ISSN 1330-3651 (Print), ISSN 1848-6339 (Online) https://doi.org/10.17559/TV-20221107051221
Original scientific paper

Modeling Selection for Credit Risk Measurement: Based on Meta Path Features

Marui DU*, Zuoquan ZHANG

Abstract: As the application scope of machine learning expands, studies of credit risk measurement have also witnessed extensive development. An increasing number of studies showed that models based on machine learning algorithms could be used as a substantive solution for credit risk modeling. Recently, path-based features showed their advantages in risk measurement of the rich semantic and relational information it contains. However, studies have yet to be probed into the field that combines meta path features and machine learning models to measure the credit risk of the enterprise. In response to this problem, we compare the performance of machine learning models in terms of meta path features to find suitable machine learning models for meta path features. This paper compares six commonly used machine learning classification models, including neural networks, support vector machine, k-nearest neighbor, random forest, AdaBoost, and GBDT. Experiments on three listed small and medium-sized enterprises datasets in China. We found that neural networks, support vector machines, and the GBDT model perform better than other machine learning models. These are potential classifiers for small and medium-sized enterprises' credit risk measurement.

Keywords: classification technique; credit risk measurement; machine learning

1 INTRODUCTION

Small and medium-sized enterprises (SMEs) are essential to the national economy. However, SMEs are often underdeveloped for need of more financial support. The primary reason is the need for a model that can accurately measure the credit risks of SMEs. Therefore, it is crucial to build a model that can accurately measure SME credit risks.

In traditional SME credit risk measurement, statistical and econometric methods are often used. For instance, Ohlson [1] was the first to apply logistic regression analysis for enterprise credit risk measurement. Many scholars conducted relevant studies using discriminant analysis and logistic regression [2, 3]. Different machine learning algorithms have been proposed with the development of machine learning techniques. McCulloc proposed the neural network [4] model in 1943 and quickly achieved many applications. It describes the prototype of machine learning theory for the first time. Subsequently, Cortes and Vapnik proposed the most widely used machine learning model support vector machine [5]. Breiman and Schapire proposed the ensemble learning algorithms bagging [6] and boosting [7]. Machine learning algorithms have made breakthroughs in image recognition, information retrieval, risk prediction, and many other fields [8, 9]. More and more machine learning methods also show the advantages of enterprise credit risk measurement. For example, Nanni [10] proved the classification capability of machine learning algorithms by using Australian, German, and Japanese public datasets. Ciampi [11] compared the neural network approach with traditional methods and found that the prediction accuracy of the model generated by the neural network was higher in small and medium-sized firms, suggesting that neural networks may improve the lending decisions of small businesses. Tobback [12] compared the classification performance of logistic regression, decision trees, artificial neural networks, support vector machines, and other algorithms and proved that the machine learning algorithm is better than the traditional enterprises' credit risk measurement methods. Wang [13] compared the classification performance of logistic regression, decision tree, artificial neural network, support vector machine, bagging, boosting, and stacking, concluding that the capability to identify defaulting enterprises of machine learning algorithms was better than traditional methods. Marques [14] compared the classification performance of logistic regression, naive Bayes, linear discriminant analysis, decision tree, K-nearest neighbors, and support vector machine based on Australian, German, and Japanese public datasets. This study also proves that machine learning algorithms outperformed traditional methods in identifying defaulting enterprises.

Machine learning methods can gain experience and knowledge from data, featuring higher flexibility during the data fitting process, more substantial explanatory power, and greater prediction ability in studies of nonlinear causal relationships. Baesens [15] compared the classification performance of several machine learning algorithms in a German public dataset, concluding that support vector machine, artificial neural network, and logistic regression outperformed others in risk measurement. Feng-Chia compared the classification of the support vector machine, K-nearest neighbors, and artificial neural network based on Australian and German public datasets and found that K-nearest neighbors showed the best performance in the Australian public dataset. In contrast, the support vector machine was the best performer in the German public dataset. There are many types of research on machine learning algorithms in risk measurement. However, due to the different data used, methods selected and parameters designed, the results of algorithm effect comparison cannot be unified. There are also mixed voices on the effectiveness of using machine learning algorithms to build risk measurement models. Some studies believe that using machine learning algorithms to build risk measurement models can improve the classification accuracy of algorithms and provide feasibility for accurate classification. Some other researchers believe that machine learning models use dynamic training, high-dimensional data, and complex nonlinear relationships, which reduces the transparency of the models and makes it more challenging to identify and evaluate risks.

As big data technology develops, massive information enterprises are recorded and accumulated, which has diversified the features used for credit risk measurement. Four main types of features are used: financial indicators, enterprises' non-financial indicators, homogeneous-path features, and meta path features. Among all feature types, financial indicators and enterprises' non-financial indicators are classified as conventional features [15-18]. Conventional features often include enterprise financial information, enterprise establishment year, enterprise scale, and other basic enterprise information. The homogeneous and heterogeneous-path features are path-based features [19, 20]. Homogeneous-path feature refers to indicators constructed based on the inter-relationships between enterprises. Such as inter-enterprise payment relationships [21], inter-enterprise joint executive relationships [22, 23], and inter-enterprise transaction information [24]. Heterogenous-path features [25] refer to indicators constructed based on multiple entities and relationships.

The studies mentioned above have the following two gaps, which serve as the direction of this study. Firstly, in existing studies, comparisons of the classification performance of various machine learning methods were usually conducted based on public datasets. Except few studies using features generated from real datasets, most of the indicators used are conventional features. Despite the sound effect machine learning algorithms showed in public datasets, the effects they achieved in real datasets still need to be verified. Secondly, multiple types of features can be used for credit risk evaluation. However, given the influence of contagion risk suffered by enterprises, the effect of conventional features was questioned. Researchers begin to turn their attention to path-based features. Although the performance of machine learning methods in conventional features has been justified, their effect on a path-based feature still awaits to be proven.

This paper will measure and test the performance of meta path features in machine learning models. The contributions of this paper are as follows:

1. The performance of machine learning classification algorithms in the path-based features is measured and compared. The advantages and disadvantages of different machine learning classification models were also analysed.
2. The effect of multiple machine learning models was measured in three listed SME datasets, including the Growth Enterprise Market (GEM) and the Small and Medium-Sized Enterprise Board (SMB) from Shenzhen Stock Exchange and the Science and Technology Innovation Board (STAR) from Shanghai Stock Exchange. According to the results, neural networks, support vector machines, and GBDT can be used as potential classifiers for small and medium-sized enterprises' credit risk evaluation.

The structure of this paper is as follows. Section 2 introduces the features and techniques used in this paper, including meta-path features and machine learning models. Section 3 presents the experiments and analysis of the results. Section 4 concludes the paper.

2 RESEARCH METHODOLOGY

2.1 Credit Scoring

Credit scoring is a set of decision support techniques that assist an enterprise's credit. It reflects the probability that an enterprise will default. The nature of credit scoring dates back to the 1950s. It uses a series of enterprise characteristics to classify enterprise into default or not. Fisher [26] was the first to distinguish groups from the population, use the size of plants to distinguish iris flowers, and the physical size of the cranium to distinguish ethnic origin. Durand [27] extended the research of Fisher and used it to distinguish loan customers. In 1968, Altman [28] introduced the credit score method into enterprise bankruptcy prediction.

For enterprise characteristics $X=(X_1, X_2, ..., X_P)$, each enterprise with the characteristic x, a credit score is defined as $s(x)$.A credit score is a sufficient statistic that describes the probability that an enterprise with characteristics will perform satisfactorily on loan (repay a loan on time). If we use the initial G of good for satisfactory performance and the initial B of bad for unsatisfactory performance, $s(x)$ has the following properties:

$$P(G \mid x) = P(G \mid s(x)),\ x \in X \tag{1}$$

Unsatisfactory performance typically occurs when a borrower defaults within a defined period. Most scores have a monotonic relationship with the probability of being good. That is, borrowers with high scores are less risky, and borrowers with low scores are more likely to default.

The most popular model of credit scoring is logistic regression. It yields a score known as a log odds score:

$$s(x) = \ln \frac{P(x \mid G) P_G}{P(x \mid B) P_B} = \ln \frac{P_G}{P_B} + \ln \frac{P(x \mid G)}{P(x \mid B)} \tag{2}$$

where P_G and P_B are the proportions of good enterprise and bad enterprise in the population.

2.2 Meta Path Feature

The meta path feature is an indicator constructed based on the meta path. It uses the logic chain to represent enterprise-associated information. It can not only represent basic enterprise information but also identify the association relationships of the enterprise. With the help of the meta path feature, we can capture the contagion credit risk transferred from enterprise-related entities.

Definition 1: The network schema $S=(\mathcal{A}, \mathcal{R})$, is a meta-level representation for an information network $G=(\mathcal{V}, \mathcal{E})$ with object type function $\tau : \mathcal{V} \to \mathcal{A}$ and relation type function $\phi : \mathcal{E} \to \mathcal{R}$, which is a directed graph over object types $\mathcal{A}$ and edges as relations from $\mathcal{R}$.

Definition 2: With a schema $S=(\mathcal{A}, \mathcal{R})$, a meta path P in the form $\mathcal{A}_1 \xrightarrow{\mathcal{R}_1} \mathcal{A}_2 \xrightarrow{\mathcal{R}_2}, ..., \xrightarrow{\mathcal{R}_n} \mathcal{A}_{n+1}$ which defines a composite relation $\mathcal{R}=\mathcal{R}_1 \circ \mathcal{R}_2 \circ , .., \circ \mathcal{R}_n$

between $\mathcal{A}_1$ and $\mathcal{A}_{n+1}$, where $\circ$ denotes the composition operator on relations.

From the above definitions, some commonly used meta path (MP) features [25] are given:

Naive meta path feature: Naive meta path feature is defined as the impact of meta path P on target object, denoted as:

$$N_P(x) = \frac{\left|\left\{x' \in D \mid \exists p_{x \rightsquigarrow x'} \in P,\, \Gamma(x') = 1\right\}\right|}{\left|\left\{x' \in D \mid \exists p_{x \rightsquigarrow x'} \in P\right\}\right|} \tag{3}$$

where D is an enterprise object collection, $p_{x_i \rightsquigarrow x_j}$ is a path instance from object x_i to object x_j and $\Gamma(x)$ is a risk inference function.

CountSim meta path weight feature: CountSim meta path feature is defined as an indicator to reveal the structure relevance impact of meta path P on enterprise x, denoted as:

$$C_P(x) = \frac{\left|\left\{x' \in D \mid \exists p_{x \rightsquigarrow x'} \in P\right\}\right|}{\left|\left\{x \in S\right\}\right| + \left|\left\{x' \in S'\right\}\right|} \tag{4}$$

where S and S' are enterprise object collections where are all links from x and to x' respectively, D is another enterprise object collection which contains all objects.

HeteSimmeta pathweight feature: HeteSim meta path feature takes HeteSim as the similarity measure to reveal the path relevance impact of meta path P on enterprise x, denoted as:

$$H_P(x) = \frac{\sum_{x' \in \{x' \mid \exists p_{x \rightsquigarrow x'} \in P, \Gamma(x')=1\}} HeteSim(x, x')}{\sum_{x' \in \{x' \mid \exists p_{x \rightsquigarrow x'} \in P\}} HeteSim(x, x')} \tag{5}$$

where $p_{x \rightsquigarrow x'}$ is a path instance from object x to object x', $HeteSim(x, x')$ is the relevance between object x and object under HeteSimand $\Gamma(x)$ is an estimating function.

2.3 Machine Learning Models in Credit Risk Measurement

2.3.1 Neural Networks

Neural Networks (NN) [4] were initially proposed to mimic the human brain for communicating and processing information. A neural network consists of a large number of input variables, each of which is multiplied by a weight, and these results are summed and transformed across neurons as input variables for the next neuron. The prediction result of a sample is judged based on the neural network's output. Mathematically, it can be expressed as:

$$y_k = F_1\left(\sum_{q=0}^{p} \omega_{kq} x_q\right) \tag{6}$$

where x_q is the q feature of enterprise, ω_{kq} is the weight of the k neuron on the q feature, F_1 indicates the first hidden layer after the input layer, $y_k (k = 1, 2, \ldots, r)$ is the output of the first hidden layer.

Since the output of the previous layer is the input of the next layer, denoted as:

$$z_v = F_2\left(\sum_{k=1}^{r} K_{vk} y_k\right) = F_2\left(\sum_{k=1}^{r} K_{vk}\left(F_1\left(\sum_{q=0}^{p} \omega_{kq} x_q\right)\right)\right) \tag{7}$$

where z_v is the output value of neuron v in the output layer, $v = 1, 2, \ldots, s$, F_2 is the activation function of the output layer, K_{vk} is applied to the weight of neuron k on the link hidden layer y and neuron v on the output layer.

The computation of the weight vector is the process of training the model. One of the most commonly used methods is backpropagation. For a sample where the input and output are known, the sample is continuously added to the network, and the weights are adjusted to minimize the error function. For the training sample t, define the error:

$$e_v(t) = o_v(t) - y_v(t) \tag{8}$$

where $o_v(t)$ is the observation of sample t on neuron v, and $y_v(t)$ is the prediction.

In backpropagation, the error is first pushed from the back to the front according to the weight. Then adjust the weight to reduce the error of each pattern. After that, a second sample is loaded, and the backpropagation process is repeated. Continue loading the samples until all samples have been loaded. The whole process is repeated many times until the stopping condition is reached.

2.3.2 Support Vector Machine

Support Vector Machine (SVM) [5] was proposed by Cortes and Vapnik. Like neural networks, SVM maps the input space into a high-dimensional feature space and then performs linear partitioning. For a training set $T = (x_1, y_1), (x_2, y_2), ..., (x_N, y_N)$, where $x_i \in \mathcal{X} = \boldsymbol{R}^n$, $y_i \in \mathcal{Y} = \{-1, 1\}$, $i = 1, 2, ..., N$. x_i is the ith instance, y_i is the label of x_i. When $y_i = 1$, x_i is positive instance, $y_i = -1$, x_i is negative instance, (x_i, y_i) called the sample points. The goal of learning is to find a separating hyperplane in the feature space that can classify instances into different classes. These parating hyperplane pair applies equation $\omega x + b = 0$, which is determined by the normal vector ω and the intercept b and can be represented by (ω, b). The separating hyperplane divides the feature space into two parts, one with positive and one with negative instances. Typically, a linear classifier can be used to segment the sample points. However, in many cases, the two types of points are mixed, making it difficult for the linear classifier to distinguish them. SVM maps the features into a high-dimensional space, then partitioned using a linear classifier. The values computed from these sample features are pushed back into the input space.

We can construct and solve constrained optimization problem:

$$\min_{\alpha} \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j K\left(x_i, x_j\right)+\sum_{i=1}^{N}\alpha_i$$
$$s.t.\sum_{i=1}^{N}\alpha_i y_i = 0,\ 0 \le \alpha_i \le c \tag{9}$$

where $K(x_i, x_j)$ is a kernel.

The class of a new sample can be determined using the following hyperplane equation:

$$y' = \mathrm{sign}\left(\sum_{i=1}^{N}\alpha_i y_i K\left(x_i, x_j\right)+b\right) \tag{10}$$

The kernel function can be one of the following:
(1) Multinomial kernel function:

$$K\left(x_i, x_j\right) = \left(\theta + x_i^T x_j\right)^d \tag{11}$$

where d and θ are hyper parameter.
(2) Radial basis kernel function:

$$K\left(x_i, x_j\right) = \exp\left(-\frac{1}{2\sigma^2}\left\|x_i - x_j\right\|^2\right) \tag{12}$$

where σ is hyper parameter.
(3) Linear kernel function:

$$K\left(x_i, x_j\right) = x_i^T \cdot x_j \tag{13}$$

(4) Multi-layer perceptron kernel function:

$$K\left(x_i, x_j\right) = \tan h\left(\delta_1 x_i^T x_j + \delta_2\right) \tag{14}$$

where δ_1 and δ_2 need to be selected within limites.

2.3.3 K-Nearest Neighbour

The K-Nearest Neighbor (KNN) [29] is a typical non-parametric method, which was first proposed by Fix and Hodges to solve the classification problem, and later was successively applied to credit scoring by Chatterjee and Henley [30]. The principle of KNN is to use a metric to measure the distance between two samples in the data space. If the past sample is used as a guide, the new sample can be judged according to the class of its neighbors, which is determined by the proportion of its nearest k nearest neighbors that are good or bad. Henley comprehensively reviewed the application of k nearest neighbor to credit scoring in his paper, where the metric used is a combination of Euclidean distance and the direction of the optimal discriminative of good and bad. If w is a $\boldsymbol{p}$ dimensional vector defined in this direction, then the distance is measured as follows:

$$d\left(x_1, x_2\right) = \left[\left(x_1, x_2\right)^T\left(\boldsymbol{I} + Dw \cdot w^T\right)\left(x_1 - x_2\right)\right]^{\frac{1}{2}} \tag{15}$$

where $\boldsymbol{I}$ is the identity matrix.

For a training set $T = \left(x_1, y_1\right), \left(x_2, y_2\right), ..., \left(x_N, y_N\right)$, where $x_i \in \mathcal{X} = \boldsymbol{R}^n$ is the eigen vector of the instance; $y_i \in \mathcal{Y} = \left\{c_1, c_2, ..., c_K\right\}$, is the category of the instance; $i = 1, 2, ..., N$; instance eigenvector is x.The process of the k-Nearest Neighbor is as follows:
(1) Based on the given distance metric, find the k points in the training set T that are closest to x, the neighborhood of x covering these k points is denoted as $N_k(x)$.
(2) Decide the class y of x in $N_k(x)$ according to a classification decision rule (such as majority voting):

$$y = arg\max_{c_j}\sum_{x_i \in N_k(x)} I\left(y_i = c_j\right) \tag{16}$$

where $i = 1, 2, ..., N$; $j = 1, 2, ..., K$; I is the indicator function, that is when $y_i = c_j$, $I = 1$, otherwise $I = 0$.

2.3.4 AdaBoost

The Boosting algorithm [7] aims to combine multiple weak classifiers into a robust classifier with higher classification accuracy. Its basic idea is that classifiers should pay more attention to the misclassified samples in the previous round. Each classifier might perform better against specific samples. Therefore, different classifiers can form a complementary classifier system.

Ada Boost [31] is the most representative algorithm in the Boosting family. Freund and Schapire proposed it in 1997. The procedure of AdaBoost is as follows:
(1) A training subset $T = \left(x_1, y_1\right), ..., \left(x_N, y_N\right)$ is randomly selected and used to train a model C_1, with equal weights applied to all samples:

$$D_1 = \left(w_{11}, ..., w_{1i}, w_{1N}\right), w_{1i} = \frac{1}{N},\ i = 1, 2, ..., N \tag{17}$$

(2) The training data set with weight distribution D_m is used for learning to obtain a classifier $G_n\left(x\right) \in \left\{-1, +1\right\}$ and the predicted category and the true category is compared.
(3) Calculate the classification error of $G_m(x)$ on the training dataset:

$$e_m = P\left(G_m\left(x_i\right) \ne y_i\right) = \sum_{i=1}^{N} w_{mi} I\left(G_m\left(x_i \ne y_i\right)\right) \tag{18}$$

(4) The weight of $G_m(x)$ is computed:

$$\alpha_m = \frac{1}{2}\log\frac{1-e_m}{e_m} \tag{19}$$

(5) The model C_2 is trained once by increasing the weight of the predicted wrong samples:

$$D_{m+1} = \left(w_{m+1,1}, ..., w_{m+1,i}, ..., w_{m+1,N}\right) \tag{20}$$

$$w_{m+1,i} = \frac{w_{mi}}{Z_m}\exp\left(-\alpha_m y_i G_m\left(x_i\right)\right),\ i = 1,\ 2,\ ...,\ N$$

where Z is the normalization factor:

$$Z_m = \sum_{i=1}^{N} w_{mi}\exp\left(-\alpha_m y_i G_m\left(x_i\right)\right)$$

(6) After t iterations, each time the weight of the wrong prediction sample goes up and the weight of the right prediction sample goes down.
(7) When predicting each new sample, the T models are used to predict a category for it, and the results of each prediction are weighted, and finally the category is determined based on the category with the largest number of weights.

2.3.5 GBDT

GBDT (Gradient Boosting Decision Tree) [32] is an iterative decision tree algorithm composed of multiple decision trees, and the conclusions of all trees are summed up as the final decision. Similar to the Adaboost algorithm, GBDT also uses the addition model of the forward distribution algorithm. However, the base classifier limits the use of the CART regression tree model, and the iteration idea is also different from Adaboost. In GBDT algorithm, assuming that the classifier obtained in the previous iteration is $f_{t-1}(x)$ and the loss function is $L(y, f_{t-1}(x))$, then the goal of this iteration is to find a weak classifier of CART regression tree model and minimize the loss function of this iteration. The procedure of GBDT:
(1) Initialize the weak classifer:

$$f_0\left(x\right) = arg\min_{\gamma}\sum_{i=1}^{N} L\left(y_i,\ \gamma\right) \tag{21}$$

(2) For each iteration: $m = 1, 2, \ldots, M$.
(a) Compute the negative gradient (residual):

$$r_{im} = -\left[\frac{\partial L\left(y_i, f\left(x_i\right)\right)}{\partial f\left(x_i\right)}\right]_{f=f_{m-1}} \tag{22}$$

(b) Fit a regression tree to the target r_{im} giving terminal regions $R_{jm}, j = 1, 2, \ldots, J_m$.
(c) For $j = 1, 2, \ldots, J_m$ compute:

$$\gamma_{jm} = arg\min_{\gamma}\sum_{x_i \in R_{jm}} L\left(y_i, f_{m-1}\left(x_i\right) + \gamma\right) \tag{23}$$

(d) Update classifer:

$$f_m\left(x\right) = f_{m-1}\left(x\right) + \sum_{j=1}^{J_m}\gamma_{jm} I\left(x \in R_{jm}\right) \tag{24}$$

(3) Output the strong classifier: $\hat{f}\left(x\right) = f_M\left(x\right)$.

2.3.6 Random Forest

Random forest (RF) [33] is an ensemble algorithm for decision trees introduced by Breiman in 2001. For the training set $T = \left(x_1, y_1\right),\ \left(x_2, y_2\right),\ ...,\ \left(x_N, y_N\right)$, from this training set, boot strap sampling is performed K times to obtain K random training subsets. A classification tree model is built on each subset, and new samples are judged based on most predicted categories' results. When a new sample comes in for each decision tree, it moves along a branch and eventually reaches an endpoint. Its category is the category of the majority in the endpoint. Similarly, many trees trained on different samples are combined to predict a new sample. This process will be repeated times. When the majority of trees attribute samples to the leaf of an endpoint, the sample is predicted to be in the majority category.

3 EXPERIMENT AND RESULTS

3.1 Datasets Information

In this paper, three SME datasets are used. One of the data comes from the growth enterprise board (GEM) of the Shenzhen Stock Exchange and one from the science and technology innovation board (STAR) of the Shanghai Stock Exchange. These two groups of data are about high-tech SMEs. The last data group comes from the small and medium-sized (SMB) board of the Shenzhen Stock Exchange and is about traditional SMEs. All the datasets were downloaded from the CSMAR database (https://www.gtarsc.com). Enterprise data from 2010 to 2020 are selected, including 27032572 samples. In addition to enterprises' information, the data also covered information about relational enterprises (subsidiary, parents, supplier, and saler enterprises), the relational person (actual controller, manager, shareholder, board member), relational products (sale, produce), relevant news (irregularity, illegal), etc.

The feature used is three state-of-the-art meta path features, including the Naive MP feature, CountSim MP feature, and HeteSim MP feature. Classifiers include six commonly used machine learning models, namely support vector machine (SVM), neural network (NN), k-nearest neighbor (k-NN), random forest (RF), AdaBoost, and GBDT. Enterprises with negative net profit at the end of the year would be regarded as default, while other enterprises would be classified as not default. All the experiments were implemented in Python 2.7.17.

3.2 Data Pre-processing

First of all, three different types of meta path features are constructed. The construction of the Naive MP feature, CountSim MP feature, and HeteSim MP feature is according to [25]. Features with more than 50% missing data were removed, and then the data were filled by the mean value. In order to eliminate the adverse effects caused by sample data, the data is normalized. The normalization formula is shown in Eq. (25):

$$X_{\text{norm}} = \frac{X - X_{\min}}{X_{\max} - X_{\min}} \tag{25}$$

Finally, the 20 most significant meta path features are selected as the input variables for machine learning models (Tab. 1 to Tab. 3).

Table 1 meta path features for GEM dataset

Feature number	Features
1	$\mathcal{A}_e \cdot \mathcal{R}_{control} \cdot \mathcal{A}_p$
2	$\mathcal{A}_e \cdot \mathcal{R}_{parent} \cdot \mathcal{A}_e$
3	$\mathcal{A}_e \cdot \mathcal{R}_{report} \cdot \mathcal{A}_n$
4	$\mathcal{A}_e \cdot \mathcal{R}_{control} \cdot \mathcal{A}_p \cdot \mathcal{R}_{control} \cdot \mathcal{A}_e$
5	$\mathcal{A}_e \cdot \mathcal{R}_{produce} \cdot \mathcal{A}_c \cdot \mathcal{R}_{report} \cdot \mathcal{A}_n$
6	$\mathcal{A}_e \cdot \mathcal{R}_{manager} \cdot \mathcal{A}_p$
7	$\mathcal{A}_e \cdot \mathcal{R}_{produce} \cdot \mathcal{A}_c$
8	$\mathcal{A}_e \cdot \mathcal{R}_{boardmember} \cdot \mathcal{A}_p$
9	$\mathcal{A}_e \cdot \mathcal{R}_{shareholder} \cdot \mathcal{A}_p$
10	$\mathcal{A}_e \cdot \mathcal{R}_{control} \cdot \mathcal{A}_p \cdot \mathcal{R}_{report} \cdot \mathcal{A}_n$
11	$\mathcal{A}_e \cdot \mathcal{R}_{parent} \cdot \mathcal{A}_e \cdot \mathcal{R}_{manager} \cdot \mathcal{A}_p$
12	$\mathcal{A}_e \cdot \mathcal{R}_{subsidiary} \cdot \mathcal{A}_e$
13	$\mathcal{A}_e \cdot \mathcal{R}_{manager} \cdot \mathcal{A}_p \cdot \mathcal{R}_{report} \cdot \mathcal{A}_n$
14	$\mathcal{A}_e \cdot \mathcal{R}_{control} \cdot \mathcal{A}_p \cdot \mathcal{R}_{relate} \cdot \mathcal{A}_p$
15	$\mathcal{A}_e \cdot \mathcal{R}_{subsidiary} \cdot \mathcal{A}_e \cdot \mathcal{R}_{report} \cdot \mathcal{A}_n$
16	$\mathcal{A}_e \cdot \mathcal{R}_{control} \cdot \mathcal{A}_p \cdot \mathcal{R}_{manager} \cdot \mathcal{A}_e$
17	$\mathcal{A}_e \cdot \mathcal{R}_{subsidiary} \cdot \mathcal{A}_e \cdot \mathcal{R}_{control} \cdot \mathcal{A}_p$
18	$\mathcal{A}_e \cdot \mathcal{R}_{parent} \cdot \mathcal{A}_e \cdot \mathcal{R}_{report} \cdot \mathcal{A}_n$
19	$\mathcal{A}_e \cdot \mathcal{R}_{parent} \cdot \mathcal{A}_e \cdot \mathcal{R}_{control} \cdot \mathcal{A}_p$
20	$\mathcal{A}_e \cdot \mathcal{R}_{supply} \cdot \mathcal{A}_e$

Table 2 Meta path features for STAR dataset

Feature number	Features
1	$\mathcal{A}_e \cdot \mathcal{R}_{control} \cdot \mathcal{A}_p$
2	$\mathcal{A}_e \cdot \mathcal{R}_{parent} \cdot \mathcal{A}_e$
3	$\mathcal{A}_e \cdot \mathcal{R}_{shareholder} \cdot \mathcal{A}_p$
4	$\mathcal{A}_e \cdot \mathcal{R}_{report} \cdot \mathcal{A}_n$
5	$\mathcal{A}_e \cdot \mathcal{R}_{subsidiary} \cdot \mathcal{A}_e$
6	$\mathcal{A}_e \cdot \mathcal{R}_{manager} \cdot \mathcal{A}_p$
7	$\mathcal{A}_e \cdot \mathcal{R}_{boardmember} \cdot \mathcal{A}_p$
8	$\mathcal{A}_e \cdot \mathcal{R}_{control} \cdot \mathcal{A}_p \cdot \mathcal{R}_{report} \cdot \mathcal{A}_n$
9	$\mathcal{A}_e \cdot \mathcal{R}_{subsidiary} \cdot \mathcal{A}_e \cdot \mathcal{R}_{report} \cdot \mathcal{A}_n$
10	$\mathcal{A}_e \cdot \mathcal{R}_{produce} \cdot \mathcal{A}_c$
11	$\mathcal{A}_e \cdot \mathcal{R}_{control} \cdot \mathcal{A}_p \cdot \mathcal{R}_{control} \cdot \mathcal{A}_e$
12	$\mathcal{A}_e \cdot \mathcal{R}_{produce} \cdot \mathcal{A}_c \cdot \mathcal{R}_{report} \cdot \mathcal{A}_n$
13	$\mathcal{A}_e \cdot \mathcal{R}_{boardmember} \cdot \mathcal{A}_p \cdot \mathcal{R}_{report} \cdot \mathcal{A}_n$
14	$\mathcal{A}_e \cdot \mathcal{R}_{supply} \cdot \mathcal{A}_p$
15	$\mathcal{A}_e \cdot \mathcal{R}_{parent} \cdot \mathcal{A}_e \cdot \mathcal{R}_{control} \cdot \mathcal{A}_p$
16	$\mathcal{A}_e \cdot \mathcal{R}_{sales} \cdot \mathcal{A}_e$
17	$\mathcal{A}_e \cdot \mathcal{R}_{parent} \cdot \mathcal{A}_e \cdot \mathcal{R}_{manager} \cdot \mathcal{A}_p$
18	$\mathcal{A}_e \cdot \mathcal{R}_{manager} \cdot \mathcal{A}_p \cdot \mathcal{R}_{report} \cdot \mathcal{A}_n$
19	$\mathcal{A}_e \cdot \mathcal{R}_{supply} \cdot \mathcal{A}_e \cdot \mathcal{R}_{report} \cdot \mathcal{A}_n$
20	$\mathcal{A}_e \cdot \mathcal{R}_{parent} \cdot \mathcal{A}_e \cdot \mathcal{R}_{report} \cdot \mathcal{A}_n$

Table 3 Meta path features for SMB dataset

Feature number	Features
1	$\mathcal{A}_e \cdot \mathcal{R}_{report} \cdot \mathcal{A}_n$
2	$\mathcal{A}_e \cdot \mathcal{R}_{parent} \cdot \mathcal{A}_e$
3	$\mathcal{A}_e \cdot \mathcal{R}_{control} \cdot \mathcal{A}_p$
4	$\mathcal{A}_e \cdot \mathcal{R}_{subsidiary} \cdot \mathcal{A}_e$
5	$\mathcal{A}_e \cdot \mathcal{R}_{subsidiary} \cdot \mathcal{A}_e \cdot \mathcal{R}_{report} \cdot \mathcal{A}_n$
6	$\mathcal{A}_e \cdot \mathcal{R}_{control} \cdot \mathcal{A}_p \cdot \mathcal{R}_{control} \cdot \mathcal{A}_e$
7	$\mathcal{A}_e \cdot \mathcal{R}_{control} \cdot \mathcal{A}_p \cdot \mathcal{R}_{manager} \cdot \mathcal{A}_e$
8	$\mathcal{A}_e \cdot \mathcal{R}_{supply} \cdot \mathcal{A}_e$
9	$\mathcal{A}_e \cdot \mathcal{R}_{boardmember} \cdot \mathcal{A}_p$
10	$\mathcal{A}_e \cdot \mathcal{R}_{parent} \cdot \mathcal{A}_e \cdot \mathcal{R}_{manager} \cdot \mathcal{A}_p$
11	$\mathcal{A}_e \cdot \mathcal{R}_{produce} \cdot \mathcal{A}_c$
12	$\mathcal{A}_e \cdot \mathcal{R}_{boardmember} \cdot \mathcal{A}_p \cdot \mathcal{R}_{report} \cdot \mathcal{A}_n$
13	$\mathcal{A}_e \cdot \mathcal{R}_{shareholder} \cdot \mathcal{A}_p$
14	$\mathcal{A}_e \cdot \mathcal{R}_{control} \cdot \mathcal{A}_p \cdot \mathcal{R}_{shareholder} \cdot \mathcal{A}_e$
15	$\mathcal{A}_e \cdot \mathcal{R}_{sales} \cdot \mathcal{A}_e$
16	$\mathcal{A}_e \cdot \mathcal{R}_{parent} \cdot \mathcal{A}_e \cdot \mathcal{R}_{produce} \cdot \mathcal{A}_p$
17	$\mathcal{A}_e \cdot \mathcal{R}_{manager} \cdot \mathcal{A}_p \cdot \mathcal{R}_{control} \cdot \mathcal{A}_e$
18	$\mathcal{A}_e \cdot \mathcal{R}_{sales} \cdot \mathcal{A}_e \cdot \mathcal{R}_{report} \cdot \mathcal{A}_n$
19	$\mathcal{A}_e \cdot \mathcal{R}_{parent} \cdot \mathcal{A}_e \cdot \mathcal{R}_{report} \cdot \mathcal{A}_n$
20	$\mathcal{A}_e \cdot \mathcal{R}_{supply} \cdot \mathcal{A}_e \cdot \mathcal{R}_{report} \cdot \mathcal{A}_n$

3.3 Evaluation Indices Of Model Performance

The classification results are compared with the actual value. False positive (*type I error*) means the model mistakenly predicted the negative as a positive. For example, the model judged that an enterprise is a default (positive), but the enterprise is not a default. This is an alarm, and the situation would be better if it could be corrected. False negative (*type II error*) means the model mistakenly predicts the positive as a negative. For example, the model judged that an enterprise is not defaulted (negative), but the enterprise is the default. This is a severe mistake. In this regard, accuracy, *type I error*, *type II error*, score, and AUC are used in this paper to measure the prediction capability of the model. As shown in Tab. 4, a confusion matrix is used to calculate the evaluation indices.

Table 4 Confusion matrix

	Predicted Positive	Predicted Negative
Actual Positive	True Positive (*TP*)	False Negative (*FN*)
Actual Negative	False Positive (*FP*)	True Negative (*TN*)

The formula of the accuracy, *Type I error, Type II error*, score, and AUC are shown in Eq.(26), Eq. (27), Eq. (28), Eq. (31) and Eq. (32).

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{26}$$

$$Type\ I\ error = \frac{FP}{FP + TN} \tag{27}$$

$$Type\ II\ error = \frac{FN}{TP + FN} \tag{28}$$

$$Recall = \frac{TP}{TP + FN} \tag{29}$$

$$Precision = \frac{TP}{TP + FP} \tag{30}$$

$$F = \frac{1 \times Precision \times Recall}{Precision + Recall} \tag{31}$$

$$AUC = \frac{1}{2}\left(1 + \frac{TP}{TP + FN} - \frac{FP}{FP + TN}\right) \tag{32}$$

3.4 Result Analysis

To measure and compare the performance of machine learning models in meta path features, six commonly used machine learning classification algorithms, namely SVM, NN, K-NN, RF, AdaBoost, and GBDT, are used for comparative analysis. The result is compared with the most popular credit scoring model, logistic regression, and an upgraded credit risk measuring model (G-LR) based on logistic regression, which is proposed in [34]. Each set of experiments was performed 50 times, and the average of each result was taken as the final result. Tab. 5 to Tab. 7 shows each method's classification performance for each dataset, with the best result in bold. Fig. 1 to Fig. 5 shows the average accuracy, type I error, type II error, average *F*-measure, and average *AUC* of all the comparison models.

Table 5 Comparison of machine learning model classification ability for Naive MP Features

Dataset	Classifier	*Accuracy*	*Type I error*	*Type II error*	*F*	*AUC*
GEM	LR	73.98	14.65	20.83	54.57	83
	G-LR	73.69	13.70	19.61	52.34	83.21
	SVM	**76.25**	11.53	18.47	**57.34**	85.37
	NN	73.55	14.72	20.77	56.74	82.54
	K-NN	67.23	18.9	25.43	50.12	74.1
	RF	69.89	11.32	15.43	51.28	76.36
	AdaBoost	72.11	5.89	14.76	51.98	88.46
	GBDT	75.82	**4.03**	**12.65**	56.27	**90.51**
STAR	LR	75.02	7.84	20.02	54.63	76.28
	G-LR	76.32	6.25	18.02	54.60	73.65
	SVM	**78.05**	5.73	20.2	**56.24**	79.01
	NN	72.67	10.54	23.76	55.23	77.38
	K-NN	65.52	12.46	25.21	45.65	75.1
	RF	67.41	4.85	19.76	48.42	67.35
	AdaBoost	70.85	4.23	15.21	50.81	83.1
	GBDT	75.42	**4.08**	**13.81**	52.1	**86.75**
SMB	LR	75.63	13.01	25.62	49.73	78.7
	G-LR	74.77	12.80	23.90	52.32	78.65
	SVM	**79.65**	12.33	24.71	56.54	82.38
	NN	75.55	15.74	24.36	55.48	80.91
	K-NN	70.2	17.28	22.54	42.38	74.85
	RF	72.86	10.54	19.38	43.25	75.76
	AdaBoost	73.64	8.91	16.25	46.53	84.67
	GBDT	78.9	**6.12**	**12.23**	**59.46**	**87.52**

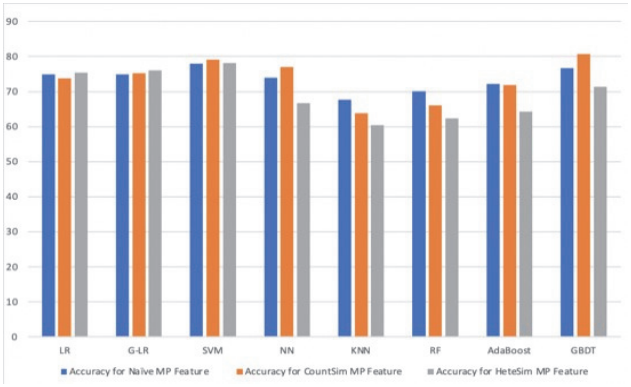

Figure 1 The average accuracy of comparison models

Table 6 Comparison of machine learning model classification ability for CountSim MP Features

Dataset	Classifier	*Accuracy*	*Type I error*	*Type II error*	*F*	*AUC*
GEM	LR	73.76	5.64	13.52	58.37	83.1
	G-LR	74.32	6.76	12.21	60.32	84.98
	SVM	78.12	13.25	19.12	62.62	90.41
	NN	76.95	16.83	25.41	61.9	87.7
	K-NN	63.72	19.74	26.73	54.91	70.38
	RF	65.36	9.52	16.76	52.73	85.42
	AdaBoost	72.46	7.67	11.41	55.47	88.7
	GBDT	**78.2**	**3.16**	**10.17**	**62.73**	**92.78**
STAR	LR	70.47	6.44	17.41	48.99	73.97
	G-LR	72.32	7.03	18.62	48.67	73.66
	SVM	75.96	15.28	24.93	55.76	84.91
	NN	75.28	19.08	25.32	52.31	82.2
	K-NN	62.24	19.43	28.43	43.7	73.26
	RF	65.37	9.39	22.71	42.75	80.4
	AdaBoost	65.37	7.56	19.18	48.62	82.75
	GBDT	**80.48**	**6.23**	**16.71**	**56.6**	**87.42**
SMB	LR	77.14	4.88	16.42	42.45	77.11
	G-LR	78.98	4.67	16.28	48.70	78.23
	SVM	83.43	17.96	24.65	65.55	90.03
	NN	78.65	17.83	24.92	65.03	84.34
	K-NN	65.42	20.38	30.47	58.19	72.36
	RF	67.7	9.72	23.17	55.73	83.65
	AdaBoost	77.6	5.63	16.8	64.91	85.19
	GBDT	**83.5**	**4.37**	**15.38**	**65.72**	**90.2**

Table 7 Comparison of machine learning model classification ability for HeteSim MP Features

Dataset	Classifier	*Accuracy*	*Type I error*	*Type II error*	*F*	*AUC*
GEM	LR	73.98	5.21	9.81	**64.79**	84.47
	G-LR	73.76	5.66	8.96	65.02	84.59
	SVM	**78.42**	**3.15**	**8.61**	65.38	90.47
	NN	68.25	10.27	16.58	61.71	76.72
	K-NN	58.79	21.63	22.71	57.79	72.08
	RF	61.6	8.65	11.93	55.08	75.35
	AdaBoost	62.86	18.18	21.15	60.01	87.47
	GBDT	71.84	9.32	17.31	63.74	**92.5**
STAR	LR	73.86	2.87	12.81	67.15	86.72
	G-LR	74.28	2.87	13.01	67.32	85.26
	SVM	**75.62**	**2.81**	**10.7**	**67.52**	89.72
	NN	70.08	11.93	19.5	62.47	79.13
	K-NN	63.98	23.6	23.47	57.02	64.78
	RF	65.29	8.6	13.7	48.36	78.16
	AdaBoost	68.63	17.52	19.58	57.17	87.37
	GBDT	11.93	8.39	17.41	62.73	**93.4**
SMB	LR	78.28	8.18	16.2	69.34	86.27
	G-LR	80.02	8.03	15.23	68.98	84.32
	SVM	**80.51**	**7.43**	**13.41**	**70.22**	90.56
	NN	61.86	10.08	17.46	67.57	76.64
	K-NN	58.73	18.41	21.25	60.25	68.45
	RF	60.03	8.77	16.33	52.09	72.37
	AdaBoost	61.27	13.65	20.7	67.37	82.78
	GBDT	70.51	9.79	17.03	69.08	**92.83**

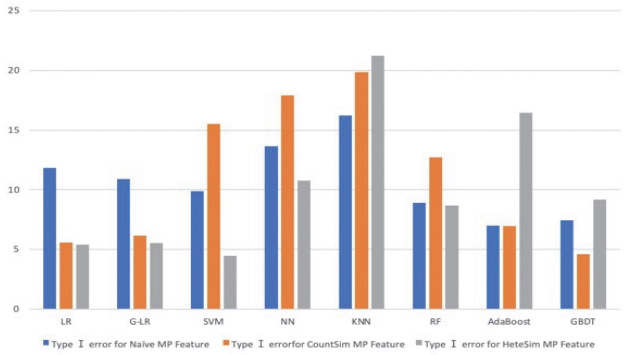

Figure 2 The average type I error of comparison models

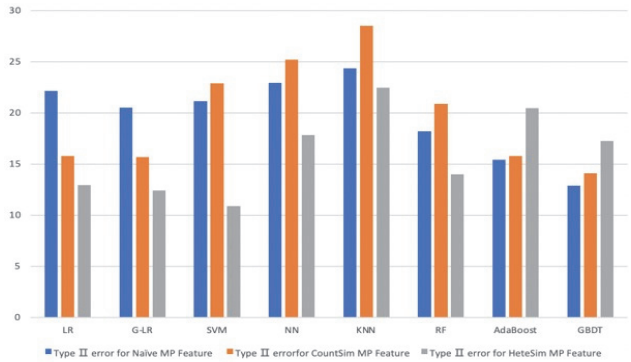

Figure 3 The average type II error of comparison models

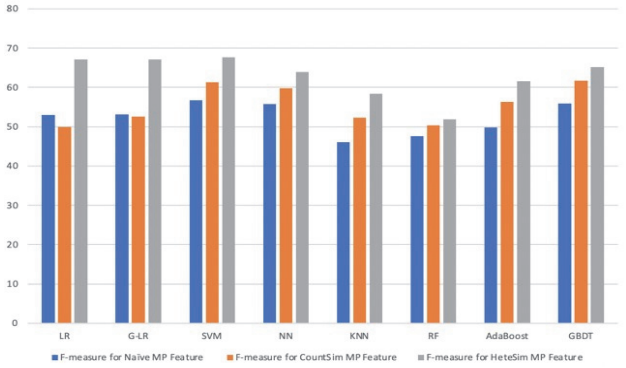

Figure 4 The average *F*-measure of comparison models

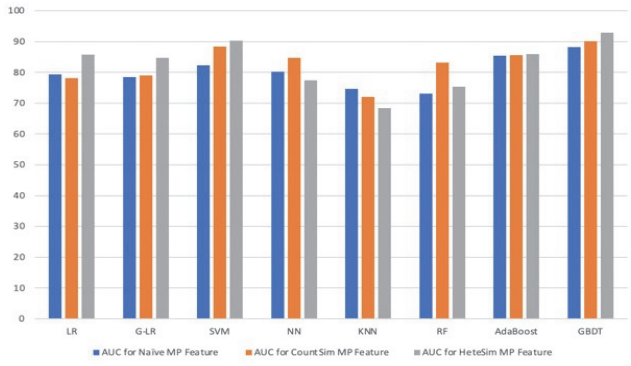

Figure 5 The average *AUC* score accuracy of comparison models

For the Naive meta path feature,SVM has the highest average accuracy of 77.98, followed by GBDT with an accuracy of 76.71, G-LR with 74.93, LR with 74.88, NN with 73.92, AdaBoost with 72.20, RF with 70.05, and KNN with 67.75; AdaBoost has the lowest average *type I error* of 7.01, followed by GBDT with errors of 7.43, RF with 8.90, SVM with 9.86, G-LR with 10.92, LR with 11.83, NN with 13.67, and KNN with 16.21; GBDT has the lowest average typeII error of 12.89, followed by AdaBoost with errors of 15.41, RF with 18.19, G-LR with 20.51, SVM with 21.13, LR with 22.16, NN with 22.96, and KNN with 24.39; GBDT has the highest average *F*-measure of 55.94, followed by NN with average *F*-measure of 55.82, SVM with 56.71, G-LR with 53.09, LR with 52.94, AdaBoost with 49.77, RF with 47.64, and KNN with 46.05; GBDT has the highest average *AUC* of 88.26, followed by AdaBoost with average *AUC* of 85.41, SVM with 82.25, NN with 80.28, LR with 79.33, G-LR with 78.5, RF with 74.68, and KNN with 73.16.

For the CountSim meta path feature, GBDT has the highest average accuracy of 80.73, followed by SVM with an accuracy of 79.17, NN with 76.96, G-LR with 75.2, LR with 73.79, AdaBoost with 71.81, RF with 66.14, and KNN with 63.79; GBDT has the lowest average *type I error* and *type II error* of 4.59 and 14.09, followed by LR with errors of 5.65 and 15.79, AdaBoost with 6.95 and 15.80, G-LR with 6.15 and 15.7, RF with 12.72 and 20.88, SVM with 15.50 and 22.90, NN with 17.91 and 25.22, and KNN with 19.85 and 28.54; GBDT has the highest average *F*-measure of 61.68, followed by SVM with average *F*-measure of 61.30, NN with 59.75, AdaBoost with 56.33, G-LR with 52.56, KNN with52.27, and RF with 50.40, LR with 49.94; GBDT has the highest average *AUC* of 90.13, followed by SVM with *AUC* of 88.45, AdaBoost with 85.55, NN with 84.75, RF with 83.16, G-LR with 78.96, LR with 78.06, and KNN with 72.

For the HeteSim meta path feature, SVM has the highest average accuracy of 78.18, followed by G-LR with an accuracy 76.02, LR with75.37, GBDT with 71.43, NN with 66.73, AdaBoost with 64.25, RF with 62.31, and KNN with 60.5; SVM has the lowest average *type I error* and *type II error* of 4.46 and 10.91, followed by LR with errors of 5.42 and 12.94, G-LR with 5.52 and 12.4, RF with 8.67 and 14, GBDT with 9.17 and 17.25, NN with 10.76 and 17.85, AdaBoost with 16.45 and 20.48, and KNN with 21.21 and 22.48; SVM has the highest average *F*-measure of 67.71, followed by G-LR with average *F*-measure of 67.11, LR with 67.09, GBDT with 65.18, NN with 63.92, AdaBoost with 61.52, KNN with 58.35, and RF with 51.84; GBDT has the highest average *AUC* of 92.91, followed by SVM with average *AUC* of 90.25, AdaBoost with 85.87, LR with 85.82, G-LR with 84.72, NN with 77.50, RF with 75.29, and KNN with 68.44.

Ensemble machine learning algorithms (Adaboost, GBDT) outperform single classifiers. Support vector machines and neural networks are the best among the single classifiers. While logistic regression is slightly worse than support vector machines and neural networks, the prediction is also satisfactory. Compare the logistic regression, and the logistic regression algorithm with a feature pre-selection step method, the method with pre-selection steps has better performance. For the three meta path features, the HeteSim meta path feature performs slightly better than the CountSim meta path feature and better than the Native meta path feature. This also verifies what has been proposed in [25] that features with weights taken into account perform better than those constructed with equal weights. For model selection, in general, logistic regression, neural networks, support vector machines, and GBDT perform better on meta path features and can be considered potential classifiers. More specifically, for Naive meta path features and HeteSim

meta path features, the GBDT model outperforms all other models for all evaluation indices, while for CountSim meta path features SVM model performs the best. The advantage of machine learning models is that they can automatically handle interactions between features and capture nonlinear relationships well. Logistic regression models contain theoretical foundations in statistics. We can run statistical tests on the features to decide whether to include them in the credit scoring system. In addition, the credit score of an enterprise needs to be given a corresponding score justification, and the interpretation of the model has to be considered. The interpretability of logistic regression models is strong. However, the interpretability of machine learning models, such as support vector machines and neural networks, is problematic. However, we can also extract some rules from these methods to improve the interpretability of the model.

4 CONCLUSION

This paper compared the performance of six classic machine learning classification algorithms in terms of meta path features. Conclusions of this study include: (1) machine learning algorithms showed the best performance in terms of meta path features and the logistic regression that is commonly used. (2) Machine learning algorithms had different effects on different features. (3) Generally speaking, logistic regression, neural networks, support vector machines, and GBDT showed better performance in terms of meta path features. With the development and broad application of machine learning models, the field of SME credit risk measurement would also witness an expanding application of machine learning models. This study has expanded the application of machine learning in risk measurement and laid a foundation for studies of machine learning based on path association indicators. SME credit risk measurement is under stable development. Despite classic classifier learning algorithms, many ensemble machine learning methods exist. Our next research direction would be how to use more complex and effective machine learning models.

5 REFERENCES

[1] Ohlson, J. A. (1980). Financial ratios and the probabilistic prediction of bankruptcy. *Journal of accounting research*, 109-131. https://doi.org/10.2307/2490395

[2] Griffin, J. M. & Lemmon, M. L. (2002). Book-to-market equity, distress risk, and stock returns. *The Journal of Finance*, *57*(5), 2317-2336. https://doi.org/10.1111/1540-6261.00497

[3] Chen, J., Chollete, L., & Ray, R. (2010). Financial distress and idiosyncratic volatility: An empirical investigation. *Journal of Financial Markets*, *13*(2), 249-267. https://doi.org/10.1016/j.finmar.2009.10.003

[4] Mitchell, T. M. & Mitchell, T. M. (1997). New York: McGraw-hill. *Machine learning*, *1*(9).

[5] Cortes, C. & Vapnik, V. (1995). Support-vector networks. *Machine learning*, *20*(3), 273-297. https://doi.org/10.1007/BF00994018

[6] Breiman L. (1996). Bagging predictors. *Machine learning*, *24*(2), 123-140. https://doi.org/10.1007/BF00058655

[7] Schapire, R. E. (1999). A brief introduction to boosting. *Ijcai*, 1401-1406.

[8] Zhang, D., Peng, Y., Zhang, Y., Wu, D., Wang, H., & Zhang, H. (2021). Train time delay prediction for high-speed train dispatching based on spatio-temporal graph convolutional network. *IEEE Transactions on Intelligent Transportation Systems*, *23*(3), 2434-2444. https://doi.org/10.1109/TITS.2021.3097064

[9] Zhang, D., Xu, Y., Peng, Y., Du, C., Wang, N., Tang, M., & Liu, J. (2022). An Interpretable Station Delay Prediction Model Based on Graph Community Neural Network and Time-Series Fuzzy Decision Tree. *IEEE Transactions on Fuzzy Systems.* https://doi.org/10.1109/TFUZZ.2022.3181453

[10] Nanni, L. & Lumini, A. (2009). An experimental comparison of ensemble of classifiers for bankruptcy prediction and credit scoring. *Expert systems with applications*, *36*(2), 3028-3033. https://doi.org/10.1016/j.eswa.2008.01.018

[11] Ciampi, F. & Gordini, N. (2013). Small Enterprise Default Prediction Modeling through Artificial Neural Networks: An Empirical Analysis of Italian Small Enterprises. *Journal of Small Business Management*, *51*(1), 23-45. https://doi.org/10.1111/j.1540-627X.2012.00376.x

[12] Tobback, E., Bellotti, T., Moeyersoms, J., Stankova, M., & Martens, D. (2017). Bankruptcy prediction for SMEs using relational data. *Decision Support Systems*, *102*, 69-81.

[13] Marqués, A. I., García, V., & Sánchez, J. S. (2012). Exploring the behaviour of base classifiers in credit scoring ensembles. *Expert systems with Applications*, *39*(11), 10244-10250. https://doi.org/10.1016/j.eswa.2012.02.092

[14] Baesens, B., Mues, C., Setiono, R., De Backer, M., & Vanthienen, J. (2003). Building intelligent credit scoring systems using decision tables. https://doi.org/10.1007/1-4020-2673-0_15

[15] Altman, E. I. (1968). Financial ratios, discriminant analysis and the prediction of corporate bankruptcy. *The journal of finance*, *23*(4), 589-609. https://doi.org/10.1111/j.1540-6261.1968.tb00843.x

[16] Chaffe-Stengel, P. & Stengel, D. (2012). Working with sample data: Exploration and inference. https://doi.org/10.4128/9781606492147

[17] Gupta, J. & Gregoriou, A. (2018). Impact of market-based finance on SMEs failure. *Economic Modelling*, *69*, 13-25. https://doi.org/10.1016/j.econmod.2017.09.004

[18] Psillaki, M., Tsolas, I. E., & Margaritis, D. (2010). Evaluation of credit risk based on firm performance. *European journal of operational research*, *201*(3), 873-881. https://doi.org/10.1016/j.ejor.2009.03.032

[19] Gupta, M., Kumar, P., & Bhasker, B. (2017). HeteClass: A Meta-path based framework for transductive classification of objects in heterogeneous information networks. *Expert Systems with Applications*, *68*, 106-122. https://doi.org/10.1016/j.eswa.2016.10.013

[20] Du, M., Ma, Y., & Zhang, Z. (2022). A Path-Based Feature Selection Algorithm for Enterprise Credit Risk Evaluation. *Computational Intelligence and Neuroscience.* https://doi.org/10.1155/2022/7650207

[21] Letizia, E. & Lillo, F. (2019). Corporate payments networks and credit risk rating. *EPJ Data Science*, *8*(1), 21. https://doi.org/10.1140/epjds/s13688-019-0197-5

[22] Moro, A. & Fink, M. (2013). Loan managers' trust and credit access for SMEs. *Journal of banking & finance*, *37*(3), 927-936. https://doi.org/10.1016/j.jbankfin.2012.10.023

[23] Tobback, E., Bellotti, T., Moeyersoms, J., Stankova, M., & Martens, D. (2017). Bankruptcy prediction for SMEs using relational data. *Decision Support Systems*, *102*, 69-81. https://doi.org/10.1016/j.dss.2017.07.004

[24] Kou, G., Xu, Y., Peng, Y., Shen, F., Chen, Y., Chang, K., & Kou, S. (2021). Bankruptcy prediction for SMEs using transactional data and two-stage multiobjective feature selection. *Decision Support Systems*, *140*, 113429. https://doi.org/10.1016/j.dss.2020.113429

[25] Du, M., Ma, Y., & Zhang, Z. (2021). A Meta-Path-Based Evaluation Method for Enterprise Credit Risk. *Computational Intelligence and Neuroscience.* https://doi.org/10.1155/2022/1783445
[26] Fisher, R. A. (1936). The use of multiple measurements in taxonomic problems. *Annals of eugenics*, *7*(2), 179-188. https://doi.org/10.1111/j.1469-1809.1936.tb02137.x
[27] Durand, D. (1941). Front matter, acknowledgments, table of contents. *Risk Elements in Consumer Instalment Financing,* 12-0.
[28] Altman, E. I. (1968). Financial ratios, discriminant analysis and the prediction of corporate bankruptcy. *The journal of finance*, *23*(4), 589-609. https://doi.org/10.1111/j.1540-6261.1968.tb00843.x
[29] Fix, E. (1985). Discriminatory analysis: nonparametric discrimination, consistency properties. *USAF school of Aviation Medicine*, *1*.
[30] Henley, W. & Hand, D. J. (1996). AK-Nearest-Neighbour Classifier for Assessing Consumer Credit Risk. *Journal of the Royal Statistical Society: Series D (The Statistician)*, *45*(1), 77-95. https://doi.org/10.2307/2348414
[31] Freund, Y. & Schapire, R. E. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences*, *55*(1), 119-139. https://doi.org/10.1006/jcss.1997.1504
[32] Friedman, J. H. (2001). Greedy Function Approximation: A Gradient Boosting Machine. *The Annals of Statistics*, *29*(5), 1189-1232. https://doi.org/10.1214/aos/1013203451
[33] Breiman, L. (2001). Random forests. *Machine learning*, *45*(1), 5-32. https://doi.org/10.1023/A:1010933404324
[34] Li, D., Ding, T., & Liu, C. (2021). Credit Risk Management of P2P Network Lending. *Tehnički vjesnik*, *28*(4), 1145-1151. https://doi.org/10.17559/TV-20200210110508

Contact information:

Marui DU
(Corresponding author)
School of Mathematics and Statistics,
Beijing Jiaotong University, China
E-mail: 17118446@bjtu.edu.cn

Zuoquan ZHANG
School of Mathematics and Statistics,
Beijing Jiaotong University, China