8-20-2021 10:00 AM

Credit Risk Measurement and Application based on BP Neural Networks

Jingshi Luo, *The University of Western Ontario*

Supervisor: Zitikis, Ricardas, *The University of Western Ontario*
Co-Supervisor: Zou, Xingfu, *The University of Western Ontario*

A thesis submitted in partial fulfillment of the requirements for the Master of Science degree in Applied Mathematics

Recommended Citation

Luo, Jingshi, "Credit Risk Measurement and Application based on BP Neural Networks" (2021). *Electronic Thesis and Dissertation Repository*. 8000.
https://ir.lib.uwo.ca/etd/8000

Abstract

The emergence of P2P(Peer-to-peer) lending has opened up a popular way for micro-finance, and the financial lending industry in many countries is growing rapidly. While it facilitates lending to individuals and small and medium-sized enterprises, improving the risk identification capability of the P2P platform is vitally necessary for the sustainable development of the platform. Especially the potential credit risk caused by information asymmetry, this may be fatal to this industry. In order to alleviate the adverse effects of this problem, this paper takes Lending Club's real loan data as the empirical research object. The random forest is used to screen the importance of features, and backpropagation neural network approach is used to establish a credit risk classification model. Before loaning, the loan applicants can be divided into default and non-default. The results show that the credit risk measurement model is effective in predicting whether the lender will default.

Keywords: Neural networks, peer-to-peer lending, credit risk assessment, information asymmetry, random forest

Summary for Lay Audience

P2P(Peer-to-peer) lending is an innovative financial mode realized through the Internet. It has the advantages of convenient transactions, applicable to a wide range of people, and improving the efficiency of financial capital circulation. Compared to traditional banks, P2P lending play an different role in the lending industry. However, the lack of risk identification capabilities of many P2P platforms has led to problems that some P2P borrowers frequently fail to pay on time or pay less than the entire principal and interest. This type of risk is defined as credit risk and is primarily caused by the information asymmetry between borrowers and lenders. If this type of micro-financing method wants to achieve sustainable development, each P2P platform should find its own appropriate credit risk model to evaluate loan applicants so as to minimize losses and bring stable benefits for lenders.

In this thesis, we will take Lending Club, the largest P2P platform in the United States, as an example and analyze its real lending data during 2018-2019. First, the missing data and outliers need to be preprocessing in its suitable way respectively to reduce the impact on model accuracy; then the random forest approach was used to rank and screen the importance of the features; finally the selected features are used to build a credit risk model using the BP neural network approach. By using this model, all loan applicants are able to be classified into two types: default and non-default. Loan applicants who are classified as default can be rejected in order to lower the credit risk. The empirical analysis results show that this model is effective for Lending Club case. This model also behaves certain reference significance for the credit risk analysis of other P2P platforms.

Contents

Abstract ii

Summary for Lay Audience iii

Chapter 1 1

1 Introduction 1

1.1 Background 1

1.2 Literature Review 4

1.3 Purpose and significance of the study 5

Chapter 2 6

2 Methodology 6

2.1 Selection of Credit Risk Assessment Modeling Methods 6

2.2 Neural Networks 7

2.3 Back Propagation (BP) Neural Networks 15

Chapter 3 22

3 Case Analysis: Lending Club 22

3.1 The Business Mode of Lending Club 22

3.2 Lending Club Platform Borrowing Process 23

3.3 Advantages of Lending Club’s Risk Control Mechanism 26

Chapter 4 28

4 Lending Club Feature Engineering 28

4.1 Sources of P2P Borrowing Data 28

4.2 Preprocessing Loan Data 30

4.3 Feature Selection 42

Chapter 5 47

5 Credit Risk Measurement by BP Neural Networks 47

5.1 Feasibility of BP Neural Networks 47

5.2 Sample Selection and Data Normalization 49

5.3 BP Neural Networks Model Training and Testing 51

Chapter 6 54

6 Conclusion 54

References 56

Appendices 59

Curriculum Vitae 60

Chapter 1

1 Introduction

1.1 Background

In recent years, with the improvement of people's living standards, the consumption level is increasing gradually. People are constantly spending on food, clothing, housing and transportation to pursue a higher standard of living quality.

In the economic system where there is demand, there is a market, banks and financial institutions have provided targeted and diversified credit services, such as housing loans, car loans, etc. A borrower in need needs to apply for a loan to the bank. The bank will grant the borrower a certain amount of loan after verifying the relevant documents submitted by the borrower and judging the borrower's ability to repay. In this way, the general public can realize their wishes through consumption in advance, and banks and financial institutions can also charge certain interests to make profits. It looks like a total win-win.

However, when the credit business is not fully mature, problems are bound to appear one after another. For example, some borrowers did not repay the bank within the prescribed period, or even absconded with money; Some provided false information or deliberately withheld information that was not conducive to the loan, and so on. This undoubtedly led to a credit crisis for the banks. To avoid such problems, banks have set higher and higher requirements for loan applications. This unintentionally rejects some potential customers who have repayment ability and good credit but cannot apply for loans because of high requirements and high threshold, which is undoubtedly a loss to the potential customers and the bank.

With the rapid development of the Internet, a new product -- P2P (Peer to Peer) arose in the Internet finance industry. P2P online lending method appeared late but developed rapidly. Compared with the traditional way, it allows people to provide funds to borrowers through the online lending platform, and it is a lending method between individuals. P2P online lending mainly consists of four important components: the regulator, the lending platform and the borrower and the lender. The online loan platform is equivalent to the role of credit intermediary played by banks in the traditional financial industry.

P2P online lending was first proposed in 1976. In the early lending process, intermediaries were used as guarantees, and borrowers offered high borrowing rates to attract lenders while they did not need a mortgage.

The world's first P2P online lending platform, Zopa, was established by Giles Andrews et al. in the UK in 2005. Zopa now has more than 700,000 registered users, has lent around £3.756 billion and has a presence in many parts of Asia.

Since Zopa was established, P2P online lending platforms have developed rapidly around the world. In the United States, Prosper, the first P2P lending platform, was launched in 2006 and now has nearly one million members and more than $200 million in loans. At present, the best developed P2P lending platform in the world is Lending Club, founded in 2007 in the United States, which has lent more than 10 billion dollars in 2019. Due to the continuous development of platforms and the increasingly perfect market management system, P2P online lending has become particularly important in many countries.

Take Prosper as an example. Borrowers on the Internet give the amount of money they need to borrow, and the highest interest rate that can be given. Willing investors can bid according to the amount of money and interest rate. Individual investors have a minimum investment limit on each loan project. As P2P platform operation compliance has been further improved, some projects have been set to require guarantees. At present, in all types of business (personal loan, enterprise loan, car mortgage loan, housing mortgage loan), the personal loan is the mainstream business type of net loan. This means that with the rise of people's consumption level and the concept of excessive consumption, personal credit will become the main business of P2P online lending in the future.

Compared with traditional loans, P2P network loans have advantages such as being fast, convenient, covenant lite and wide coverage. However, the borrower's credit risk is the biggest risk faced by P2P network loans for the current mainstream personal credit business. The credit risk in P2P network loans is also known as default risk. In other words, the borrower fails to perform the terms contained in the loan contract due to various reasons and causes the default, which may lead to economic losses for the P2P online lending platform, the lender or other stakeholders.

The virtual nature of the network leads to information asymmetry between borrowers and lenders, which is the main cause of this problem. The only way for lenders to know borrowers is through the online lending platform and decide whether or not to lend. At the same time, because the borrowers do not need the collateral, the risk that the borrowers will not repay are greatly increased. The credit risk of P2P online lending mainly studied in this paper refers to the possibility that borrowers may default due to some reasons for failing to repay the debt in time and in full. Therefore, finding an appropriate algorithm to accurately assess and predict the credit risk of borrowers in P2P online lending is a necessary condition for the normal operation of P2P online lending, which is also the focus of this paper.

1.2 Literature Review

Greiner et al. (2009) made an empirical analysis on the data of the American website Prosper. He found that the historical default record would significantly increase the borrower’s credit risk, so the lender would consider whether to add additional risk compensation according to the historical default record of the borrower.

The study of Freedman and Jin (2008) also described the existence of default records, and the credit rating of borrowers would be lowered, thus increasing the credit risk of borrowers.

In addition, scholars Herzenstein and Andrews (2008) studied the data of Prosper website in 2008, found that financial factors such as the borrower's economic income and existing property were more important than the borrower's different race, gender, skin color, address, age and other basic personal information for the borrower's default behavior.

Lin et al. (2011) proposed that the P2P online lending industry, similar to the traditional financial industry, has the problem of information asymmetry. That is, lenders can only rely on the limited data of borrowers provided by the platform to judge whether to lend funds.

Berger and Gleisner (2014) analyzed the historical transaction data of American P2P lending platform Prosper and found that the degree of information disclosure and credit behavior are two important factors for the risk control of P2P online Lending. Lin also pointed out that to reduce the loss rate of bad debts on P2P platforms, the government credit investigation agencies can be used to increase the openness and transparency of borrowers' credit.

Emekter et al. (2015) used Lending Club's historical data to establish a Logistic regression model to predict credit risk. The empirical results show that the borrower's credit score significantly affects the borrower's default behavior.

Milad and Vural (2015) published a credit risk assessment model based on a random forest machine-learning algorithm in the same year. The model selected 15 feature data provided by borrowers and introduced non-standardized financial data variables into the model to improve predictions accuracy. The research results show that compared with the known credit risk assessment based on FICO credit scores, the credit risk assessment model based on the random forest algorithm is more effective in predicting whether the borrower will default.

Wang and Li (2019) used decision support tools to establish a credit risk assessment model and estimated credit risk using some support vector machine models and linear regression to analyze data and the empirical results show that decision support tools can better assess the credit risk of borrowers.

Byanjankar and Viljanen (2020) proposed an idea. He believes that borrowers' unsecured loans, information asymmetry, and the lack of useful information from borrowers have led to increasing credit risks with the rapid economic growth. Therefore, a survival analysis method is proposed to predict the default probability of P2P network borrowers in different periods, and the survival analysis results are classified by neural networks.

1.3 Purpose and Significance of the Study

In the current environment, there may be serious information asymmetry between investors and borrowers in P2P platforms, which will bring huge credit risks to the entire platform and is the most significant problem currently facing the industry. Improving big data analysis technology is an important way to improve the credit risk assessment capabilities of P2P platforms in various countries.

This paper aims to study how to use the BP neural network model to measure the credit risk of P2P network credit data and reduce the information asymmetry between investors and borrowers of the P2P credit platform by improving the model's accuracy. Take the Lending Club platform as an example, which is the most successful platform in the P2P industry. They have the most sufficient data and the most advanced models, hopefully its model can be used as a reference for other P2P platforms.

Chapter 2

2 Methodology

2.1 Selection of Credit Risk Assessment Modeling Methods

P2P credit risk assessment can be regarded as a binary classification problem. Therefore, in theory, all classification algorithms can be used for credit risk measurement modeling. However, there are differences in the implementation of each algorithm, which also leads to its applicable conditions for each algorithm. Therefore, it is necessary to analyze the advantages and disadvantages of each before modeling and select the appropriate algorithm for modeling according to the actual situation. Table 1 is a summary of several classification algorithms that are currently more commonly used.

Method	Advantages	Disadvantage
Logistic Regression	1. Simple and efficient execution 2. Good interpretability	1. Due to the simplicity of the model, the model's ability to fit the data is weak 2. Because of the weak model expression ability, more features are needed to be constructed in feature engineering 3. Avoid multicollinearity between variables when modeling, which may result in less available data and information dimensions, resulting in information loss

Support Vector Machine	1. The model fits the data well 2. A large classification accuracy can be achieved on a small sample of data by selecting a suitable kernel function	1. Model training is slower in the case of large samples 2. If the "radial basis" kernel function is used, the model may be difficult to interpret
Neural Networks	1. Strong fitting ability. It can theoretically fit any complex data by constructing a complex network structure 2. Insensitive to the multicollinearity of the variables in the training data	1. 1.It is troublesome to adjust parameters during model training, and it is difficult to reflect its advantages under small data volume 2. 2.The internal structure of the model is complex, so its interpretability is poor. Often being treated as a black-box model
Decision Tree	1. No need to consider multicollinearity. Good accuracy and robustness on classification problems 2. Good interpretability	Overfitting problems are prone to occur
Random Forest	1. Upgrade of the decision tree model; Strong fitting ability 2. Compared with a single decision tree, random forest can reduce overfitting	Random forest is a model integrated from multiple decision trees, the structure is more complicated, and many random processes are involved in the training process. When there are many internal trees, the prediction mechanism inside the model is difficult to explain

Table 1 Summary of Several Classification Algorithms

Considering their respective advantages and disadvantages, as well as a large amount of personal credit data and the complicated correlation between the original features, this paper finally chooses to use the neural network model to measure credit risk.

2.2 Neural Network

Neural network originated from the human nervous system. It is a physical mechanism based on the knowledge of network topology, and through a large number of computing units (neurons) to simulate the way of human brain processing transactions and then carry out a distributed transmission of a mathematical model. Neural networks have the ability of self-repeated learning. To a certain extent, it imitates the intelligent functions of the human brain nervous system, such as filtering useless knowledge thinking, information processing and storage, recognition and classification.

Neural network has many advantages over human brains. For example, hardware and software can be implemented together, non-linear image processing ability, the ability to stimulate the brain's associative memory without limitations, avoid complex mathematical models only need input and output network topology knowledge, easy distribution, fault tolerance and high storage capacity, and so on. It is precisely because it is through learning and memory, rather than through hypothesis, in the execution of problems can imitate the human brain to establish a unique information processing system, so people often use it to deal with some engineering or difficult to implement problems.

2.1.1 Perceptron

In a biological neural network, each neuron is connected to other neurons. When it gets "excited," it sends chemicals to connected neurons that change the electrical potential in those neurons. If a neuron becomes more electric than a threshold, it gets "excited," and sends chemicals to other neurons.

The artificial neural network was first proposed by biologist McCulloch and mathematician Pitts (1943). This MCP neuron model can perform logical operations like and, or, not. The weight and bias of the MCP model are fixed, so there is no possibility of learning. The problem was solved a few years later by Frank Rosenblatt (1958). He developed the first neural network capable of learning and called it the Perceptron.

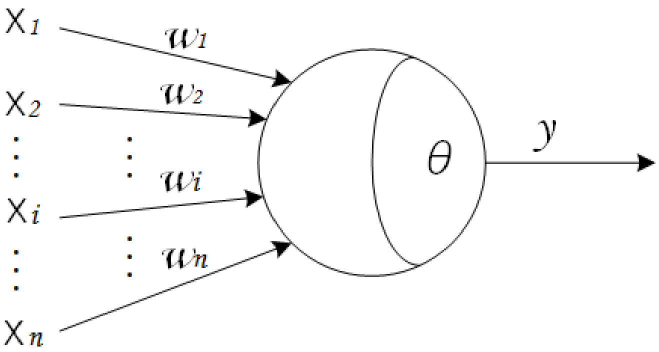


Figure 1 Perception Model

Perceptron is the most basic machine learning algorithm, which can be regarded as an artificial neural network containing only one node to understand the self-learning process of a single neuron. In a perceptron model with only one node, each input variable enters the activation function according to its own initial weight. The weight of each variable is adjusted continuously through the training set. The neuron receives the input signal x from n other neurons, which is transmitted through a connection with weight ω, and the total input received by the neuron $\sum_{i=1}^{n} \omega_i x_i$ will be compared to the threshold θ of the neuron. Then, after "activation function" processing, the output of the neuron is generated. The specific formula is as follows:

$$y = \varphi(\sum_{i=1}^{n} \omega_i x_i - \theta)$$

Name	Symbol
Input variables	x_i
Output value	y
Weights	ω_i
Threshold	θ
Activation Function	φ

Figure 2 Abbreviation Descriptions

The learning process stops when the accuracy of output values reaches the preset requirements. At this point, the weight is the final value after the training set is learned, which is the parameter of the perceptron model.

2.1.2 Activation Functions

There is a functional relationship between the output of the upper level node and the input of the lower level node, and this function is called the activation function. Without the activation function, the input of each layer of nodes is a linear function of the output of the upper layer. It is easy to verify that no matter how many layers you have in the neural network, the output is a linear combination of the inputs, which is nearly the same to the effect of no hidden layers, and this situation is the most primitive perceptron, then the approximation ability of the network is quite limited. Because of the above reason, some nonlinear functions are used as activation functions, so that the deep neural network is no longer a linear combination of inputs, but can approximate almost any function.

The ideal activation function is the Heaviside step function, as shown in Figure 3. The input value is mapped to an output value of 0 or 1 (1 for neuronal excitation and 0 for neuronal inhibition). Its mathematical form is shown below:

$$H(x) = \begin{cases} 1, x \geq 0 \\ 0, x < 0 \end{cases}$$

The Heaviside step function cannot be used as the activation function because it is discontinuous and unsmoothed. Therefore, Sigmoid function is often been used as the activation function. The Logistic function is a typical Sigmoid function. As shown in Figure 4, the input value that may vary over a wide range is set within the output value range of (0,1), so it is also known as the "squashing function". Its mathematical form and derivative form are as follows:

$$f(x) = \frac{1}{1 + e^{-x}}$$

$$f'(x) = f(x)(1 - f(x))$$

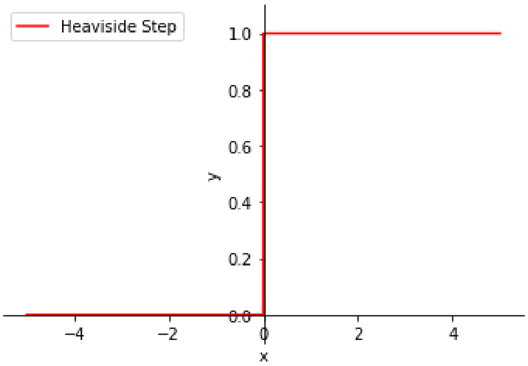


Figure 3 Heaviside Step Function

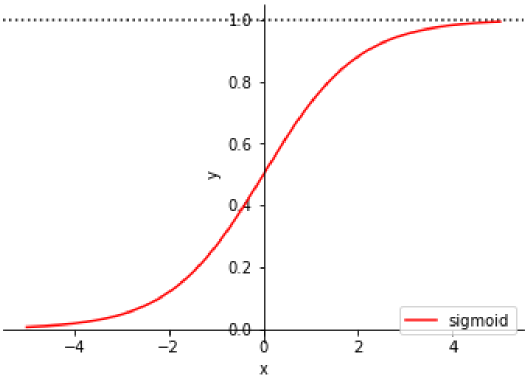


Figure 4 Sigmoid Function

Although the sigmoid function has some advantages like it it continuous and it is very easy to calculate its derivative, it still has three drawbacks: first, it is prone to gradient vanishing: when the activation function approaches the saturation zone, the change is too slow and the derivative is close to 0. According to the mathematical basis of backward transfer is the chain rule of calculus, the current derivative needs the product of the derivatives of the previous layers, and several relatively small numbers are multiplied together, and the derivative results are very close to 0, thus making it impossible to complete the training of the deep network.

Second, the output of sigmoid is not 0-mean (zero-centered): this leads to a non-zero-mean signal for the input of the neurons in the later layers, which has an impact on the gradient. Taking $y = w^T x + b$ as an example, assuming that the inputs x are all positive (or negative), the derivative of w is always positive (or negative), so that in the backpropagation process either all update in the positive direction or all update in the negative direction, resulting in a binding effect that makes the convergence slow.

Third, its analytic formula contains power operations, which is relatively timeconsuming to solve by computer. For large scale deep networks, this can increase the training time significantly.

Except the sigmoid function, there are also some common activation functions, each of which is quite different in nature.

The hyperbolic tangent function is similar to sigmoid but slightly different, mainly in that its range is (-1, 1). It is centrally symmetric about the origin. Its mathematical form, derivative form and the transformation relationship between it and sigmoid are as follows:

$$f(x) = tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

$$f'(x) = 1 - f(x)^2$$

$$tanh(x) = 2sigmoid(x) - 1$$

It solves the problem that the Sigmoid function is not zero-centered output, however, the problem of gradient vanishing and the problem of power operation still exist.

In addition, the Rectified Linear Unit (ReLU) is an activation function that has been introduced in the last few years and is often used as an inter-layer activation function. Its mathematical form and derivative form are shown as follows, and the value range is [0, +∞).

$$f(x) = \max(0, x)$$

$$f'(x) = \begin{cases} 0, x < 0 \\ 1, x \geq 0 \end{cases}$$

ReLU has several advantages: first, it does not have the gradient vanishing problem. Second, it is very fast to compute, only need to determine whether the input is greater than 0. Moreover, the convergence is much faster than sigmoid and tanh.

Softplus is the smooth version of ReLU. The mathematical form and derivative form are,

$$f(x) = \ln(1 + e^x)$$

$$f'(x) = \frac{e^x}{e^x + 1} = \frac{1}{1 + e^{-x}}$$

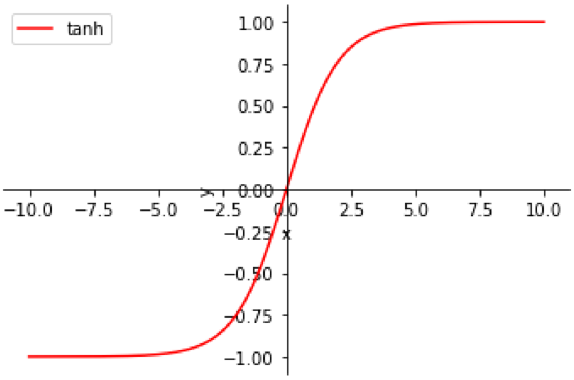


Figure 5 Tanh function

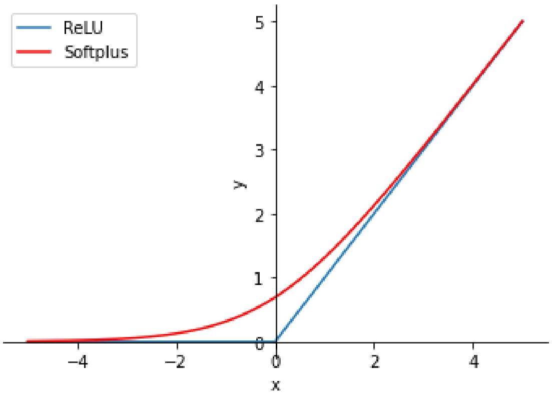


Figure 6 ReLU Function and Softplus Function

The output of the ReLU function is equal to the input if the input is not negative. Otherwise, it is zero. In addition to the above, ReLU has many derivative functions such as Noisy ReLUs, Leaky ReLUs, etc. ReLU is simple to calculate and does not have the problem of gradient saturation, but it also has some limitations. During the use process, extra attention should be paid to avoid dead ReLU problem: The phenomenon of ReLU being killed in the negative region is called dead ReLU. The gradient of this neuron and subsequent neurons will always be 0 and will not respond to any data, resulting in the corresponding parameters never being updated. Therefore, it should be noted during initialization that if all of them are initialized to 0, then all of the neurons will be in a silent state and cannot be trained. At the same time, when setting the learning rate, it should be noted that too high a learning rate will lead to the direct "death" of many neurons.

2.1.3 Layered Structure

The main structure of neural network is composed of layers and nodes. Input variables and output values constitute the input layer and output layer, and the layer between the input layer and output layer is called the hidden layer. Each layer contains nodes. The nodes of each layer are connected to the nodes of the adjacent layer. Each node needs to set weight, and the function connecting each node is the activation function.

When only one layer of such neurons exists, it is called a perceptron. The input layer is called the zeroth layer because it simply buffers the input. The only layer of neurons that exists forms the output layer. Each neuron in the output layer has its own weight and threshold.

When many such layers exist, the network is called a multilayer perceptron (MLP). MLP has one or more hidden layers. These hidden layers have different numbers of hidden neurons. Neurons in each hidden layer have the same activation function:

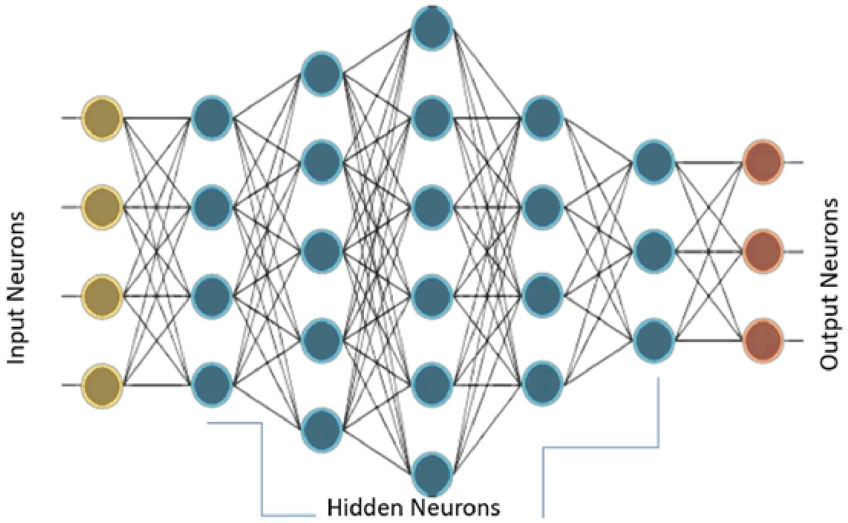


Figure 7 Multilayer Perceptron

The MLP in Figure 7 has one input layer with four inputs, five hidden layers with four, five, six, four, and three neurons, and one output layer with three neurons. In this MLP, all neurons in the lower layer are connected to all neurons in the upper layer adjacent to them. Therefore, MLP is also known as the full connection layer. The flow of information in an MLP is usually from input to output and currently has no feedback or jump, so these networks are also called feed-forward networks.

2.3 Back Propagation (BP) Neural Networks

Rumelhart et al. (1986) proposed a multilayer feed-forward networks, namely Back Propagation neural networks. The whole networks are similar to the general neural networks, except that there is an error backpropagation process. The unit of each layer will input the signal to the next layer after processing. The final model structure is a network structure with the full connection among processing units of each layer and mutual independence among processing units of the same layer. The details are shown in figure 8.

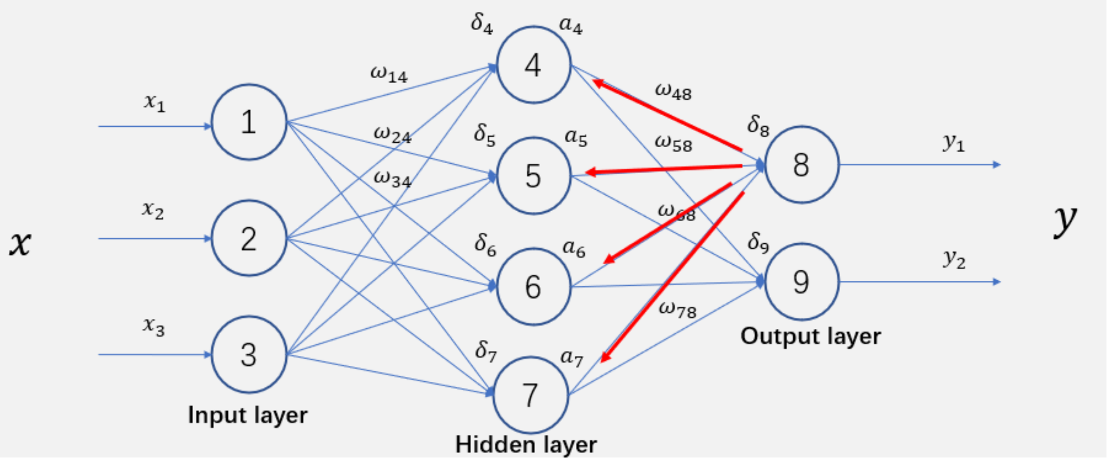


Figure 8 Back Propagation Neural Network Model

The flow chart in figure 9 is the procedure of BP neural network. The solid line represents forward propagation and the dashed line represents backward propagation.

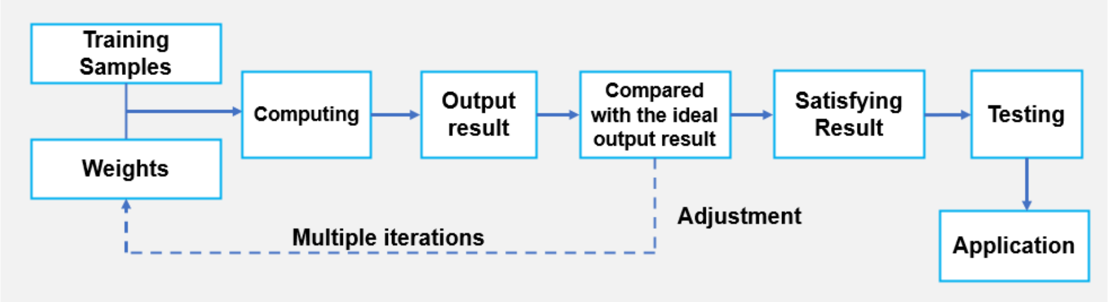


Figure 9 The procedure of BP neural network

Back-propagation refers to comparing the final output value with the ideal output value. The error between them is being sent backward, by using gradient descent method and doing multiple iterations to continuously adjust all the weights, this process ends if the final output reaches the required accuracy.

The specific process is to continuously adjust all the weights between each node on the network through a process of multiple iterations, and the method of weight adjustment adopts the gradient descent method.

Hence, we can see, the connection method, the number of layers and the number of nodes in each layer are not learned, but artificially set, which we call hyper-parameters. The weights are what the model needs to learn.

Here is the mathematical analytic process of BP neural networks algorithm. The first consideration is the forward propagation of the signal. Suppose ω_{ij} represents the weight transferred from node i to node j, and the size of the information transmitted from node i is y_i (if neuron i is in the input layer, then $y_i = x_i$), then the activated value of each node j is α_j:

$$\alpha_j = \sum_i \omega_{ij} y_i$$

According to the activation function φ of node j, the amount of its information is calculated,

$$y_j = \varphi(\alpha_j) = \varphi(\sum_i \omega_{ij} y_i)$$

In the BP algorithm, the gradient descent method is used to update the weight, and the activation function is required to be differentiable. Therefore, if the Sigmoid function is used as activation function,

$$\varphi(z) = \frac{1}{1 + e^{-z}}$$

The first derivative is,

$$\frac{\partial \varphi}{\partial z} = \varphi(z)(1 - \varphi(z))$$

Then the actual output o_j of the output layer can be calculated, and the error E generated by node j can be defined as

$$E_j = \frac{1}{2}\sum_{j=1}^{l}(d_j - o_j)^2$$

Assume the output layer has l nodes, where d_j represents the expected output of the output layer and y_j represents the actual output of the hidden or output layer. The weight ω_{ij} is gradually updated according to the gradient descent method to minimize the error E. Therefore, calculate the first derivative of E with respect to ω_{ij},

$$\frac{\partial E_j}{\partial \omega_{ij}} = \frac{\partial E_j}{\partial o_j}\frac{\partial o_j}{\partial \omega_{ij}}$$

For the sake of notation, we will use y_j to denote o_j (y_j represents the actual output of the hidden or output layer, o_j represents the actual output of the output layer),

$$\frac{\partial E_j}{\partial \omega_{ij}} = \frac{\partial E_j}{\partial y_j}\frac{\partial y_j}{\partial \omega_{ij}}$$

According to the above formula, it can be seen that the information value y_j is a function of the activation value α_j, and the activation value α_j is directly related to ω_{ij}, so

$$\frac{\partial E_j}{\partial \omega_{ij}} = \frac{\partial E_j}{\partial y_j}\frac{\partial y_j}{\partial \omega_{ij}} = \frac{\partial E_j}{\partial y_j}\frac{\partial y_j}{\partial \alpha_j}\frac{\partial \alpha_j}{\partial \omega_{ij}}$$

Then, do the separate calculations for each term.

$$\frac{\partial \alpha_j}{\partial \omega_{ij}} = \frac{\partial(\sum_i \omega_{ij} y_j)}{\partial \omega_{ij}} = y_i$$

This derivative is the value of the information that's being sent to the node i. The second term can be written as,

$$\frac{\partial y_j}{\partial \alpha_j} = \varphi(\alpha_j\left(1 - \varphi(\alpha_j)\right) = y_j(1 - y_j)$$

Finally, the first term $\frac{\partial E_j}{\partial y_j}$, is needed to be discuss in two cases,

(1) When $y_j$is the output layer, $y_j = o_j$,

$$\frac{\partial E}{\partial y_j} = \frac{\partial E}{\partial o_j} = o_j - d_j$$

$$\frac{\partial E_j}{\partial \omega_{ij}} = \frac{\partial E_j}{\partial y_j}\frac{\partial y_j}{\partial \alpha_j}\frac{\partial \alpha_j}{\partial \omega_{ij}} = \frac{\partial E_j}{\partial y_j}\frac{\partial y_j}{\partial \alpha_j} y_i = (o_j - d_j) \cdot o_j(1 - o_j) \cdot y_j$$

(2) When y_j is the hidden layer, the error caused by the output value of node j, y_j,

is equal to the sum of the error values caused by all the downstream nodes. Assume the set of the downstream node j is L,

$$\frac{\partial E_j}{\partial y_j}=\sum_{l\in L}\frac{\partial E_l}{\partial y_j}=\sum_{l\in L}\frac{\partial E_l}{\partial y_l}\frac{\partial y_l}{\partial \alpha_l}\frac{\partial \alpha_l}{\partial y_j}=\sum_{l\in L}\frac{\partial E_l}{\partial y_l}\frac{\partial y_l}{\partial \alpha_l}\omega_{jl}$$

From the above, it can be seen that both of these cases satisfy the following formula (the difference lies in $\frac{\partial E}{\partial y_j}$ of both cases),

$$\frac{\partial E_j}{\partial \omega_{ij}}=\frac{\partial E_j}{\partial y_j}\frac{\partial y_j}{\partial \alpha_j}\frac{\partial \alpha_j}{\partial \omega_{ij}}=\frac{\partial E_j}{\partial y_j}\frac{\partial y_j}{\partial \alpha_j}y_i$$

For convenience, let

$$\delta_j=\frac{\partial E_j}{\partial y_j}\frac{\partial y_j}{\partial \alpha_j}$$

Then the first derivative of weight is,

$$\frac{\partial E_j}{\partial \omega_{ij}}=\delta_j y_i$$

δ_{ij} can be written as

$$\delta_{ij}=\begin{cases}(o_j-d_j)\cdot o_j(1-o_j)\\ \left(\sum_{l\in L}\frac{\partial E_l}{\partial y_l}\frac{\partial y_l}{\partial \alpha_l}\omega_{ij}\right)y_j\left(1-y_j\right)=(\sum_{l\in L}\delta_l\omega_{ij})y_j(1-y_j)\end{cases}$$

BP algorithm uses gradient descent method to reduce the error. Therefore, in the process of iteration, the following methods are used to update the weight:

$$\omega_{ij}{}'=\omega_{ij}-\eta\frac{\partial E_j}{\partial \omega_{ij}}$$

η is the learning rate of gradient descent, and the weight is updated along the negative gradient direction.

Here is the example of BP neural network model mentioned in figure 8. The above mathematical analytic process is used to calculate errors and update weights in figure 10.

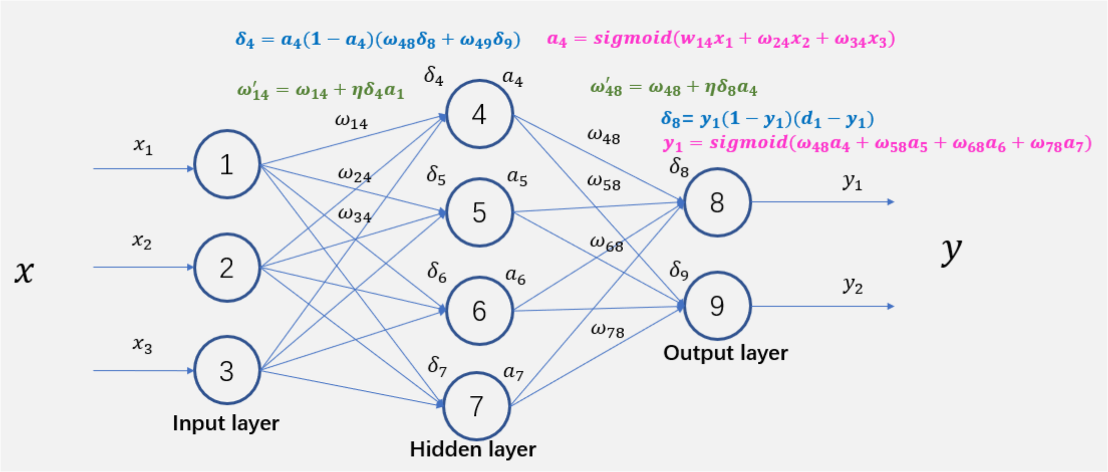

$\delta_4 = a_4(1-a_4)(\omega_{48}\delta_8 + \omega_{49}\delta_9)$
$a_4 = sigmoid(w_{14}x_1 + \omega_{24}x_2 + \omega_{34}x_3)$
$\omega'_{14} = \omega_{14} + \eta\delta_4 a_1$
$\omega'_{48} = \omega_{48} + \eta\delta_8 a_4$
$\delta_8 = y_1(1-y_1)(d_1-y_1)$
$y_1 = sigmoid(\omega_{48}a_4 + \omega_{58}a_5 + \omega_{68}a_6 + \omega_{78}a_7)$
x
x_1
x_2
x_3
1
2
3
Input layer
ω_{14}
ω_{24}
ω_{34}
δ_4
a_4
4
δ_5
a_5
5
δ_6
a_6
6
δ_7
a_7
7
Hidden layer
ω_{48}
ω_{58}
ω_{68}
ω_{78}
δ_8
8
δ_9
9
Output layer
y_1
y_2
y

Figure 10 Back propagation neural network model (weight update)

In summary, BP neural network can be divided into the following five steps.

Step 1: Assign random values to all weights ω_{ij} (including threshold), usually small real numbers. Where t represents the learning round and is initialized to 0.

Step 2: The forward propagation of signals. The output signals of hidden layer and output layer y_j are calculated layer by layer

$$\alpha_j = \sum_i \omega_{ij} y_i$$

$$y_i = \varphi(\sum_i \omega_{ij} y_i)$$

Step 3: If the error does not reach the set accuracy, it needs to get in the error back propagation process. Firstly, δ_j is calculated for each neuron in the output layer

$$\delta_j = (y_j - d_j)y_j(1 - y_j)$$

Then δ_j is calculated for the hidden layer is,

$$\delta_j = (\sum_{l \in L} \delta_l \omega_{jl}) y_j(1 - y_j)$$

Step 4: Calculate the weight correction. The error is allocated to each node of each layer,

$$\Delta\omega_{ij}(t) = \eta\delta_j y_i$$

Step 5: Update weights:

$$\omega_{ij}(t + 1) = \omega_{ij}(t) + \Delta\omega_{ij}(t)$$

The updated networks' output is redetermined by using the updated weights in the previous step until the required accuracy is reached. Otherwise, repeat from Step2 to Step 5.

In this paper, BP neural network is applied to the risk assessment of the P2P lending platform. Compared with the traditional prediction method, BP neural network does not need to analyze the samples and choose a data model that the samples match in advance. It can achieve more accurate predictions through its own internal learning. Its advantages can be summarized as the following three points,

1. Non-linear mapping capability: BP neural networks prediction can approximate nonlinear functions with arbitrary accuracy, which breaks out of the limitation that the traditional methods are mostly linear functions. For the problems with complex internal mechanism, BP neural networks can also be well solved.
2. Generalization: BP neural network is widely used. That is, it can not only ensure

the networks classification in training, but also achieve better classification for new data after the networks training is completed for prediction. Therefore, BP neural networks also has many applications in prediction.

3. Self-learning: The learning process of BP neural networks are completely autonomous. It extracts the internal rules of training data through self-learning and approaches the optimal effect by constantly adjusting weights and thresholds until the learning of the network is realized. It can be said that BP neural networks has a very high degree of self-learning and self-adaptive ability.

Chapter 3

3 Case Analysis: Lending Club

This chapter will analyze the risk control of Lending Club, a P2P platform in the United States. As a leading company in the P2P industry, it has a lot of successful experience in credit risk control management, especially the credit risk control system based on big data, which has always been regarded as the industry benchmark.

3.1 The Business Mode of Lending Club

Founded in 2006, Lending Club is the largest P2P online Lending platform in the United States and a leader in the global P2P industry. Under the requirements of American regulation, the business model has undergone three changes, including the promissory note mode, the quasi- bank mode and the securities mode.

From June 2006 to December 2007, the promissory note mode, Lending Club acts as a pure information intermediary. The borrower issues a loan promissory note to Lending Club, which issues a loan to the borrower and then transfers the promissory note signed by the borrower to the investor. As an intermediary, the platform collects service fees as a source of profit and does not undertake loan risks in the process of trading. Many small P2P platforms are still at this stage.

From January 2008 to March 2008, Lending Club was in the quasi-bank mode, cooperating with the commercial bank WebBank. WebBank makes loans to borrowers and transfers the promissory loan notes to Lending Club on a non-recourse basis at par. Finally, Lending Club transfers the promissory loan note obtained from the borrower to the subscribed investors, who become the creditors of the borrower.

Since Lending Club's registered at the United States Securities and Exchange Commission, the platform officially entered the securities mode. This mode is similar to the quasi-bank mode. The essential difference is that what investors buy is no longer the borrower's promissory note but a "bond" (member repayment support bond) issued by Lending Club. Investors become creditors of Lending Club, and there is no direct debtor- creditor relationship between investors and borrowers under the securities mode. Flow chart in figure 11 showed the operation of this mode.

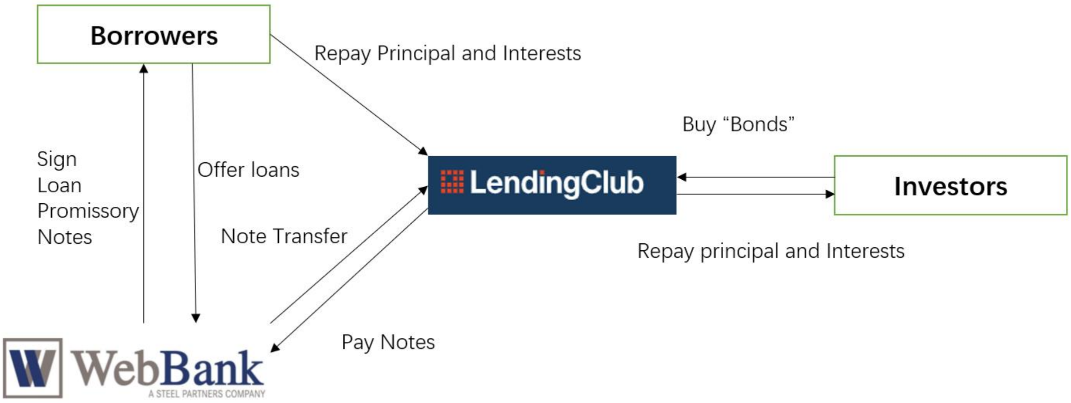


Figure 11 Securities Mode of Lending Club

3.2 Lending Club Platform Borrowing Process

In order to reduce the credit risk of borrowers, Lending Club hopes to screen out high-quality borrowers through an effective credit evaluation system, retain general borrowers, and reject borrowers with higher risks. Moreover, according to different credit ratings, to achieve differentiated pricing of borrowing interest rates. Therefore, Lending Club has developed a strict, rigorous and effective credit evaluation system that combines external and internal ratings to avoid bad debt risks to the greatest extent.

First of all, the borrower needs to register with the real name on the platform before applying for a loan and provide relevant certification information, such as occupation, annual income, real estate, loan purpose.

After the borrower applies for a loan from Lending Club, Lending Club will conduct a preliminary screening of the borrower. The selection criteria include:

i. FICO score greater than 660 points;
ii. Debt-to-income (DTI) ratio less than 35%;
iii. At least two revolving credit accounts;
iv. Provide credit history within 36 months

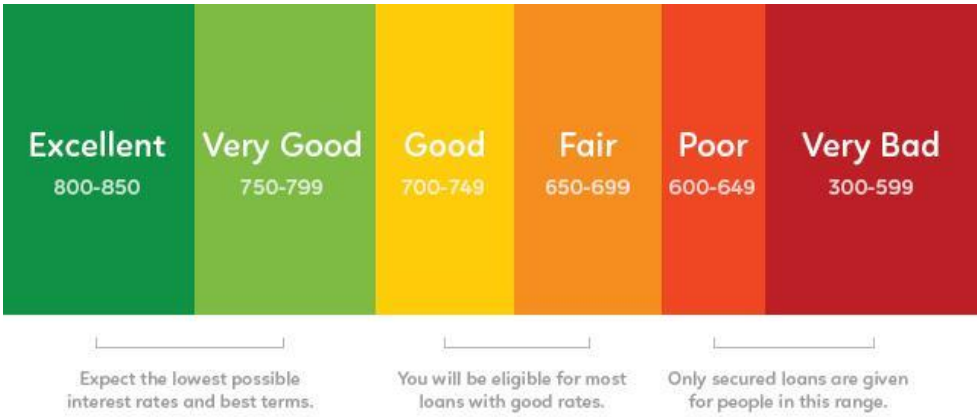


Figure 12 FICO score

FICO credit score is a personal credit rating method developed by the American Personal Consumer Credit Rating Company, which has been widely accepted by the society. The credit score obtained by the FICO scoring system ranges from 300 to 850 points. Figure 12 show the specific credit status reflected by the score**.** The higher the score, the smaller the customer's credit risk. However, the score itself does not indicate whether a customer is good or bad. The lender usually uses the score as a reference to make loan decisions.

Third, after passing the preliminary screening, Lending Club uses Proprietary Scoring Models to conduct an in-depth assessment of the borrower to determine the borrower's credit rating and the corresponding borrowing interest rate.

Combining the borrower's FICO score and other related characteristics, a Model Rank will be obtained through the Initial Scoring Model, which is divided into 35 levels, and each level corresponds to the basic risk sub-grade of A1-G5. A1 borrowers have the lowest risk, but the corresponding interest rate is also the lowest. G5 borrowers have the highest credit risk, and the corresponding interest rate is also the highest. Investors can choose investment objects according to their own risk preferences. Next, Lending Club will use Loan Grades and Risk Modifiers to determine the borrower's final credit sub-grade. The adjustment indicators include the borrower's loan amount and loan period.

The borrower's borrowing interest rate is determined by the credit sub-level, and the interest rate of each level is calculated as follows:

$$Borrowing\ interest\ rate = \ benchmark\ interest\ rate\ (5.05\%) + \ risk\ fluctuation\ adjustment\ (0.88\%{\sim}21.01\%)$$

Different credit sub-grades correspond to different risk volatility adjustment rates. The higher the credit level, the lower the risk volatility adjustment rate and the lower the corresponding borrowing interest rate.

Due to the strict loan conditions, only 10% of borrowers can meet the Lending Club's requirement, thus ensuring the quality of Lending Club's customers and reducing the level of default on the platform. The figure below shows the statistics of Lending Club's loan amount to borrowers of various credit ratings in recent years. From the table below, it can be seen that the proportion of Lending Club's loan amount to low-risk borrowers is increasing, while the loan amounts of high-risk borrowers is decreasing. It intuitively reflects Lending Club's risk control policy of selecting high-quality borrowers as much as possible.

Credit Rating	2013	2014	2015	2016	2017	2018	2019
A	13.66%	14.90%	16.79%	15.84%	16.66%	26.87%	31.94%
B	30.05%	24.10%	26.12%	28.17%	27.94%	28.85%	30.56%
C	28.60%	27.50%	27.70%	30.33%	33.79%	25.86%	23.02%
D	14.30%	19.68%	15.57%	14.84%	13.54%	13.84%	13.88%
E	8.14%	9.86%	10.06%	7.25%	5.17%	3.66%	0.59%
F	4.26%	2.98%	3.07%	2.80%	1.80%	0.76%	0.01%
G	0.99%	0.95%	0.69%	0.78%	1.10%	0.17%	0%

Table 2 Lending Club Percentage of Loans by Rating

3.3 Advantages of Lending Club’s Risk Control Mechanism

Lending Club has always attached great importance to technology research and development in the field of financial technology, among which the most widely known is the credit risk assessment system based on big data. Data is the foundation of all big data analysis technologies. Lending Club has accumulated a large number of user data samples by accessing data from the US Credit Statistics Bureau and personal information provided by platform users, which meets the prerequisites for conducting big data analysis. At the same time, the platform already has the ability to process these data automatically. Through big data technology and risk analysis algorithms, it can efficiently perform credit ratings for borrowers, which greatly reduces the degree of information asymmetry between transaction parties, improves platform operation efficiency, and helps investors reduce transaction risk. Compared with traditional risk control mechanisms, risk control based on big data technology has the following advantages:

(1) Big data analysis technology can use more dimensions of data to evaluate the credit risk of borrowers in P2P credit risk control. Compared with traditional financial risk control, big data risk control can effectively use a large amount of non-traditional financial data. For example, the Internet attributes of P2P online loans include the borrower's network equipment information, network transaction behavior, social network and other data. In this way, the user profile and risk assessment of the borrower can be carried out, and the degree of information asymmetry between the parties to the transaction can be reduced.

(2) Data risk control has realized the automation and standardization of approval through computer programs, and the approval results can be obtained quickly, which greatly improves the approval efficiency.

(3) Big data risk control, on the one hand, eliminates the impact on the consistency of approval results due to the experience and subjective awareness of risk control personnel. On the other hand, big data analysis technology based on artificial intelligence algorithms can quickly learn and evolve through the iteration of data samples, making credit risk control more and more accurate.

Overall, in the current environment, it is quite necessary to study the risk control system of the Lending Club for the entire industry. Meanwhile, improving big data analysis technology is an important way to improve the credit risk assessment capabilities of P2P platforms in various countries.

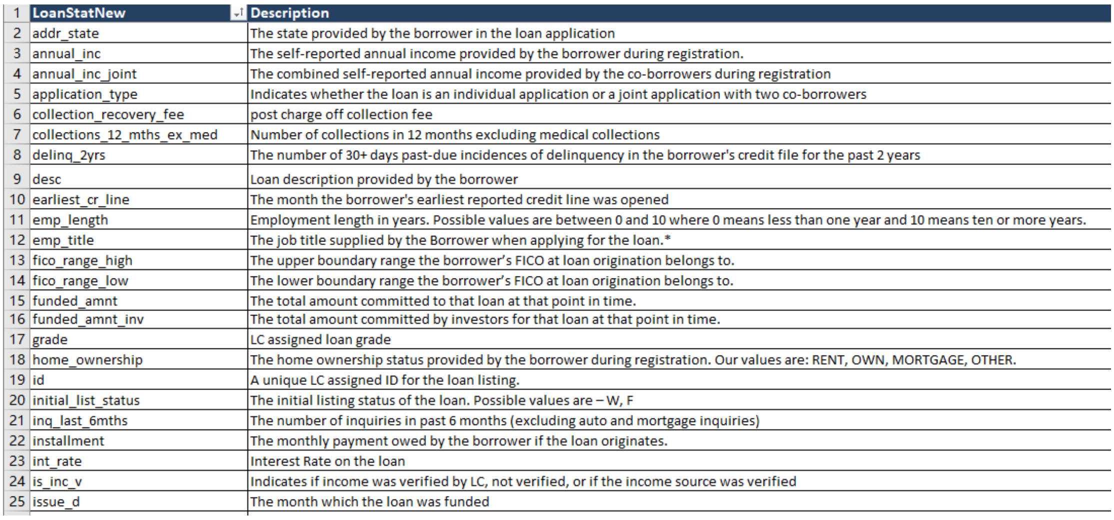
Chapter 4

4 Lending Club Feature Engineering

There is a saying widespread in the industry: For a machine learning problem, data and features determine the upper limit, and models and algorithms just approach this upper limit. Feature engineering, as the name suggests, is a series of engineering processing on raw data. It is refined into features and used as input for algorithms and models. In essence, feature engineering is a process of representing data. In actual work, feature engineering aims to remove impurities and redundancies in the original data, then design more efficient features to characterize the relationship between the solved problem and the prediction model. This chapter is mainly to clean the data and construct features. Dealing with the missing data and outliers in the data is quite important to the subsequent modelling.

4.1 Sources of P2P Borrowing Data

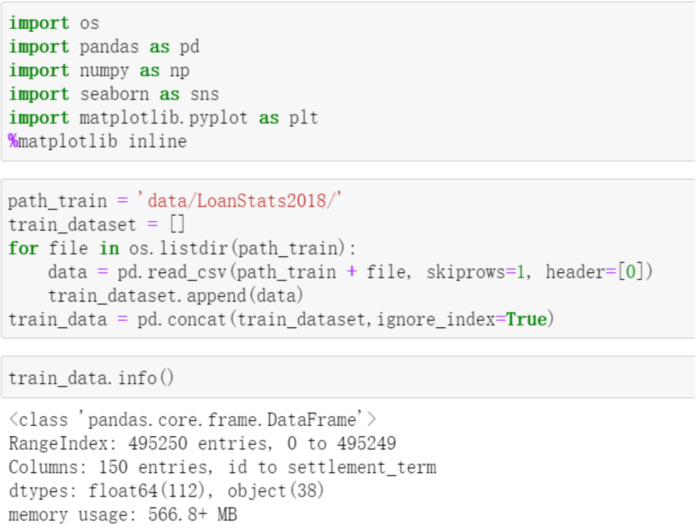
The data for the empirical analysis in this paper comes from the real loan transaction data on the Lending Club platform. Lending Club publishes its transaction data on its official website for download every quarter, with the data feature description attached. This description contains an explanation of the features contained in the dataset, such as the borrower's annual income, employment length, their funded amount. The data used in this paper span from the first quarter of 2018 to the fourth quarter of 2019 (the latest data available). There are 1,004,958 samples and 150 features in total. The figure below shows the names of some features and their corresponding feature descriptions.



	LoanStatNew	Description
2	addr_state	The state provided by the borrower in the loan application
3	annual_inc	The self-reported annual income provided by the borrower during registration.
4	annual_inc_joint	The combined self-reported annual income provided by the co-borrowers during registration
5	application_type	Indicates whether the loan is an individual application or a joint application with two co-borrowers
6	collection_recovery_fee	post charge off collection fee
7	collections_12_mths_ex_med	Number of collections in 12 months excluding medical collections
8	delinq_2yrs	The number of 30+ days past-due incidences of delinquency in the borrower's credit file for the past 2 years
9	desc	Loan description provided by the borrower
10	earliest_cr_line	The month the borrower's earliest reported credit line was opened
11	emp_length	Employment length in years. Possible values are between 0 and 10 where 0 means less than one year and 10 means ten or more years.
12	emp_title	The job title supplied by the Borrower when applying for the loan.*
13	fico_range_high	The upper boundary range the borrower's FICO at loan origination belongs to.
14	fico_range_low	The lower boundary range the borrower's FICO at loan origination belongs to.
15	funded_amnt	The total amount committed to that loan at that point in time.
16	funded_amnt_inv	The total amount committed by investors for that loan at that point in time.
17	grade	LC assigned loan grade
18	home_ownership	The home ownership status provided by the borrower during registration. Our values are: RENT, OWN, MORTGAGE, OTHER.
19	id	A unique LC assigned ID for the loan listing.
20	initial_list_status	The initial listing status of the loan. Possible values are – W, F
21	inq_last_6mths	The number of inquiries in past 6 months (excluding auto and mortgage inquiries)
22	installment	The monthly payment owed by the borrower if the loan originates.
23	int_rate	Interest Rate on the loan
24	is_inc_v	Indicates if income was verified by LC, not verified, or if the income source was verified
25	issue_d	The month which the loan was funded

Figure 13 Feature Descriptions

The process of data processing and model construction is applied in the Jupyter notebook. Before modelling, the data of 2018 and 2019 is needed to be cleaned and selected, respectively. Below is the data processing process for 2018 shown. The data processing process for 2019 is similar to that for 2018. Here is the information of the raw data in 2018, including 49250 pieces of information and 150 features. These features have 112 float64 types and 38 object types.



```
import os
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
```

```
path_train = 'data/LoanStats2018/'
train_dataset = []
for file in os.listdir(path_train):
    data = pd.read_csv(path_train + file, skiprows=1, header=[0])
    train_dataset.append(data)
train_data = pd.concat(train_dataset,ignore_index=True)
```

```
train_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 495250 entries, 0 to 495249
Columns: 150 entries, id to settlement_term
dtypes: float64(112), object(38)
memory usage: 566.8+ MB
```

Figure 14 Information of original data

4.2 Preprocessing Loan Data

Absolutely, most of the original data cannot be used directly. They should be analyzed and preprocessed first. Data preprocessing is an indispensable step before data modelling. The main purpose is to clean the messy original data into a format that meets the requirements of the modelling algorithm. Here are three steps of preprocessing. First of all, appropriate methods are used to deal with various types of missing values. Then, faced with some categorical variables, they need to be converted into numerical data so as to facilitate the following process. Third, some obvious outliers are identified and adjusted with appropriate methods if some extreme value may bias the model results.

4.2.1 Missing Value

There are many reasons why data is missing. On the one hand, because of the data entry equipment problems, failing data acquisition and storage. For instance, the transmission of data collection fails, or the storage medium is damaged. On the other hand, the data is incomplete because of human error, or deliberate concealment for a certain purpose, or because some data is not important enough to be necessary. Besides, due to inconsistent data collection standards, there are some missing feature values in the data collected in the early stage when compared with the data collected in the later period.

About the data of Lending Club, the main reasons for missing data are, for one thing, artificial concealment and low importance of data, resulting in not filling in; for another thing, inconsistent standards for data collection.

Facing the missing values, there are different processes between numerical variables and categorical variables, and the following methods are usually used,

(1) Delete: If the number of missing features is too large, the features are removed directly. Otherwise, it may bring in a large noise, which will negatively effect on

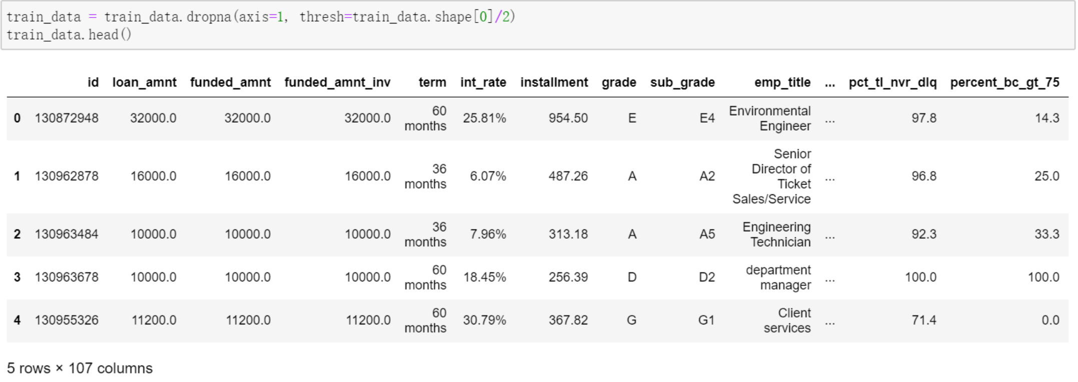
the results.

(2) Fill with statistical values: The missing values are filled by its statistical features, such as the mean value and mode of the non-missing parts of the features. Interpolation methods and machine learning algorithms, like the random forest, also can be used to predict and fill the missing values. In addition, the missing values can be filled manually by using expert experience. These methods are suitable for a small number of missing values.

(3) Derived feature: Generate new features or introduce new values. If the current feature is missing, add a new feature. The corresponding new feature will have a value of 1; otherwise, the value will be 0. In this way, a new binary feature is generated using the missing feature. Alternatively, replace the missing value with some new value in the category variable like filling in the missing value with "Unknown".

Next, some processing of the original data is started. The purpose is to clean the messy original data into a format that meets the algorithm's requirements. Here are four steps to find some features that needed to be deleted.

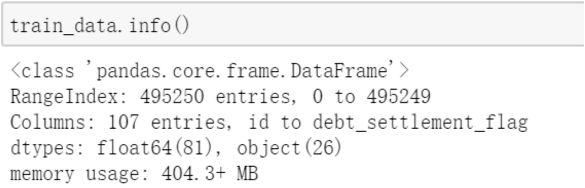
First, features that have a missing value are deleted if the missing ratio is more than 50%. After doing this, there are only 107 features left, with 81 float64 types and 26 object types.



```
train_data = train_data.dropna(axis=1, thresh=train_data.shape[0]/2)
train_data.head()
```

	id	loan_amnt	funded_amnt	funded_amnt_inv	term	int_rate	installment	grade	sub_grade	emp_title	...	pct_tl_nvr_dlq	percent_bc_gt_75
0	130872948	32000.0	32000.0	32000.0	60 months	25.81%	954.50	E	E4	Environmental Engineer	...	97.8	14.3
1	130962878	16000.0	16000.0	16000.0	36 months	6.07%	487.26	A	A2	Senior Director of Ticket Sales/Service	...	96.8	25.0
2	130963484	10000.0	10000.0	10000.0	36 months	7.96%	313.18	A	A5	Engineering Technician	...	92.3	33.3
3	130963678	10000.0	10000.0	10000.0	60 months	18.45%	256.39	D	D2	department manager	...	100.0	100.0
4	130955326	11200.0	11200.0	11200.0	60 months	30.79%	367.82	G	G1	Client services	...	71.4	0.0

5 rows × 107 columns

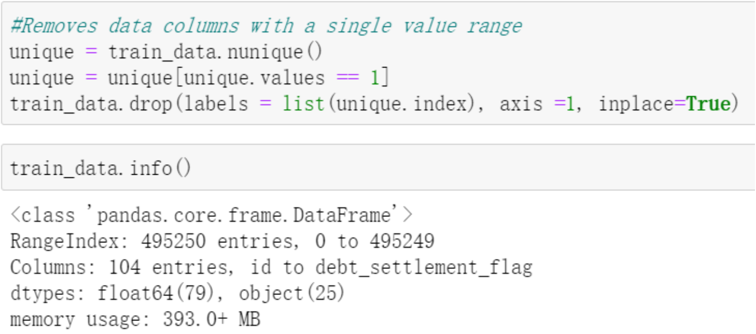


```
train_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 495250 entries, 0 to 495249
Columns: 107 entries, id to debt_settlement_flag
dtypes: float64(81), object(26)
memory usage: 404.3+ MB
```

Figure 15 Information after missing value ratio of more than 50% are deleted

Second, features with only one possible value are deleted. Because the values of these features are too single, there is no distinction between the data. By using "unique" in pandas to filter these variables, we have 104 features left.



```
#Removes data columns with a single value range
unique = train_data.nunique()
unique = unique[unique.values == 1]
train_data.drop(labels = list(unique.index), axis =1, inplace=True)
```

```
train_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 495250 entries, 0 to 495249
Columns: 104 entries, id to debt_settlement_flag
dtypes: float64(79), object(25)
memory usage: 393.0+ MB
```

Figure 16 Delete feature variables with unique value

Third, meaningless values or categorical features that have too many values are deleted. Meaningless features, for example, the feature "desc" in the original data, is a description of the loan, which cannot be directly modelled and has been deleted in the previous step due to too many missing values. The "zip code" is also repeated with the feature that represents the region. When it comes to the categorical features with too many kinds of values, "employment title ", which describes the borrower's job information, is the representative. Although these features contain much important information, it has too many values and is inconvenient to process.

Fourth, features with correlation coefficients approaching 1 are deleted. Figure 17 show the correlation coefficients of each feature of the data. The right side of the matrix heat map is the color band, which represents the mapping from value to a color. The value from small to large corresponds to the color from blue to red. In the upper left corner, three features "loan_amnt", "funded_amnt_inv ", "funded_amnt" have a highly significant correlation.

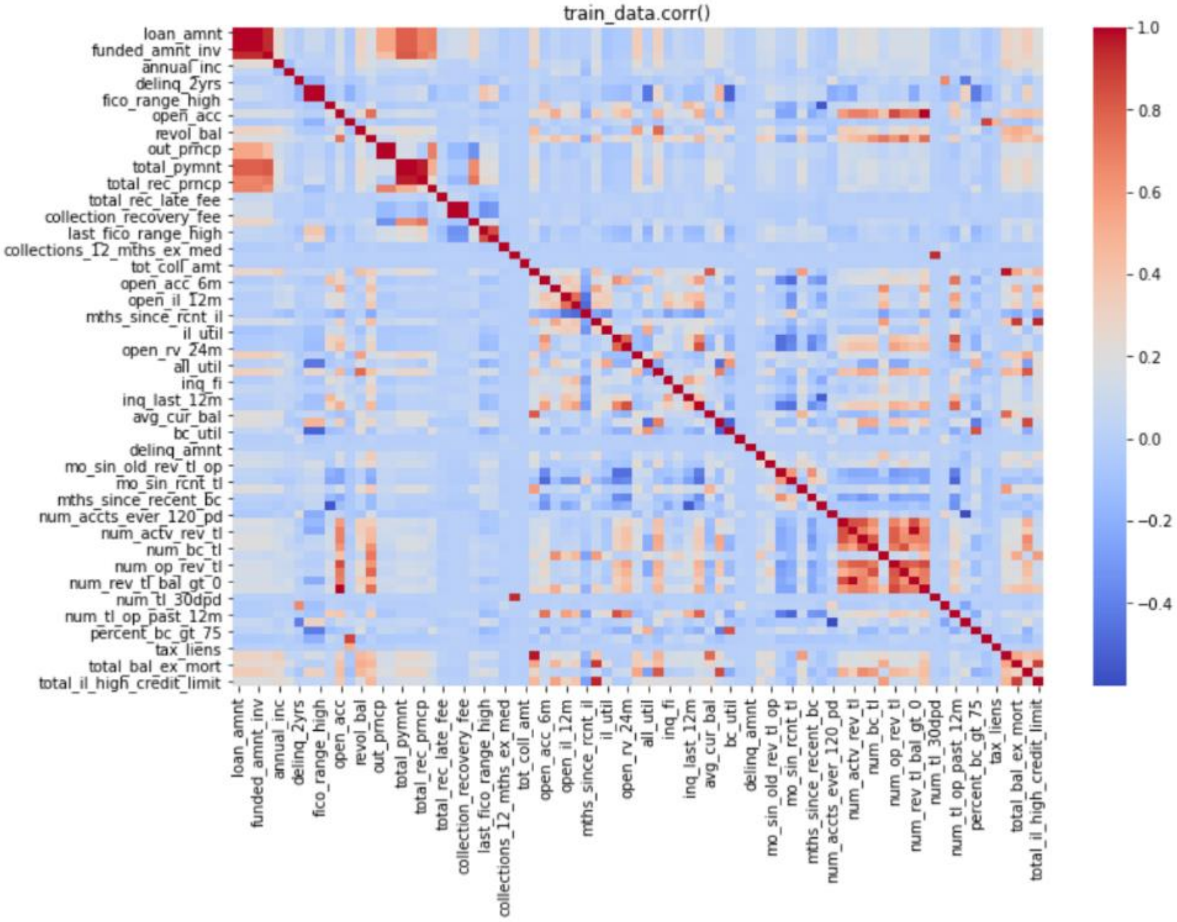
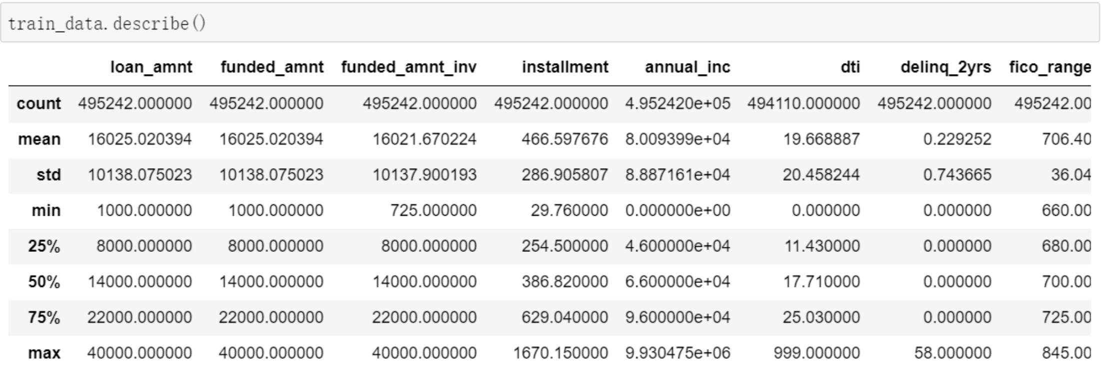


Figure 17 Features' correlation



```
train_data.describe()
```

	loan_amnt	funded_amnt	funded_amnt_inv	installment	annual_inc	dti	delinq_2yrs	fico_range
count	495242.000000	495242.000000	495242.000000	495242.000000	4.952420e+05	494110.000000	495242.000000	495242.00
mean	16025.020394	16025.020394	16021.670224	466.597676	8.009399e+04	19.668887	0.229252	706.40
std	10138.075023	10138.075023	10137.900193	286.905807	8.887161e+04	20.458244	0.743665	36.04
min	1000.000000	1000.000000	725.000000	29.760000	0.000000e+00	0.000000	0.000000	660.00
25%	8000.000000	8000.000000	8000.000000	254.500000	4.600000e+04	11.430000	0.000000	680.00
50%	14000.000000	14000.000000	14000.000000	386.820000	6.600000e+04	17.710000	0.000000	700.00
75%	22000.000000	22000.000000	22000.000000	629.040000	9.600000e+04	25.030000	0.000000	725.00
max	40000.000000	40000.000000	40000.000000	1670.150000	9.930475e+06	999.000000	58.000000	845.00

Figure 18 Statistical descriptions

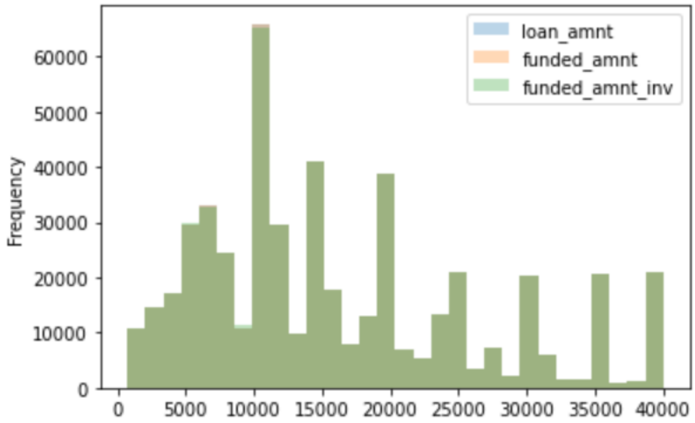


Figure 19 Histogram of three features' frequency

From the above data description, it can be seen that the mean, variance, maximum and minimum indicators of the "loan_amnt", "funded_amnt_inv", "funded_amnt" columns are almost the same. Therefore, two of the columns need to be deleted. Observe by drawing a histogram, "funded_amnt_inv" covers the other two columns of data. Hence, "funded_amnt_inv", "funded_amnt" are been deleted, keep only "loan_amnt".

Last, if the proportion of missing values in not particularly high, use the mean value to fill in these features. The following table counts the number of missing values and the corresponding proportions.

Feature	Number of missing values	Missing rate
Id	0	0
Loan_amnt	8	0.000016
Term	8	0.000016
Int_rate	8	0.000016
Installment	8	0.000016
…	…	…
Total-bal_ex_mort	8	0.000016
Total_bc_limit	8	0.000016
Total_il_high_credit_limit	8	0.000016
Hardship_flag	12952	0.026152
Debt_settlement_flag	8	0.000016

Table 3 Number of missing values and missing rate

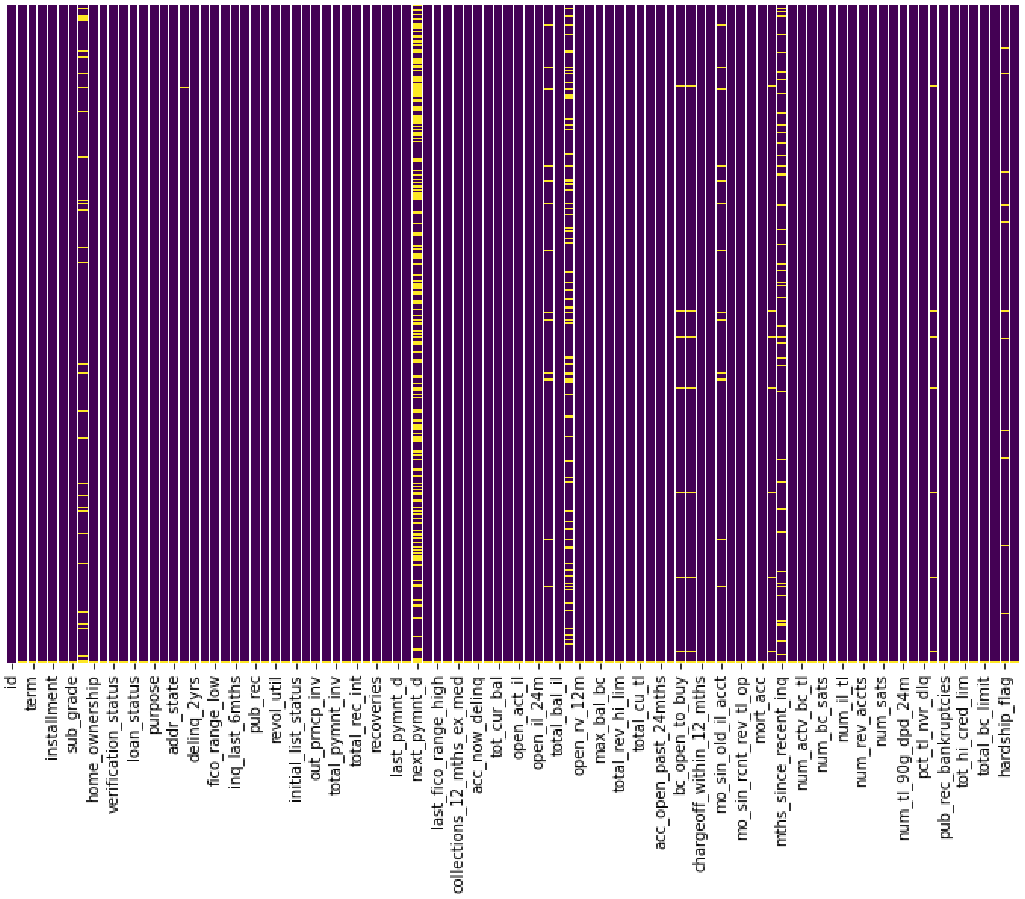


Figure 20 Heatmap of missing values

As shown in Figure 20, the following heat map make the missing values more visible. Yellow represents NaN.

Then, use the mean values to fill in these features with missing values, including 'loan_amnt', 'annual_inc','dti', 'inq_last_6mths', 'open_acc', 'revol_bal', 'total_acc', 'total_pymnt', 'total_rec_int' .

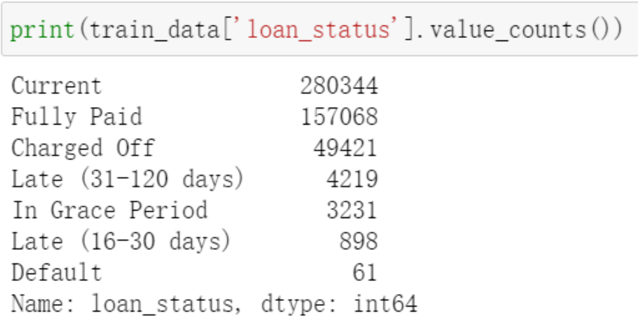
4.2.2 Feature Abstraction

In machine learning models, all data should be numerical because machine learning is based on mathematical function methods. When categorical data appears in the data set we want to analyze, the data is not ideal because we cannot process them mathematically.

After processing the missing values of the data in the previous steps, there are still 23 categorical variables in the data set. According to observation, these categorical variables are mainly divided into two types. One is nominal categorical variables, and the other is ordinal categorical variables. Now, these features need to be digitized with different methods.

First, some features are belonged to nominal categorical variables, such as "term", "home_ownership", "purpose", "verification_status".

The borrower's loan status can be divided into the following states, as shown below.

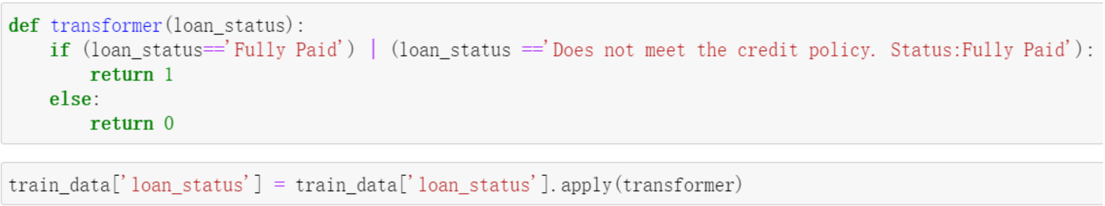


```
print(train_data['loan_status'].value_counts())
```

```
Current               280344
Fully Paid            157068
Charged Off            49421
Late (31-120 days)      4219
In Grace Period         3231
Late (16-30 days)        898
Default                   61
Name: loan_status, dtype: int64
```

Figure 21 loan status of borrowers

This paper is mainly to evaluate the default risk of P2P network lending. Therefore, it is necessary to classify the default samples and the performance samples in order to select the well-performing borrowers as the customers better. This is the main step to achieve the purpose of downgrading the default risk of the borrowers. This paper selects fulfilling customers whose transaction records are in the Fully Paid status and digitizes them as 1. The rest of the status is regarded as the default category, digitized them as 0.

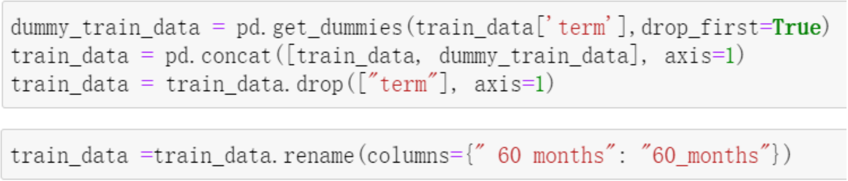


```
def transformer(loan_status):
    if (loan_status=='Fully Paid') | (loan_status =='Does not meet the credit policy. Status:Fully Paid'):
        return 1
    else:
        return 0
```

```
train_data['loan_status'] = train_data['loan_status'].apply(transformer)
```

Figure 22 Abstract Feature "loan_status"

categorical variables. Take "term" as an example. Its variable values are 36 months and 60 months. I used the "get dummies" method in pandas to create virtual features. Each column of the virtual feature represents a category of variable attributes. Then use the pandas "concat" method to splice the new virtual feature with the original data.

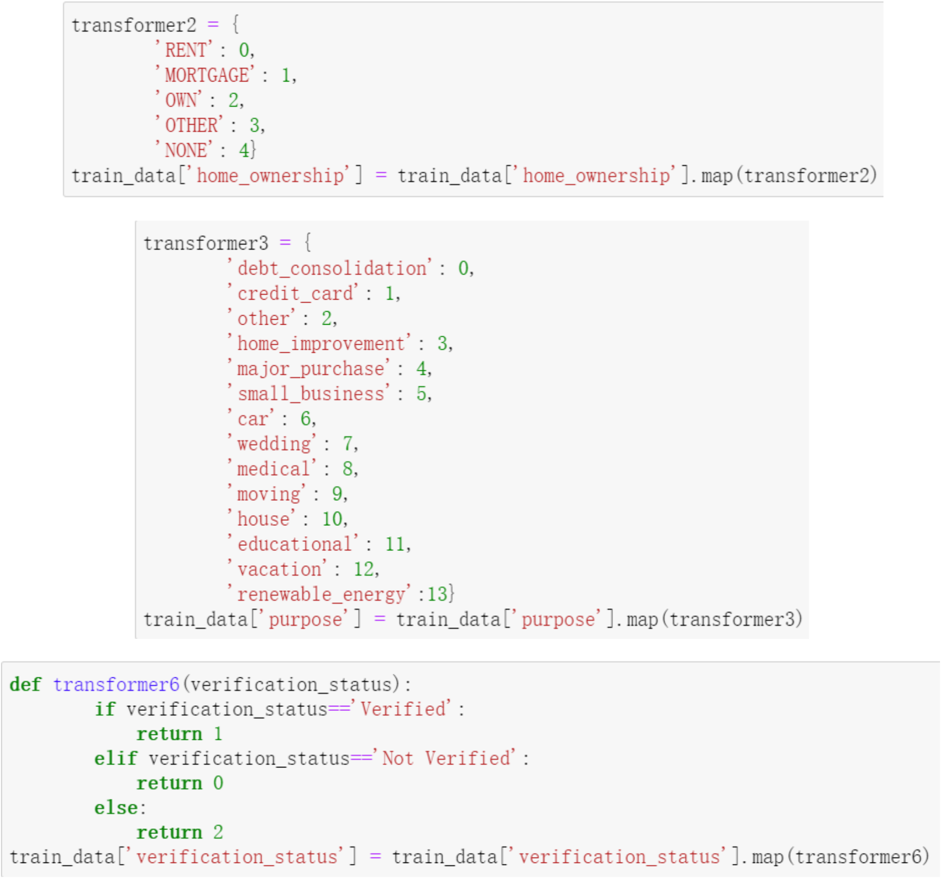


```
dummy_train_data = pd.get_dummies(train_data['term'],drop_first=True)
train_data = pd.concat([train_data, dummy_train_data], axis=1)
train_data = train_data.drop(["term"], axis=1)
```

```
train_data =train_data.rename(columns={" 60 months": "60_months"})
```

Figure 23 Abstract Feature "term"

The rest of nominal categorical variables above mentioned are used label encoding. The details are shown below.



```
transformer2 = {
        'RENT': 0,
        'MORTGAGE': 1,
        'OWN': 2,
        'OTHER': 3,
        'NONE': 4}
train_data['home_ownership'] = train_data['home_ownership'].map(transformer2)
```

```
transformer3 = {
        'debt_consolidation': 0,
        'credit_card': 1,
        'other': 2,
        'home_improvement': 3,
        'major_purchase': 4,
        'small_business': 5,
        'car': 6,
        'wedding': 7,
        'medical': 8,
        'moving': 9,
        'house': 10,
        'educational': 11,
        'vacation': 12,
        'renewable_energy':13}
train_data['purpose'] = train_data['purpose'].map(transformer3)
```

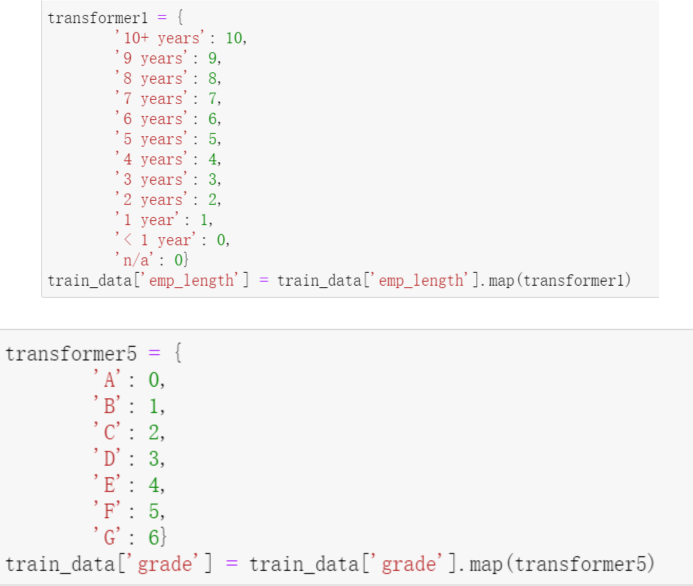
```
def transformer6(verification_status):
        if verification_status=='Verified':
            return 1
        elif verification_status=='Not Verified':
            return 0
        else:
            return 2
train_data['verification_status'] = train_data['verification_status'].map(transformer6)
```

Figure 24 Abstract Features "home_ownership", "purpose", "verification"_status

The ordinal categorical variables are also called rank variables. The characteristics of the original variables should be preserved in the digitization process. In the Lending Club data, "grade", "emp_length" are belongs to this type.

Lending Club categorizes the credit ratings of loan applicants from A to G and matches loan interest rates according to different credit ratings accordingly. Customers with grade A have better credit scores than customers with grade B.

Here I used the label encoder method to abstract these two features. The processes are shown below:

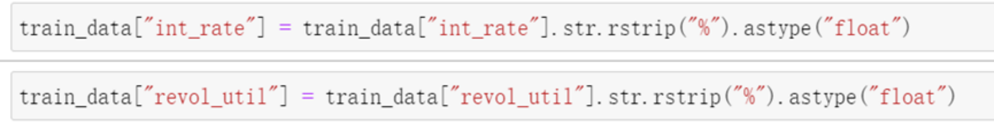


```
transformer1 = {
        '10+ years': 10,
        '9 years': 9,
        '8 years': 8,
        '7 years': 7,
        '6 years': 6,
        '5 years': 5,
        '4 years': 4,
        '3 years': 3,
        '2 years': 2,
        '1 year': 1,
        '< 1 year': 0,
        'n/a': 0}
train_data['emp_length'] = train_data['emp_length'].map(transformer1)
```

```
transformer5 = {
        'A': 0,
        'B': 1,
        'C': 2,
        'D': 3,
        'E': 4,
        'F': 5,
        'G': 6}
train_data['grade'] = train_data['grade'].map(transformer5)
```

Figure 25 Abstract ordinal features "emp_length", "grade"

We noticed that in categorical variables, the attributes of "int_rate", "revol_util" are essentially numerical values, but pandas misrecognize them as characters because they contain the "%" symbol or there are commas between the numbers. In order to facilitate subsequent processing, we first reclassify their data types.

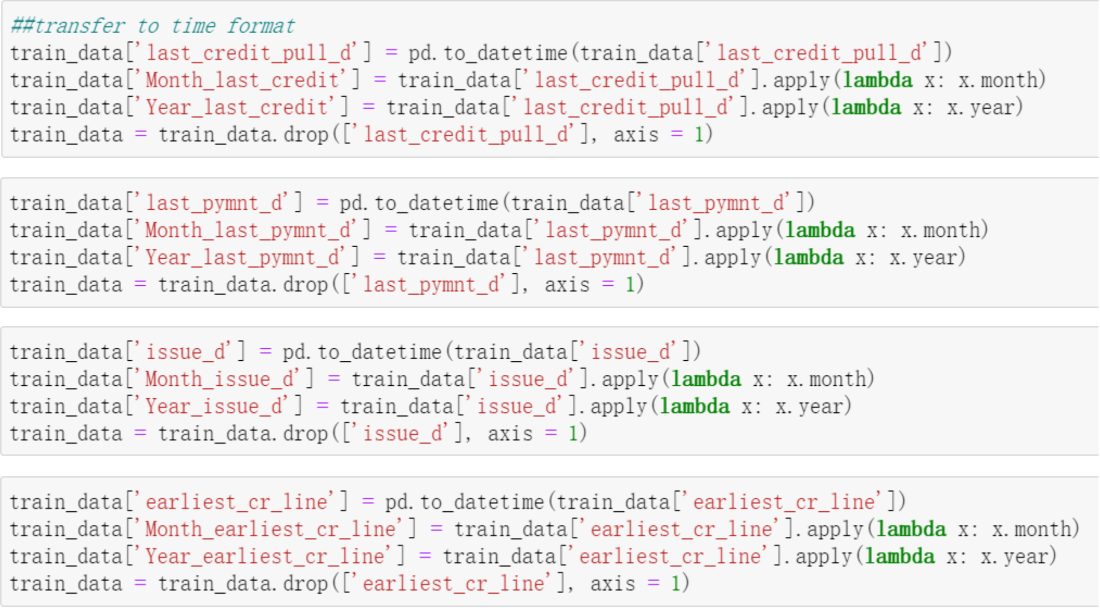


```
train_data["int_rate"] = train_data["int_rate"].str.rstrip("%").astype("float")
```

```
train_data["revol_util"] = train_data["revol_util"].str.rstrip("%").astype("float")
```

Figure 26 Transform "int_rate", "revol_util" to numerical variables

For some variables representing time, use the python toolkit "datetime" to process date features and convert the original character data into monthly and annual numerical data.



```
##transfer to time format
train_data['last_credit_pull_d'] = pd.to_datetime(train_data['last_credit_pull_d'])
train_data['Month_last_credit'] = train_data['last_credit_pull_d'].apply(lambda x: x.month)
train_data['Year_last_credit'] = train_data['last_credit_pull_d'].apply(lambda x: x.year)
train_data = train_data.drop(['last_credit_pull_d'], axis = 1)
```

```
train_data['last_pymnt_d'] = pd.to_datetime(train_data['last_pymnt_d'])
train_data['Month_last_pymnt_d'] = train_data['last_pymnt_d'].apply(lambda x: x.month)
train_data['Year_last_pymnt_d'] = train_data['last_pymnt_d'].apply(lambda x: x.year)
train_data = train_data.drop(['last_pymnt_d'], axis = 1)
```

```
train_data['issue_d'] = pd.to_datetime(train_data['issue_d'])
train_data['Month_issue_d'] = train_data['issue_d'].apply(lambda x: x.month)
train_data['Year_issue_d'] = train_data['issue_d'].apply(lambda x: x.year)
train_data = train_data.drop(['issue_d'], axis = 1)
```

```
train_data['earliest_cr_line'] = pd.to_datetime(train_data['earliest_cr_line'])
train_data['Month_earliest_cr_line'] = train_data['earliest_cr_line'].apply(lambda x: x.month)
train_data['Year_earliest_cr_line'] = train_data['earliest_cr_line'].apply(lambda x: x.year)
train_data = train_data.drop(['earliest_cr_line'], axis = 1)
```

Figure 27 Abstract date features

4.2.3 Outlier Identification and Processing

Outliers are those far away from the vast majority of sample points. Usually, such data points show unreasonable characteristics in the data set. If these outliers are ignored, it will lead to wrong conclusions in some modeling scenarios. In general, the identification of outliers can be done with the help of graphical methods (such as box plots, normal distribution diagrams) and modelling methods (such as linear regression, clustering algorithm, K-nearest neighbor algorithm).

When the outliers are identified, there are usually the following three methods to deal with them.

(1) Delete. If you want to find out the general laws of the data, and there are not too many outliers, you can consider deleting them because outliers may affect the conclusion. This explains why in many matches, the highest score and the lowest

score are removed from the platers' final scores.

(2) Keep the outliers. Because the outliers also represent real events, and there are specific behaviors behind them. Even if some values are abnormal, they will not affect the model.

(3) Fill outliers with mean or mode value. Because rashly deleting data may lose information, but if you leave it alone, it may affect the model, so you can consider filling in with the mean or mode value.

This paper will use box plot, the most common method, to identify outliers. In Python, the "matplotlib" module can be used to realize data visualization. In Figure 28, it shows the relationship between employment length and annual income, which has a lot of obvious outliers, especially when "employment length=10".

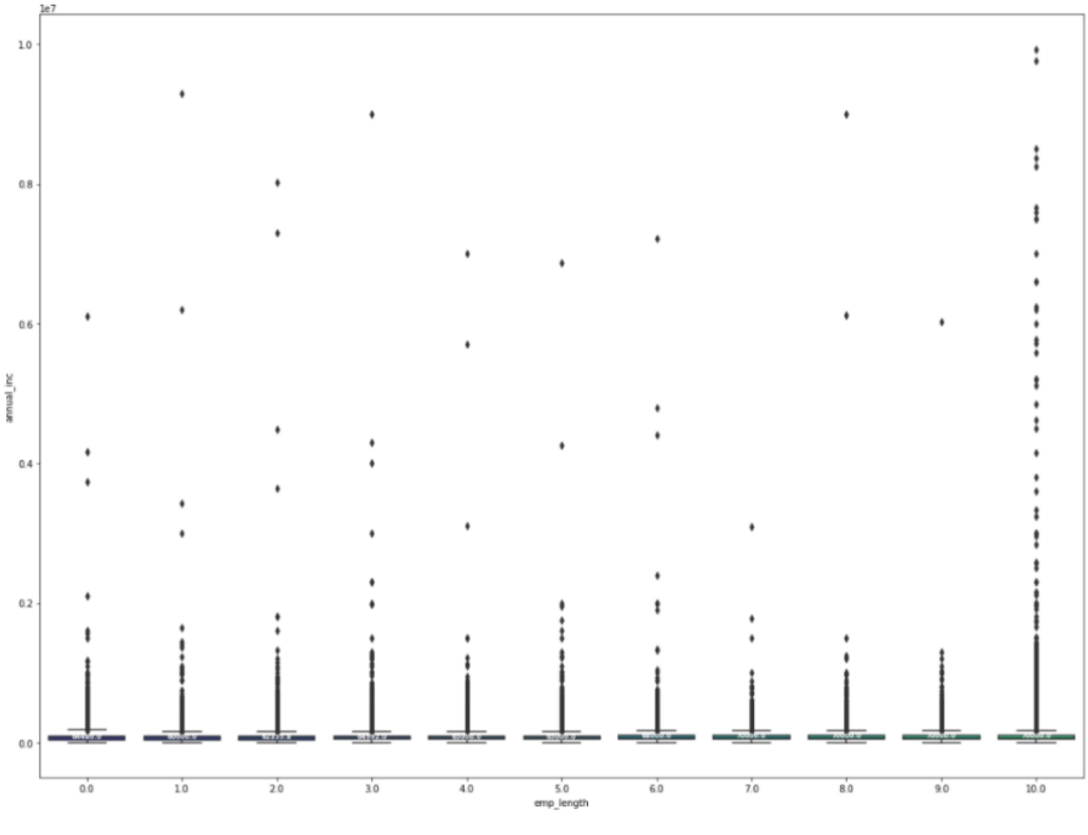


Figure 28 Identify "employment length" 's outliers by boxplot

It is shown in Figure 29, the annual income data has many noticeable outliers. In order to avoid these outliers having a more obvious impact on subsequent model estimates, the annual income data will be ranked from high to low, and the top one-thousandth of the data will be excluded. After processing, we can see in Figure 30, the extreme value in annual income has been significantly reduced.

Other two features "revol_bal" and "inq_last_6mths" have quite similar distribution, the boxplot of the revolving balance and the employment length is shown below. For these outliers in these two features, they will be regarded as missing values, and the mean value will be used to fill in. After doing this, the outliers reduced obviously.

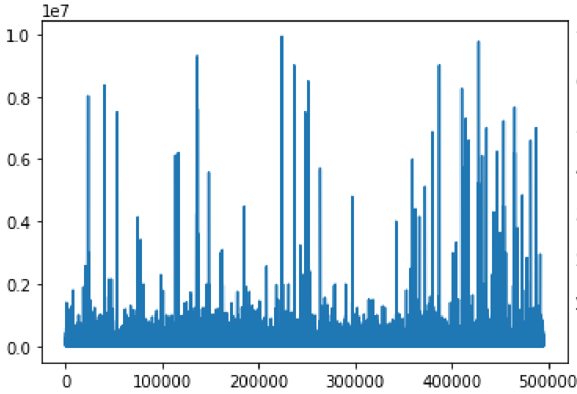


Figure 29 Annual income (before removing outliers)

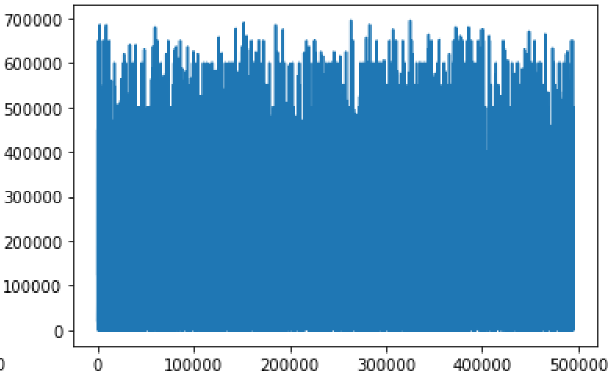


Figure 30 Annual income (after removing outliers)

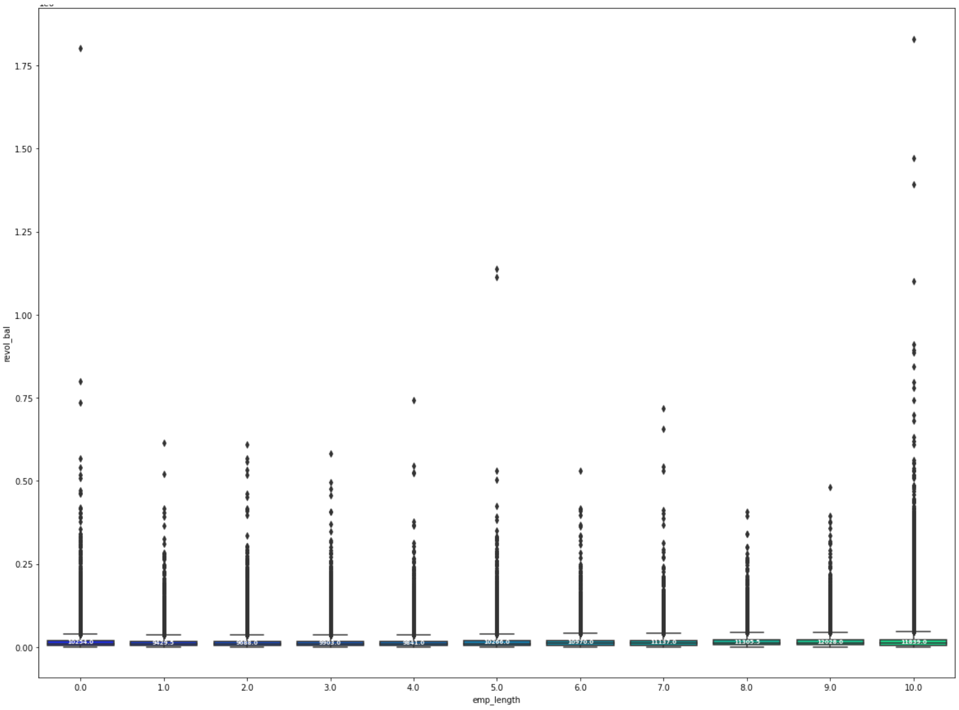


Figure 31 Identify outliers by boxplot

4.3 Feature Selection

After data cleaning, the sample data still contains 124 features. The number of features has a certain impact on the final prediction accuracy of the model. Under the same prediction accuracy, the fewer features the model chooses, the better the model's effectiveness. This chapter will select features according to their importance.

4.3.1 Overview of Random Forest Algorithm

As the name implies, random forest, is an ensemble learning model that takes decision trees as base learners. Its basic principle is to do simple random sampling with replacement from the original data set and obtain several subsets. Then, the subsets are used to train different base classifiers, and finally use the voting method to select the best classification result.

(1) Bootstrap

In random forest, the method of sampling from the original data set and obtaining a new data set with random replacement is called bootstrap. Its basic rule is: suppose there are m samples in the original data set D, randomly take out a sample from D every time to copy, and then put the copied sample into the new data set D', and take it. The original sample is put back into the original data set D, so as to ensure that there is still a chance to get the sample next time. This sampling process is repeated m times, and finally a new data set D' contains m samples.

The above method shows that there is a part of the original data set D in the new data set D', and other samples have not appeared. Assuming that the probability that the sample in the original data set has not been sampled is P,

$$P = (1 - \frac{1}{m})^m$$

When $n \to \infty, P = \frac{1}{e} \approx 0.368$. This means, during the bootstrap process, about 36.8% of the samples in the original data set did not appear in the new data set.

(2) Random Forest Model Construction

Step 1: According to the number of samples in the original sample set, bootstrap is used to sample so that the number of samples in the new data set is the same as the original sample set. Moreover, do N samplings to generate N new data subsets.

Step 2: Train the new N data subsets separately to obtain N decision tree models.

Step 3: For each decision tree model, when there are n features in the corresponding data subset, the best feature is selected according to the Gini index for each splitting until all samples of the node belong to the same category. At this time, the trained decision tree does not need to be pruned.

Step 4: Combine all the trained decision tree models into a random forest model. For classification problems, use the voting results of each decision tree to determine the final classification result. For regression problems, the average value of the results predicted by all decision trees is used to determine the final prediction result.

4.3.2 Use random forest to select important feature

In practical applications, a data set often contains many features, and inputting too many features for model training will not only increase the training time but also overfit the training results of the model. Therefore, this paper uses the random forest algorithm to evaluate the importance of features. Commonly used evaluation indicators are the Gini coefficient and out-of-bag (OOB) error rate.

Suppose now there are m variables $x_1, x_2, \dots, x_m$, the feature importance measures are represented by VIM, and the Gini coefficient and the OOB error of x_i are represented by VIM_i^{Gini} and VIM_i^{OOB}, respectively.

(1) Gini Coefficient

VIM_i^{Gini} represents the average change in the impurity of node splitting in all decision trees of the random forest for the i^{th} variable. The formula is shown below:

$$GI_m = \sum_{k=1}^{K} p_{mk}(1 - p_{mk})$$

K is number of classes, p_{mk} is the estimated probability that the sample of node m belongs to the k^{th} class.

When K=2, the Gini coefficient at node m is:

$$GI_m = 2p_{mk}(1 - p_{mk})$$

At this time, p_{mk} is the probability estimate that the sample belongs to any category at node m.

Therefore, the change in the Gini coefficient of node m before and after branching is $GI_m - GI_l - GI_r$, where GI_l and GI_r are the Gini coefficients of two new nodes that after node m is classified into left and right. This is the feature importance measure of x_i on node m.

$$VIM_{im}^{Gini} = GI_m - GI_l - GI_r$$

If the variable x_i appears M times on the j^{th} tree, then its VIM is

$$VIM_{ij}^{Gini} = \sum\nolimits_{m=1}^{M} VIM_{im}^{Gini}$$

The VIM of variable x_i in the entire random forest is

$$VIM_i^{Gini} = \frac{1}{n}\sum\nolimits_{m=1}^{M} VIM_{ij}^{Gini}$$

With n is the number of decision trees in the random forest.

(2) Out-of-bag (OOB) Error Rate

The formula for calculating the OOB error rate is to use a new sample training set drawn from random replacement to generate a decision tree, and calculate the OOB prediction error rate. Then, after randomly replacing the observed value of the variable x, a new decision tree is generated and the new out-of-bag data prediction error rate is calculated. Finally, the average value of the difference between the two error rates in all trees is called the feature importance of the variable x_i. Then the VIM of the variable x_i in the j^{th} tree is

$$VIM_{ij}^{OOB} = \frac{\sum_{p=1}^{n_j} I(Y_p = Y_p^j)}{n_j} - \frac{\sum_{p=1}^{n_j} I(Y_p = Y_{p,\sigma_i}^j)}{n_j}$$

With n_j is the number of observations on the j^{th} tree. $I(x)$ is piecewise function, when the two values are equal, the function value is 1, and when the two values are not equal, the function value is 0. Y_p is the true result of the p^{th} observation. Y_p^j is the p^{th} observation result of the j^{th} tree on the OOB data before doing simple random sampling with replacement. Y_{p,σ_i}^j is the p^{th} observation result of the j^{th} tree on the OOB data after doing simple random sampling with replacement. If variable x_i does not appear on the j^{th} tree, $VIM_{ij}^{OOB} = 0$. Therefore, the VIM of variable x_i is defined as

$$VIM_i^{OOB} = \frac{1}{n}\sum_{j=1}^{n} VIM_{ij}^{OOB}$$

With n is the number of decision trees in the random forest.

VIM_i^{Gini} has a wide range of applications in estimating the importance of variables. When the variables are continuous and uncorrelated, the estimation of VIM_i^{Gini} is unbiased, and its stability will be higher. But when there are continuous variables and categorical variables or different levels of categorical variables in the original data set, the VIM_i^{Gini} estimation is not accurate enough. This is because even if some variables have no categorical effect, there is the possibility of reducing the Gini coefficient, which means that VIM_i^{Gini} is overestimated.

In practical applications, VIM_i^{OOB} is also widely used, because VIM_i^{OOB} calculations use out-of-bag data, which can be regarded as the classification ability of variables. If it is a variable without classification ability, the OOB error rate will not change before and after the observation value is replaced. At the same time, when there are categorical variables and continuous variables in the original data set, or the level of categorical variables is different, it will not affect the accuracy of VIM_i^{OOB}. Considering their respective characteristics, and this paper has converted all categorical features into numerical features, it is enough to use the default Gini coefficient to calculate VIM.

After using the random forest algorithm to measure the importance of Lending Club credit data features, top 20 important features are retained. Through the histogram below, the importance of these features can be visualized in descending order. Obviously, from the results, it can be concluded that financial features are more influencing than demographic features in determining credit risk.

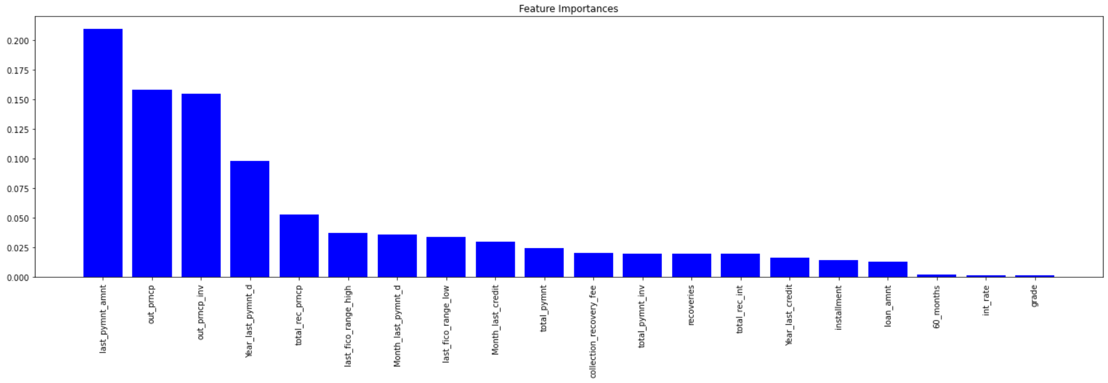


Figure 32 20 Important Features in Descending Order

The following table is a display of these 20 important features and their respective descriptions are detailed in the appendix.

Number	Feature	Number	Feature
1	loan_amnt	11	collection_recovery_fee
2	int_rate	12	last_pymnt_amnt
3	installment	13	last_fico_range_high
4	grade	14	last_fico_range_low

5	out_prncp	15	60_months
6	out_prncp_inv	16	Month_last_credit
7	total_pymnt	17	Year_last_credit
8	total_pymnt_inv	18	Month_last_pymnt_d
9	total_rec_prncp	19	Year_last_pymnt_d
10	total_rec_int	20	recoveries

Table 4 20 Selected Features

The above is the result of 2018 data processing. The results after data processing in 2019 are similar to those in 2018, which have 22 features being selected. Compared with the features of 2018, their 20 common features will be used as the final features of the model introduced in the next chapter.

Chapter 5

5 Credit Risk Measurement by BP Neural Networks

5.1 Feasibility of BP Neural Networks

With the rapid development of information technology, the application of artificial intelligence provides a powerful means for credit risk assessment, the most representative of which is the assessment model based on artificial neural networks. The artificial neural network was first proposed by biologist McCulloch and mathematician Pitts in 1943. It consists of a large number of neurons connected to each other to form a complex nonlinear network (Zhang, L., & Zhang, B. 1999). The network has strong self-organization, self-adaptability and robustness. Neural network is a simulation of the biological nervous system. It is good at imitating the human brain to analyze and predict data and can use the structure of the network itself to express the relationship between input and output. The neural network model regards credit risk assessment as a classification problem of pattern recognition, and establishes a discriminant model by extracting characteristic features of defaulting borrowers and non-defaulting borrowers, so as to make classification predictions on the credit status of other borrowers.

P2P network lending is an emerging Internet financial model. When using neural networks for credit risk assessment (Guo, Y. 2020), on the one hand, it uses its non-linear mapping ability to find the credit evaluation indicators and credit status of network borrowers. On the other hand, it uses its generalization ability to infer the characteristic attributes of the borrower sample. The neural network adjusts the network structure through continuous learning of the borrower's historical credit data. It can accurately process the non-linear mapping relationship between credit risk assessment indicators and credit ratings (Zhang, S., Hu, Y., & Wang, C. 2019), so as to classify the credit rating of borrowers based on credit indicators. In addition, the artificial neural network has a powerful parallel processing mechanism and a distributed storage structure. It is good at handling uncertain information in credit risk assessment, and its modeling process does not need to determine a clear functional relationship between variables, nor does it require strict data. Therefore, it is faster and more convenient in the application of credit risk assessment.

The neural network model has powerful non-linear processing capabilities and generalization capabilities. It does not need to make strict assumptions about the distribution of variables and can directly obtain knowledge from the training data set, so it has the following advantages in credit risk assessment applications:

(1) BP neural network has self-learning and self-adaptive capabilities. In other words, it can learn from the environment and improve its performance. There are a large number of adjustable parameters inside, so the system is more flexible.
(2) BP neural networks model is essentially a mapping from input to output. The mathematical theory proves that neural network can approximate any non-linear

mapping relationship with arbitrary precision. Different countries and regions have different starting points for personal credit rating assessment, and relevant information is not comprehensive. The ability of the BP network to approximate arbitrary non-linear mapping relations is very suitable for solving problems with almost no rules, multiple constraints or incomplete data.

(3) The acquired learning ability of the BP neural networks enables it to learn as the environment changes continuously. When carrying out credit rating evaluation and prediction, restricted by the fuzziness of the evaluation data, laws can be discovered from a large number of complex data with unknown patterns. Compared with traditional evaluation methods, the performance of the BP neural networks is significantly better.

(4) The BP neural network method is a natural non-linear modeling process. It overcomes the complexity of the traditional analysis process and the difficulty of choosing an appropriate model function form. It does not need to distinguish what kind of non-linear relationship exists, which brings great convenience to establish models and analysis.

(5) The BP neural network method better guarantees the objectivity of the evaluation and prediction results. In traditional personal credit risk assessment, most of the determinants are the subjective judgments of loan officers. By contrast, the BP neural network can reproduce the expert's experience, knowledge and intuitive thinking, which better guarantees the objectivity of the evaluation and prediction results.

5.2 Sample Selection and Data Normalization

5.2.1 Sample Selection

The data of this model comes from the real loan transaction data of the Lending Club platform from 2018 to 2019. In this paper, the sample data set was partitioned in to two subsets: training samples and test samples. Let the 749,770 pieces of data from January 2018 to June 2019 as the training set of the model (nearly 75% of the observations), and the 255,187 pieces of data from July 2019 to December 2019 as the testing set (nearly 25% of the observations) to construct the BP neural network model. The process of data processing and model construction is implemented in the Jupyter notebook software.

5.2.2 Normalization

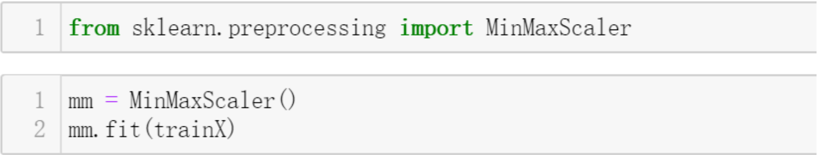
The economic significance and units among the various indicators of the data are not consistent. In order to avoid large defects caused by the numerical difference between the indicators and ensure the accuracy of the data analysis results, it is necessary to eliminate the dimensional influence between the indicators. The dimensional index expression is transformed into a dimensionless index expression (Nawi, NM, Atomi, WH, & Rehman, MZ. 2013), making the data's absolute value become a relative value. After normalization, each indicator data can be processed to the same magnitude, and each data indicator can be comparable, which is suitable for subsequent comprehensive comparison and analysis of data.

When the data X is centered according to the minimum value and scaled by the range ($maximum - minimum$), the data moves by the minimum unit and is converged to between [0,1]. This process is called Normalization (also known as Min-Max Scaling). The normalized data obey normally distributed. In this paper, "sklearn.preprocessing" is implemented to realize the Min-Max Scaler function. The default range that Min-Max Scaler compresses the data to is [0,1].

Before building the model, first and foremost, it is necessary to normalize all data to ensure the accuracy of model training. The method used is the maximum and minimum interval mapping method, its mathematical formula is shown below, and the processing code descriptions are shown in Figure 31.

$$X' = \frac{X - X_{min}}{X_{max} - X_{min}}$$

with X' is the feature after normalization, X is the feature before normalization, X_{max} and X_{min} are the maximum and minimum value.



```
from sklearn.preprocessing import MinMaxScaler
```

```
mm = MinMaxScaler()
mm.fit(trainX)
```

Figure 33 Max-Min Scaler Normalization

5.3 BP Neural Networks Model Training and Testing

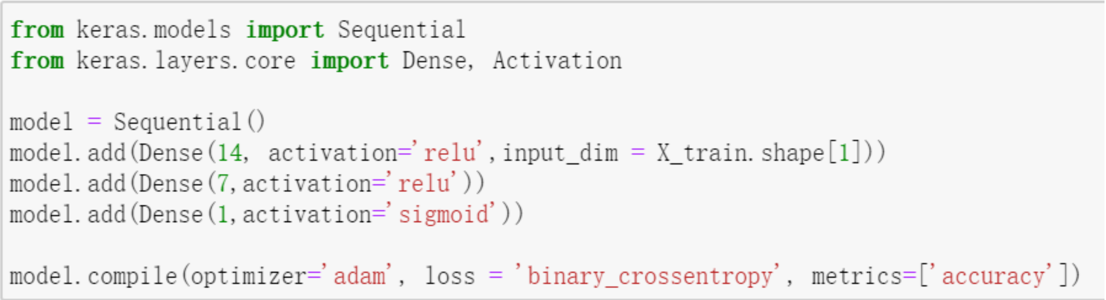
5.3.1 Parameter Choices and Model Training

In this study, data from July 2019 - December 2019 period was selected as the testing sample.

The BP neural network model constructed in this paper adopts a two hidden layers structure. The first hidden layer has 14 nodes, and the second layer has 7 nodes. Both take ReLU as the activation function. The classification layer has only one node and is using Sigmoid as the activation function.

ADAM optimizer is used and the loss function is binary cross-entropy. In order to ensure the stability of the system, it is generally preferred to choose a smaller learning rate. Here the built-in learning rate of the system is 0.001. Binary cross entropy is the default loss function of binary classification problem. That is why the classification layer must use the sigmoid activation function. The sigmoid function can convert any real value to the range (0,1).

Epochs are defined as all training samples propagated forward and backward in the neural network. This means that one cycle is a single forward and backward pass of the entire input data. Simply speaking, epochs refer to how many times the data will be "rounded" during the training process. In this study, we set the epochs to 20 when training the model, and the process is shown in Figure 34.



```
from keras.models import Sequential
from keras.layers.core import Dense, Activation

model = Sequential()
model.add(Dense(14, activation='relu',input_dim = X_train.shape[1]))
model.add(Dense(7,activation='relu'))
model.add(Dense(1,activation='sigmoid'))

model.compile(optimizer='adam', loss = 'binary_crossentropy', metrics=['accuracy'])
```

Figure 34 BP neural network model training process

5.3.2 Test Error and Analyze Accuracy

When training the model, we often check the model based on the training loss and the validation loss, we hope to optimize the parameter training at a better model. In case underfit or overfit will happen. This better means the model has strong generalization ability (generalization).

This general strategy is to use validation data to measure the effect of the selection of different hyperparameters (such as training rounds, learning rate, best network architecture, etc.). We use this method to find suitable values for hyperparameters.

Because validation data is a part of the training set, it can be used to verify the validation set after each epoch, or every few epochs, to find problems early, such as overfitting, or problems with hyperparameter settings.

Figure 35 shows the training loss has a declining trend, and the validation loss is also in a declining trend as a whole. Training loss and validation loss are almost all within 0.013, indicating that the network model learning has a good effect.

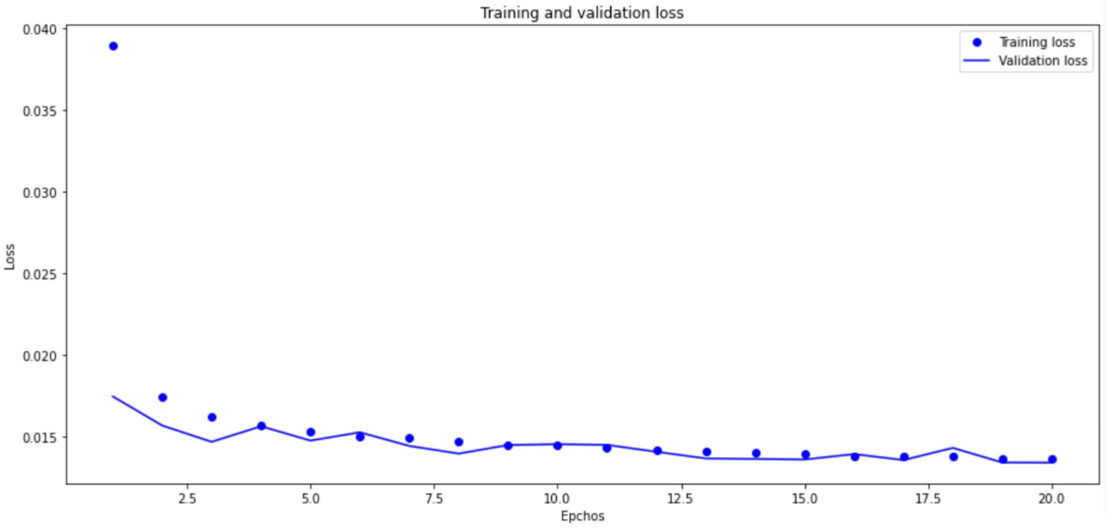


Figure 35 Training loss and validation loss

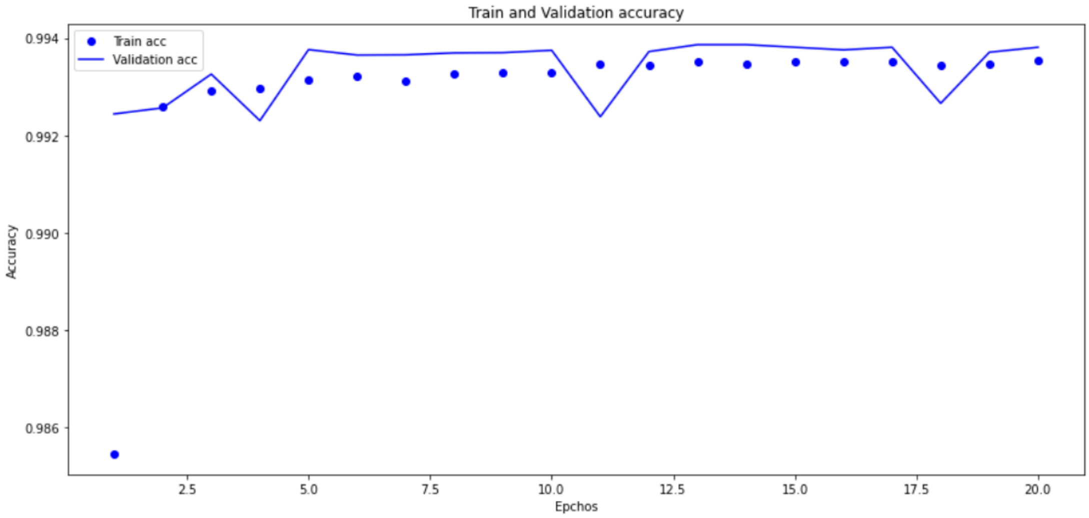
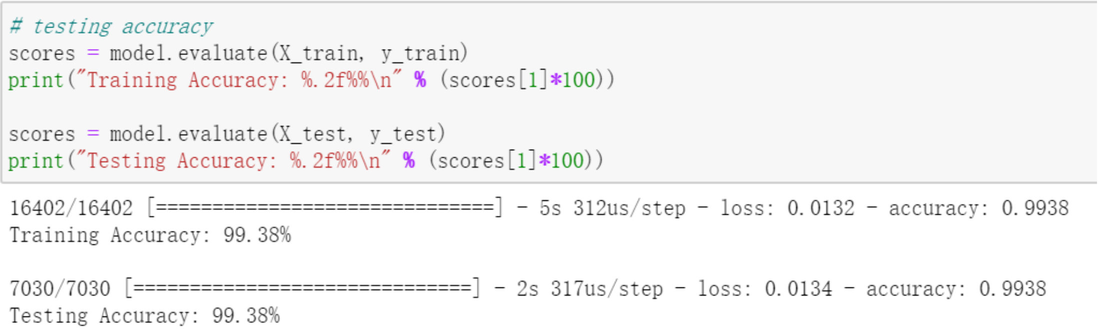


Figure 36 Training Accuracy and Validation Accuracy

As the number of training increases, training accuracy and validation accuracy are also increasing in general. At 20 iterations of epochs, the training results are relatively stable, and the training effect is better than those of which epochs are less than 20. The corresponding trend is shown in Figure 36.

5.3.3 Empirical Results

According to the comparison of the actual original values and the predicted output result, it can be seen that the simulation of the constructed measurement system works well, which means this neural network credit risk model was successful in classifying default and non- default loans. The error of the test sample is controlled within 0.0132, and the accuracy reaches 99.38%. This result shows that the built risk evaluation network has a good simulation effect, good stability and practicability, and can be widely used in P2P lending platform risk evaluation. The specific output results are shown in figure 37.



```
# testing accuracy
scores = model.evaluate(X_train, y_train)
print("Training Accuracy: %.2f%%\n" % (scores[1]*100))

scores = model.evaluate(X_test, y_test)
print("Testing Accuracy: %.2f%%\n" % (scores[1]*100))
```

```
16402/16402 [==============================] - 5s 312us/step - loss: 0.0132 - accuracy: 0.9938
Training Accuracy: 99.38%

7030/7030 [==============================] - 2s 317us/step - loss: 0.0134 - accuracy: 0.9938
Testing Accuracy: 99.38%
```

Figure 37 Training and testing output results

Chapter 6

5 Conclusion

For credit risk, the fundamental source is the information asymmetry between the platform and the borrower. It can be seen that the best way to reduce credit risk is to eliminate the information asymmetry between the two. This paper uses Lending Club's relatively complete borrower's credit-related data and uses a neural network credit model to classify credit applications for credit data, showing a good result. Therefore, by identifying default loans in advance, lenders can reduce their financial losses by avoiding investment in bad applicants.

In the actual users’ credit risk assessment process, the research ideas in this paper and the selection of borrower user characteristics have a certain reference value. On this basis, a credit scoring model can also be established to build a more accurate credit situation for the P2P network platform.

However, the research in this paper still has some limitations: First, this research only considers one P2P lending case, but there may be more or less similarities and differences between different P2P lending platforms. Hence, it is necessary for different platforms to analyze the similarities and differences of the results of other P2P loan cases operating in their respective financial environments. Secondly, this paper only uses the BP neural network model to predict user defaults and does not compare and analyze with other classification models, such as logistic regression, random forest. If several classification methods can be tried and compared, and analyzed, the ability of these models to reflect user default behavior can be improved. The third point is that the interpretability of the neural network model is poor, and it is treated as a black-box model. How to optimize a neural network is a difficult problem.

References

Hui, W., Greiner, M. E., & Aronson, J. E. (2009). People-to-People Lending: The Emerging E-Commerce Transformation of a Financial Market. *Springer Berlin Heidelberg*.

Freedman, S., & Jin, G. Z. (2008). Do social networks solve information problems for peer-to-peer lending? evidence from prosper.com. Working Papers.Granovetter, M. (1985). Economic Action and Social Structure: The Problem of Embeddedness. *American Journal of sociology*, 481-510.

Herzenstein, M., & Andrews, R. L.. (2008). The democratization of personal consumer loans? determinants of success in online peer-to-peer loan auctions. *Bulletin of the University of Delaware.*

Lin, M., Prabhala, N., & Viswanathan, S. (2011). Judging borrowers by the company they keep: friendship networks and information asymmetry in online peer-to-peer lending. S*ocial Science Electronic Publishing.*

SC Berger, & F Gleisner. (2014). Emergence of financial intermediaries in electronic markets: the case of online p2p lending. *Business Research.*

Milad Malekipirbazari and Vural Aksakalli. (2015). Risk assessment in social lending via random forests. *Expert Systems With Applications*, 42(10), pp. 4621-4631.

Emekter, R., Tu, Y., Jirasakuldech, B., & Lud, M. (2015). Evaluating Credit Risk and Loan Performance in Online Peer-to-Peer (p2p) Lending. *Applied Economics*, 47, 54-70.

Wang, T., & Li, J. (2019). An improved support vector machine and its application in p2p lending personal credit scoring. *IOP Conference Series Materials Science and*

Engineering.

Byanjankar, A., & Viljanen, M.9. (2020). Predicting expected profit in ongoing peer-to-peer loans with survival analysis-based profit scoring. *Intelligent Decision Technologies* 2019.

McCulloch, W.S., Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics* 5, 115–133.

Rosenblatt, Frank. (1957). The perceptron, a perceiving and recognizing automaton Project Para.

DE Rumelhart, Hinton, G. E., & Williams, R. J. (1986). Learning representations by back propagating errors. *Nature*, 323(6088), 533-536.

Guo, Y. (2020). Credit risk assessment of P2P lending platform towards big data based on BP neural network. *Journal of Visual Communication and Image Representation*, 71, 102730.

Nawi, N. M., Atomi, W. H., & Rehman, M. Z. (2013). The effect of data pre-processing on optimized training of artificial neural networks. *Procedia Technology,* 11, 32-39.

Sula, A., Spaho, E., Matsuo, K., et al. (2014). A new system for supporting children with autism spectrum disorder based on IoT and P2P technology. *Int. J. Space-Based Situated Comput.* 4(1), 55–64.

Zhang, S., Hu, Y., & Wang, C. (2019). Evaluation of borrower's credit of P2P loan based on adaptive particle swarm optimisation BP neural network. *International Journal of Computational Science and Engineering*, 19(2), 197-205.

Zhang, L., & Zhang, B. (1999). A geometrical representation of McCulloch-Pitts neural model and its applications. *IEEE Transactions on Neural Networks*, 10(4), 925-929.

Pacelli, V., & Azzollini, M. (2011). An artificial neural network approach for credit risk management. Journal of Intelligent Learning Systems and Applications, 3(2), 103-112.

Liaw, A. , & Wiener, M. . (2002). Classification and regression by randomforest. R News, 23(23).

Haykin, S. . (1998). Neural Networks: A Comprehensive Foundation (3rd Edition). Macmillan.

Berger, S. C. , & Gleisner, F. . (2009). Emergence of financial intermediaries in electronic markets: the case of online p2p lending. Social Science Electronic Publishing, 2(1), 39-65.

Appendices

Number	Feature name	Feature descriptions
1	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.
2	int_rate	Interest Rate on the loan
3	installment	The monthly payment owed by the borrower if the loan originates.
4	grade	LC assigned loan grade
5	out_prncp	Remaining outstanding principal for total amount funded
6	out_prncp_inv	Remaining outstanding principal for portion of total amount funded by investors
7	total_pymnt	Payments received to date for total amount funded
8	total_pymnt_inv	Payments received to date for portion of total amount funded by investors
9	total_rec_prncp	Principal received to date
10	total_rec_int	Interest received to date
11	recoveries	post charge off gross recovery
12	collection_recovery_fee	post charge off collection fee
13	last_pymnt_amnt	Last total payment amount received
14	last_fico_range_high	The upper boundary range the borrower's last FICO pulled belongs to.
15	last_fico_range_low	The lower boundary range the borrower's last FICO pulled belongs to.
16	60_months	The number of payments on the loan. Values are in months and can be either 36 or 60.
17	Month_last_pymnt_d	Last month payment was received
18	Year_last_pymnt_d	Last year payment was received
19	Month_last_credit	The most recent month LC pulled credit for this loan
20	Year_last_credit	The most recent year LC pulled credit for this loan

Table 5 selected feature descriptions

Curriculum Vitae

Name: Jingshi Luo

Post-secondary Education and Degrees:

Shenzhen University
Shenzhen, Guangdong, China
2015-2019 B.Econ in Finance

Shenzhen University
Shenzhen, Guangdong, China
2015-2019 B.Sc in Mathematics and Applied Mathematics

The University of Western Ontario
London, Ontario, Canada
2019-2021(currently) M.Sc. Applied Math

Honors and Awards:

National Scholarship
Ministry of National Education
2018

Outstanding Star of Liyuan
Shenzhen University
2018
Honorable Mention for The Mathematical Contest In Modelling
COMAP
2018

Related Work Experience:

Teaching Assistant
The University of Western Ontario
2019-2021