Research on Personal Loan Default Risk Assessment Based on Machine Learning

Guangsen Liu

Aberdeen Institute of Data Science and Artificial Intelligence, South China Normal University, Foshan, China

Abstract. In the present era of rapid development of the Internet and big data, the scale of personal loans and the complexity of personal credit data are growing rapidly. Accurately assessing personal credit rating and personal loan default risk has become an important topic in the financial field. This paper analyzes the current research status of other scholars on machine learning in personal loan default risk assessment in recent years, and selects Logistic Regression, Support Vector Machine, Naïve Bayes and Deep Neural Networks as research model. Meanwhile, this paper selects the Kaggle website data of a bank and credit information bureau in India, preprocesses the dataset and applies it to the training and testing of the models, and finally derives the performance results of the four models. The results of the study show that the machine learning models have better accuracy and higher efficiency in analyzing personal credit data and assessing the risk of personal loan default. Among them, the Deep Neural Network has the best overall performance compared to the other three machine learning models. The research in this paper has certain research significance for the research of machine learning in personal loan default risk assessment.

1 Introduction

In the field of finance, the Personal Loan means borrowing of money from an organization such as Bank, Credit Cooperatives, etc. Banks grant loans to individuals to help them alleviate their financial difficulties and promote socio-economic development, and also allow the banks to earn interest from borrowers.

However, personal loans create an important financial problem, that is, loan default. Loan default is when a lender fails to repay a loan within a specified time period. Loan default can be harmful: For lending institutions, this can result in a loss of funds. If the amount of overdue loans is large, it can even result in a breakdown of the lending institution's financial chain. If the amount of overdue loans is large, it can even lead to a break in the lending institutions' financial chain. Therefore, predicting the risk of personal loan is essential for banks, which affects their decisions on whether or not to allow lending loan to borrowers [1].

The traditional method of evaluating the risk of default on personal loan is an expert's evaluation based on experience [2]. However, as the development of the Internet and big data,

Corresponding author: 20213801012@m.scnu.edu.cn

online loans are gradually emerging and increasing year by year. Also, the threshold for individuals to apply for loans has been lowered compared to the past. At the same time, there has been a quantitative growth in the amount of personal credit data by leaps and bounds, and the types of data become more diversified [3]. At this point it is difficult to adapt the use of expert experience in evaluating personal credit is no longer sufficient for this situation. Therefore, in the context of combining the lending business with the Internet, how to assess the default risk of individual loans from multi-dimensional and ever-changing credit data has become a challenge for the lending industry.

In the face of the current situation where the number of personal credit data has increased, the types have become complex, and the pressure on lending institutions to evaluate personal credit is growing, constructing machine learning models to predict personal credit rating as well as the risk of loan defaults will effectively improve the efficiency of evaluation [4]. Machine learning algorithms can accurately and efficiently assess the risk of personal loan defaults by calculating the relationship between individual features of personal credit data, thereby reducing the likelihood of personal loan defaults and maintaining financial market stability. In addition, using machine learning algorithms to study and predict the risk of individual loan defaults in the lending business helps to promote the development of individual credit assessment models [5].

For the personal loan default risk assessment model, many scholars have conducted a lot of in-depth research on it in recent years. In the lending industry's big data, individual credit data often suffers from the disadvantage of imbalance, that is, far fewer defaults occur than no defaults. It leads to a tendency for models to bias towards non-defaulting data when using machine learning for classification [6]. Luo, Yan, Tian et al. proposed an unsupervised, kernel-free Quadratic Surface Support Vector Machine (QSSVM) model. It can avoid the selection of kernel and related kernel parameters, thus improving the accuracy of classification [7]. For the situation of diverse forms of loans where a single prediction model is difficult to adapt to all situations, Yang, Qiao, Huang et al. presented a new automated credit scoring strategy (ACSS). This method shows significant performance improvement in automatically constructing scoring models for several different credit datasets [8]. On this basis, Jin, Cai et al. proposed multi-SVM-KNN, which is optimized on the basis of a Support Vector Machine (SVM), combined with feature extraction of Principal Component Analysis (PCA), and well solve the problem of evaluating personal credit in an automated way [9].

The rest of the paper is structured as follows: In section 2, this article describes the research methodology, including the data sources and algorithms. In Section 3, this article describes the research process. Then, the article presents and discusses the results of study in section 4. Finally, in section 5, the article summarizes the conclusions.

2 Research Methodology

2.1 Algorithm Introduction

2.1.1 Logistic Regression

Logistic regression is a type of linear regression. It is similar to other linear regression models that predict values after inputting a given data set. What is different is that logistic regression produces results as probability values, so the logistic regression results take values between 0 and 1. Logistic regression predicts which class the data belongs to base on the features and determines the confidence of the result based on the magnitude of the resulting probability value.

For binary classification tasks, logistic regression often uses a Sigmoid activation function to convert the results into probabilities. In contrast, in multi-classification tasks, logistic regression uses the Softmax activation function. The formula for the two is as follows:

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

$$Softmax(x_i) = \frac{e^{x_i}}{\sum_{j=1}^{N} e^{x_j}} \tag{2}$$

Here, the Softmax activation function is equivalent to the sigmoid function when N=2.

The loan default risk levels in the dataset of this paper are categorized into 4 classes, so this paper uses Softmax function to calculate the probability.

2.1.2 Support Vector Machine

Support Vector Machines, or SVMs, are another machine learning algorithm that can also be used to solve classification tasks. Its principle is that, when solving the task of classification of linearly divisible data, a number of hyperplanes can be found to separate the partition of two classes of data in the feature space. The expression of the hyperplane in the feature space is:

$$W^T X + B = 0 \tag{3}$$

where W and B are the normal vector and intercept of the hyperplane.

When the distance from any data point (X_i, Y_i) in the feature space to the hyperplane is greater than 1, the dataset can be considered linearly divisible, at this point there are:

$$W^T X_i + b \geq 1 \tag{4}$$

$$W^T X_i + b \leq -1 \tag{5}$$

To find an optimal hyperplane in the feature space, two parallel vectors, called interval boundaries, are created separately and moved in two opposite directions until they just touch a data point. The two data points that are touched are the support vectors. In this case an interval exists between the two interval boundaries parallel to the hyperplane. The support vector machine works by finding the case where the interval is broadest, when the hyperplane is in the middle of the two interval boundaries.

SVMs can also solve nonlinear regression problems by simply introducing kernel methods and kernel functions. The role of the kernel function is to make the data linearly differentiable in a high-dimensional space by mapping the data to a high-dimensional space. Thus, SVM is a powerful machine learning algorithm with excellent generalization capabilities. However, it also has the drawbacks of high computational cost and long training time. For the same dataset, SVM tends to take longer training time than other types of machine learning algorithms.

2.1.3 Naïve Bayes

Based on Bayes' theorem, the Naïve Bayes algorithm predicts the probability of a feature belonging to a certain type by calculating the conditional probability. Unlike the Bayes' Theorem, Naïve Bayes assumes that all features are independent to each other. Although this assumption ignores the correlation between individual features, it can make the algorithm simpler and stable in performance.

The Bayes Rule is:

$$P(Y|X) = \frac{P(X|Y) * P(Y)}{P(X)} \tag{6}$$

P(X), P(Y), P(X|Y), P(Y|X) are evidence, prior probability, likelihood, and posterior probability, respectively.

Naïve Bayes, under the assumption that the individual features are independent of each other, can be expanded into a Bayesian formulation:

$$P(X|Y = y) = \prod_{i=1}^{d} P(x_i|Y = y) \tag{7}$$

The posterior probability is then obtained:

$$P_{post} = P(Y|X) = \frac{P(Y)\prod_{i=1}^{d} P(x_i|Y)}{P(X)} \tag{8}$$

When comparing features belonging to different classes, since P(X) is the same, only the numerator part of the above equation needs to be compared in the comparison.

2.1.4 Deep Neural Networks

Deep Neural Network (DNN) is an algorithmic model that simulates the neuronal connectivity of the human brain, and belongs to an important branch of the field of machine learning. DNN simulates multiple layers of neurons, including input, hidden and output layers. Among them, there may be one or multiple hidden layers. Each layer is fully connected to the neurons in each layer and passes data through a nonlinear transformation. As data is gradually passed backward, the neural network abstracts the data and extracts features from it.

Before training, DNN initializes the weights and bias sizes of its individual layers by setting them randomly. During the training process, the data first experiences a pass from the input layer to the output layer, called Forward propagation. At this point, DNN produces a prediction. Then, calculate the error between the predicted value and the actual value, which is the loss of the DNN. Then the neural network passes the error backward and calculates the error contribution of each layer of neurons. At the same time, the neural network updates the weights and biases using gradient descent algorithm. The process of passing errors from the output layer to the input layer while updating the weights and biases of the neurons is called Backward propagation.

In classification tasks, DNN shows strong feature learning ability as well as high accuracy. Meanwhile, DNN has good flexibility and scalability because of its flexible network structure. However, DNN also has the disadvantages of being computationally intensive and data-dependent. The training of deep neural networks requires a sufficient amount of data. The lack of amount of data can easily lead to a decrease in the generalization ability of DNN [10].

2.2 Model Optimization Method

2.2.1 Grid Search

When building the model, various hyperparameters need to be set, such as learning rate, number of training rounds, batch size, etc. When modeling for the first time, it is common to choose to use the estimated hyperparameters. These initial hyperparameters may not be optimal for the dataset, so in order to train the best model, the model needs to be continually adjusted at the subsequent stage to try different hyperparameters.

By trying various combinations of hyperparameters, finding the optimal hyperparameters. The grid search creates a hyperparameter grid, where each element of the dictionary is a

hyperparameter as well as a value to be tested. When conducting a grid search, this dictionary is traversed through a loop, different models are created and trained, and finally the best performing model is selected. At this point the hyperparameters corresponding to this model are the optimal hyperparameters. After the preliminary identification of the optimal hyperparameters, the range of hyperparameters on the hyperparameter grid is adjusted and narrowed to find more accurate optimal hyperparameters. Finally, when the identified hyperparameter combination is stable, it can be considered that the optimal hyperparameter combination is identified.

Grid searches have the advantage of simple operation and easy implementation, and find out the optimal combination of hyperparameters by enumerative search. However, this also leads to the disadvantage that it is computationally large and time consuming. Especially when the mesh is large and the model is complex, the time consumption grows tremendously.

This paper conducts a mesh search for the above models separately. The hyperparametric meshes of the above models are shown in Table 1:

Table 1. Hyper-parameters and results of grid search.

Model type	Hyper-parameter	Data range	Optimal parameter
Logistic Regression	Inverse of regularization strength	[0.001, 0.01, 0.1, 1, 10, 100]	1
	Regularization	[L1, L2]	L1
	Optimization algorithm	['liblinear', 'saga']	‘saga’
Support Vector Machine	Inverse of regularization strength	[0.1, 1, 10, 100]	1
	Kernel function	['linear', 'rbf']	'linear'
	Kernel coefficient	['scale', 'auto']	'scale'
Naïve Bayes	-	-	-
Deep Neural Network	Learning rate	[0.01, 0.001, 0.0001]	0.001
	Epoch	[10, 20, 30]	30
	Batch size	[32, 64, 128]	64
	Dropout rate	[0, 0.1, 0.2, 0.5]	0.1

2.2.2 Regularization

Overfitting is a common problem in machine learning. The use of regularization methods, which add a penalty term to the computation of the model's loss function, can be used to limit the complexity of the model and thus alleviate the overfitting problem.

The expression is:

$$R(f) = L(f) + \lambda * \Omega(f) \tag{9}$$

where L(f) is the original loss function, λ*Ω(f) is the penalty term, λ is the regularization parameter, and R(f) is the new loss function.

The principle of regularization is to minimize the bias of the model without significantly increasing the variance, so that the model achieves a good balance of variance-bias.

Depending on the penalty term, regularization is divided into L1 regularization and L2 regularization. L1 regularization is also called Lasso regularization. It adds a penalty term, which is the sum of the absolute values of the coefficients, that is, Ω(f) = ∑|w|. L2

regularization, also known as Ridge regularization, adds a penalty term for the coefficients of for the sum of squares of the weights. That is, $\Omega(f)=\sum(w)^2$.

2.3 Model Evaluation Criteria

2.3.1 Confusion Matrix

Confusion matrix is a tool that commonly used in machine learning for evaluating classification problems by counting the number of data predicted to be true or false corresponding to the number that are actually true or false, namely: true-positive (TP), false-positive (FP), true-negative (TN), false-negative (FN). The form of the confusion matrix is shown in Table 2:

Table 2. Confusion Matrix.

Margin	**mm**
Top	24
Bottom	16
Left	20
Right	20

With these four values, it can further calculate four common performance metrics, namely Accuracy, Precision, Recall, and F1-Score.

However, evaluation on loan default risk is a multi-classification task in this article, so the characteristics of the dataset cannot be simply classified into true and false samples. To do this, the four metrics can be calculated by first calculating the average of the four quantities for each class, thereby resulting in a confusion matrix for the multi-classification task.

When there is an imbalance in the number of individual values of the dataset's objectives, the method of calculating the averages can also affect the results of the final four indicators. There are usually three methods of calculating averages: one-to-one, unweighted average, and weighted average. One-to-one means that each category pair of values in a multicategory is treated as a single category. This method produces a larger number of results and does not easily yield comprehensive performance. The unweighted average is also calculating each pair of categories and then calculating the average. This method does not take the number of individual classes into account. On the basis of the unweighted average, the weighted average calculates a weighted average based on the number of each category. The weighted average calculation better reflects the performance of the model classification, so the weighted average is used in this paper.

2.3.2 ROC Curve and AUC Value

The Receiver Operating Characteristic curve, or ROC curve in abbreviation, is a visual indicator of the model's evaluation. Based on the calculation of the confusion matrix, ROC curves can be plotted and AUC values can be calculated. The ROC curves show the relationship between true positive and false positive rates at different lower thresholds.

By calculating the true positive and false positive rates at different thresholds, a roughly smooth curve can be plotted. The curve ends at the lower left and upper right corners, with an overall bulge to the upper left. The closer the ROC curve is to the upper left corner, the better the performance of the model is

However, ROC curves can only give a rough indication of model performance, not an accurate one. As the ROC curve gets closer to the upper left corner, the larger the area enclosed by the ROC curve with the lower and right edges. This area is called Area Under the Curve (AUC), and usually ranges from 0.5 to 1. The closer the AUC value is to 1, the better the performance of the model classification.

Similarly, in a multi-classification task, it cannot straightforward to plot the ROC curve and compute the AUC value. It is required to do the computation for each pair of classes separately and then compute the average value in a weighted manner.

2.3.3 kappa coefficient

Kappa coefficient is another measure of machine learning classification accuracy based on confusion matrix. It is used to assess agreement between two raters or classifiers when assigning classification labels to a set of items. The range of values for the Kappa coefficient is between -1 and 1. As the Kappa coefficient gets closer to 1, it indicates that the model classification is more accurate. When the kappa coefficient is 0, it indicates that the model effect is equivalent to a random guess. And a kappa coefficient of less than 0 indicates that the model is predicting even less than a random guess.

3 Research Result

3.1 Data Introduction

3.1.1 Data Source

This paper uses publicly available datasets from the Kaggle platform. The dataset is internal data of an Indian bank and external information of Credit Information Bureau India Limited (CIBIL). The dataset is divided into two parts: Internal Bank Dataset and External Cibli Dataset.

3.1.2 Data Information

In this paper, Internal Bank Dataset and External Cibli Dataset are merged and used as training dataset for training the model. The merged Internal Bank Dataset and External Cibli Dataset resulted in a dataset containing 78 features, 1 target, and about 50,000 rows of data.

After merging Internal Bank Dataset and External Cibli Dataset, the merged dataset is obtained, and the partial feature names and their meanings are shown in Table 3.

Table 3. Partial variables name and description.

Variable Name	Description
Total_TL	Total trade lines/accounts in Bureau
Tot_Closed_TL	Total closed trade lines/accounts
Tot_Active_TL	Total active accounts
Total_TL_opened_L6M	Total accounts opened in last 6 Months
Tot_TL_closed_L6M	Total accounts closed in last 6 months

3.1.3 Data Distribution

In data mining, there are various ways to view the distribution of data. For numerical variables, the data distribution can be shown by plotting a Box Graph and a Kernel Density Estimation Graph. For categorical variables, the quantitative distribution and the percentage of each value can be shown by plotting histograms and pie charts.

Fig. 1, Fig. 2 show the distribution of feature Tot_Active_TL and tag Approved_Flag respectively:

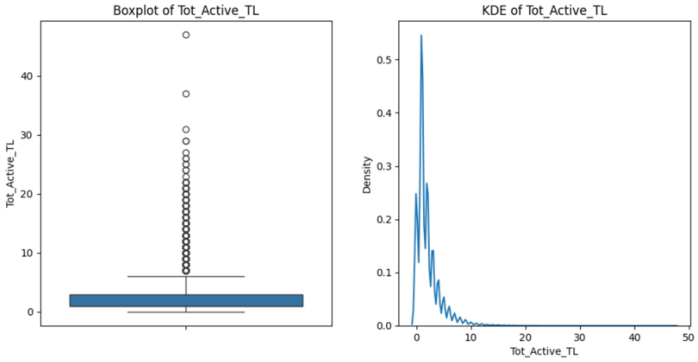

Fig. 1. Box plot and KDE variable Tot_Active_TL (Photo/Picture credit : Original)

Fig.1 shows the distribution of the variable Tot_Active_TL. The boxplot shows the upper and lower edges and median of the feature Tot_Active_TL, where some of the data is clustered around 1, while a larger number of strongholds exist above the upper edge. It shows that there may be an outlier in the data in the column. The KDE plot then visualizes the distribution pattern of the variable Tot_Active_TL. The figure shows that the variable Tot_Active_TL can be roughly considered to have a single-peaked distribution, and the data are more centrally distributed around 1.

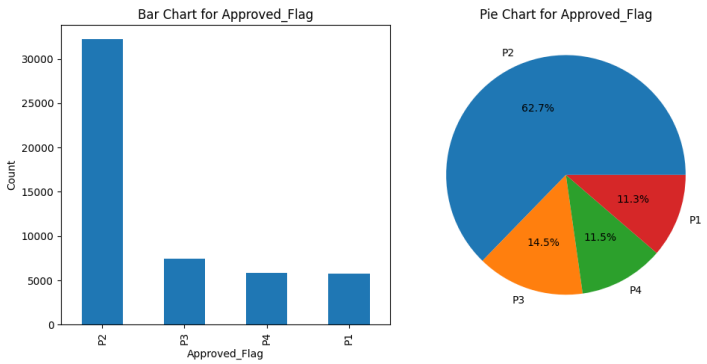

Fig. 2. Bar Chart of variable Approved_Flag (Photo/Picture credit : Original)

Approved_Flag is the target of the dataset. Observing the variable Approved_Flag not only helps to understand the type of data and its distribution, but also contributes to the subsequent processing of the data. It can be seen from the bar and pie charts in Fig.2 that type P2 occupies the largest proportion, while the remaining three types account for a smaller proportion. In addition, in Approved_Flag, P1, P2, P3, and P4 represent the four levels of default score first from low to high, with P1 being the lowest and P4 being the highest.

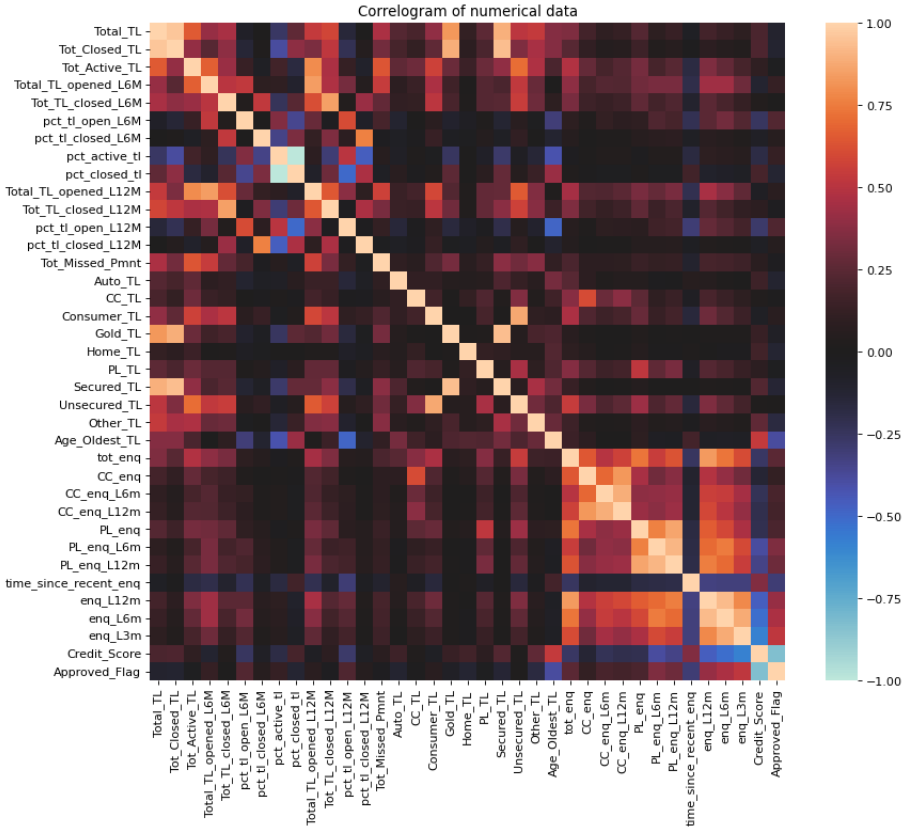

Fig. 3. Correlogram of partial numerical data (Photo/Picture credit : Original)

3.1.4 Feature Correlation

In addition to focusing on the distribution of individual data, it can also observe the correlation between any two features. By calculating the correlation coefficients between each numerical variable in the dataset, obtaining the correlation matrix and then plotting the correlation heat map based on the matrix. The correlation heat map is shown in Fig.3.

This heat map shows the correlation between some attributes. As two attributes get closer to the light orange color, the more positive the correlation between them is; as they get closer to the light blue color, the more negative the correlation between them is. Black color indicates a weak or no correlation.

3.1.5 Feature Importance

Feature importance is another key concept of data, and is used to measure the magnitude of a feature's impact on a target. Feature importance ranking is obtained after calculating the feature importance of each feature and sorting it according to the value size of the feature importance. The higher a feature is ranked, the more influence that feature has on the target and the more important it is in predicting the target.

In machine learning, calculating feature importance helps to understand the degree to which different features influence the model and contribute to the prediction task. There are multiple methods to calculate feature importance. The paper evaluates the importance of features by building a Random Forest Classifier (RFC), and using it to evaluate the

importance of features by calculating the average of the impurity reductions brought about by the splitting of each feature at the decision tree nodes. In this method, if the mean impurity reduction of a feature is higher on average, the importance of this feature is also higher.

Using Random Forest Classifier to train the model, the importance feature ordering of the top ten features in the dataset is obtained as shown in Table 4:

Table 4. Rank the top ten by feature importance.

Feature	Importance
Credit_score	0.491434
Age_Oldest_TL	0.053827
enq_L3m	0.046026
enq_L6m	0.029678
time_since_recent_enq	0.024207
num_std	0.021821
num_std_12mts	0.019306
num_std_6mts	0.015193
pct_PL_enq_L6m_of_ever	0.013362
enq_L12m	0.013791

3.2 Data Processing

Preprocessing the training data is necessary before training the machine learning model. Raw data is usually messy and may include missing values, outliers, and duplicates, and the feature classes in the data are more complex. If the original data is used directly for model training, it may lead to problems such as model bias, increase in training time, and reduction of model generalization ability. Thus, the paper preprocesses the data.

Fill in missing values. In terms of missing values, the dataset used in this paper uses -99999 to represent missing values, so -99999 is first replaced with a null value to facilitate the next step. Then, check the number of missing values in the dataset. For features containing more than 10,000 rows of missing values, they can be considered to have too many missing values to get enough information from them and are dropped directly [11]. For data containing fewer than 10,000 rows of missing values, use the padded average to replace a value of -99,999 with the average value of the column.

Remove outliers. Another step in data preprocessing is removing outliers. Calculate the mean and standard deviation for each column of data and consider data that exceed the mean plus or minus three times the standard deviation as outliers. Then, remove these outlier data. However, some of the features in the dataset with special value ranges, such as percentages between 0 and 1, or features with small distribution ranges, are not processed.

Coding. There are data of string type in the dataset, for which they need to be converted to numerical type so that the model can process them. If there is a size ordering relationship between the individual values of the feature, it is coded from 1, starting from the lowest to the highest. If there is no size ordering relationship, then using the dummy coding. To be specific, data of type string is converted into N status columns, each corresponding to each type of the original data. The status column uses a Boolean value of true to indicate that the attribute in the changed row of data belongs to that value, and the rest of the columns are set to false. This method is suitable for string properties with fewer types. However, when this column has too many value types, it creates also too many state columns, which may cause

an increase in the complexity of the model, thus decrease the classification performance of the model [11].

Data Normalization. Data normalization is the final step in data preprocessing. Because the data distribution ranges vary in size for each feature of the dataset and these gaps can lead to unfairness when analyzing the data, that is, data with large distribution ranges will dominate while data with small distribution ranges will be obscured, there is a need to narrow down the ranges of all the data to the same size. This paper uses maximum-minimum normalization to normalize the data. The process involves subtracting the minimum value from the data as a whole and then dividing it by the difference between the maximum and minimum values of the data, thereby scaling the data to a range between 0 and 1 [11].

After the data preprocessing is completed, dividing the previously obtained merged dataset into train set and test set. Training set is used to train the model to learn the features and patterns in the data. Test set is used to test the model to evaluate the performance of the model on unseen data and then to check the generalization ability of the model. This paper divides the dataset as 80% for training set and 20% for testing set.

3.3 Results

The performance of the different models was obtained through the various steps mentioned above. The performance of each model is shown in Table 5 and Fig. 4.

As shown in Table 5, the Deep Neural Network performs best on the prediction of the personal loan default risk dataset, where it is first in all metrics, reflecting its strong ability to fit high-dimensional data. The second is Logistic Regression. The Support Vector Machines are ranked third in terms of performance. Finally, the Naïve Bayes with Accuracy, Precision, Recall, F1-Score, AUC and Kappa coefficients are the lowest among the four models. And in Fig. 4, it can also be observed by comparing the ROC curves of each model that the ROC curve of the deep neural network is closest to the upper left corner. This concludes that the Deep Neural Network has the best performance and the highest prediction accuracy.

In particular, Deep Neural Networks have the characteristic of obtaining data features by training the data layer by layer and have high generalization ability. Also, Deep Neural Networks can handle nonlinear data well. Therefore, the Deep Neural Network has the best performance among the four models.

Table 5. Performance for each model.

Model	Accuracy	Precision	Recall	F1-Score	AUC	Kappa
LR	0.83	0.81	0.83	0.81	0.95	0.66
SVM	0.79	0.76	0.79	0.76	0.95	0.57
NB	0.55	0.67	0.55	0.57	0.79	0.33
DNN	0.94	0.94	0.94	0.94	0.99	0.92

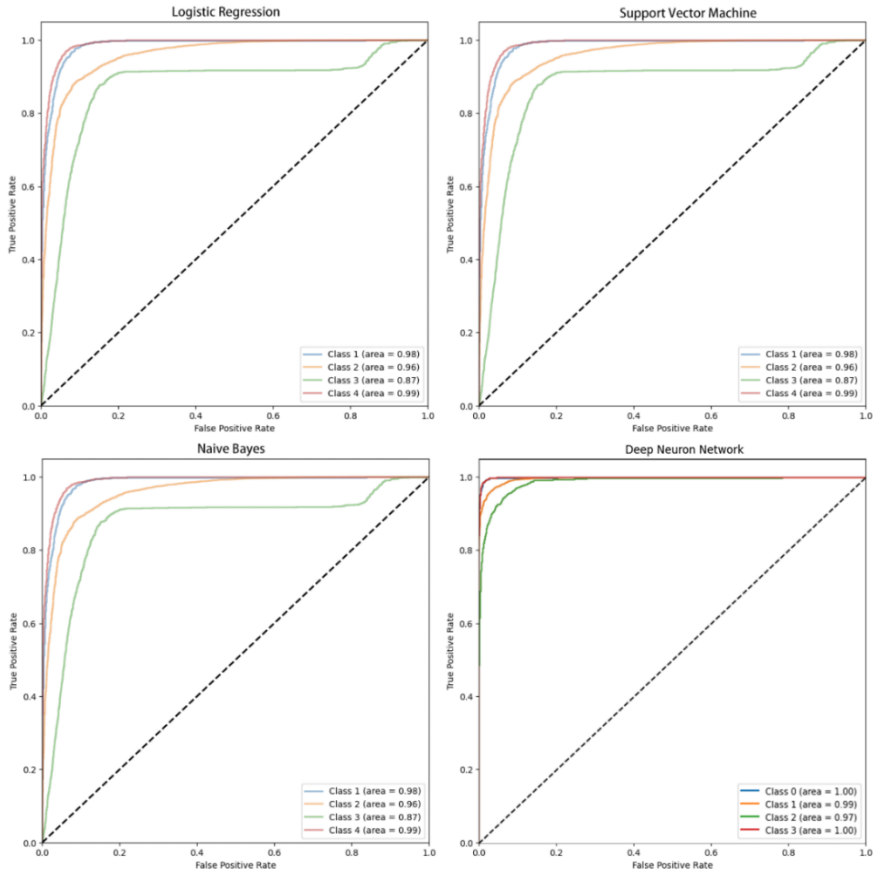

Fig. 4. ROC curve for each model (Photo/Picture credit : Original)

In addition, the complexity of the model and the training time are also indicators for evaluating the performance of the model. Detect the run lengths of the four models separately, where the run includes a combination of the model's training and prediction lengths. The hyperparameters of the model have been given by Table 1. The results are shown in Table 6.

Table 6. Run time of each model.

Model	LR	SVM	NB	DNN
Runtime(sec)	0.91	129.29	0.11	38.00

It should be noted that the actual runtime of the model is affected by the performance of the computer and that the hyperparameters of the model also affect the final runtime of the model. Therefore, the above training hours are relative. From the above results, Support Vector Machines took the most time to train, 129.29 seconds. Deep Neural Networks took less time, at 38 seconds. Logistic Regression and Naïve Bayes took an extremely short runtime of less than a second.

4 Conclusion

The evaluation of the risk of default on personal loan is an important issue in the financial field. This paper constructs four machine learning models, Logistic Regression, Support

Vector Machine, Simple Bayes and Deep Neural Networks, based on data from Kaggle platform originating from a bank in India and data from the Credit Information Bureau India Limited, and investigates the predictive effectiveness of these models for the dataset. After the model predicts the results, using metrics such as confusion matrix, ROC curve and kappa value to evaluate the model performance.

In terms of personal loan default risk evaluation, machine learning algorithms have the advantage of high prediction accuracy, speed and efficiency over traditional methods based on expert experience assessment. However, there are also performance differences between different machine learning algorithms. Among the four models mentioned above, Deep Neural Networks have the best comprehensive performance. About the Support Vector Machine, although it ranked second in metrics such as prediction accuracy, is not suitable for classification and prediction when there are time constraints due to the excessive training time. In regard to logistic regression, the model is fast and accurate, so it can also be a choice for classification tasks. Finally, although the running time of Naïve Bayes is also extremely short, the model is simpler and performs less well in the face of high-dimensional datasets. Thus, Naive Bayes has an advantage when confronted with datasets of lower complexity.

Data preprocessing methods have a significant impact on indicators such as the accuracy of the final model predictions. This paper uses the feature of filling the missing values with the mean and removing the missing values with excess 10,000 values. Data that were more than three times the standard deviation from the mean were then considered outliers and removed. The next step is to encode the string types in the dataset, where features with size ordering relationships are encoded from 1 and those without size ordering relationships are encoded as dummy codes. Finally, the dataset is max-min normalized to scale the dataset to between 0 and 1. When the above steps were not conducted, indicators such as the accuracy of the predictions of each model decreased significantly.

Apart from that, there are still limitations in this study. First, there are few types of machine learning models selected in this study. In this paper, only four types of machine learning algorithms have been selected for study, while there are many more algorithms to be further researched in the field of machine learning. Second, there is no further validation of the impact of various data processing methods on the accuracy of model predictions. Finally, all of the research in this paper is based on the dataset obtained by merging the two datasets, and no exploration of the performance of individual machine learning algorithms has been conducted on other datasets. Therefore, this paper has limitations in the selection of data sets.

In summary, although there are limitations in this study, the results have some theoretical implications for machine learning in multi-classification tasks as well as in the evaluation of personal loan default risk.

References

1. N.T. Luu and P.D. Hung, Loan Default Prediction Using Artificial Intelligence for the Borrow - Lend Collaboration, Cooperative Design, Visualization, And Engineering (CDVE 2021), vol. 12983, pp. 256-270, (2021)
2. K. Du, Q. Huang and Y. Zhang, A Review of Personal Credit Assessment Models in China, Management and Administration, (01):166-172 (2021), doi: 10.16517/j.cnki.cn12-1034/f.2021.01.036
3. Z. Qiu and B. He, Research on the Construction of Credit Risk Assessment System under Machine Learning Algorithm--Analysis of Personal Credit Risk Evaluation Based on China UnionPay Data, Price: Theory & Practice, (10):89-92+194(2021). doi: 10.19851/j.cnki.cn11-1010/f.2021.10.358

4. S. Z. H. Shoumo, M. I. M. Dhruba, S. Hossain, N. H. Ghani, H. Arif and S. Islam, Application of Machine Learning in Credit Risk Assessment: A Prelude to Smart Banking, TENCON 2019 - 2019 IEEE Region 10 Conference (TENCON), Kochi, India, 2019, pp. 2023-2028, doi: 10.1109/TENCON.2019.8929527
5. G. Zhang, Research on personal credit risk assessment based on machine learning method, University of International Business and Economics, 2023. doi: 10.27015/d.cnki.gdwju.2023.000062
6. F. Shen, X. Zhao, Z. Li, K. Li and Z. Meng, A novel ensemble classification model based on neural networks and a classifier optimisation technique for imbalanced credit risk evaluation, Physica A: Statistical Mechanics and its Applications(Journal), vol. 526 (2019), https://doi.org/10.1016/j.physa.2019.121073
7. J. Luo, X. Yan and Y. Tian, Unsupervised quadratic surface support vector machine with application to credit risk assessment, European Journal of Operational Research, vol. 280, pp. 1008-1017 (2020), doi: 10.1016/j.ejor.2019.08.010
8. F. Yang, Y. Qiao, C. Huang, S. Wang and X. Wang, An Automatic Credit Scoring Strategy (ACSS) using memetic evolutionary algorithm and neural architecture search, Applied Soft Computing, vol. 113 Part A (2021), doi: 10.1016/j.asoc.2021.107871
9. K. Jin and C. Zhu, A research on feature optimization of personal credit evaluation based on svm, Sun Yat-Sen University (People's Republic of China), (2009)
10. J. Zhao and L. Jiao, Fast Sparse Deep Neural Networks: Theory and Performance Analysis, in IEEE Access, vol. 7, pp. 74040-74055 (2019), doi: 10.1109/ACCESS.2019.2920688
11. Y. Wu and Y. Pan, Application Analysis of Credit Scoring of Financial Institutions Based on Machine Learning Model, (2021), https://doi.org/10.1155/2021/9222617