An Evaluation Model for College Student 's Loan Repayment Ability based on NNN

Yanghai Fan, Ming Cheng and Lijun Yang*
College of computer science and technology, Southwest Minzu University, Chengdu, China
f0122@vip.qq.com

Abstract. This paper designs a risk model suitable for student loans. It is proposed to use NNN algorithm assess the student loan repayment ability. By establishing an evaluation index system, the student's relevant indicator data is compared with the evaluation index standard, and the closeness value of the evaluation index is calculated, and finally the evaluation conclusion is obtained. It provides a reference for the final establishment and improvement of the student loan system in China.

1. Introduction
A loan is a form of credit activity in which a bank or other financial institution lends monetary funds at a certain interest rate and must be returned. At the same time, it has strong benefits, which alleviates the lag of production and consumption to a certain extent, and improves the consumer's propensity to consume. [1] College students' consumer credit has entered the exploration period. In the short period of one or two years, the main mode of welcoming major e-commerce and capital layouts is to provide a lot of campus loans by providing installment loans or small cash loans for college students.[2]

As a classic statistical pattern recognition method, KNN is also one of the best classification methods, and the KNN method mainly relies on the surrounding limited samples, rather than relying on the method of discriminating the class domain to determine the category, so for the class domain For large data with overlapping or overlapping, the KNN method is more suitable than other methods.[3]

But the KNN algorithm is a lazy learning method. It does not establish a classification model in advance, but simply saves all training samples until a new sample needs to be classified. [4]

Although the K-nearest neighbor algorithm is simple and effective, the determination of the K value is the most important step in KNN. The choice of K value has a great influence on the result. The data affects the choice of K value. When the value of K is large, the influence of noise can be reduced, but the boundary between different classes may be less obvious. When the value of K is small, the data point may be Stripped in the original class.
In order to solve this problem, the natural neighbor method was adopted in the construction of the model. This is an algorithm that does not require the parameter K and can reduce the number of noise points. At the same time, it can improve the accuracy of classification and reduce the error.

2. Model overview
This paper mainly uses Principal Component Analysis and NNN algorithm to correctly classify and classify students' loan ability of different students.

3. Principal Component Analysis
principle: Hypothesis Random vector $x' = (x_1 + x_2 + \ldots + x_p)$ Correlation coefficient matrix $R$ (Covariance matrix $\Sigma$) $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_p$ is Reigenvectors $e_1, e_2, \ldots, e_p$ is the corresponding standard orthogonal eigenvector. The $i$-th principal component is $y_i = e_i' x = e_{i1} x_1 + e_{i2} x_2 + \ldots + e_{ip} x_p$, $i = 1, 2, \ldots, p$ while $\text{Var}(y_i) = e_i'R e_i = \lambda_i, i = 1, 2, \ldots, p$ $\text{Cov}(y_i, y_k) = e_i'R e_k = 0, i \neq k$ If some $\lambda_i$ have a repeated root, then the coefficient vectors $e_i$ and $y_i$ are not unique. Normalized orthogonal eigenvectors $e_1, e_2, \ldots, e_p$ always exist. Actually, if the eigenvalues $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_p$ are not equal, then $e_i$ is naturally orthogonal. If there are some eigenvalues with repeated roots, it is also possible to select the eigenvectors corresponding to these eigenvalues so that they are orthogonal. The principal component of $x_1, x_2, \ldots, x_p$ is a linear combination of $R$'s eigenvectors as coefficients, they are not related to each other, and their variance is the eigenvalue of $R$.

Let the ratio of the variance of the $k$th principal component to the total variance be $p_k$

$$
p_k = \frac{\lambda_k}{\sum_{i=1}^{p} \lambda_i} \quad (1)
$$

From formula (1) we can proceed with the following steps. When the number of variables $p$ is large, if the sum of the variances of the first few principal components accounts for a large part of the total variance (such as 85% or more), replacing the original $P$ variables with these principal components will not lose too much information. Coefficient vector $e_i' = (e_{i1}, e_{i2}, \ldots, e_{ip})$ The component of $e_i' = (e_{i1}, e_{i2}, \ldots, e_{ip})$ also has a certain meaning. $e_{ki}$ characterizes the importance of the $k$th variable to the $i$th principal component. The principal component is calculated as formula (2)

$$
\begin{align*}
    y_1 &= e_{i1} x_1 + e_{i2} x_2 + \ldots + e_{im} x_m \\
    y_2 &= e_{i1} x_1 + e_{i2} x_2 + \ldots + e_{m} x_m \\
    \vdots \\
    y_1 &= e_{pi} x_1 + e_{p2} x_2 + \ldots + e_{pm} x_m
\end{align*}
\quad (2)
$$

4. KNN algorithm

4.1 Principal
In the KNN algorithm, the distance between samples is calculated based on all the characteristics of the sample. However, because the importance of each feature is different, if the less important feature is overused, it will mislead the classification result, and the KNN algorithm is very sensitive to this situation. For this, we use the Manhattan distance formula (3) to calculate the distance between each data, and adjust the weight coefficient of each feature to highlight the difference in importance between the attributes.

$$
y = \sqrt{\sum_{i=1}^{n} (x_{i1} - x_{i2})^2} \quad (3)
$$
4.2 Algorithm

In this paper, we use the KNN algorithm as follows: on the basis of the data and labels already defined in the training set, the student loan will be required to be accurately input as the data to be tested, and finally compared with the standard features given in the training set. Thus, the first k data with the highest phase collection speed are quickly acquired and trained from the training set. The category corresponding to the test data will appear most frequently in the k data. In the algorithm, it is first necessary to accurately calculate the specific distance value of the test data from each training data, and then determine the k points with the smallest distance value according to the distance increasing relationship. At this time, after validating the category corresponding to the k points with the smallest distance value and the number of occurrences thereof, returning to the first k points, and selecting the category with the most occurrences, the category is the prediction test data type.

KNN algorithm

```
function KNN(K,A)
figure();
hold on
[N,dim]=size(A);
x=A(:,1);
y=A(:,2);
dist=zeros(N,N);
for i=1:N
    for j=1:N
        dist(i,j)=sqrt((x(i)-x(j))^2+(y(i)-y(j))^2);
    end
end
[sdist,index]=sort(dist,2);
```

In this paper, we calculate the scores of each student's loan quota, loan credit, and repayment ability through certain rules, and get the training set and the data of the test set. By calculating the distance between each student in the set T to be tested and all the training data in the training data set S1 and the test set data set S2, and sequentially sorting according to the calculated distance, k nearest neighbors are found, corresponding to k neighbors. The results are merged, sorted, and returned to the most frequently occurring classification result. In view of the characteristics of the KNN algorithm without training process, this paper innovatively introduces a training process, that is, firstly, the linear complexity clustering method is used to block the big data. In the test process, for each test sample, first find the cluster with the cluster center closest to the sample to be tested as the new training sample set, and then classify the new training set to classify the model.[5]

The KNN algorithm is a typical passive learning method. Only all training instances are stored in the training phase, and all calculations are delayed until the classification phase. Therefore, it constructs a classification model for each instance to be classified, once for modeling. In contrast, active learning methods complete the modeling process during the training phase. Therefore, it builds a classification model for all instances to be classified, and one model is used repeatedly.[6]
5. NNN algorithm

5.1 Principal

Although the KNN algorithm is simple and effective, the noise points and other interferences still have some influence on the final information analysis. In order to solve these problems, we adopt the natural neighbor algorithm.

When it is necessary to determine whether the two points X and Y are natural neighbors, it is only necessary to traverse the neighbor list of X to confirm whether Y exists and traverse the neighbor list X of Y. When both points consider that the other party is its own neighbor, it corresponds to Natural neighbors, so that the calculation process of the number of neighbors per point does not require parameters, and can be generated by algorithm adaptive calculation. This results in a densely distributed data object having more natural neighbors than a sparsely distributed natural neighbor, and the degree of density of the data object is determined by the number of nearest neighbors covering that point.

That is, there are four points A, B, C, and D. A regards B as his neighbor, B as C as his neighbor, C as B as his neighbor, and D as A as his neighbor. A is treated as a neighbor once, B is treated as a neighbor 2 times, C is treated as a neighbor once, and D is treated as a neighbor 0 times, where B treats C as his neighbor and C treats B as his neighbor. Here, it is called B and C are natural neighbors.

5.2 Algorithm

Algorithm: Natural neighbor search algorithm

Input: Data Set X

Output: NaN_Edge, NaN_Num(x_i)

1: r=1, flag=0, NaN_Edge=∅
2: Create a k-d tree $\tilde{T}$ from data set X
3: $\forall x_i \in X, \text{NaN}_\text{Num}(x_i) = 0$
4: While flag=0 do
5: for all $x_i \in X$ do
6: $\text{KNN}(x_i) = \text{findKNN}(x_i,r,T)$
7: $\text{KNN}_r(x_i) = \text{KNN}_r(x_i) \cup \{\text{KNN}_r(x_i)\}$
8: if $x_i \in \text{KNN}_r(x_i) \land \{\text{KNN}_r(x_i), x_i\} \notin \text{NaN}_\text{Edge}$ then
9: $\text{NaN}_\text{Edge} = \text{NaN}_\text{Edge} \cup \{\text{KNN}_r(x_i), x_i\}$
10: $\text{NaN}_\text{Num}(x_i) = \text{NaN}_\text{Num}(x_i) + 1$
11: $\text{NaN}_\text{Num}(\text{knn}_r(x_i)) = \text{NaN}_\text{Num}(\text{knn}_r(x_i)) + 1$
12: end if
13: end for
14: cnt = count($\text{NaN}_\text{Num}(x_i) = 0$)
15: rep = repeat(cnt)
16: if all ($\text{NaN}_\text{Num}(x_i) \neq 0 \lor \text{rep} \geq \sqrt{r-re\text{peat}}$) then
17: flag = 1
18: end if
19: r = r - 1
20: end while
21: λ = r - 1
22: Return: λ, NaN_Edge, NaN_Num(x_i)

Where NaN_Num is the number of natural neighbors of each point in the set X. NaN_Edge is a natural neighbor edge set. find KNN(x_i, r, T) is the rth nearest neighbor of the i-th point, repeat(cnt) records the number of times the variable cnt is repeated.

6. Result
Before the principal component analysis method, in order to determine whether the extracted factors can explain the original data well, SPSS23 is used for analysis, and the following total variance interpretation table is obtained. When the dimensionality reduction is reduced to eight dimensions, the cumulative square sum reaches 91.26%, which can explain the original data well, can reduce the dimensionality of the data. The result of the operation is shown in Table 1 of the following table.

In the process of using KNN algorithm, we use the cross-validation method to select the optimal k value. For each k value 1-10, use the verification set to calculate the number of errors corresponding to k, and take the average k. The value, K=0.909090909, is implemented by NNN algorithm correspondingly. The accuracy of NNN algorithm is k=0.964. By comparison, it can be found that the accuracy of using NNN algorithm is higher in the process of classification and clustering. The result is shown in Fig. 1.

![Algorithm accuracy](image)

Fig. 1 Algorithm accuracy

7. Conclusions
This paper provides a repayment ability framework model, which aims to construct a scoring system for student loan repayment ability through student credit score, loan quota, repayment time and so on. This paper provides a reference for students' repayment ability and provides a reference for student loan program design. In this paper, NNN and KNN are used to obtain repayment ability, and the NNN algorithm has great advantages for data acquisition, analysis and prediction. And the more data, the more machine learning samples, the more the test results will be closer to the true value.
Table 1. Total variance interpretation

|     | Initial eigenvalue | Extracting the sum of squared loads | Sum of squared rotational loads |
|-----|--------------------|-------------------------------------|---------------------------------|
|     | total percentage of variance | total percentage of variance | total percentage of variance |
| 1   | 2.378               | 23.779                             | 2.378                           |
| 2   | 1.890               | 18.896                             | 1.890                           |
| 3   | 1.064               | 10.455                             | 1.064                           |
| 4   | .997                | 9.973                              | .997                            |
| 5   | .886                | 8.861                              | .886                            |
| 6   | .666                | 6.661                              | .666                            |
| 7   | .643                | 6.433                              | .643                            |
| 8   | .618                | 6.177                              | .618                            |
| 9   | .484                | 4.482                              | .484                            |
| 10  | .392                | 3.922                              | 100.00                          |

method of extraction : Principal Component Analysis

8. Reference

[1] Fan Ruizhi. "Student loan risk model based on KNN algorithm" [J]. *Electronic test*, 2019 (Z1): 80-81+84.

[2] Zheng Chunmei, Jia Shanshan. "Comparison and Regulation of Campus Credit Platforms at Home and Abroad" [J]. *Finance and Economics (Academic Edition)*, 2016 (17): 349-351.

[3] Gen Lijuan, Li Xingyi. "Research on KNN algorithm for big data classification" [J]. *Application Research of Computers*, 2014, 31(05): 1342-1344+1373.

[4] Feng Guohe, "Wu Jingxue. Research Progress of KNN Classification Algorithm Improvement" [J]. *Library and Information Service*, 2012, 56(21): 97-100+118.

[5] Su Yijuan, Deng Zhenyun, Cheng Debo, Zong Ming. "Fast KNN classification algorithm under big data" [J]. *Journal of Computer Applications*, 2016, 33(04): 1003-1006+1023.

[6] LU Weiwei, LIU Jing. "A new algorithm for improving the efficiency of K-nearest neighbor algorithm" [J]. *Computer Engineering and Applications*, 2008(04): 163-165+178.

Acknowledgements

These authors contributed equally to this work.