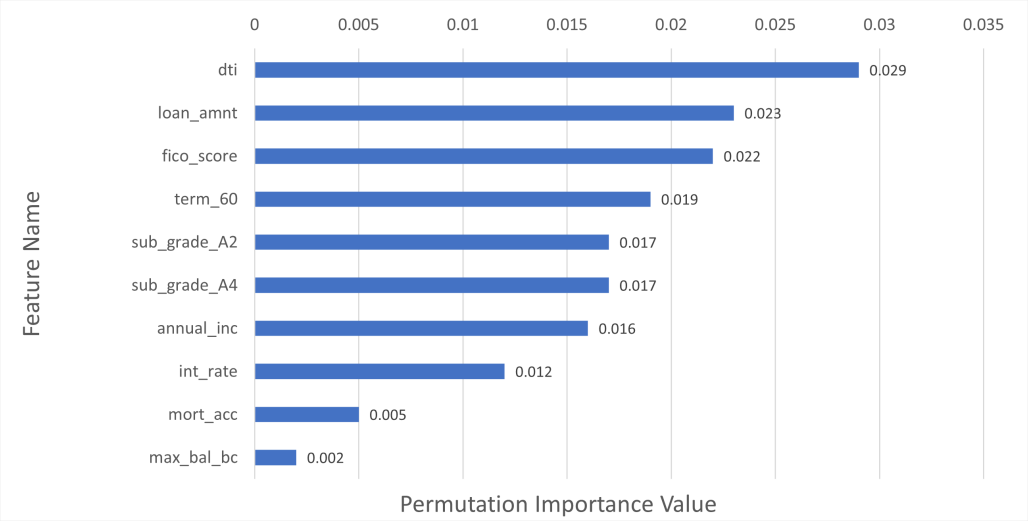


Figure 10: Permutation Feature Importance of Random Forest(1)

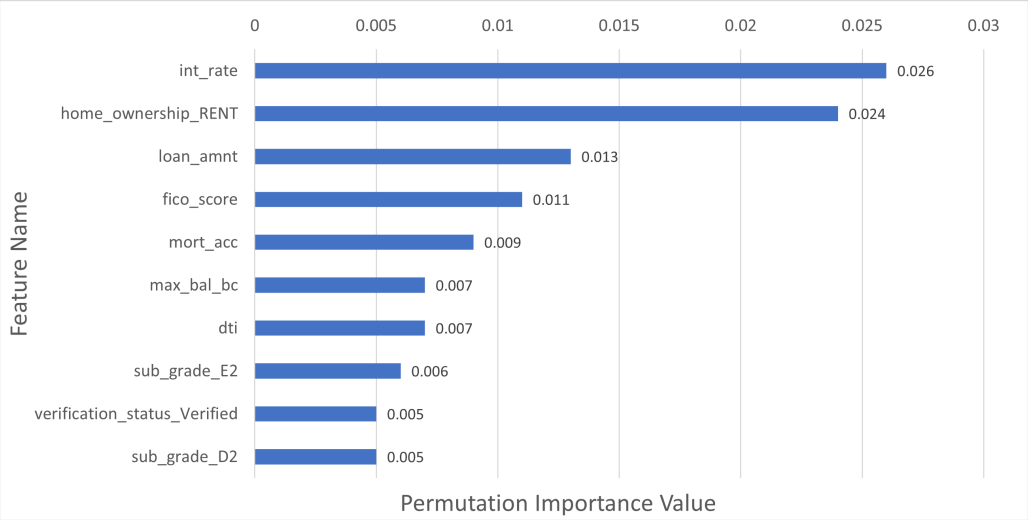
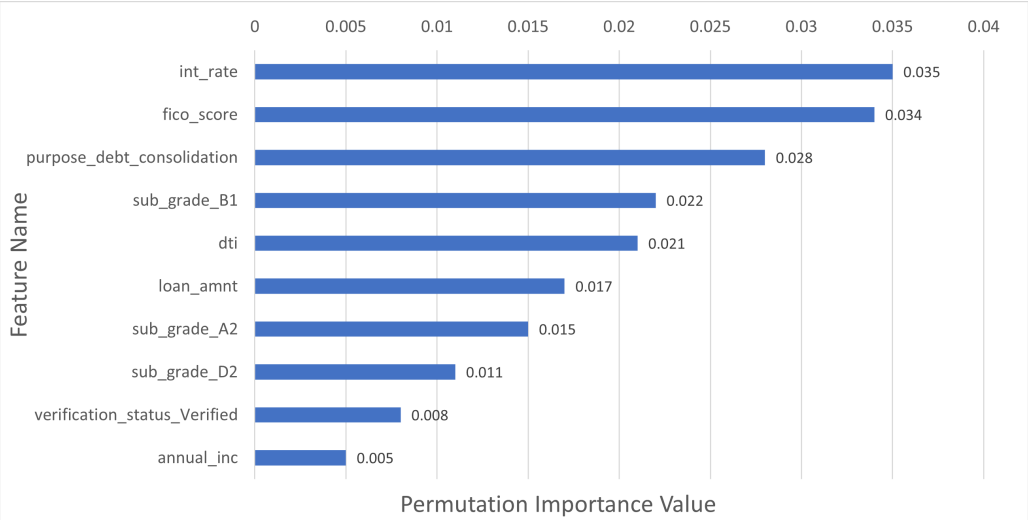


Figure 11: Permutation Feature Importance of Deep Neural Network(1)



6 Conclusion

Compared with traditional bank lending, online P2P lending utilizes the information from both borrowers and lenders more efficiently and realizes transactions with lower transaction cost.With the improvement of the regulatory system and artificial intelligence application, online P2P lending will develop healthier and play a more important role in the financial market.

In this thesis, I manage to deal with unbalanced credit default data from LendingClub and successfully built up various machine learning models to predict borrowers default. Only pre-loan information is used and several data clean and feature engineering process are implemented. The final results provide two main aspect information. First, deep neural network model performs the best compared to logistic regression and random forest model in predicting loan default action. Second, the top important features that affect default prediction are analysis. The loan grading and FICO score calculated by LendingClub are good indicators for investors to identify quality of a loan. In addition, interest rate and borrowers' debt to income ratio also affect loan default a lot.

To my best knowledge, this is the first paper adopting Matthews Correlation Coefficient (MCC) as performance metric on LendingClub default analysis. Other relevant studies on LendingClub default prediction adopt accuracy or ROC-AUC as their main performance measure, which is critical since the minority class(default) should with more interest. Overall, this thesis contributes to a better understanding of the literature with an improved predicting method, and it contributes to a better feature selection method, especially in the spirit of timing of the features.

For the future study, I may approach two aspects. First, I might use a more customized performance measure, such as not simply calculating whether defaults can be accurately predicted, but adding a cost-benefit analysis to the performance measure. Second, I may study how some behaviors of borrowers after getting a loan affect default behavior. In this way, a more dynamic loan risk alert control mechanism may be established.

References

Bachmann, A., Becker, A., Buerckner, D., Hilker, M., Kock, F., Lehmann, M., ... Funk, B. (2011). Online peer-to-peer lending-a literature review. *Journal of Internet Banking and Commerce*, *16*(2), 1.

Basha, S. A., Elgammal, M. M., & Abuzayed, B. M. (2021). Online peer-to-peer lending: A review of the literature. *Electronic Commerce Research and Applications*, *48*, 101069.

Benesty, J., Chen, J., Huang, Y., & Cohen, I. (2009). Pearson correlation coefficient. In *Noise reduction in speech processing* (pp. 1–4). Springer.

Breiman, L. (2001). Random forests. *Machine learning*, *45*(1), 5–32.

Calabrese, R., Osmetti, S. A., & Zanin, L. (2019). A joint scoring model for peer-to-peer and traditional lending: a bivariate model with copula dependence. *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, *182*(4), 1163–1188.

Cao, C., Chicco, D., & Hoffman, M. M. (2020). The mcc-f1 curve: a performance evaluation technique for binary classification. *arXiv preprint arXiv:2006.11278*.

Carmichael, D. (2014). Modeling default for peer-to-peer loans. *Available at SSRN 2529240*.

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, *16*, 321–357.

Chicco, D., & Jurman, G. (2020). The advantages of the matthews correlation coefficient (mcc) over f1 score and accuracy in binary classification evaluation. *BMC genomics*, *21*(1), 1–13.

Chorzempa, M., & Huang, Y. (2022). Chinese fintech innovation and regulation. *Asian Economic Policy Review*.

Dhaigude, R., & Lawande, N. (2022). Impact of artificial intelligence on credit scores in lending process. In *2022 interdisciplinary research in technology and management (irtm)* (pp. 1–5).

Du, S., Lee, J., Li, H., Wang, L., & Zhai, X. (2019). Gradient descent finds global minima of deep neural networks. In *International conference on machine learning* (pp. 1675–1685).

Duan, J. (2019). Financial system modeling using deep neural networks (dnns) for effective risk assessment and prediction. *Journal of the Franklin Institute*, *356*(8), 4716–4731.

Emekter, R., Tu, Y., Jirasakuldech, B., & Lu, M. (2015). Evaluating credit risk and loan performance in online peer-to-peer (p2p) lending. *Applied Economics*, *47*(1), 54–70.

Fu, Y. (2017). Combination of random forests and neural networks in social lending. *Journal of Financial Risk Management*, *6*(4), 418–426.

Haykin, S. (2009). *Neural networks and learning machines, 3/e*. Pearson Education India.

He, H., & Ma, Y. (2013). Imbalanced learning: foundations, algorithms, and applications.

Jiang, C., Wang, Z., Wang, R., & Ding, Y. (2018). Loan default prediction by combining soft information extracted from descriptive text in online peer-to-peer lending. *Annals of Operations Research*, *266*(1), 511–529.

Kelly, R., O'Toole, C., et al. (2016). *Lending conditions and loan default: What can we learn from uk buy-to-let loans?* (Tech. Rep.). Central Bank of Ireland.

Kim, H., Cho, H., & Ryu, D. (2020). Corporate default predictions using machine learning: Literature review. *Sustainability*, *12*(16), 6325.

Kim, J.-Y., & Cho, S.-B. (2019a). Predicting repayment of borrows in peer-to-peer social lending with deep dense convolutional network. *Expert Systems*, *36*(4), e12403.

Kim, J.-Y., & Cho, S.-B. (2019b). Towards repayment prediction in peer-to-peer social lending using deep learning. *Mathematics*, *7*(11), 1041.

Kumar, M., Goel, V., Jain, T., Singhal, S., & Goel, L. (2018). Neural network approach to loan default prediction. *International Research Journal of Engineering and Technology (IRJET)*, *5*(4), 4–7.

Lemaître, G., Nogueira, F., & Aridas, C. K. (2017). Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning. *Journal of Machine Learning Research*, *18*(17), 1-5. Retrieved from http://jmlr.org/papers/v18/16-365.html

Li, X., Ergu, D., Zhang, D., Qiu, D., Cai, Y., & Ma, B. (2022). Prediction of loan default based on multi-model fusion. *Procedia Computer Science*, *199*, 757–764.

Li, Z., Li, S., Li, Z., Hu, Y., & Gao, H. (2021). Application of xgboost in p2p default prediction. In *Journal of physics: Conference series* (Vol. 1871, p. 012115).

Matthews, B. W. (1975). Comparison of the predicted and observed secondary structure of t4 phage lysozyme. *Biochimica et Biophysica Acta (BBA)-Protein Structure*, *405*(2), 442–451.

McKinney, W., et al. (2010). Data structures for statistical computing in python. In *Proceedings of the 9th python in science conference* (Vol. 445, pp. 51–56).

Merenda, M., Porcaro, C., & Iero, D. (2020). Edge machine learning for ai-enabled iot devices: A review. *Sensors*, *20*(9), 2533.

Midi, H., Sarkar, S. K., & Rana, S. (2010). Collinearity diagnostics of binary logistic regression model. *Journal of interdisciplinary mathematics*, *13*(3), 253–267.

Milne, A., & Parboteeah, P. (2016). The business models and economics of peer-to-peer lending.

Namvar, A., Siami, M., Rabhi, F., & Naderpour, M. (2018). Credit risk prediction in an imbalanced social lending environment. *arXiv preprint arXiv:1805.00801*.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., ... Chintala, S. (2019). Pytorch: An imperative style, high-performance deep learning library. In *Advances in neural information processing systems 32* (pp. 8024–8035). Curran Associates, Inc. Retrieved from http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830.

Rousseeuw, P. J., & Hubert, M. (2011). Robust statistics for outlier detection. *Wiley interdisciplinary reviews: Data mining and knowledge discovery*, *1*(1), 73–79.

Serrano-Cinca, C., Gutiérrez-Nieto, B., & López-Palacios, L. (2015). Determinants of default in p2p lending. *PloS one*, *10*(10), e0139427.

Smith, L. N. (2018). A disciplined approach to neural network hyper-parameters: Part 1–learning rate, batch size, momentum, and weight decay. *arXiv preprint arXiv:1803.09820*.

Tiwari, A. K. (2018). Machine learning application in loan default prediction. *Machine Learning*, *4*(5).

Wang, H., Chen, K., Zhu, W., & Song, Z. (2015). *A process model on p2p lending* (Vol. 1) (No. 1). Springer.

Wang, H., Greiner, M., & Aronson, J. E. (2009). People-to-people lending: The emerging e-commerce transformation of a financial market. In *Sigebiz track of the americas conference on information systems* (pp. 182–195).

Wardrop, R., Rosenberg, R., Zhang, B., Ziegler, T., Squire, R., & Burton, J. (2016). Breaking new ground. *The Americas alternative finance*.

Xia, Y., He, L., Li, Y., Liu, N., & Ding, Y. (2020). Predicting loan default in peer-to-peer lending using narrative data. *Journal of Forecasting*, *39*(2), 260–280.

Xia, Y., Liu, C., Da, B., & Xie, F. (2018). A novel heterogeneous ensemble credit scoring model based on bstacking approach. *Expert Systems with Applications*, *93*, 182–199.

Yao, J., Chen, J., Wei, J., Chen, Y., & Yang, S. (2019). The relationship between soft information in loan titles and online peer-to-peer lending: evidence from renrendai platform. *Electronic Commerce Research*, *19*(1), 111–129.

Ying, L. (2018). Research on bank credit default prediction based on data mining algorithm. *The International Journal of Social Sciences and Humanities Invention*, *5*(6), 4820–4823.

Zhang, W., Wang, C., Zhang, Y., & Wang, J. (2020). Credit risk evaluation model with textual features from loan descriptions for p2p lending. *Electronic Commerce Research and Applications*, *42*, 100989.

Zhou, J., Li, W., Wang, J., Ding, S., & Xia, C. (2019). Default prediction in p2p lending from high-dimensional data based on machine learning. *Physica A: Statistical Mechanics and its Applications*, *534*, 122370.

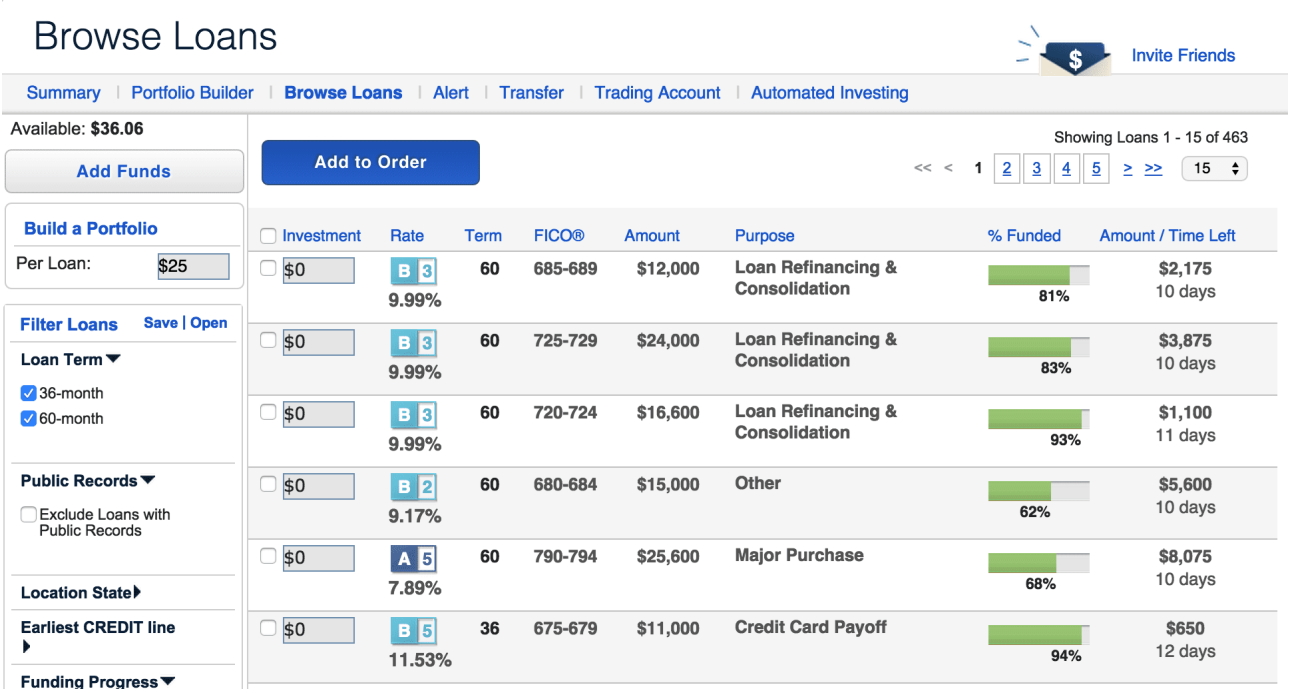
Zhu, L., Qiu, D., Ergu, D., Ying, C., & Liu, K. (2019). A study on predicting loan default based on the random forest algorithm. *Procedia Computer Science*, *162*, 503–513.

Appendices

There are two parts of of the Appendices. Appendix A provides additional information about data and Appendix B provides additional information about techniques.

A Additional Data Information

A.1 Application Screenshot



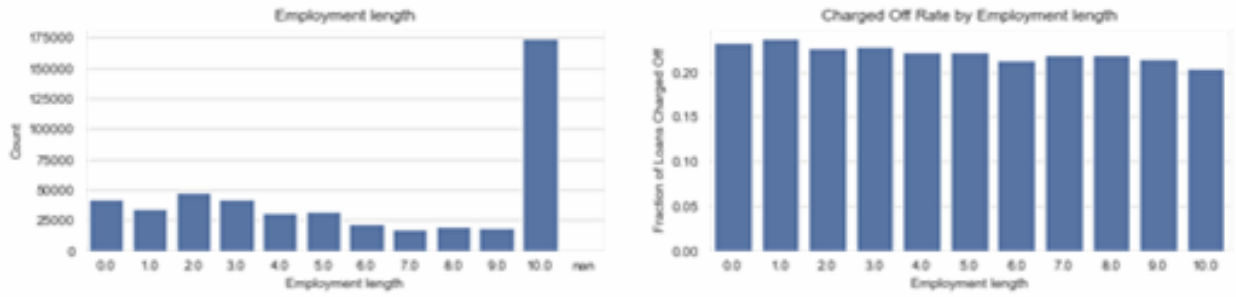
Source: https://www.lendacademy.com/lending-club-review/

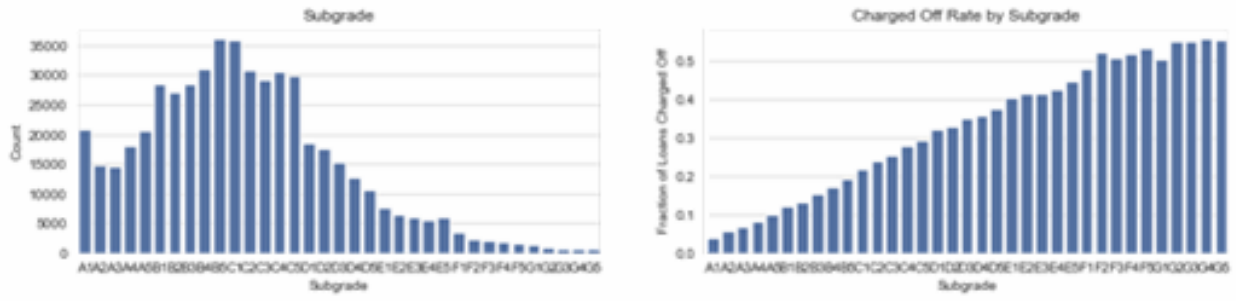
A.2 Description of Selected Features in The Analysis

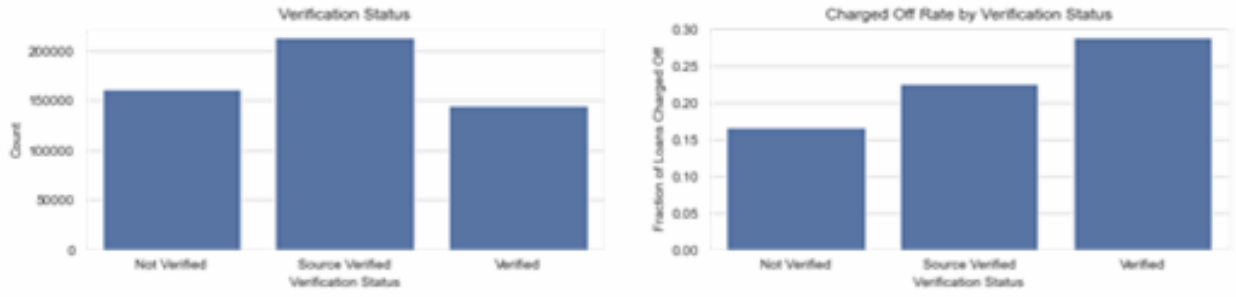
Variable Name	Definition
dti	A ratio calculated using the borrower's total monthly debt payments on the total debt obligations, excluding mortgage and the requested LC loan, divided by the borrower's self-reported monthly income.
loan_amnt	The listed amount of the loan applied for by the borrower. If at some point in time, the credit department reduces the loan amount, then it will be reflected in this value.
int_rate	Interest Rate on the loan.
fico_range_low	The lower boundary range the borrower's FICO at loan origination belongs to.
fico_range_high	The upper boundary range the borrower's FICO at loan origination belongs to.
fico_score	Constructed variable, which is the average score of *fico_range_high* and *fico_range_low*.
term	The number of payments on the loan. Values are in months and can be either 36 or 60.
installment	The monthly payment owed by the borrower if the loan originates.
sub_grade	LendingClub assigned loan subgrade
emp_length	Corrected version: Employment length in years. (The actual data does not match the description from LendingClub data dictionary, which is possible values are between 0 and 10 where 0 means less than one year and 10 means ten or more years.)
home_ownership	The home ownership status provided by the borrower during registration or obtained from the credit report. Values are: RENT, OWN, MORTGAGE, OTHER
annual_inc	The self-reported annual income provided by the borrower during registration.
verification_status	Indicates if income was verified by LC, not verified, or if the income source was verified.
purpose	A category provided by the borrower for the loan request. Possible values are: wedding, credit card repayment, mortgage repayment and student loan repayment.

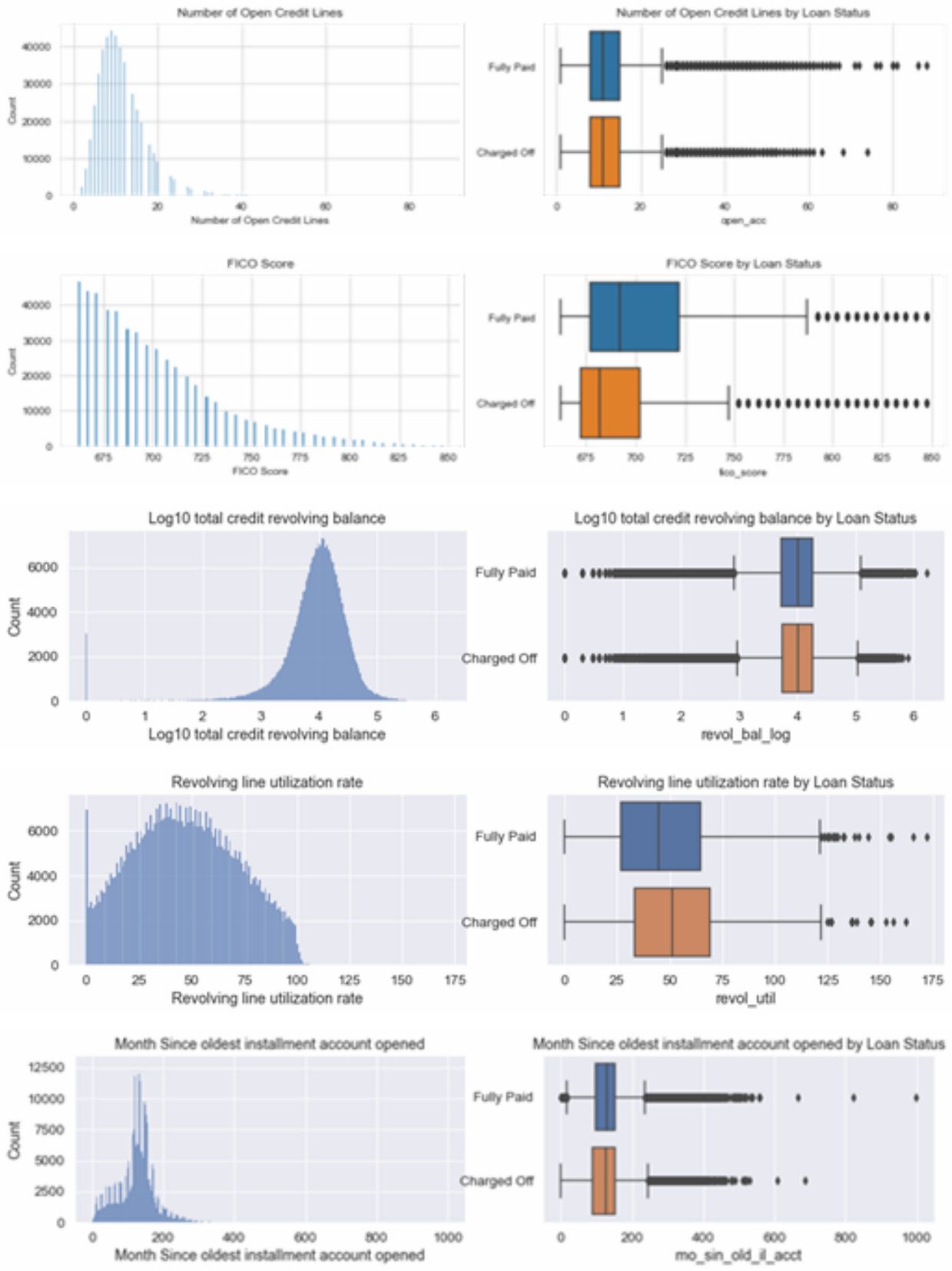
Variable Name	Definition
addr_stat	The state provided by the borrower in the loan application.
total_acc	The total number of credit lines currently in the borrower's credit file.
pub_rec_bankruptcies	Number of public record bankruptcies.
mort_acc	Number of mortgage accounts.
mo_sin_old_rev_tl_op	Months since oldest revolving account opened.
mo_sin_old_il_acct	Months since oldest bank installment account opened.
total_cu_tl	Number of finance trades.
inq_fi	Number of personal finance inquiries.
all_util	Balance to credit limit on all trades.
max_bal_bc	Maximum current balance owed on all revolving accounts.
revol_bal	Total credit revolving balance.
revol_util	Revolving line utilization rate, or the amount of credit the borrower is using relative to all available revolving credit.
il_util	Ratio of total current balance to high credit/credit limit on all install acct.
total_bal_il	Total current balance of all installment accounts.
initial_list_status	The initial listing status of the loan. Possible values are – W, F.

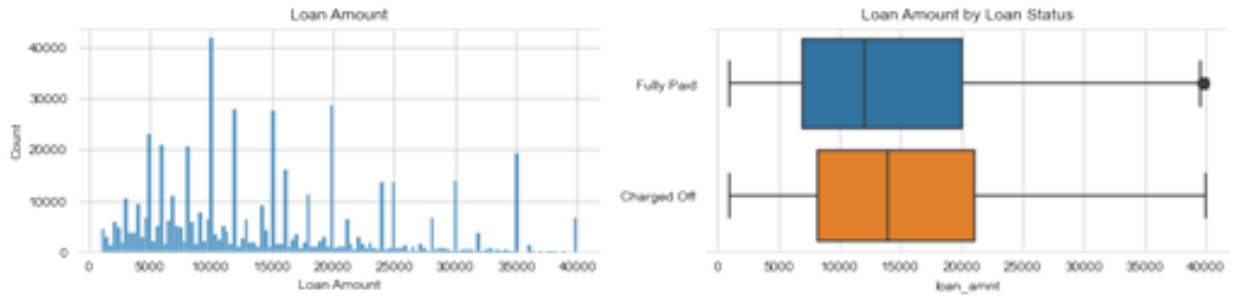
A.3 Distribution and Box Plots of Multiple Features

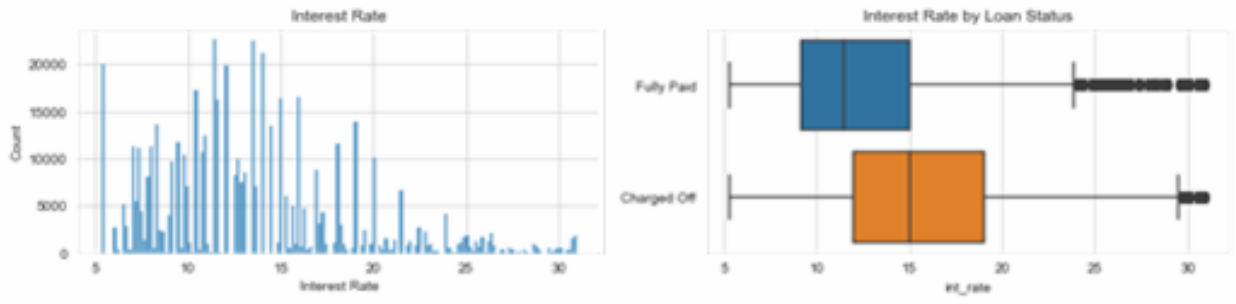


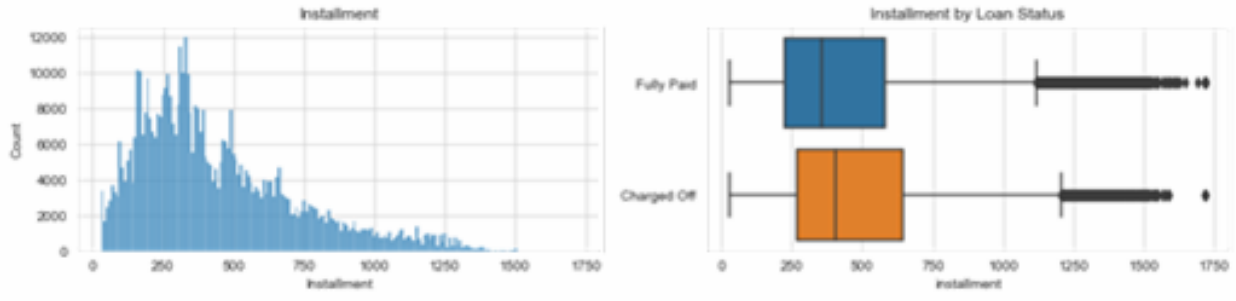


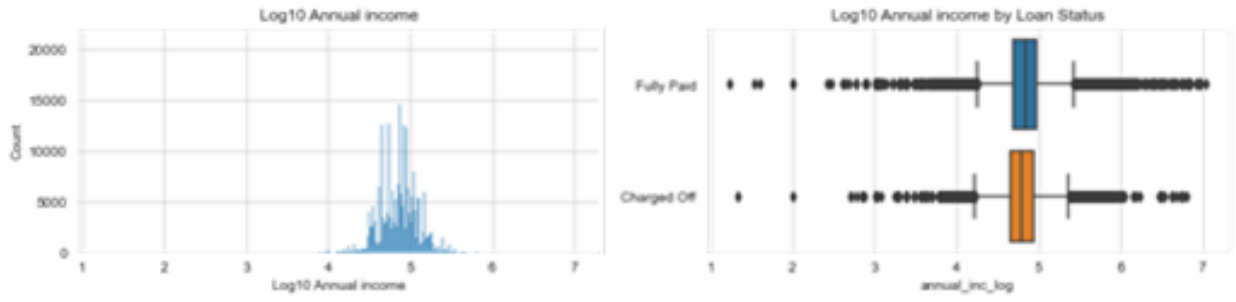


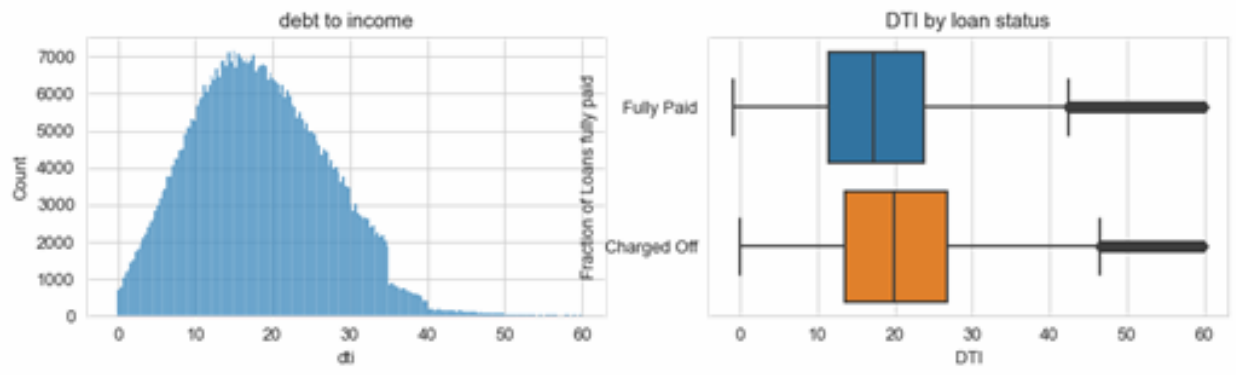

Number of Open Credit Lines
Count
Number of Open Credit Lines
Number of Open Credit Lines by Loan Status
Fully Paid
Charged Off
open_acc
FICO Score
Count
FICO Score
FICO Score by Loan Status
Fully Paid
Charged Off
fico_score
Log10 total credit revolving balance
Count
Log10 total credit revolving balance
Log10 total credit revolving balance by Loan Status
Fully Paid
Charged Off
revol_bal_log
Revolving line utilization rate
Count
Revolving line utilization rate
Revolving line utilization rate by Loan Status
Fully Paid
Charged Off
revol_util
Month Since oldest installment account opened
Count
Month Since oldest installment account opened
Month Since oldest installment account opened by Loan Status
Fully Paid
Charged Off
mo_sin_old_il_acct


Loan Amount
Count
Loan Amount
Loan Amount by Loan Status
Fully Paid
Charged Off
loan_amnt


Interest Rate
Count
Interest Rate
Interest Rate by Loan Status
Fully Paid
Charged Off
int_rate


Installment
Count
Installment
Installment by Loan Status
Fully Paid
Charged Off
installment


Log10 Annual income
Count
Log10 Annual income
Log10 Annual income by Loan Status
Fully Paid
Charged Off
annual_inc_log


debt to income
Count
dti
Fraction of Loans fully paid
DTI by loan status
Fully Paid
Charged Off
DTI

B Additional Technique Information

B.1 SMOTE Algorithm

Algorithm *SMOTE*(T, N, k)
Input: Number of minority class samples T; Amount of SMOTE $N\%$; Number of nearest neighbors k
Output: $(N/100)$ * T synthetic minority class samples

(∗ *If N is less than 100%, randomize the minority class samples as only a random percent of them will be SMOTEd.* ∗)
if $N < 100$
  then Randomize the T minority class samples
    $T = (N/100) * T$
    $N = 100$
endif
$N = (int)(N/100)$ (∗ *The amount of SMOTE is assumed to be in integral multiples of 100.* ∗)
k = Number of nearest neighbors
$numattrs$ = Number of attributes
$Sample[\,][\,]$: array for original minority class samples
$newindex$: keeps a count of number of synthetic samples generated, initialized to 0
$Synthetic[\,][\,]$: array for synthetic samples
(∗ *Compute k nearest neighbors for each minority class sample only.* ∗)
for $i \leftarrow 1$ **to** T
    Compute k nearest neighbors for i, and save the indices in the $nnarray$
    Populate(N, i, $nnarray$)
endfor

Populate(N, i, $nnarray$) (∗ *Function to generate the synthetic samples.* ∗)
while $N \neq 0$
    Choose a random number between 1 and k, call it nn. This step chooses one of the k nearest neighbors of i.
    for $attr \leftarrow 1$ **to** $numattrs$
      Compute: $dif = Sample[nnarray[nn]][attr] - Sample[i][attr]$
      Compute: gap = random number between 0 and 1
      $Synthetic[newindex][attr] = Sample[i][attr] + gap * dif$
    endfor
    $newindex{+}{+}$
    $N = N - 1$
endwhile
return (∗ *End of Populate.* ∗)
End of Pseudo-Code.

Source: (Chawla et al., 2002)

B.2 Additional Performance Measures

Confusion matrix is a specific table layout that allows visualization of the performance of an algorithm. There are four terminologies of a confusion matrix.

- True positive (TP): A test result that correctly indicates the presence of a condition or characteristic.
- True negative (TN): A test result that correctly indicates the absence of a condition or characteristic.
- False positive (FP): A test result which wrongly indicates that a particular condition or attribute is present.
- False negative (FN): A test result which wrongly indicates that a particular condition or attribute is absent.

A typical confusion matrix table and relevant performance measures are presented as following.

		True diagnosis		
		Positive	Negative	Total
Predict	Positive	TP	FP	$TP+FP$
	Negative	FN	TN	$FN+TN$
	Total	$TP+FN$	$FP+TN$	N

Metric	Explaination	Formula
AUC	AUC is the probability that a random chosen positive instance will be ranked ahead of a randomly chosen negative instance.	$\frac{\Sigma_1^{N_{D0}} \Sigma_1^{N_{D1}} S(D_0, D_1)}{N_{D0} * N_{D1}}$
Sensitivity	Sensitivity is 1 - Type I error. It is the rate at which a positive prediction is indeed positive.	$\frac{TP}{TP+FN}$
Specificity	Specificity is 1 - Type II error. It is the rate at which a negative prediction is indeed negative.	$\frac{TN}{FP+TN}$
Accuracy	Overall accuracy is the ratio of all correct predictions to all predictions made.	$\frac{TP+TN}{TP+FP+TN+FN}$

B.3 Confusion Matrix of Multiple Models

		True Diagnosis	
		Positive	Negative
Predict	Positive	5,650	3,178
	Negative	16,242	30,991

Logistic regression(1)

		True Diagnosis	
		Positive	Negative
Predict	Positive	765	8,063
	Negative	988	46,245

Logistic regression(2)

		True Diagnosis	
		Positive	Negative
Predict	Positive	4,625	2,553
	Negative	13,908	25,208

Logistic regression(3)

		True Diagnosis	
		Positive	Negative
Predict	Positive	39	7,139
	Negative	83	39,033

Logistic regression(4)

		True Diagnosis	
		Positive	Negative
Predict	Positive	1,677	7,151
	Negative	3,142	44,091

Random forest(1)

		True Diagnosis	
		Positive	Negative
Predict	Positive	829	7,999
	Negative	1,064	46,169

Random forest(2)

		True Diagnosis	
		Positive	Negative
Predict	Positive	1,155	6,023
	Negative	2,312	36,804

Random forest(3)

		True Diagnosis	
		Positive	Negative
Predict	Positive	484	6,694
	Negative	615	38,501

Random forest(4)

		True Diagnosis	
		Positive	Negative
Predict	Positive	5,737	3,091
	Negative	11,263	34,970

Deep neural network(1)

		True Diagnosis	
		Positive	Negative
Predict	Positive	310	8,518
	Negative	371	46,862

Deep neural network(2)

		True Diagnosis	
		Positive	Negative
Predict	Positive	4,202	2,976
	Negative	11,019	28,097

Deep neural network(3)

		True Diagnosis	
		Positive	Negative
Predict	Positive	75	7,103
	Negative	114	39,002

Deep neural network(4)