

assignment2

by Jingpei Li

Submission date: 29-Apr-2023 06:25AM (UTC+0100)

Submission ID: 203968675

File name: 41273_Jingpei_Li_assignment2_493224_328825378.docx (1.29M)

Word count: 3412

Character count: 18792

1 Data Set Description

The data set mentioned in this article is selected from the Alibaba Tianchi competition data set. The task of the competition is to predict whether the user's loan defaults. The financial risk control data set mainly refers to the data set used for risk management and credit evaluation in the financial industry. These data usually include customers' personal information, loan records, transaction records, etc., as well as other factors that may have an impact on risk and credit assessment, such as balance sheet, income status, work experience, education, etc. In addition, the financial risk control data set may also contain some warning signs for fraud and improper transactions, such as abnormal transaction patterns, fraudulent behavior, etc., to help financial institutions better protect themselves and the interests of customers.

The financial risk control data set can be used as an important data resource to help financial institutions carry out risk management and credit evaluation, and can provide rich information to support decision-making and risk control. The data comes from the loan records of a credit platform. The total data volume exceeds 5w, including 47 columns of variable information, of which 15 Listed as anonymous variables. This dataset desensitizes information such as employmentTitle, purpose, postCode, and title. The data set is from TIANCHI(<https://tianchi.aliyun.com/competition/entrance/531830/information>).

The letter recognition data set contains a total of 47 columns, and the specific descriptions of each attribute are as follows:

- id: the unique letter of credit identifier assigned to the loan list, numerical type.
- loanAmnt: loan amount, numerical type.
- term: loan term (year), nominal type.
- interestRate: loan interest rate, numerical type.
- installment: installment payment amount, numerical type.
- grade: loan grade, numerical type.
- subgrade: sub-grade of loan grade, numerical type.
- employmentTitle: employment title, nominal type.
- employmentLength: employment length (years), numerical type.
- homeownership: the home ownership status provided by the borrower at the time of registration, nominal type.
- purpose: the type of loan purpose of the borrower at the time of loan application, nominal type.
- postCode The first 3 digits of the postal code provided by the borrower in the loan application, numerical type.
- issueDate: the month the loan was issued, numerical type.
- regionCode: area code, numerical type.
- ficoRangeHigh: the upper limit range of the borrower's fico at the time of loan issuance, numerical type.

- openAcc: The number of open credit lines in borrower's credit file, numerical type.
- pubRec: the number of deprecated public records, numerical type.
- pubRecBankruptcies: number of public record purges, numerical type.
- revolBal: total credit revolving balance, nominal type.
- interestRate: loan interest rate, numerical type.
- installment: installment payment amount, numerical type.
- grade: loan grade, numerical type.
- subgrade: sub-grade of loan grade, numerical type.
- employmentTitle: employment title, nominal type.
- initialListStatus: the initial list status of the loan, numerical type.
- applicationType: indicates whether the loan is an individual, nominal type.
- purpose: the type of loan purpose of the borrower at the time of loan application, nominal type.
- postCode: The first 3 digits of the postal code provided by the borrower in the loan application, numerical type.
- n-series anonymous features: anonymous features n0-n14, processing for counting features for some lender behavior, numerical type.

2 Data Pre-processing

Data preprocessing is a very important step in machine learning and data mining. It refers to the process of cleaning, converting, integrating, and reducing raw data to better meet the needs of machine learning algorithms. Data preprocessing can improve data quality: the original data may have various problems such as outliers, missing values, noise, etc. After preprocessing, these problems can be removed and the quality of the data can be improved. Through statistical analysis and visual analysis of the data, hidden patterns and laws in the data can be identified, and a reference can be provided for subsequent modeling. When training the model, data preprocessing can reduce the redundant information of the data and prevent the model from overfitting the training data, thereby improving the generalization ability of the model. For some large-scale or high-dimensional data sets, preprocessing can also reduce the complexity of the data and speed up the operation of the algorithm, thereby improving the efficiency and performance of the model.

In addition, by performing reasonable feature selection, feature scaling, and other operations on the data, the accuracy of the model can be improved to achieve better prediction results.

2.1 Data Cleaning

Data cleaning is part of the data preprocessing process, which refers to removing invalidity, duplication, error, missing and other problems in the original data to ensure

the quality and reliability of the data. The role of data cleaning is mainly to improve data quality. There may be various problems in the original data, such as invalid values, duplicate records, error messages, missing data, etc. These problems seriously affect the quality and validity of the data. Data cleaning can remove these problems, improve the quality of data, and make it more real and reliable. Improve the data structure. In the original data, there may be problems such as irregular format, inconsistent fields, and mismatched data types. These problems will make the data difficult to process and analyze. By cleaning the data, the data structure can be improved to make it more standardized and easier to process. Reduce data errors. In the process of data collection and transmission, various errors may occur, such as human errors, equipment failures, network delays, etc. These errors can be removed through data cleaning, thereby reducing the impact of data errors on model training and analysis. Accelerate the data processing speed. In the big data environment, the data volume is huge and the processing speed is very slow, which affects the efficiency. Data cleaning can reduce redundant data and invalid data, thereby speeding up data processing.

Selected attribute			
Name: employmentLength		Type: Nominal	
Missing: 11742 (6%)		Unique: 0 (0%)	
Distinct: 11			
No.	Label	Count	Weight
1	10+ years	65772	65772
2	2 years	18207	18207
3	4 years	11833	11833
4	< 1 year	15989	15989
5	6 years	9328	9328
6	3 years	16011	16011
7	1 year	13182	13182
8	5 years	12543	12543
9	8 years	8976	8976

Figure 1: Attribute of employmentLength

Through data analysis, we can see that some variables have a large number of missing values. As shown in Figure 1, the employmentLength variable has 11,742 missing values, which will affect the accuracy when making predictions, so we fill in the missing values. Missing value handling There are mainly several ways to remove missing values: If the number of missing values is small, it may be easiest to directly remove these samples or attributes. However, if the proportion of missing values is high, then deletion may lead to problems such as small sample size and underrepresentation of the sample. Imputation, imputation is a method of inferring missing values based on known information. Common imputation methods include mean, median, mode filling, and regression analysis.

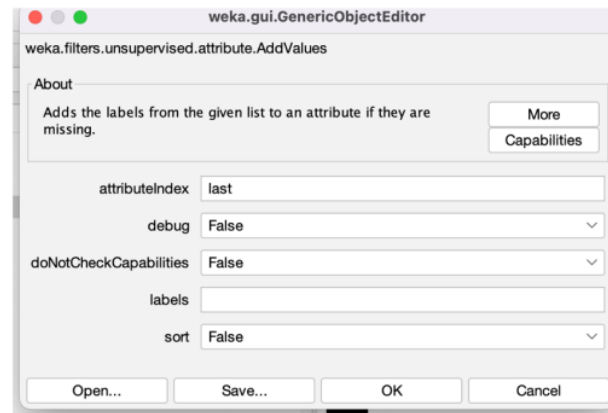


Figure 2. Setting of Replace Missing Values

In addition, there are a lot of sparse data in the data set as shown in the figure follow. For this kind of data, I use the method of direct deletion. The role of deleting sparse data in data preprocessing is to reduce the noise and redundant information of the data set, and improve the training speed and accuracy of the model. Sparse data refers to a large number of features that are missing or have very few values in the data. If these features are retained, it will lead to problems such as increased training time and overfitting. Therefore, we usually perform sparse data deletion to optimize the data set and only keep the features that are useful to the model, thereby improving the efficiency and predictive ability of the model. At the same time, by removing sparse data, it can also reduce storage and computing costs and speed up model operation.

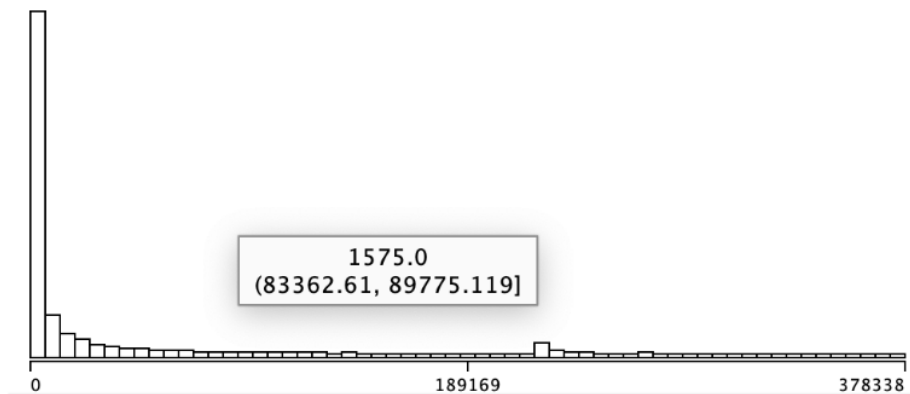


Figure 3. Sparse Data Values

2.2 Data Conversion

Data type conversion in data preprocessing is to convert the data type of some attributes in the original data set from one form to another. Typically, this is caused by the raw data coming from a different source or stored in a different format. To enable data to be correctly interpreted

and processed by computer programs: Different data types are stored in different ways in the computer, and need to be converted into data types that the computer can recognize. Some data types are better suited to storing corresponding data than others, for example converting an integer to a byte can greatly reduce the size of a dataset. Some analysis methods require specific data types. For example, linear regression models require numerical variables, classifiers require discrete variables, etc.

Selected attribute			
Name: earliesCreditLine		Type: Nominal	
Missing: 0 (0%)		Distinct: 668	
		Unique: 34 (0%)	
No.	Label	Count	Weight
1	Nov-1974	25	25
2	Jul-2001	1044	1044
3	Aug-2006	1038	1038
4	Jul-2002	1087	1087
5	Dec-2000	1004	1004
6	Jul-2000	1016	1016
7	Jan-1998	705	705
8	Feb-2007	582	582
9	May-2004	1021	1021

Figure 4. attribute of earliesCreditLine

As shown in Figure 3, many attributes appear in Chinese descriptions. Such data will take up a lot of storage space and cannot be combined with other attributes. It is also difficult for our model to extract useful feature information, so it is necessary to analyze such data. Conversion, the conversion method in this article is to directly extract the year information, for example, Nov-1974 is converted to 1974. This article also involves converting text data to numeric data, such as "yes/no" to 0/1, or converting datetime data (such as '2021-09-01') to timestamp. Encoding of categorical data: Numerical encoding of categorical data (such as color, gender), for example, converting "male" to 1 and "female" to 2.

2.3 Data bucketing

Data bucketing is a method of discretizing continuous numerical features, dividing continuous features into several intervals (also called "buckets" or "boxes"), and each interval has a value range. Data bucketing can solve the problem of overfitting: data bucketing can reduce the impact of noise in the data on model predictions, and prevent the model from over-fitting due to excessive attention to some abnormal data. Solve the sparse sample problem: If some feature in the data set has a very wide range of values, then this feature will have the same value on most samples, resulting in very sparse data. Data bucketing can discretize this continuous feature, increasing the number of samples in the same bucket, thereby alleviating the sparse sample problem. Make the model easier to understand: In some cases, for some continuous features, we don't need to know the value of it precisely, but only need to

know the interval it is in. Through data bucketing, continuous features can be transformed into discrete features, making the model easier to understand and interpret. Improve model prediction ability: For some models, such as decision trees and random forests, using data bucketing can improve the prediction ability of the model, because this discretization method can make the model better learn the interaction relationship between features.

2.4 Normalization

Data standardization is the process of converting data from different sources and formats into the same format and units for easy comparison, analysis and sharing. Data standardization can eliminate inconsistencies and errors in data, thereby improving the accuracy, completeness and consistency of data. Reduce data management costs: Standardized data is easier to manage, maintain and update, thereby reducing data-related management costs. Improve data readability: Data standardization can transform data into an understandable and usable form, making it easier for human users to understand and use. Improve data analysis efficiency: Standardized data is easier to perform operations such as data mining, analysis, and modeling, thereby improving the efficiency and accuracy of data analysis. The following are the values after normalization of the relevant attributes.

'generation other renewable'	'generation solar'	'generation waste'	'generation wind onshore'	'forecast solar day ahead'	'wind onshore day ahead'	'total load forecast'	'total load actual'	'price day ahead'	'price actual'
0.613445	0.008460	0.549020	0.365795	0.002913	0.360554	0.344127	0.319666	0.480737	65.41
0.596639	0.008633	0.546218	0.337807	0.002742	0.326819	0.293279	0.276008	0.460723	64.92
0.613445	0.008633	0.549020	0.313203	0.001371	0.303437	0.232338	0.204274	0.453017	64.48
0.630252	0.008633	0.535014	0.300413	0.000343	0.285814	0.194846	0.141247	0.402382	59.32
0.621849	0.007251	0.529412	0.283035	0.001542	0.268947	0.158042	0.096762	0.363755	56.04
0.621849	0.005870	0.526611	0.264854	0.000685	0.254755	0.143268	0.081135	0.336836	53.63
0.621849	0.005870	0.521008	0.252179	0.000514	0.234921	0.136569	0.085706	0.330932	51.73
0.605042	0.006043	0.529412	0.228952	0.002056	0.218519	0.147735	0.101680	0.341839	51.43
0.613445	0.009323	0.554622	0.208133	0.006683	0.195719	0.143354	0.089362	0.303613	48.98
0.621849	0.128280	0.554622	0.176245	0.134339	0.173036	0.148379	0.112997	0.345642	54.20

Figure 5. Normalized data

3 Data Analysis

After preprocessing, we can see that the distribution of variables gradually conforms to the normal distribution, and there are few outliers, and the data gap will not be too large, but a small amount of data such as installment still shows a long-tail

distribution. In the subsequent processing, this phenomenon is alleviated by using the log function. We can see the distribution of some variables in the figure below.

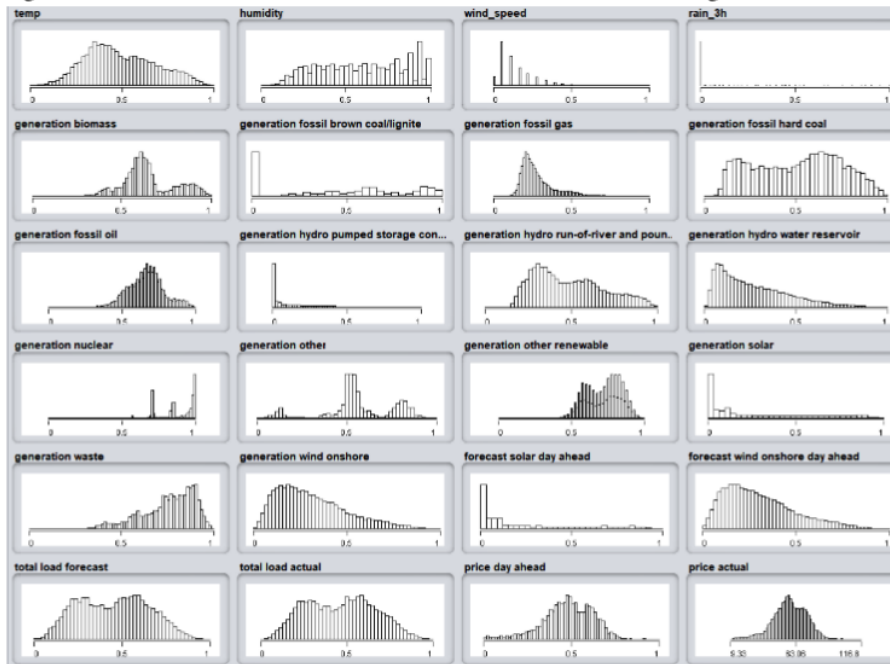


Figure 6. Distribution of attributes

Then, using Weka's visualization feature, we can create scatter plots using the data from the two separate activities mentioned above. The correlation between the data may be examined from these scatter plots, and some early inferences can be drawn to confirm the validity of the following data mining experiment. In Figure 7, for instance, the link between interest Rate characteristics and grades is depicted, with temperature on the x-axis and price real on the y-axis. We can see from the scatter plot that the higher the interest rate, the greater the risk of default.

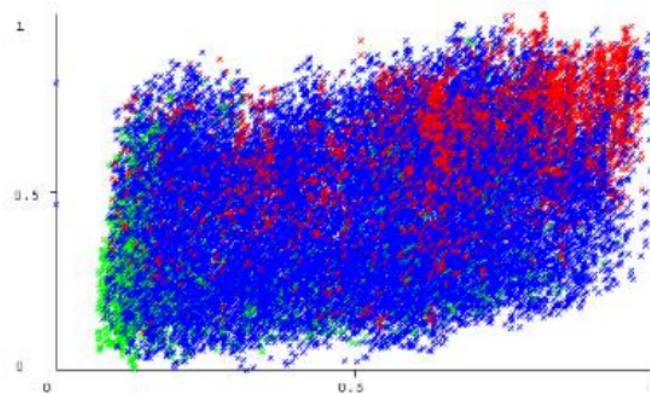


Figure 8. interest Rate characteristics and grades

There are some intriguing correlations between additional traits in addition to the attributes that need to be anticipated. Figure 9 on the left makes it very evident that the 'installment' will rise as the 'employment' does. It may be that the longer the employment time, the more capital accumulation, the larger the loan amount, and thus the larger repayment amount. The link between subgrade and loanAmnt is depicted in Figure 9 to the right. It is obvious that the link between the subgrade and loan amount is quite sufficient since the bigger the subgrade, the smaller the loan amount and the less prominent the red area is in the histogram.

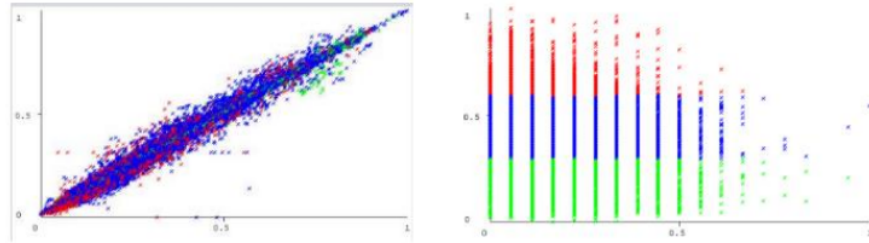


Figure 9. Relationship between employment and installment, subgrade and loanAmnt

4 Experiments

This data set involves a classification problem. After data preprocessing and correlation analysis, I removed some attributes. These attributes may not help the experiment, or even have a bad effect. Finally, there are three Ten attributes, the goal of classification is to judge whether the record is a breach of contract. The training set selects 80% of the data in the data set, and the remaining 20% is used as the test set. The selected classification models mainly include Bayesian network, simple regression, logistic regression, multi-layer perceptron, random forest, etc. , the following is the experiment I conducted.

Bayes Net

A Bayesian network is a probability-based graphical model that uses probabilistic inference to model relationships between variables and uses these relationships for classification. Specifically, Bayesian networks treat each instance in the dataset as a node and connect them with their feature variables. Then, it uses a conditional probability table to represent the relationship between each variable and its parent node, and uses Bayesian formula to calculate the posterior probability from the prior probability and the sample data. In classification problems, Bayesian networks can use the Naive Bayes classifier for classification. A Naive Bayesian classifier assigns each instance in the dataset to the class with the highest posterior probability, $P(Y|X)$,

where Y represents the class and X represents the input feature vector. During training, we need to use training examples of known categories to estimate the prior probability $P(Y)$ of each category and the conditional probability $P(X|Y)$ between the feature variables of a given category. These probabilities can be obtained through statistical calculations on sample data. During testing, we can assign new instances to one of the classes by computing the posterior probability for each class. The root mean squared error of Bayes net is 0.4784, and the prediction accuracy is 58.1615%. The model classification results are shown in the figure below.

```

=== Summary ===

Correctly Classified Instances      13935           39.745 %
Incorrectly Classified Instances    21126           60.255 %
Kappa statistic                    0.1456
Mean absolute error                 0.4017
Root mean squared error             0.6337
Relative absolute error             150.2466 %
Root relative squared error         173.3175 %
Total Number of Instances          35061

=== Detailed Accuracy By Class ===

               TP Rate  FP Rate  Precision  Recall   F-Measure  MOC     ROC Area  PRC Area
0.302    0.167    0.847    0.302    0.445    0.131    0.598    0.797
0.707    0.391    0.227    0.707    0.343    0.221    0.668    0.208
0.668    0.252    0.241    0.668    0.354    0.281    0.718    0.203
Weighted Avg.   0.397    0.207    0.696    0.397    0.421    0.160    0.620    0.651

=== Confusion Matrix ===

  a    b    c  <-- classified as
7977 11058  7391 |    a = m
  922   3452   509 |    b = h
  518    728  2506 |    c = l

```

Figure 10. Results of Bayes net

Simple Logistic

Weka's Simple Logistic algorithm makes use of LogitBoost, which is created by maximizing the loglikelihood function. Each iteration of the algorithm updates the model using the Newton descent approach. As illustrated in Figure 14, the Simple Logistic's Root Mean Squared Error is 0.3251 and its Correctly Classified Rate is 78.7998%.

```

=== Summary ===

Correctly Classified Instances      27628           78.7998 %
Incorrectly Classified Instances    7433           21.2002 %
Kappa statistic                    0.3044
Mean absolute error                0.2139
Root mean squared error            0.3251
Relative absolute error            79.9821 %
Root relative squared error        88.9273 %
Total Number of Instances         35061

=== Detailed Accuracy By Class ===

                TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area
                0.962   0.740   0.799     0.962   0.873     0.329   0.746   0.882
                0.237   0.018   0.678     0.237   0.352     0.352   0.831   0.500
                0.281   0.016   0.683     0.281   0.399     0.400   0.853   0.495
Weighted Avg.   0.788   0.562   0.770     0.788   0.749     0.340   0.769   0.787

=== Confusion Matrix ===

      a    b    c  <-- classified as
25413  534  479 |    a = m
3713  1159  11 |    b = h
2679   17  1056 |    c = 1

```

Figure 11. Results of Simple Logistic

Logistic

Logistic regression is a common binary classification algorithm that is mainly used to predict which category a given input data belongs to. The basic idea is to use a linear model to weight the input features and map them to a probability value, and then judge which category the sample belongs to according to the probability value. Specifically, logistic regression usually uses the sigmoid function to transform the linear prediction result, limiting it between 0 and 1, expressed as the probability that the sample belongs to the positive example. The goal of logistic regression is to make the prediction result as close as possible to the real label by adjusting the weights and bias items, usually using maximum likelihood estimation or cross-entropy loss function to measure the prediction error. In the training process, optimization algorithms such as gradient descent are usually used to solve the optimal parameters to minimize the loss function. When predicting, the input features are directly brought into the trained model, and the probability of belonging to the positive example is obtained through the sigmoid function. If the probability is greater than 0.5, it is determined that the sample belongs to the positive example category, otherwise it belongs to the negative example category. According to Figure 12, the classification impact of Logistic is marginally superior to Simple Logistic, with a Correctly Classified Rate of 79.008% and a Root Mean Squared Error of 0.3244. The accuracy of logistic regression is not particularly great and it is more susceptible to multicollinearity data. It cannot accurately reflect the true distribution of the data for electrical data sets with more properties.

```

=== Summary ===

Correctly Classified Instances      27701          79.008 %
Incorrectly Classified Instances    7360          20.992 %
Kappa statistic                    0.3194
Mean absolute error                0.2116
Root mean squared error            0.3244
Relative absolute error             79.1392 %
Root relative squared error        88.7359 %
Total Number of Instances         35061

=== Detailed Accuracy By Class ===

                TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area
                0.959   0.723   0.802     0.959   0.874     0.341   0.749    0.883
                0.259   0.020   0.677     0.259   0.374     0.368   0.831    0.504
                0.289   0.016   0.680     0.289   0.406     0.405   0.857    0.498
Weighted Avg.   0.790   0.550   0.772     0.790   0.754     0.352   0.772    0.789

=== Confusion Matrix ===
      a    b    c  <-- classified as
25353  582  491 |    a = m
 3600 1263   20 |    b = h
 2646   21 1085 |    c = l

```

Figure 12. Results of Logistic

Multilayer Perceptron

The artificial neural network model known as the multilayer perceptron is effective at simulating the nonlinearity of complicated data. The multilayer perceptron performed well in this experiment. According to Figure 16, this model's root mean squared error is 0.2848, and its prediction accuracy is 84.371 percent.

```

=== Summary ===

Correctly Classified Instances      29581          84.3701 %
Incorrectly Classified Instances    5480          15.6299 %
Kappa statistic                    0.5475
Mean absolute error                0.1543
Root mean squared error            0.2848
Relative absolute error             57.6875 %
Root relative squared error        77.8986 %
Total Number of Instances         35061

=== Detailed Accuracy By Class ===

                TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area
                0.956   0.493   0.856     0.956   0.903     0.547   0.840    0.918
                0.517   0.023   0.785     0.517   0.623     0.593   0.875    0.698
                0.477   0.017   0.771     0.477   0.589     0.572   0.887    0.646
Weighted Avg.   0.844   0.377   0.837     0.844   0.831     0.556   0.850    0.859

=== Confusion Matrix ===
      a    b    c  <-- classified as
25268  651  507 |    a = m
 2334 2523   26 |    b = h
 1923   39 1790 |    c = l

```

Figure 13. Results of Multilayer Perceptron

Random Forest

The random forest is composed of multiple decision trees, and each decision tree is trained on a different subset (bootstrap). At the same time, when each node selects a partition attribute, it selects the optimal one from several randomly selected attributes for to avoid overfitting problems. Usually, we will set some parameters to control the random forest generation process, such as the number of decision trees, the maximum depth of each decision tree, etc. The Root mean squared error of the model is 0.2148, and the Correctly Classified rate is 90.9907%, as shown in Figure 14.

```
=== Summary ===

Correctly Classified Instances      10847          90.9907 %
Incorrectly Classified Instances    1074           9.0093 %
Kappa statistic                    0.7467
Mean absolute error                0.1233
Root mean squared error            0.2148
Relative absolute error            46.2743 %
Root relative squared error        59.2062 %
Total Number of Instances         11921

=== Detailed Accuracy By Class ===

                TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area
                0.981   0.314   0.908     0.981   0.943     0.744   0.966   0.988
                0.680   0.009   0.923     0.680   0.783     0.766   0.980   0.909
                0.693   0.008   0.916     0.693   0.789     0.777   0.987   0.916
Weighted Avg.   0.910   0.240   0.911     0.910   0.905     0.750   0.970   0.969

=== Confusion Matrix ===

  a    b    c  <-- classified as
8876  89   81 |    a = m
 514 1090   0 |    b = h
 388   2   881 |    c = l
```

Figure 14. Results of random forest

5 Summary

This paper conducts a preliminary analysis and processing of the financial default data set. Through the analysis and modeling of historical borrowing data, it identifies which factors will increase the probability of loan default, thereby predicting whether the borrower may default. We can collect the borrower's personal information, credit history, financial status and other data, and train a machine learning model to predict whether the borrower is likely to default based on these factors. Through such a predictive model, banks or financial institutions can assess risks more accurately, formulate more reliable lending strategies and risk management plans, thereby ensuring the stability and sustainable development of the financial system.

This article uses a variety of machine learning models, including Bayesian networks: Bayesian networks are probabilistic graphical models that can be classified by establishing conditional probability relationships between variables. In the financial loan default classification task, the Bayesian network can handle the dependencies between variables well, and has good accuracy and interpretability for complex loan risk assessment. Logistic regression: Logistic regression is a linear classifier that maps input features to a probability value that represents the probability that a sample belongs to the positive class. In the task of classifying financial loan defaults, logistic regression is simple, fast, and has good interpretability, and it also has good performance on large-scale data sets. Random Forest: Random Forest is an ensemble learning algorithm that improves classification accuracy by combining multiple decision trees. In the task of classifying financial loan defaults, random forests have advantages in dealing with high-dimensional data and outliers, and are also strong in modeling nonlinear relationships. Multilayer Perceptron: A multilayer perceptron is a deep neural network model that can learn non-linear relationships between features. In the financial loan default classification task, the multi-layer perceptron improves the classification accuracy through the combination of multi-layer hidden neurons, but requires a large amount of data and computing resources.

References

- [1] Han Xiao, Kashif Rasul, and Roland Vollgraf, "*Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms.*" arXiv preprint arXiv:1708.07747, pp. 1–6, 2017.
- [2] Witten, Ian H, Frank, Eibe and Hall, Mark A. Data Mining: Practical Machine Learning Tools and Techniques: Practical Machine Learning Tools and Techniques. s.l. : Elsevier, 2001.
9780080890364
- [3] Cover T, Hart P. Nearest neighbor pattern classification[J]. IEEE transactions on information theory, 1967, 13(1): 21-27.
- [4] 周志华. 机器学习, 2016
- [5] Browne M W. Cross-validation methods[J]. Journal of mathematical psychology, 2000, 44(1): 108-132.
- [6] El-Henawy I , Bakry H , Hadad H , et al. A New Muzzle Classification Model Using Decision Tree Classifier[J]. International Journal of Electronics & Information Engineering, 2017.

assignment2

GRADEMARK REPORT

FINAL GRADE

GENERAL COMMENTS

Instructor

61 /100

PAGE 1

PAGE 2

PAGE 3

PAGE 4

PAGE 5

PAGE 6

PAGE 7

PAGE 8

PAGE 9

PAGE 10

PAGE 11

PAGE 12

PAGE 13

PAGE 14

KNOWLEDGE AND UNDERSTANDING*Dataset description (15 marks)*

-  Tianchi competition dataset appraised and user default prediction previously undertaken. You have not made your rational clear. Why did you choose this dataset ? What are you hoping to determine having examined it ?

11**COGNITIVE/ INTELLECTUAL SKILLS***Primary and Secondary data analysis, reflection (35 marks)*

-  Performed analyses are depicted clearly and many instances of self reflection (significance of results and relevance to discipline collected dataset is taken from) demonstrated

19**PRACTICAL APPLICATION SKILLS***Primary and Secondary data processing (20 marks)*

-  Description of pre-processing listed sufficiently well and connections are mapped to utilised algorithms very well.

17**TRANSFERABLE SKILLS**



Submission offers evidence of independent learning and an ability convey technical content in an articulate manner overall.

14