Using Machine Learning Models to Assess Users' Credit Default Risk

Yuxi Huang(✉)

School of Mechanical and Electrical Engineering, Guangzhou Huali College, Guangzhou, China
740783840@qq.com

Abstract. As far as the current social situation is concerned, more and more people choose credit cards as a way to pay for their daily expenses. When it comes to credit cards, we have to mention the credit default risk associated with them. We know that when a person's credit default risk is too high, his credit card may be frozen. It means that the credit default risk likes a kind of judgment for whether a person can apply for a credit card, and it is an important factor in continuing to use a credit card. In this paper, we analyze how to assess a user's credit default risk. First, we use principal component analysis (PCA) to extract several key factors for judging credit default risk, then we use BP neural network to evaluate and analyze the extracted key factors, and finally, we analyze the user's credit default risk through the results obtained.

Keywords: Credit Cards · Credit Default Risk · Principal Component Analysis Techniques · BP Neural Networks · Logistic Regression

1 Introduction

A credit card is a non-cash transaction method, which means people do not need to pay cash when using a credit card but keep a unified account, then settle and repay until the repayment date. In addition, the credit card also has a certain amount overdraft function, which is more and more popular with the public. However, because of the special consumption form of credit cards, the consumption of many people is far greater than their repayment ability, which will fail to repay in time when the repayment date comes, and it will produce credit default risk.

With the increasing use of credit cards, people are more and more concerned about the risk of credit default. There are many aspects to the assessment of credit default risk. In this paper, we set up three aspects to evaluate and analyze the credit default risk of the user: whether the loan is paid off, whether the installment is paid off, and the credit card balance and then we can obtain the user's credit default risk level, judge whether the user can hold a credit card again.

J. Yen et al. (Eds.): ICBIS 2023, AHCS 14, pp. 297–323, 2023.
https://doi.org/10.2991/978-94-6463-198-2_33

2 Methods

2.1 Introduction of PCA [1]

In this section, we introduce a statistical method called principal component analysis to analyze the obtained data. This method can transform a set of potentially correlated variables into a set of linearly uncorrelated variables through an orthogonal transformation. For example, we choose whether the loan is paid off and whether the installment is paid off as a set of variables, both of which have a certain correlation, they all relate to the balance of the credit card.

We then used projection tracking to assign weights to each of the principal component metrics from the main analysis. We should try our best to achieve that only one projection is searched for each projection pursuit to ensure that the extracted projections are non-Gaussian distribution and reduce errors. Projection pursuit is a commonly used method for processing and analyzing high-dimensional data. Its basic idea is to project data in high-dimensional space onto low-dimensional space, to study and analyze high-dimensional data. When the available data is available, the corresponding projection can be obtained to realize the separate extraction and analysis of the data.

2.2 Selection of Indicators

There are many aspects to judge the credit default risk level of a user, such as whether the loans of different credit types are repaid. To identify the reliable key factors in assessing credit default risk, we analyze it through three aspects: whether the loan of different credit types is paid off, whether the installment applied by the credit card is paid off, and the balance of the credit card.

2.3 Data Collection

To further investigate the relationship between credit default risk levels and the three selected indicators, we collected some data related to the three indicators for analysis. To make the data more intuitive to reflect the relationship with the risk level, we first preprocessed the data (Table 1).

Table 1. List of data obtained by preprocessing 1

REDIT_SUM_DEBT	AMT_CREDIT_SUM_LIMMIT	AMT_CREDIT_SUM_OVERDUE	CREDIT_TYPE	DAYS_CREDIT_UPDATE	AMT_ANNUITY	0_prop	MONTHS_BALANCE
0.0	67500.0	0.0	Credit Card	−183	0.0	0.296296	0
0.0	67500.0	0.0	Credit Card	−183	0.0	0.296296	−1
0.0	67500.0	0.0	Credit Card	−183	0.0	0.296296	−2
0.0	67500.0	0.0	Credit Card	−183	0.0	0.296296	−3
0.0	67500.0	0.0	Credit Card	−183	0.0	0.296296	−4
…	…	…	…	…	…	…	…
0.0	NaN	0.0	Consumer Card	−787	NaN	0.000000	−75
0.0	NaN	0.0	Consumer Card	−787	NaN	0.000000	−76
0.0	NaN	0.0	Consumer Card	−787	NaN	0.000000	−77
0.0	NaN	0.0	Consumer Card	−787	NaN	0.000000	−78
0.0	NaN	0.0	Consumer Card	−787	NaN	0.000000	−79

By preprocessing the data, it was found that it was difficult to find correlations from the bulk of the data, so we decided to take the next step, which is to use principal component analysis to analyze the preprocessed data (Table 2).

2.4 The Realization Process of PCA

As mentioned in the previous section, the preprocessed data has some kind of correlations, but it is difficult to find it only from the data, so it needs to be converted into a principal component, and the information on the correlation between the data can be obtained through principal component analysis.

Following are the steps regarding the principal component analysis implementation process:

Step 1: Construct a data matrix X.

$$\mathrm{X} = \begin{bmatrix} x_{11} & \cdots & x_{1n} \\ \vdots & \ddots & \vdots \\ x_{y1} & \cdots & x_{yn} \end{bmatrix}$$

Each row represents the imported indicator data about credit cards, and each column represents the indicator.

Step 2: Use the following formula to centralize the obtained data.

Calculated by the average formula: $\overline{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$ (n for the number of samples).

Calculated by the variance formula: $var(x) = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \overline{x})^2$ (n for the number of samples).

Step 3: Calculate the sample correlation index matrix.

$$\mathrm{Y} = \begin{bmatrix} y_{11} & \cdots & y_{1n} \\ \vdots & \ddots & \vdots \\ y_{y1} & \cdots & y_{yn} \end{bmatrix}$$

Calculated by the covariance formula: $cov(x, y) = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - x)(y_i - y)$

Step 4: Calculate the eigenvalues of the relevant index matrix and its eigenvalue vector.

Set eigenvalues: $z_1, z_2, z_3, \cdots, z_n$

Set the eigenvalue vector: $v_i = (v_{i1}, v_{i2}, v_{i3}, \cdots, v_{in})$

Step 5: Obtain the final principal components through principal component analysis. The principal component contribution rate of each index can be calculated by the following formula, that is, the proportion of the variance of the principal component of the index in the total variance

$$Cr = \frac{z_i}{\sum_{i=1}^{n} z_i}$$

Step 6: Calculate Cr by Python and get the following data [Fig. 1].

Table 2. List of data obtained by preprocessing 2

SK_ID_CURR	NUM_INSTALMENT_VERSION	NUM_INSTALMENT_NUMBER	DAYS_INSTALMENT	DAYS_ENTRY_PAYMENT	Paid_on_time	Max_paid_on_time
161674	1.0	6	−1180.0	−1187.0	1	1
151639	0.0	34	−2156.0	−2156.0	1	1
193053	2.0	1	−63.0	−63.0	1	1
199697	1.0	3	−2418.0	−2426.0	1	1
167756	1.0	2	−1383.0	−1366.0	0	1
…	…	…	…	…	…	…
428057	0.0	66	−1624.0	NaN	0	1
414406	0.0	47	−1539.0	NaN	0	1
402199	0.0	43	−7.0	NaN	0	1
409297	0.0	43	−1986.0	NaN	0	1
434321	1.0	19	−27.0	NaN	0	1

Table 3. List of obtained data

REDIT_SUM_DEBT	AMT_CREDIT_SUM_LIMMIT	AMT_CREDIT_SUM_OVERDUE	CREDIT_TYPE	DAYS_CREDIT_UPDATE	AMT_ANNUITY	0_prop	MONTHS_BALANCE
0.0	67500.0	0.0	Credit Card	−183	0.0	0.296296	0
0.0	67500.0	0.0	Credit Card	−183	0.0	0.296296	−1
0.0	67500.0	0.0	Credit Card	−183	0.0	0.296296	−2
0.0	67500.0	0.0	Credit Card	−183	0.0	0.296296	−3
0.0	67500.0	0.0	Credit Card	−183	0.0	0.296296	−4
…	…	…	…	…	…	…	…
0.0	NaN	0.0	Consumer Card	−787	NaN	0.000000	−75
0.0	NaN	0.0	Consumer Card	−787	NaN	0.000000	−76
0.0	NaN	0.0	Consumer Card	−787	NaN	0.000000	−77
0.0	NaN	0.0	Consumer Card	−787	NaN	0.000000	−78
0.0	NaN	0.0	Consumer Card	−787	NaN	0.000000	−79

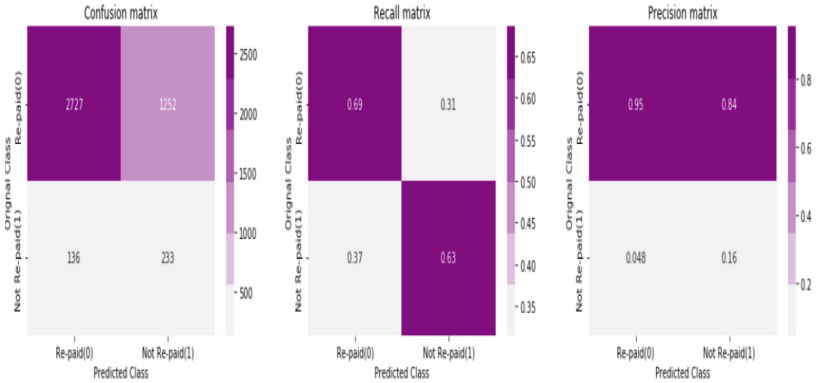

Fig. 1. Data obtained by calculating Cr

3 Implementation Steps

Step 1: Import the dataset.

Use the read_csv method in pandas to read the CSV file and we can obtain six sets of data: application_train, bureau, bureau_balance, POS_CASH_balance, credit_card_balance, and installments_payments. After reading the data in the CSV file, then output the shape of each dataset.

Step 2: Transform and clean the acquired data.

Some special data whose values are relatively deviated from other values in the data set, if they are still retained, will affect subsequent data processing and other operations, resulting in large deviations in the results of data processing. Therefore, such data needs to be converted and processed. Preprocessing operations such as cleaning.

The first is to remove constant values from the dataset and data that are not related to or cannot be explained by credit default risk. After removing the relevant values, start processing the missing data in the dataset. After processing all the data, calculate the proportion of each data to its total data dichotomy, convert the categorical variables to continuous variables, and finally create a new variable.

(1) the bureau_balance dataset

1. Import the bureau_balance data set, calculate the proportion of the value of 0 in the variable and judge whether a new variable needs to be created. If some data does not contain 0, the data needs to be converted; or use sum or c to get a non-zero scale, you can get a non-zero scale of and add three columns of status_sum, status_total, and 0_or_not to the data list.
2. Remove months_balance, status_sum, status_total, and 0_or_not as this data is not related to customer behavior.
3. bureau_balance
   and bureau through SK_ID_BUREAU to generate bureau_combine to get the following data list (Table 3).

(2) the bureau dataset

1. For each SK_ID_CURR, calculate the ratio of closed and active in its CREDIT_ACTIVE and output the number of closed, active, old, and bad debt in CREDIT_ACTIVE respectively.
2. Set a choice, if CREDIT_ACTIVE is lost or active, output clos_active_or_not = 1, otherwise clos_active_or_not = 0.
3. a: Regroup the data of CREDIT_ACTIVE and count the number of times it appears in SK_ID_CURR, then rename it to status_total and reassign it to an index starting from 0; b: Regroup the data of clos_active_or_not and sum it, then rename it to status_sum and reassign it to 0-based index; c: combine a, b, and on = ' SK_ID_CURR', set the formula c [' clos_active_or_not '] = c[' status_sum ']/ c[' status_total '], calculate the obtained data, and the final output called bureau_combine is the merger of c, the bureau_combine obtained in step (1) and on = ' SK_ID_CURR '.
4. Remove irrelevant variables in the output data list: first remove the three intermediate variables status_sum, status_total, and clos_active_or_not in the data list, then remove the constant variable and five uncorrelated variables. Finally, output the shape of bureau_combine (Table 4).
5. Analysis of the CREDIT_TYPE in bureau_combine and output the account of consumer credit, credit card, and different loans.
6. Set an option, if CREDIT_TYPE is consumer credit or credit card, output cons_cred_card_or_not = 1, otherwise cons_cred_card_or_not = 0.
7. Calculate the proportion of consumer credit and credit card; a: Regroup the data of CREDIT_TYPE and count the number of times it appears in SK_ID_CURR, then rename it to status_total and reassign as a 0-based index; b: regroup and sum the data cons_cred_card_or_not, then rename it status_sum and reassign it as a 0-based index; c: merge a and b with on = 'SK_ID_CURR ', set the formula c [' cons_cred_card_prop '] = c[' status_sum ']/ c[' status_total '] to calculate the obtained data, output bureau_combine (Table 7) in the combination of c and on = ' SK_ID_CURR ' obtained in step (1).
8. Regroup the data of bureau_combine and count the number of times it appears in SK_ID_CURR, calculate the average value of each value, and then reassign it to an index starting from 0, name bureau_num, and output.
9. Group the data in it according to the numerical features of SK_ID_CURR, and output the shape of bureau_num.
10. Filter the value of object_dtypes in bureau_combine for one - hot encoding and name it bureau_cat, but this will ignore the most important column of SK_ID_CURR, so we need to replace it; regroup and count the data of bureau_cat The number of times it appears in SK_ID_CURR, calculate the average value of each value, and then reassign it to an index starting from 0, replace the result obtained in the previous step with bureau_cat (Table 5), and output the shape of bureau_cat (Table 6).

Table 4. List of obtained data

SK_ID_BUEREAU	CREDIT_ACTIVE	AMT_CREDIT_MAX_OVERDUE	CNT_CREDIT_PROLONG	AMT_CREDIT_SUM	AMT_CREDIT_SUM_DEBT	AMT_CREDIT_SUM_LIMIT
5896630	Closed	NaN	0	112500.0	0.0	0.0
5896630	Closed	NaN	0	112500.0	0.0	0.0
5896630	Closed	NaN	0	112500.0	0.0	0.0
5896630	Closed	NaN	0	112500.0	0.0	0.0
5896630	Closed	NaN	0	112500.0	0.0	0.0
…	…	…	…	…	…	…
5126337	Closed	16618.5	0	450000.0	NaN	NaN
5126337	Closed	16618.5	0	450000.0	NaN	NaN
5126337	Closed	16618.5	0	450000.0	NaN	NaN
5126337	Closed	16618.5	0	450000.0	NaN	NaN
5126337	Closed	16618.5	0	450000.0	NaN	NaN

Table 5. List of Bureau_cat Data

CREDIT_ACTIVE_Active	CREDIT_ACTIVE_Bad debt	CREDIT_ACTIVE_Closed
0.186047	0.0	0.813953
0.181818	0.0	0.818182
0.380952	0.0	0.619048
0.500000	0.0	0.500000
0.000000	0.0	1.000000
…	…	…
0.146875	0.0	0.853125
0.620690	0.0	0.379310
0.470085	0.0	0.529915
0.000000	0.0	1.000000
0.317406	0.0	0.682594

CREDIT_TYPE_Consumer credit	CREDIT_TYPE_Credit card	…	CREDIT_TYPE_Real estate loan	CREDIT_TYPE_Unknown type of loan	STATUS_0
1.000000	0.000000	…	0.0	0.0	0.180233
0.472727	0.527273	…	0.0	0.0	0.409091
0.380952	0.619048	…	0.0	0.0	0.666667
0.500000	0.000000	…	0.0	0.0	0.277778
0.473913	0.000000	…	0.0	0.0	0.343478
…	…	…	…	…	…
0.887500	0.037500	…	0.0	0.0	0.206250
0.678161	0.321839	…	0.0	0.0	0.137931
0.735043	0.264957	…	0.0	0.0	0.401709
1.000000	0.000000	…	0.0	0.0	0.216216
0.843003	0.156997	…	0.0	0.0	0.215017

(3) the installment_payment dataset

1. AMT_INSTALMENT and AMT_PAYMENT into the same variable to show if a credit card user has a bill on time for installment payments. Set a new data named pay diff, pay diff = AMT_PAYMENT – AMT_INSTALMENT, when pay diff is greater than or equal to 0, paid_on_time = 1, otherwise paid_on_time = 0.
2. Rename the data of paid_on_time to the percent of paid on time and output; remove the n a value in pay diff and calculate its maximum value, rename it to maximum paydiff and output.
3. Get a new installment data list named installments_payments_new; calculate the maximum value of aid_on_time; combine max_paid_on_time and installments_payments_new to get installments_payments_new (Table 9) data list.

(4) Preprocessing the credit_card_balance dataset

1. Set the na value in the variables to 0.
2. Calculate the average value of each data in the credit_card_balance data set, and then regroup it and assign to an index starting from 0, named ccb_num data list (Table 10).
3. Filter the value of object_dtypes in credit_card_balance for one-hot encoding and name it ccb_cat, but this will ignore the most important column of SK_ID_CURR, so we need to replace it; regroup the data of ccb_cat (Table 11) and count it in SK_ID_CURR the number of occurrences, calculate the average of each value in it, and then reassign it to a 0-based index.

(5) the POS_CASH_balance dataset (Table 12)

1 Convert the na value in the variables to 0, and then set a new dummy variable according to the following conditions: if the original variable is 0, keep its output equal to 0, otherwise output the variable equal to 1.

2 Calculate the mean of this new dummy variable data, then regroup it and combine with the data list of POS_CASH_balance.

3 Change the na value to 0, then take the average value in SK_ID_CURR, and finally obtain the processed POS_CASH_balance (Table 13) data list.

Table 6. List of Bureau_num data

Clos_active_prop	SK_ID_BUREAU	AMT_CREDIT_MAX_OVERDUE	CNT_CREDIT_PROLONG	AMT_CREDIT_SUM	AMT_CREDIT_SUM_DEBT	AMT_CREDIT_SUM_LIMIT
1.0	5.896632e + 06	NaN	0.000000	161516.250000	23893.770349	0.000000
1.0	6.1152350e + 06	1312.010357	0.000000	111388.838727	70223.142857	3198.856500
1.0	6.735201e + 06	0.000000	0.000000	175903.714286	132923.785714	0.000000
1.0	5.576631e + 06	NaN	0.000000	495000.000000	174003.750000	0.00000
1.0	5.922081e + 06	19305.000000	0.000000	532530.923478	0.000000	NaN
…	…	…	…	…	…	…
1.0	6.353728e + 06	4612.367773	0.031250	367670.334375	417788.571429	0.000000
1.0	6.817237e + 06	0.000000	0.000000	971625.227586	676051.789655	18753.043448
1.0	5.864376e + 06	NaN	0.000000	914615.384615	378608.923077	0.000000
1.0	6.669849e + 06	NaN	0.000000	45000.000000	0.000000	NaN
1.0	5.126332e + 06	13629.984961	0.112628	342536.498294	179242.109531	0.000000

Table 7. List of Bureau_combine data

SK_ID_BUREAU	AMT_CREDIT_MAX_OVERDUE	CNT_CREDIT_PROLONG	AMT_CREDIT_SUM	AMT_CREDIT_SUM_DEBT	AMT_CREDIT_SUM_LIMIT
5.896632e + 06	NaN	0.000000	161516.250000	23893.770349	0.000000
6.1152350e + 06	1312.010357	0.000000	111388.838727	70223.142857	3198.856500
6.735201e + 06	0.000000	0.000000	175903.714286	132923.785714	0.000000
5.576631e + 06	NaN	0.000000	495000.000000	174003.750000	0.00000
5.922081e + 06	19305.000000	0.000000	532530.923478	0.000000	NaN
…	…	…	…	…	…
6.353728e + 06	4612.367773	0.031250	367670.334375	417788.571429	0.000000
6.817237e + 06	0.000000	0.000000	971625.227586	676051.789655	18753.043448
5.864376e + 06	NaN	0.000000	914615.384615	378608.923077	0.000000
6.669849e + 06	NaN	0.000000	45000.000000	0.000000	NaN
5.126332e + 06	13629.984961	0.112628	342536.498294	179242.109531	0.000000

Table 8. List of Bureau_combine data (2)

AMT_ANNUITY	0_prop	MONTHS_BALANCE
1236.244186	0.180233	−16.279070
0.000000	0.409091	−24554545
608.785714	0.666667	−4.333333
NaN	0.277778	−46.000000
0.000000	0.343478	−29.373913
…	…	…
4837.617904	0.206250	−27.281250
170124.655862	0.137931	−14.149425
58369.500000	0.401709	−14.282051
0.000000	0.216216	−18.000000
1059.695565	0.215017	−26.392491

4 Logistic Regression

4.1 Basic Concepts of Logistic Regression [2]

For some data with a linear relationship, we generally use a straight line to fit these data, the fitting process is called regression. The main idea of logistic regression is to establish a regression formula for the boundary line of the classification according to the obtained data: $\pi(x) = \frac{1}{1+e^{-w^T x}}$.

4.2 Steps to Perform Logistic Regression

1. Split the combined dataset into 70% training set and 30% training set
2. Convert the input data list to matrix format and divide it into a numerical list and a categorical list
3. Characterize digital data and classified data
4. Define the function cv_plot and set two parameters alpha and cv_auc; assign the return value obtained by the plt. Subplots() function to the two variables fig and ax respectively
5. In the ax variable, plot the logarithm of the alpha parameter which is based in 10
6. Set up a for loop, round the value of the floating point number in the cv_auc parameter, put the entire tuple into the loop, and add the subscript i to the alpha parameter and the cv_auc parameter in the ax variable. Set the grid lines and coordinates of the chart, name the chart Cross Validation Error for each alpha, name the x-axis of the chart Alpha i's, name the y-axis of the chart as Error measures, and finally display the chart.

Table 9. List of installment_payment data

	SK_ID_PREV	SK_ID_CURR	NUM_INSTALMENT_VERSION	NUM_INSTALMENT_NUMBER	DAYS_INSTALMENT	DAYS_ENTRY_PAYMENT	Paid_on_time
0	1054186	161674	1.0	64	−1180.0	−1187.0	1
1	1330831	151639	0.0	34	−2156.0	−2156.0	1
2	2085231	193053	2.0	1	−63.0	−63.0	1
3	2452527	199697	1.0	3	−2418.0	−2426.0	1
4	2714724	167756	1.0	2	−1383.0	−1366.0	0
…	…	…	…	…	…	…	…
13605396	2186857	428057	0.0	66	−1624.0	NaN	0
13605397	1310347	414406	0.0	47	−1539.0	NaN	0
13605398	1308766	402199	0.0	43	−7.0	NaN	0
13605399	1062206	409297	0.0	43	−1986.0	NaN	0
13605400	2448869	434321	1.0	19	−27.0	NaN	0

Table 10. List of ccb_num data

AMT_DRAWIMGS_POS_CURRENT	AMT_INST_MIN_REGULARITY	…	AMT_RECIVABLE	AMT_TOTAL_RECIVABLE	CNT_DRAWIMGS_ATM_CURRENT
877.5	1700.325	…	0.000	0.000	0.0
0.0	2250.000	…	64875.555	64875.555	1.0
0.0	2250.000	…	31460.085	31460.085	0.0
0.0	11795.760	…	233048.970	233048.970	1.0
11547.0	22924.890	…	453919.455	453919.455	0.0
…	…	…	…	…	…
0.0	0.000	…	0.000	0.000	NaN
0.0	0.000	…	0.000	0.000	0.0
0.0	2250.000	…	273093.975	273093.975	2.0
0.0	0.000	…	0.000	0.000	NaN
0.0	0.000	…	0.000	0.000	0.0

Table 11. List of ccb_cat data

NAME_CONTRACT_STATUS_Active	NAME_CONTRACT_STATUS_Approved	NAME_CONTRACT_STATUS_Completed	NAME_CONTRACT_STATUS_Demand
1.000000	0.0	0.000000	0.0
1.000000	0.0	0.000000	0.0
1.000000	0.0	0.000000	0.0
0.411765	0.0	0.588235	0.0
1.000000	0.0	0.000000	0.0
…	…	…	…
0.878049	0.0	0.121951	0.0
1.000000	0.0	0.000000	0.0
1.000000	0.0	0.000000	0.0
1.000000	0.0	0.000000	0.0
1.000000	0.0	0.000000	0.0

Table 12. List of obtained data

	SK_ID_PREV	SK_ID_CURR	MONTHS_BALANCE	CNT_INSTALMENT	CNT_INSTALMENT_FUTURE	NAME_CONTRACT_STATUS	SK_DPD	SK_DPD_DEF
0	1803195	182943	−31	48.0	45.0	Active	0	0
1	1715348	367990	−33	36.0	35.0	Active	0	0
2	1784872	397406	−32	12.0	9.0	Active	0	0
3	1903291	269225	−35	48.0	42.0	Active	0	0
4	2341044	334279	−35	36.0	35.0	Active	0	0
…	…	…	…	…	…	…	…	…
100001353141565	2448283	226558	−20	6.0	0.0	Active	1	0
100001354	1717234	141565	−19	12.0	0.0	Active	1	0
100001355	1283126	315695	−21	10.0	0.0	Active	1	0
100001356	1082516	450255	−22	12.0	0.0	Active	1	0
100001357	1259607	174278	−52	16.0	0.0	Completed	0	0

Table 13. List of POS_CASH_balance data

	SK_ID_PREV	SK_ID_CURR	MONTHS_BALANCE	CNT_INSTALMENT	CNT_INSTALMENT_FUTURE	NAME_CONTRACT_STATUS	SK_DPD	SK_DPD_DEF
0	1803195	182943	−31	48.0	45.0	Active	0	0
1	1715348	367990	−33	36.0	35.0	Active	0	0
2	1784872	397406	−32	12.0	9.0	Active	0	0
3	1903291	269225	−35	48.0	42.0	Active	0	0
4	2341044	334279	−35	36.0	35.0	Active	0	0
…	…	…	…	…	…	…	…	…
100001353141565	2448283	226558	−20	6.0	0.0	Active	1	0
100001354	1717234	141565	−19	12.0	0.0	Active	1	0
100001355	1283126	315695	−21	10.0	0.0	Active	1	0
100001356	1082516	450255	−22	12.0	0.0	Active	1	0
100001357	1259607	174278	−52	16.0	0.0	Completed	0	0

4.3 Select Features According to the WOE Table

List the table mylist (Table 14) according to the required data, and form a table named x _taset_final and output it.

4.4 Calculating Feature Importance with Boosting Method

(1) Solve the naming problem of train_features, valid_features, and X_train_final
(2) Introduce pickle and lightgbm to convert the obtained data list into a histogram
(3) Introduce plot, add the obtained histogram [Fig. 2] to the coordinates and label and output

4.5 Fit the Logistic Regression Model According to the Smf Package

Introduce the statsmodels [5] package to process the x_train_in2 dataset, and output the list (Table 15) sum of logistic regression.

5 Model Evaluation [3]

5.1 Definition of the Confusion Matrix

A confusion matrix is one of the standard formats for evaluating the accuracy of classification models. Each column of the confusion matrix represents the predicted class, and each row represents the true attribution class of the data [Fig. 3].

True Positive: The sample category is a positive class, and the model recognizes it as a positive class.

False Positive: The sample category is a positive class, and the model recognizes it as a negative class.

True Negative: The sample category is a negative class, and the model recognizes it as a positive class.

False Negative: The sample category is a negative class, and the model recognizes it as a negative class.

5.2 Results and Results Analysis of Performing Confusion Matrix

1. In the Precision Matrix [Fig. 4], we can see that nearly 90% of the data is Re_paid, so the conclusion will be that the two cases of False are higher than the two cases of True.
2. When running the confusion matrix this time, there are two sets of data that, by definition, deviate from actual life situations. The definition of False Positive is that the customer cannot repay the loan, but the model still believes that it can be paid; the definition of False Negative is: the customer can pay, but the model thinks that it cannot pay.

Table 14. Data list of mylist

LIVINGAPARTMENT_MEDI	COMMONAREA_MODE	DAYS_EMPLOYED	FLAG_EMP_PHONE	…	NAME_HOUSING_TYPE	OWN_CAR_AGE	AMT_INCONE_TOTAL
−0.151474	−0.168658	−0.454628	0.468667	…	NaN	−0.146282	−0.124284
−0.151474	−0.168658	−0.462952	0.468667	…	NaN	−0.287706	−0.042241
−1.438169	−0.601949	−0.462542	0.468667	…	NaN	−0.146282	−0.206328
0.292278	−0.173419	−0.454019	0.468667	…	NaN	−0.146282	−0.255554
−0.151474	−0.168658	−0.462499	0.468667	…	NaN	−0.146282	−0.370415
…	…	…	…	…	…	…	…
−1.024124	2.700112	−0.465848	0.468667	…	NaN	0.419415	0.121846
−0.183038	0.686022	−0470272	0.468667	…	NaN	−0.146282	0.450021
−0.151474	−0.168658	−0.456766	0.468667	…	NaN	1.692232	−0.124284
−0.151474	−0.168658	−0.459017	0.468667	…	NaN	−0.146282	−0.370415
−0.151474	−0.168658	−0.455286	0.468667	…	NaN	−0.146282	0.367977

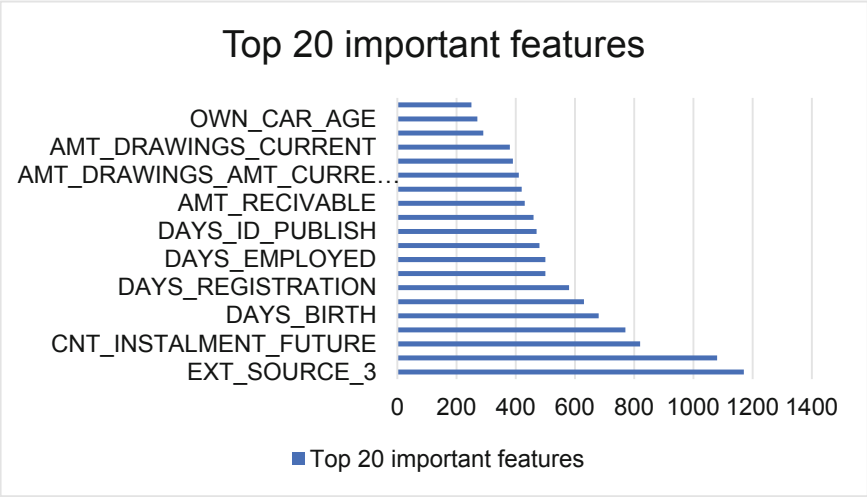

Fig. 2. Histograms Obtained

Table 15. List Sums

Dep.Variable: target	**No. Observations:** 20286
Model: GLM	**Df Residuals:** 20212
Model Family: Binomial	**Df Model:** 73
Ling Funtion: Logit	**Scale:** 1.0000
Method: IRLS	**Log-Likelihood:** -5312.6
Date: Thu, 10 Feb 2022	**Deviance:** 100625.
Time: 21:14:16	**Person chi2:** 1.95e + 04
No. Iterations: 22	**Pseudo R-squ. (CS):** 0.05482
Convariance Type: nonrobust	

5.3 Model Diagram for the Confusion Matrix

Plot the ROC [Fig. 5] curve that can show the performance of the model.

In the ROC curve, the closer the drawn model is to the rated 45° diagonal line, the lower the accuracy of the drawn image. The AUV value of the ROC curve obtained this time can be known from the annotations in the figure is 0.709, which means that the image obtained this time is more accurate.

5.4 Computing Performance Indicators

Using the test dataset, evaluate the performance of the logistic regression model by calculating the precision, recall, F_1 score, precision, and total misclassification rate [Fig. 6].

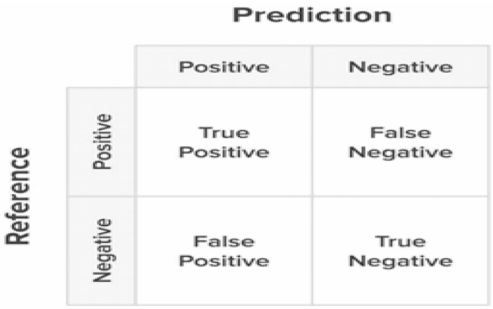

Fig. 3. A basic form of the confusion matrix

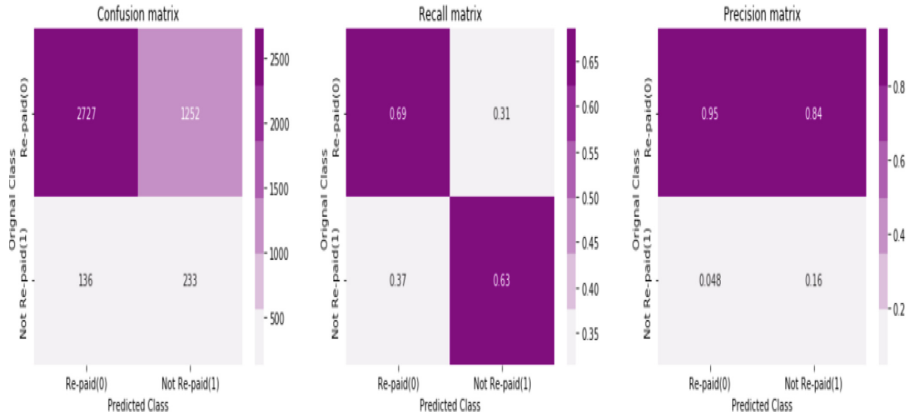

Fig. 4. Results of performing confusion matrix

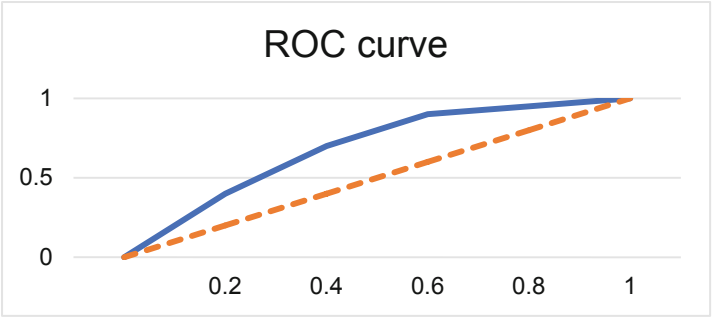

Fig. 5. ROC Curve for model diagram

	precision	recall	f1-score	support
0	0.95	0.69	0.80	3979
1	0.16	0.63	0.25	369
accuracy			0.68	4348
macro avg	0.55	0.66	0.52	4348
weighted avg	0.88	0.68	0.75	4348

The total misclassification rate 31.92% :

Fig. 6. Calculated precision, recall, F_1 score, precision, and total misclassification

6 Random Forest [4]

6.1 Definitions of Random Forest

Random forest mainly refers to the use of multiple decision trees to train, classify and predict the sample data obtained. While classifying the data, it can also evaluate the role of each variable in the classification.

6.2 Executing Random Forests

(1) By changing the number of trees and the maximum depth of the tree, the accuracy of AUC in different situations is obtained [Fig. 7].

(2) By performing step (1) multiple times, the optimal number of trees is 200 and the maximum depth of the optimal tree is 7. After setting the number of trees and the maximum depth of the tree to the optimal values, show the training of AUC value, the verification of AUC value, and the test of AUC value in the figure below [Fig. 8].

(3) Put the obtained data into the confusion matrix to obtain the evaluation result of the confusion matrix [Fig. 9].

```
For n_estimators 200, max_depth 7 cross validation AUC score 0.7120111563971722
For n_estimators 200, max_depth 10 cross validation AUC score 0.7064391042101467
For n_estimators 500, max_depth 7 cross validation AUC score 0.7119107652130204
For n_estimators 500, max_depth 10 cross validation AUC score 0.7067901318880645
For n_estimators 1000, max_depth 7 cross validation AUC score 0.7113493941016423
For n_estimators 1000, max_depth 10 cross validation AUC score 0.7066487647103817
For n_estimators 2000, max_depth 7 cross validation AUC score 0.711550859403171
For n_estimators 2000, max_depth 10 cross validation AUC score 0.7072292579520745
```

Fig. 7. Accuracy of AUC under the number of different trees and the maximum depth of the tree

```
The optimal values are: n_estimators 200, max_depth 7
For best n_estimators 200 best max_depth 7, The Train AUC score is 0.8398387088279861
For best n_estimators 200 best max_depth 7, The Validation AUC score is 0.7120111563971722
For best n_estimators 200 best max_depth 7, The Test AUC score is 0.7093977800798366
The test AUC score is : 0.7093977800798366
The percentage of misclassified points 08.49% :
```

Fig. 8. The value of AUC when the number of trees is 200 and the maximum depth of the tree is 7

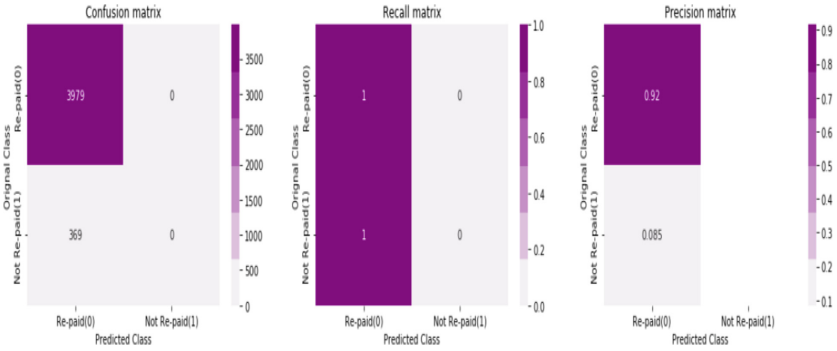

Fig. 9. Results of the confusion matrix

6.3 Model Illustration for Random Forest

Plot the ROC curve [Fig. 10] that can show the performance of the model.

Precision, recall, F_1 score, and support based on the evaluation results of the confusion matrix [Fig. 11].

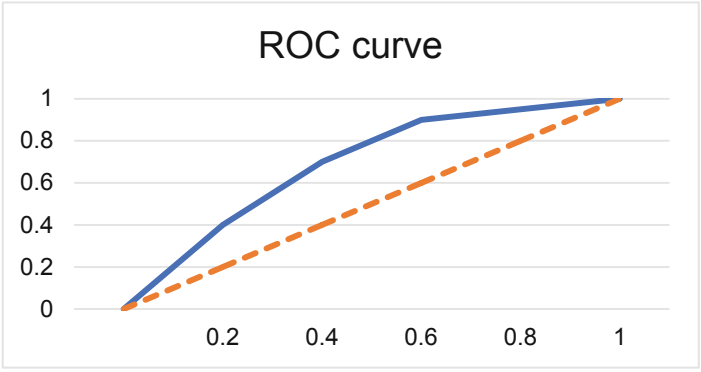

Fig. 10. ROC Curve for model illustration

	precision	recall	f1-score	support
0	0.92	1.00	0.96	3979
1	0.00	0.00	0.00	369
accuracy			0.92	4348
macro avg	0.46	0.50	0.48	4348
weighted avg	0.84	0.92	0.87	4348

The total misclassification rate 08.49% :

Fig. 11. Calculated precision, recall, F_1 score, precision, and total misclassification

7 Conclusion [5–14]

With the development of science and technology, machine learning and deep learning models have been widely used in different industries, and credit default risk analysis is no exception. Machine learning models play an important role in analyzing the credit default risk of users. In the process of data preprocessing, machine learning can help us eliminate data irrelevant to risk analysis, constant quantities, and data that deviate too much from other data. In addition, we can also use the logistic regression method in the machine learning model to fit the preprocessed data set model to obtain the fitting curve corresponding to each data set; using the confusion matrix method, by calculating the accuracy, recall rate, F _1 score, accuracy rate and total misclassification rate to evaluate the performance of the logistic regression model and determine the accuracy rate of the logistic regression model; use the random forest method to train, classify and predict the acquired data. Make a classification and assess the role each data plays in that classification. The machine learning model evaluates the user's credit default risk based on the user's loan amount, whether the loan and installment are repaid on schedule and whether the installment is repaid on schedule, and obtains the user's credit default risk level. Compared with the previous method of assessing risks by paper questionnaires, the risk level assessed by the machine learning model is more objective and more reliable. Therefore, the assessment of users' credit default risk based on the machine learning model can be implemented in the market.

References

1. Rasmus Bro, Age K. Smilde. Principal component analysis. Royal Society of Chemistry. 2014, 10.1039
2. David W. Hosmer, Jr., Stanley Lemeshow, Rodney X. Sturdivant. Applied Logistic Regression. 2013
3. Robert Susmaga. Confusion Matrix Visualization. Poznan University of Technology. 60–965
4. Leo Breiman. Random Forests. Machine Learning. 2014, 45, 5-32
5. Patrick J Beck. Summer 2021 CS 687 Capstone Project Progress Report Predicting Loan Default Likelihood Using Machine Learning. City University of Seattle. 2021
6. Somayeh Moradi, Farimah Mokhatab Rafiei. A dynamic credit risk assessment model with data mining techniques: evidence from Iranian banks. Financial Innovation. (2019)5: 15

7. Hussain Ali Bekhet, Shorouq Fathi Kamel Eletter, Credit risk assessment model for Jordanian commercial banks: Neural scoring approach. Review of Development Finance.4(2014)20-28
8. Nasser Mohammdi, Maryam Zangeneh, Customer Credit Risk Assessment using Artificial Neural Networks. MECS.2016.10.5815
9. Aida Krichene Abdelmoula. Bank credit risk analysis with k-nearest-neighbor classifier: Case of Tunisian banks. Accounting and Management Information Systems, 2015, Vol.14, No.1,79-106
10. Mohsen Nazari, Mojtaba Alidadi, Measuring Credit Risk of Bank Customers Using Artificial Neural Networks, Journal of Management Research. 2013, 1941–899X
11. Peter Martey Addo, Dominique Guegan, Bertrand Hassani. Credit Risk Analysis Using Machine and Deep Learning Models. MDPI. 2018
12. Ozdemir Ozlem, Boran Levent. An Empirical Investigation on Consumer Credit Default Risk. ECONSTOR. 2004/20
13. Yiping Huang, Longmei Zhang, Zhenhua Li, Hna Qiu, Tao Sun, Xue Wang. Fintech Credit Risk Assessment for SMEs: Evidence from China. IMF Working Paper. 2020. WP/20/193
14. Fisnik Doko, Slobodan Kalajdziski, Igor Mishkovski. Credit Risk Model Based on Central Bank Credit Registry Data. Risk and Financial Management. 2021. 14:138