Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 174 (2020) 150–160

Procedia Computer Science

www.elsevier.com/locate/procedia

2019 International Conference on Identification, Information and Knowledge in the Internet of Things (IIKI2019)

Credit Risk Assessment based on Gradient Boosting Decision Tree

Zhenya Tian[a], Jialiang Xiao[a], Haonan Feng[a], Yutian Wei[a]

School Of Electronic Engineering And Computer Science, Peking University, Beijing, 100871, China
E-mail: 1600012944@pku.edu.cn
School of Foreign Languages And Literature, Wuhan University, Wuhan, 430072, China
E-mail: 2015300340016@whu.edu.cn
School Of Economics, Liaoning University, Shenyang, 110000, China
E-mail: andyfenghaonan@outlook.com
Tabor Academy, Marion, Massachusetts, 02738, United States
E-mail: ywei20@taboracademy.org

These authors are contributed equally to this work

Abstract

Regarding to the increasingly attention on the credit risk rating system, the tradition way to evaluate the credibility of any given individual or company using machine learning is based on methods like SVM, decision tree or MLP. In our research, a more efficient method is introduced, which is known as the Gradient Boosting Decision Tree. With proper data preprocessing and feature selection, models are compared due to their performance. Our model, Gradient Boosting Decision Tree, has been proved to be one of the best that obtain the highest accuracy (92.19%), f1 score (91.83%) and AUC value (0.97). The experiment proved that this model has the best ability of classification and generalization.

Keywords: credit risk assessment; classification; machine learning; Gradient Boosting Decision Tree

1. Introduction

Along with the continuous changes in the global financial situation, the credit risk caused by the subprime loans of commercial credit companies and commercial banks has gradually become the primary source of liquidity crisis for various enterprises, especially for financial institutions. Moreover, as the volume of transactions continues to grow, the expansion of the scale of transactions tends to evolve from a regional financial crisis to a global financial crisis. For instance, the Asian financial turmoil in 1997 caused a large number of bankruptcies in the global economy. This expanding financial risk has aroused widespread concern about financial risk management and highlighted the importance of financial risk management. Some regulations would be pivotal to improve this current miserable situation [14].

Credit risk mainly refers to the potential possibility that the individuals who take loans or the transaction object cannot fulfill their obligations in accordance with the agreement reached in advance [15]. The severe credit risk is likely to cause the bankruptcy of the commercial credit company and would possibly cause a serious of the butterfly effect. Regarding the problem addressed above, the credit rating system for the commercial credit company's customers is crucial. There are three main methods of credit risk measurement, the expert system, the rating system, and the credit scoring systems [3]. Those systems mainly relied on artificial

10.1016/j.procs.2020.06.070

subjective judgment and supplemented by the modern quantitative model to verify the credit quality of borrowers [5]. However, traditional credit risk measurement methods are challenging to meet the growing demand for credit behavior. Those subjective decisions are always blended with the bias that could negatively affect the efficiency and accuracy while the development of data science and machine learning could plausibly reduce the problem of low efficiency and low precision of traditional risk measurement methods[11].

In the era of big data, a large amount of data information can contribute to the determination of a person's credit quality. We can train the mathematical model through the historical data of the borrower's credit behavior and its final credit quality classification in a certain period, and apply this mathematical model to the identification of customer credit quality[1]. Since the classification of customer credit risk is completely judged by historical data, artificial subjective identification is avoided, labor cost and judgment error are reduced, and the problem of low efficiency and high cost of traditional risk measurement method is solved.

This paper starts with the historical loan data of a lending company, carries out structured data processing, and performs data filtering and feature extraction based on the relationship between the number of missing values and the data. Because in the actual credit behavior, there is always a tendency to be more trustworthy than untrustworthy behavior, but we tend to pay more attention to the prediction of untrustworthy behavior. Therefore, in the processing of data, we adopted the SMOTE oversampling algorithm for data balancing. In the choice of model, we investigated the current research on machine learning processing of credit risk and found that existing research mostly uses SVM, Decision Tree, Random Forest, and other algorithms to fit the historical data of lending, while in our model [6]. During the training process, it was found that using Gradient Boosting Decision Tree 's integrated learning algorithm to fit credit history data can improve the accuracy and recall rate compared with traditional models such as SVM. Therefore, we finally chose the Gradient Boosting Decision Tree model and Using the Grid Search algorithm for parameter adjustment, a better solution to credit quality classification is achieved.

2. Related Work

There are several models already exists to solve the problem while dealing with problem about credit risk assessment. They are got some significance but contain some vital drawbacks. The commonly used models are listed below:

Support Vector Machine (SVM) is a traditional method used to deal with binary classification problems. SVM, which is a generalized linear classifier, classifies data by supervised learning. The decision boundary is the Hyper-margin hyperplane for solving the learning samples. To calculate empirical risk, SVM uses the hinge loss function and adds a regularization term to the solution system to optimize structural risk. It is a classifier with sparsity and stability. Since the SVM was first introduced in 1964, this method is relatively inefficient in time consumption and performed poorly when compared to those newly developed algorithms [7].

Logistic regression is the corresponding regression analysis when the corresponding variable is dichotomous (binary). Similar to all the other regression analyses, logistic regression is a predictive analysis. This method is used to explain the relationship between differential binary variables. However, Logistic regression is sensitive to multivariate collinearity of independent variables in the model, that one variable would largely affect others. While dealing with too many variables, the performance would not be that satisfactory.

Decision Tree is the most common inductive reasoning method in machine learning. By analyzing training samples to summarize concepts and knowledge, it is a method to approximate discrete objective function, which is represented by a decision tree. However, at the same time, the result of the decision tree may be unstable, because a little change in the data may lead to the creation of a completely different tree, a problem that can be solved by using an ensemble decision tree. If the data for each category are with inconsistent sample size, information gain may be biased towards features with more values, making it challenging to process unbalanced data.

MLP is a multi-layer perceptron, an artificial neural network that tends to structure, mapping a set of input vectors to a set of output vectors [9]. MLP can be thought of as a directed graph, consisting of multiple node layers, each connected to the next layer. In addition to the input nodes, each node is a neuron with a nonlinear activation function. MLP has strict requirements for feature selection and normalization of data [4].

AdaBoost selects the weak classifiers for weighted combination step by step. The weak classifier with the lowest weighted error rate is selected to combine each time, and the weighted error rate determines its weight. However, AdaBoost is sensitive to the outlier data, which has a great influence on the accuracy of prediction results.

Random Forest random selects the features and the data to generate many decision trees and then summarize the results of the decision trees. Random Forest improves the prediction accuracy without significant increase of the calculated amount, but it may generate many similar decision trees that may cover up the real results.

Gradient Boosting Decision Tree is an algorithm that ensembles several weak classifiers (decision tree) together to form strong and effective classifier. The method contains rounds of iteration that entrust higher weight to the negative samples and lower weight to the positive samples. It’s a process of emerge all weak classifiers together to have a model with better performance.

Regarding to all the problems addressed above about the traditional used models, the method we choose, the Gradient Boosting Decision Tree, would have the best performance with both high accuracy and efficiency. This model would be the best fit for this specific type of problem.

3. Gradient Boosting Decision Tree

3.1 Boosting

In the Boosting algorithm, we have ranked each weak classifier and weighted each sample. Initially, we assign the same weight for each sample and use the first sample to train the first weak classifier. After the learning process, we increase the weight of the mistaken samples, and at the same time, decrease the weight of the correct sample. Then we get our second weak classifier to learn from the data sample with this kind of actions repeatedly. At last, we get m weak classifiers, after merging these m classifiers, we get our final classifier [8].

Gradient Boosting Decision Tree is a kind of ensemble algorithms which is based on multiple weak classifiers. The result was then assayed by weighting method. Weak classifiers are commonly adopting Regression Decision Tree employing several times' iteration. Every weak classifier is trained based on previous weak classifier's residual error in which way GBDT can reach the classification target by decreasing the residual error in the training process.

Gradient Boosting Decision Tree is a sort of boosting machine learning method. Therefore, we adopt the addition model and forward stagewise algorithm. The following formula represents the mth weak classifier, which the M represents the number of the classifiers and the Θ_m indicates the parameter of specific classifiers.

$$f_M(x) = \sum_{m=1}^{M} T(x;\Theta_m)$$

The forward stagewise algorithm refers to the model iterative process, from front to back, learning only one weak classifier with its parameters at a time. The learning process of the current weak classifier is based on all the weak classifiers that have been trained before. Therefore, the mth step of the boosting classifier model can be expressed as follows.

$$f_m(x) = f_{m-1}(x) + T(x;\Theta_m)$$

Loss Function:

$$L(f_m(x),y) = L(f_{m-1}(x) + T(x;\Theta_m),y)$$

In the training process, we train only one classifier $T(x;\Theta_m)$ in each rounds which guarantee Loss Function $L(f_m(x),y)$ to be minimum.

3.2 Gradient Boosting Decision Tree Algorithm Processs

Gradient Boosting Decision Tree is commonly regarding Negative binomial logarithm likelihood $\log(1 + e^{-2yF})$, $y \in (-1, +1)$. It's simple to optimize this form of loss function, but it's difficult to optimize the general function by the common gradient descend method. In response to this kind of problems, Freidman put forward an algorithm which uses negative gradient of loss function to fit the approximation of loss this round and then fit a Classification and Regression Tree. The specific procedure is shown below:

Assume there are N sample sets and initialize the weak classifiers:

$$f_0(x) = \arg\min_c \sum_{i=1}^{N} L(y_i,c)$$

The tth round ith sample loss's Negative Gradient Direction can bey expressed as:

$$r_{ti} = -\left[\frac{\partial L(y_i,f(x_i))}{\partial f(x_i)}\right]_{f(x) = f_{t-1}(x)}$$

We can use (x_i,r_{it}) $i = 1,2,...m$ to fit the tth Decision Regression Tree, the corresponding leaf node region is $R_{tj}, j = 1,2,...,J$. J is the number of the leaf nodes. As to the sample in every node, we seek out the output value c_{tj} to minimize the loss function.

$$c_{tj} = \arg\min_{c} \sum_{x_i \in R_{tj}} L(y_i,f_{t-1}(x_i) + c)$$

Refresh the round and get a stronger classifier.

$$f_t(x) = f_{t-1}(x) + \sum_{j=1}^{J} c_{tj} I(x \in R_{tj})$$

At last, we get our final strong classifier.

$$f(x) = f_M(x) = f_0(x) + \sum_{t=1}^{M}\sum_{j=1}^{J} c_{tj} I(x \in R_{tj})$$

When it comes to the classification problem, we can adopt the Negative binomial logarithm likelihood $\mathrm{L(y,F)} = \log(1 + e^{-2yF})$, $y \in (-1, +1)$. So we have the negative gradient error that time is

$$r_{ti} = -\left[\frac{\partial L(y_i,f(x_i))}{\partial f(x_i)}\right]_{f(x) = f_{t-1}(x)}$$

$$= \frac{y_i}{1 + \exp(y_i f(x_i))}$$

So we have the Best Residual Error value for each nodes

$$c_{tj} = \arg\min_{c} \sum_{x_i \in R_{tj}} log(1 + \exp(-y_i(f_{t-1}(x_i) + c)))$$

And then we can have the majorization which uses the approximate value as the substitute.

$$c_{tj} = \frac{\sum_{x_i \in R_{tj}} r_{ti}}{\sum_{x_i \in R_{tj}} |r_{ti}|(1 - |r_{ti}|)}$$

3.3 Grid Search

Grid Search is a fundamental parameter majorization method which substantially divide the parameter into the grids with same length in the certain range of coordinate system. Every point in the coordinate system represent a set of parameters, and then we can adopt every point in a certain interval into our model to verify the performance of the algorithm. The point that performs best is called best parameter. In other words, the algorithm of Grid Search is to traverse the points corresponding to all grids [13].

4. Experiment

4.1 Question Raised

In the experiment session, the data we use is collected from a credit assessment company. The original data has around 50,000 rows and 350 columns containing discrete numeric value.

The amount of data and the amount of features we obtain are large, because different features have different effects on positive and negative samples, and some bad features can interfere with our predictions, so we need to solve how to select the data that represents the overall data. A problem with good features. Then, the data imbalance we obtained is more significant, and the experimental results may not reflect the accuracy of the model. Therefore, we need to solve the problem of how to adjust the ratio of the two categories to one to one by adjusting the sampling method. In addition, after our research, the accuracy of the existing model for credit risk assessment is still insufficient, so we want to find a new model to accurately predict the credit risk of the user through historical data of known credit behavior.

4.2 Data Cleaning

The initial data obtained includes some missing units that have to be filled; otherwise, the further analysis will not work out. To deal with the missing units, we remove the columns that are missing a significant amount of data and delete some specific samples which lack integrity. The columns represent a feature, those features which are missing a considerable amount of data is considered as bad features that would be eliminated. For the rows, they represent each sample, for those that are missing much information are considered as bad samples that will not contribute to the model and even worsen the final result.

Therefore, the data is cleaned both vertically and horizontally to guarantee the viability for further analysis. The scale of the data removal is tiny when compared to the entire amount of the data collected. In our case, the sample size does not drop dramatically due to these data cleaning process. The sufficient numbers of samples collected ensure these minor removals will not affect the final result drastically.

In our data set, there some blank that didn't contain any type of value. Considering the fact that the amount of instance which are of no value is significantly low regarding total amount of data, we decide to delete the NULL value from data set. In order to protect the hidden correlations among data from being removed by deletion, we choose to perform data deletion based on the number of NULL value of lines and columns. After data cleaning, our data has around 51,400 rows with 345 columns. At this stage, we have cleaned our data. However, before we choose machine learning model to fit, we have to select features that are important to our final model's build.

4.3 Feature Selection

The data collected have a ton of meaningless features that should be removed. Most of the features will not contribute to the result of the prediction; some of them even worsen the situation[2]. For those features that represent some very irrelevant aspects, the filtration is vital. Other than figuring out each feature and estimated their relativeness through their meanings, a mathematic analysis would be more convincing and efficient. For instance, in our case, the data collected is kind of ambiguous that some of the features cannot be interpreted. Therefore, a mathematical way to evaluate the features is necessary.

Also, efficiency is also a significant factor to be considered, too many features would cause the model very inefficient and lack of competitiveness. To select the right feature is also a crucial part of the data preprocessing. While choosing between a couple of dozens of features and a couple of hundreds of them, a smaller number of features would be much more attractive, if this filtration will not dramatically affect the final results in terms of the model accuracy. Therefore, a specific coefficient is introduced which is known as Pearson Correlation Coefficient. This coefficient somehow reveals the relativeness between the features and the result, helping us to better identify those meaningful features.

In our experiment, the usage of the Pearson Correlation Coefficient as a method reduces the total amount of feature that needs to be considered in the model. The Pearson Correlation Coefficient presents the relativeness of the two variables. The formula to calculate the Pearson Correlation Coefficient is defined as the quotient of the covariance and standard deviation:

$$\rho_{X,Y} = \frac{cov(X,Y)}{\sigma_X \sigma_Y} = \frac{E[(X-\mu_X)(Y-\mu_Y)]}{\sigma_X \sigma_Y}$$

While for specific samples, a more intuitive method could be written as the following:

$$r = \frac{N\sum x_i\, y_i - \sum x_i \sum y_i}{\sqrt{N\sum {x_i}^2 - (\sum x_1)^2}\sqrt{N\sum {y_i}^2 - (\sum y_i)^2}}$$

The coefficient is in the range of -1 to 1, and the higher the absolute value is, the higher the relativeness is. While changing the threshold value would help us to limit the different amount of the most related features. This filtration enhanced efficiency and improved accuracy.

After all these steps of data preprocessing, all the data sets are now fully prepared to apply to the further analytic models. The data preprocessing is always one of the most significant steps to take. This process enables the further analysis and makes the result much more persuasive and objective.

To achieve accurate and precise selection of features, we use Pearson's Correlation Coefficient to evaluate the importance of each feature in our data set. Then, we label every feature with its Pearson's correlation coefficient range from -1 to 1. In order to narrow down our feature set for the future benefit in training models, preventing negative influence caused by introducing uncorrelated features into ultimate model and improving the generalization performance of our model, we set several thresholds with different numeric value to select from features. By comparing several performance indices such as accuracy, f1 score and AUC value, we can find the best threshold and its corresponding number of features. Considering the coefficient may be below zero, here we use the absolute value of all the coefficient for feature selection. The result of different number of features and their corresponding performance index is shown in Figure 1.

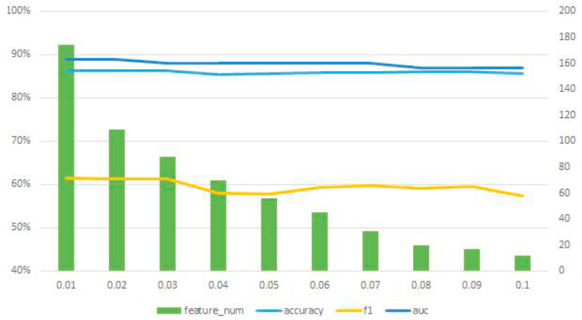

Figure 1 the performance result of different threshholds in Pearson's Correlation Coefficient

In Figure 1, we can find that there is a noticeable decrease of f1 value from threshold of 0.03 to 0.04. Meanwhile we can perceive that the accuracy and AUC value of threshold 0.03 is relatively high comparing to that of threshold 0.01 and 0.02. From the comparison we decide to use Pearson correlation coefficient of 0.03 as our threshold for feature selection because at this threshold the number of features is relatively low which is good for future model training and the accuracy of the model would receive low influence under this threshold. Next, we will move on to deal with the data imbalance.

4.4 Data Imbalance

The data obtained is significantly unbalanced that more than 70 percent is classified in one category. Therefore, some techniques have to be used to adjust the data to make samples more balanced. The traditional ways include random under-sampling and random over-sampling [10].

The under-sampling method is to randomly decrease the amount of the majority ones collected to make the sample seeming much more balanced. While for the over-sampling method, the minority samples get a higher weight to balance out the asymmetric representation of the data collected.

In our experiment, SMOTE (Synthetic Minority Oversampling Technique) is eventually chosen, which provides the best performance [12]. This method is an improved method based on oversampling which first introduced in 2002 by Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall, and W. Philip Kegelmeyer. In their journal, they introduce the SMOTE method is an expanded method of over-sampling with replacement, while the traditional ones have been proved as cannot significantly improve the recognition of the minority.

For algorithm SMOTE, the minority class is over-sampled while introducing synthetic examples along with the line segments that connect minority class nearest neighbors. The usage of SMOTE provides synthetic examples which cause the classifier to have a new region, which is considered as more extensive and less specific. A more general regions are now presented as a representation of the minority class samples instead of being submerged by the surrounding majority class samples. The usage of the SMOTE algorithm helps us to adjust the data sample in a more balanced way. A simple flow chart is presented in Figure 2.

This flow chart briefly describes how this SMOTE algorithm works. The usage of this algorithm helps us to balance the data so that the result for our model would be much more reasonable. Therefore, for these considerations, the SMOTE algorism is applied.

The data set we use is a highly imbalanced one in which the label for classification work innate a 4 to 1 ratio in two categories. The data imbalance in data set will have negative influence on recall rate of our ultimate model. There are two ways to address this problem namely, under-sampling and over-sampling. We use different sampling measures(raw sampling, random under-sampling, random oversampling and SMOTE Tomek) to train a default Gradient Boosting Decision Tree model for selection of best sampling method. The result is shown in Figure 3.

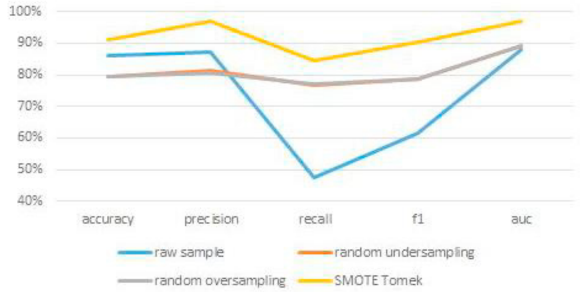

Figure 2 the performance result of raw samples, random undersampling, random oversampling and SMOTE Tomek

From Figure 3we can see that the performance of SMOTE Tomek is the best among all of sampling ways. And as we expected without any sampling(raw sample) the recall value is significantly lower than others. Considering the fact that random under-sampling and random over-sampling have the same performance which is slightly worse than SMOTE Tomek, we determine to use SMOTE Tomek as our final sampling method.

4.5 Model Comparison

Since the data gathered is from some commercial credit company, features with different units are all included in the same database. For example, several columns are representing the date and a couple of other columns representing the location of the clients. The problem of having differences among features are imperative. The method we used is known as min-max normalization, which convert all the data sample in a range between 0 and 1. More importantly, the process of this normalization removes the dimension of all the data collected. With a more standardized and generalized data set, most models would have a relatively decent performance. While for some of the models, the normalization would not help to enhance the performance, such as Decision Trees. Even though the final model chosen is Gradient Boosting Decision Tree, which is somehow based on the decision tree, the process of applying the normalization to the database is still pivotal. The reason for still using normalization as a step in our experiment is because we want all models are competing in their best performance, regardless of the possible fruitless attempt for those models based on decision trees. In this case, the normalization guarantees the viability to compare different models and help us to find the best one to predict the creditability of the samples.

To verify the effectiveness of our Gradient Boosting Decision Tree, we use LR, SVM, CART, MLP, AdaBoost and Random Forest six algorithms to tst and compare the correct classification of our dataset.

As can be seen from table 1, under normalized conditions, the results of LR, SVM, CART, MLP, AdaBoost, Random Forest and Gradient Boosting Decision Tree are compared by accuracy, f1 score and AUC value. From the dataset, the application of Boosting has brought a substantial improvement for Decision Tree: Gradient Boosting Decision Tree(90.99%, 90.37%, 0.97) outperform four base learners LR(74.43%, 74.37%, 0.84), SVM(77.64%, 77.94%, 0.87), CART(84.68%, 84.71%, 0.85), MLP(84.61%, 83.45%, 0.93) in terms of three indicators. Compared with other ensemble learning methods AdaBoost(87.67%, 87.37%, 0.95) and Random Forest(88.96%, 88.45%, 0.96), the Gradient Boosting Decision Tree(90.99%, 90.37%, 0.97) also has a bigger improvement in terms of three indicators. The better performance about the dataset in terms of three indicators shows that our Gradient Boosting Decision Tree model has a higher accuracy rate and stronger generalization ability.

Table 1 the performance result of different machine learning models

	accuracy	f1 score	AUC
Logistic Regression	74.43%	74.37%	0.84
SVM	77.64%	77.94%	0.87
Decision Tree	84.68%	84.71%	0.85
MLP	84.61%	83.45%	0.93
AdaBoost	87.67%	87.37%	0.95
Random Forest	88.96%	88.45%	0.96

Gradient Boosting Decision Tree	90.99%	90.37%	0.97

4.6 Gradient Boosting Decision Tree Model Parameters Adjustment

The parameters we adjust for our Gradient Boosting Decision Tree model are number of estimators, learning rate and minimum impurity decrease. Number of estimators represents the maximum number of iteration. The default number of weak classifiers is 100. Generally speaking, lack of number of iteration would cause under-fitting of ultimate model while too much iteration would cause over-fitting. In our experiment, we increase the value of the number of weak classifiers 100 each time from default value of 100 to 500 in the end and record the accuracy score, precision score, recall value and f1 value of models with different number of maximum iteration(other parameters set as default). The result is shown in Figure 4.

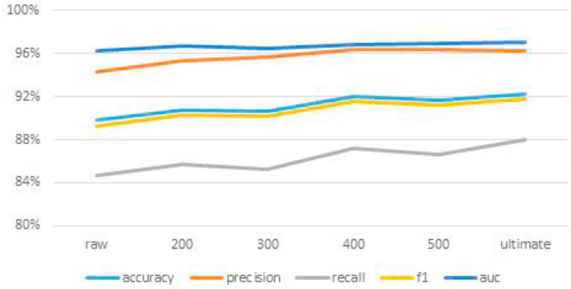

Figure 3 the performance result of different numbers of iterations in Gradient Boosting Decision Tree

According to Figure 4 we can find that when the number of weak classifiers equals 400, performance of our model is the best among all other models. We would choose 400 as the number of iterations of our Gradient Boosting Decision Tree model.

The we have to adjust learning rate parameter for our model. Learning rate is also called step size. It is also in charge of the iteration time of Gradient Boosting Decision Tree model. Usually learning rate and the number of weak classifiers are need be adjusted at the same time. Here we start adjust trial from the default value 0.1 to 0.3 of learning rate. For each step we increase learning rate by 0.05 and record the performance index of model with different learning rate. The result is shown in Figure 4.

From Figure 5 we can find that when we adopt 0.25 as learning rate of our model, all the indices of the corresponding model reach the peak. In our final model we would use 0.25 for the model's learning rate.

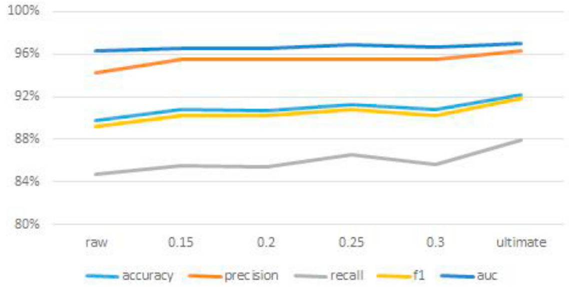

Figure 4 the performance result of different learning rate in Gradient Boosting Decision Tree

At the third step we would adjust the minimum impurity split for our Gradient Boosting Decision Tree model. This is a parameter which confine the growth of weak classifier. Here the weak classifier is decision tree. If the impurity of one node is lower than the threshold we set, this node would not be split and have child nodes. Here in parameter adjustment, we record the performance of Gradient Boosting Decision Tree models with different minimum impurity split range from 1.00E-05 to 1.00E-09. Each step the value of threshold would be nine times smaller than the former value. And the performance of models with different minimum impurity split threshold is shown in Figure 6.(raw value is 1.00E-07)

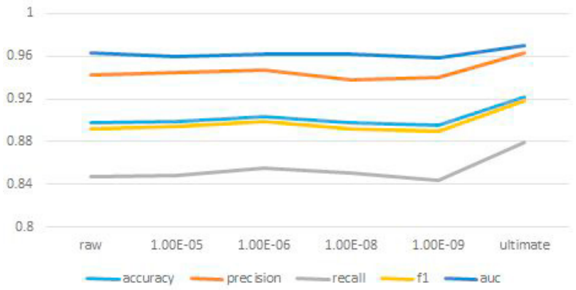

Figure 5 the performance result of different minimum impurity split value in Gradient Boosting Decision Tree

We can find that when the minimum impurity split value is set as 1.00E-06, the performance of Gradient Boosting Decision Tree is the best. So we use 1.00E-06 as the value of minimum impurity split parameter's value for our model.

5. Conclusion

The credibility of individuals or companies is a pivotal aspect that needs to be considered. It would be beneficial for banks and other financial companies to acquire an effective way to predict credibility. Therefore, having such a prediction would help society to avoid many significant economic crises, such as the subprime crisis in 2008. During that time, too much highly unstable loans were sent out, which eventually caused the collapse of the economies. Starting in the United States and radiated affecting the entire world. The lack of regulation and some effective manner to evaluate the credibility level contributed to such a miserable situation. Thus, due to that consideration, society is urging for some kinds of prediction models to solve the problem.

The machine learning is one of the most popular ways that being introduced so far. The usage of big data and machine learning largely helped the situation. As the passage of the time, more and more new models are introduced. From a single model such as SVM, decision tree, logistic regression, and neural network, to more advanced ensembled models, including random forest and gradient boosting decision trees. There are many different types of models, and each of them would be optimal in specific circumstances. In our research, most of the models are compared to pick the most efficient one for our specific set of data about credibility. The selection of the model is primarily based on the final result for precision and some consideration of the efficiency.

For further research, there are still many aspects that we could improve on as the advancement of the algorithms, more and more outstanding ones would be innovated. We could assume that while using those more advanced models, a better result would be expected. Thus, developing new models would be a way to improve our research further. Also, while dealing with different models, the Gradient Boosting Decision Tree might not be the optimal one to use. Those other models could be much more efficient and precise. The selection of the model would largely depend on the data collected, and the field the problem is about. Furthermore, some more sophisticated way of data preprocessing would possible improve the final result in some circumstances. For instance, the SMOTE algorithm still got space to improve on, such as using the algorithm combined with under-sampling.

Anyway, to the best of our knowledge, there are very few previous studies that used Gradient Boosting Decision Tree to deal with this specific type of problem. As future work, we intend to improve the methodology used in the data preprocessing steps, including the data cleaning and the sample collections. Such as having a more complex model for dealing with the imbalanced data set; applying more different types models; assort models due to different specific situations.

Acknowledgements

The completion of the thesis is attributed to many people's support and encouragement. Without of the help of many people,it could never be completed, thus here we would like to express my sincere gratitude towards them.

First and foremost, we owe our heartfelt thanks to my distinguished and cordial supervisor,Professor Yingjie Tian, who influenced us with his insightful ideas and meaningful inspirations, guided us with practical academic advice and feasible instructions, and enlightened us while we was confused during the writing procedure. His thought-provoking comments and patiently encouragements are indispensable for my accomplishment of this thesis.

Ultimately, thanks go to our parents who are our mentor and guardian from the very beginning in primary school. Without their refined education and care,we could never grow up in such a joyous and cozy environment nor have the courage to confront any obstacles on our way to success.

References

[1] B. Zhu, W. Yang, H. Wang and Y. Yuan, "A hybrid deep learning model for consumer credit scoring," 2018 International Conference on Artificial Intelligence and Big Data (ICAIBD), Chengdu, 2018, pp. 205-208.

[2] Y. Liu, A. Ghandar and G. Theodoropoulos, "A Metaheuristic Strategy for Feature Selection Problems: Application to Credit Risk Evaluation in Emerging Markets," 2019 IEEE Conference on Computational Intelligence for Financial Engineering & Economics (CIFEr), Shenzhen, China, 2019, pp. 1-7.

[3] C. R. D. Devi and R. M. Chezian, "A relative evaluation of the performance of ensemble learning in credit scoring," 2016 IEEE International Conference on Advances in Computer Applications (ICACA), Coimbatore, 2016, pp. 161-165.

[4] L. Wang, Y. Chen, Y. Zhao, Q. Meng and Y. Zhang, "Credit Management Based on Improved BP Neural Network," 2016 8th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC), Hangzhou, 2016, pp. 497-500.

[5] S. Kalaycı, M. Kamasak and S. Arslan, "Credit risk analysis using machine learning algorithms," 2018 26th Signal Processing and Communications Applications Conference (SIU), Izmir, 2018, pp. 1-4.

[6] T. N. Pandey, A. K. Jagadev, S. K. Mohapatra and S. Dehuri, "Credit risk analysis using machine learning classifiers," 2017 International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS), Chennai, 2017, pp. 1850-1854.

[7] L. Wei, W. Li and Q. Xiao, "Credit Risk Evaluation Using: Least Squares Support Vector Machine with Mixture of Kernel," 2016 International Conference on Network and Information Systems for Computers (ICNISC), Wuhan, 2016, pp. 237-241.

[8] A. Lawi, F. Aziz and S. Syarif, "Ensemble GradientBoost for increasing classification accuracy of credit scoring," 2017 4th International Conference on Computer Applications and Information Processing Technology (CAIPT), Kuta Bali, 2017, pp. 1-4.

[9] S. Dahiya, S. S. Handa and N. P. Singh, "Impact of bagging on MLP classifier for credit evaluation," 2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom), New Delhi, 2016, pp. 3794-3800.

[10] L. E. Boiko Ferreira, J. P. Barddal, H. M. Gomes and F. Enembreck, "Improving Credit Risk Prediction in Online Peer-to-Peer (P2P) Lending Using Imbalanced Learning Techniques," 2017 IEEE 29th International Conference on Tools with Artificial Intelligence (ICTAI), Boston, MA, 2017, pp. 175-181.

[11] A. Gahlaut, Tushar and P. K. Singh, "Prediction analysis of risky credit using Data mining classification models," 2017 8th International Conference on Computing, Communication and Networking Technologies (ICCCNT), Delhi, 2017, pp. 1-7.