The Application of Tree-based model to Unbalanced German Credit Data Analysis

Zhengye Chen[1]

[1]Allendale Columbia School, 519 Allens Creek Road, Rochester 14618, NY, USA

Abstract. With the development of financial consumption, demand for credit has soared. Since the bank has detailed client data, it is important to build effective models to distinguish between high-risk groups and low-risk groups. However, traditional credit evaluation methods including expert opinion, credit rating and credit scoring are very subjective and inaccurate. Moreover, the data are highly unbalanced since the number of high-risk groups is significantly less than that of low-risk groups. Progress in machine learning makes it possible to conduct accurate credit analysis. The tree-based machine learning models are particularly suitable for the unbalanced credit data by weighting the credit individuals. We apply a series of tree-based machine learning models to analyze the German Credit Data from the UCI Repository of Machine Learning Databases.

1 Introduction

With the development of economic and information globalization, a lot of business data and customer information data are collected and saved in banking system. How to analyze and utilize these data effectively and discover new business patterns are important issues. To support decision making and risk management in banking system, a sound personal credit analysis system is essential. Moreover, a reliable and accurate method to distinguish between high-risk groups and low-risk groups can not only brings great convenience to the credit loan business, but also create great economic benefits for banks.

There are several traditional credit evaluation methods [1,2], such as expert method, credit rating and credit scoring. Expert method is a qualitative analysis method, which relies mainly on the subjective judgment of experts. The evaluation conclusions of different experts are often different and difficult to form a unified standard, which makes it hard to apply in practice. The rating method is a method of quantifying customer credit rating after comprehensive analysis of customer information. However, the credit related factors and features for different customers are not uniform, which also makes the evaluation results highly subjective. Credit scoring method looks for the main cause or factors of the default, and then computes the score by a weighted or comprehensive assessment. The scores are used to assess credit risk for different individuals. Similar to credit rating, credit scoring method is also subjective since the risk factors and weights of these factors are subjective setting. Generally speaking, traditional credit evaluation methods are subjective in different degree, which reduced the accuracy of credit evaluation. Moreover, these traditional credit evaluation methods do not work with the huge amount of data, which is often the case in the big data area. The mass credit data in bank system also make the traditional methods inefficient in credit evaluation. In summary, traditional credit evaluation methods are lack of accuracy and efficiency in the current situation.

With the increasing diversification of financial assets and complication of risk, the traditional credit evaluation methods, which are mainly qualitative, and rely heavily on individual subjective ideas and experience knowledge, have failed to meet the needs of risk management. With the development of Internet big data [3], the traditional credit evaluation methods are also severely limited in new data portraits and business scenarios. In recent years, machine learning has developed rapidly, and have made great breakthroughs in information retrieval, language recognition and disease identification. In simple terms, machine learning refers to build models using existing data and make predictions about the future. Machine learning [4] transforms the process of human thinking and induction into computer learning and modeling. With the development of computational science, Machine learning algorithms and techniques can handle tens of thousands of data very efficiently, which increase computational efficiency greatly in comparison to traditional methods. With the help of Machine learning techniques, the data mining technology is used to construct the personal credit analysis model and assist the bank's personal credit decision. Machine learning models are more comprehensive, scientific, objective and fair, which can improve the accuracy of credit evaluation significantly. In summary, credit evaluation methods based on machine learning models are more accurate and efficient than traditional credit evaluation methods.

Credit evaluation analysis can be understood as a representative classification problem in machine learning, since our goal is distinguishing between high-risk groups and low-risk groups with the information like gender, age and savings account. Simply speaking, we want to build a model to classify clients into high-risk groups who are more likely to default or low-risk groups who are more reliable, with some available information data in banking system. There are many machine learning models to deal with this classification problem, such as K-Nearest Neighbor Categorization Algorithm (KNN) [5], Naive Bayes (NB) [6], and Artificial Neural Network (ANN) [7]. However, there are two unique features in credit evaluation analysis. The amount of credit data is usually large, while the number of high-risk individuals is significantly less than low-risk individuals. This is quiet intuitional and reasonable, since most individuals have good credit in real life and few individuals are not credible. Although the number of high-risk individuals is smaller, identification of high-risk individuals is more important than identification of low-risk individuals for banks, since high-risk individuals may bring huge loss to the bank. This kind of data is called unbalanced data in machine learning field. The unbalanced feature presents great challenge in credit evaluation analysis, because the number of important high-risk individuals is smaller and the corresponding information is less. In this paper, we employ several tree-based machine learning models to conduct credit evaluation, because these tree-based models can handle the unbalanced feature of credit data by weighting on the individual samples.

Tree-based models are classic algorithms in machine learning fields, and are widely used in artificial intelligence, biomedical and automation. Tree-based models are known and popular for their interpretability, accuracy and efficiency. A decision tree [8] is a decision support tool that uses a tree-like graph or model of decisions and their possible consequences. Decision trees can be seen as generative models of induction rules from empirical data. An optimal decision tree [9] is then defined as a tree that uses least levels to account for most information. Several algorithms have been proposed to build the decision tree, such as ID3, ID4 and CART [10]. Decision tree is a white box model and simple to understand and interpret. A single decision tree is unstable and low-accuracy. Therefore, machine learning ensemble meta-algorithms are used to improve the stability and accuracy. Two representative machine learning ensemble meta-algorithms are Adaptive Boosting (AdaBoost) [11] and Bootstrap aggregating (Bagging) [12]. Adaptive Boosting is formulated by Yoav Freund and Robert Schapire. The output of weak learners (for example, decision trees classifiers) is combined into a weighted sum that represents the final output of the boosted classifier. Bootstrap aggregating is also a machine learning ensemble meta-algorithm designed to improve the stability and accuracy of machine learning algorithms used in statistical classification. Bootstrap aggregating was proposed by Leo Breiman in 1994 to improve classification by combining classifications of randomly generated training sets. Models (for example, decision trees classifiers) are fitted using the bootstrap samples and combined by voting [13]. Random forest [14, 15] is also an ensemble learning method for classification and it is an extension of the bagging algorithm. Random forest is just slightly different from bagging: they select both a random subset of the features ("feature bagging") and a random subset of samples in the learning process.

The following of this paper is organized as follows. In Section 2, we describe the data and background briefly. In Section 3, we build credit evaluation models with a series of aforementioned tree-based machine learning models, including decision tree, Adaboost, Bagging and random forest. We also compare the performance of these models in different perspective. Finally, we make a brief conclusion about our results and discuss the advantages and disadvantages of our methods.

2 Data Description

The German Credit Data from the UCI Repository of Machine Learning Databases is public available at https://archive.ics.uci.edu/ml/datasets/statlog+(german+credit+data). This dataset classifies people described by a set of attributes as good or bad credit risks. There are 700 samples belonging to good credit risk and 300 samples belonging to bad credit risk. As we stated in the introduction, the number of bad credit individuals are more than that of good credit individuals, which embody the unbalanced feature of credit data. There are 20 attributes are measured in the German Credit Data, including 7 numerical attributes and 13 categorical attributes. These attributes are the risk factors used to distinguish between high-risk groups and low-risk groups. The detailed information about these attributes is listed in Table 1. We divided the data into two sets, one for training, and the other one for testing. To ensure the practicality of the model, we randomly picked 700 samples out of 1000 and put them into the set for training. The 300 samples that were left automatically went into the set for testing. In order to find an optimal method the divide the samples, we applied various methods on the data we gathered without changing the structure above.

Table 1. The detailed information about data attributes..

	Attribute	Type
1	Status of existing checking account	qualitative
2	Duration in month	numerical
3	Credit history	qualitative
4	Purpose	qualitative
5	Credit amount	numerical
6	Savings account/bonds	qualitative
7	Present employment since	qualitative
8	Installment rate in percentage of disposable income	numerical
9	Personal status and sex	qualitative
10	Other debtors / guarantors	qualitative
11	Present residence since	numerical
12	Property	qualitative
13	Age in years	numerical

14	Other installment plans	qualitative
15	Housing	qualitative
16	Number of existing credits at this bank	numerical
17	Job	qualitative
18	Number of people being liable to provide maintenance for	numerical
19	Telephone	qualitative
20	Foreign worker	qualitative

3 Models

3.1.Decision Tree

Decision tree model is commonly used in data mining [16] and machine learning for its interpretability and visualization. A tree is built or trained to predict the value of dependent variable with several independent variables (namely features) [17]. Each node in the tree represents an input variable, and edges to children nodes imply possible values of that independent variable. The dependent variables are divided into leaf nodes, and the path from the root to the leaf is a predictive progress. A tree is learned by splitting the data set into subsets based on specific criterion. This process is repeated on each subset in a recursive manner called recursive partitioning. Several decision-tree algorithms have been proposed for tree learning, including ID3, C4.5, CART, MARS and so on. Decision tree is a white box model, and simple to understand and interpret. It requires little data preparation and still performs well with large datasets. However, trees do not tend to be as accurate as other machine learning methods [18], and a single tree is very non-robust and stable.

We first build a decision tree to analyze our data. The tree model we built divided the samples into good or bad depending on the values of different attributes like duration, employment, purpose, credit, and etc. However, the problem of overfitting always comes along with an excessively detailed tree. In order to prevent overfitting, we need to control the number of branches within a reasonable range by cutting the extra branches (called pruning). The complexity parameter was introduced for the tradeoff between predictive ability and overfitting in the pruning progress. The complexity parameter (cp) is used to control the size of the decision tree and to select the optimal tree size. When the error is at its lowest point, the corresponding cp value is the number we should adopt to build our model.

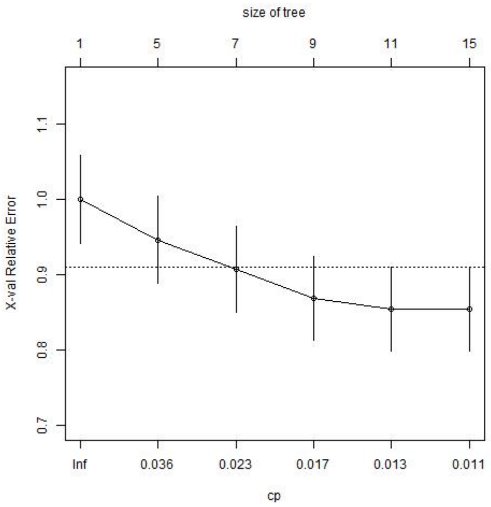


Figure 1: The Relative Error for different cp parameter.

As shown in Figure 1, the error is lowest when the cp value is 0.012. We applied this cp value to prune the original tree and obtained a new tree as shown in Figure 2. The accuracy of the final tree model on the training data and testing set are presented in Table 2. The prediction of bad customers tended to be very ineffective, which was totally different from the prediction of good ones.

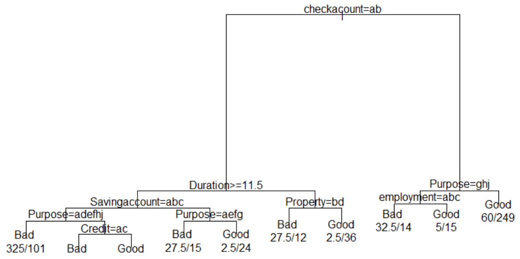
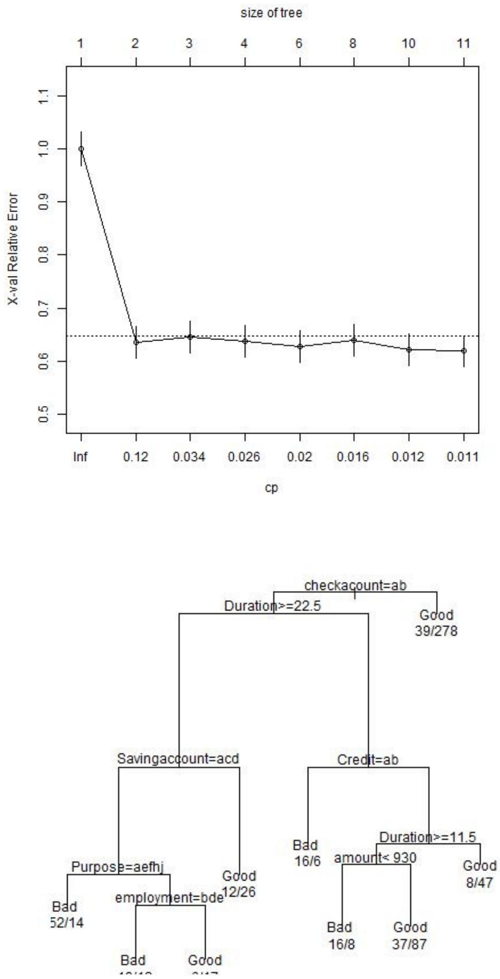


Figure 2: The final tree model after pruning.

Table 2: The accuracy of tree model on the training data and testing set.

Training data			Testing data		
	Bad	Good		Bad	Good
Bad	0.50244	0.497561	Bad	0.34737	0.65263
Good	0.08081	0.919192	Good	0.09756	0.90244

The emergence of unbalanced predictions was owing to the inadequate number of bad samples. To solve the problem of unbalanced predictions, we put a weight on bad customers, which are set to be 2.5: 1. Similar to the process of building unweighted tree, we first picked the appropriate cp value according to the cp diagram. Then, we used the cp value to prune the weighted tree and finally arrive at the tree model shown in Figure 3. The accuracy of the weighted tree model on the training data and testing set are shown in Table 3. The prediction accuracy for the bad customers is improved significantly with this method. It is easy to see that with a little loss of accuracy for good customers prediction, the accuracy of predicting bad customers increase significantly. Comparing to the unweight method and the one without pruning, giving some weight to the bad customers rendered the best prediction overall.



Figrure 3: The cp diagram and weighted tree model.

Table 3: The accuracy of weighted tree model on the training data and testing set.

Training data			Testing data		
	Bad	Good		Bad	Good
Bad	0.85854	0.141463	Bad	0.63158	0.36842
Good	0.31313	0.686869	Good	0.34146	0.65854

3.2.Adaboost

In machine learning, an effective method to improve the performance and stability is ensemble learning [19]. Adaboost[11] is a representative ensemble learning algorithm, which combines the predictions of several trees with a weight sum to form the final prediction. In the training process, the weights of samples are updated according to those misclassified samples in previous training process. The weights of weak classifiers (trees) are assigned according to their prediction accuracy. Specifically, classifiers with higher accuracy are more reliable and assigned with higher weights. The individual learners (trees) can be weak, but as long as the performance of weak learnings is slightly better than random guessing, the final model can converge to a strong learner.

We also applied adaboost into our analysis of the credit dataset. To establish an adaboost model, multiple decision trees were needed. Similar to the number of branches in the decision tree model, there is also an optimal number of trees for the adaboost model to achieve its best performance. We compute the traing error and testing error of adaboost models with different number of trees, and the results are plotted in Figure 4. We could easily determine the optimal number of trees as 20. The accuracy of adaboost model on the training data and testing set are listed in Table 4. The prediction accuracy is considered reasonably effective. If a random number was used, as shown in Figure 4, the accuracy of test data would not reach its highest point, although we could reduce the error of train data by keep on adding the amount of trees, which would not contribute to the performance to the final model.

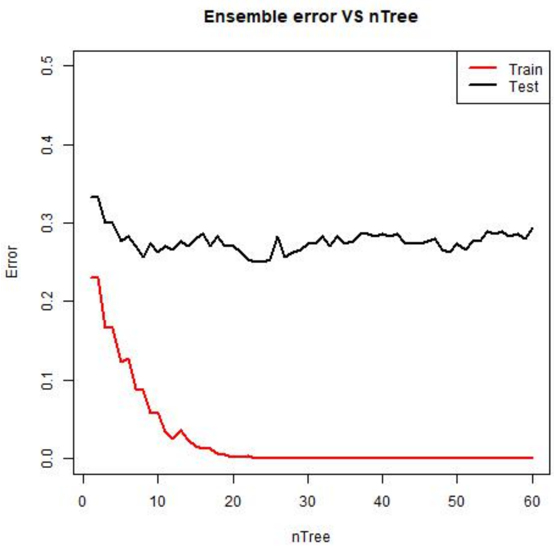


Figure 4: The traing error and testing error of adaboost models with different number of trees.

Table 4: The accuracy of adaboost model on the training data and testing set.

Training data			Testing data		
	Bad	Good		Bad	Good
Bad	0.85854	0.141463	Bad	0.60318	0.39683
Good	0.31313	0.686869	Good	0.24051	0.75949

Generally speaking, adaboost outperforms the decision trees in prediction for the good customers, but it has a slightly lower accuracy for the bad customers prediction. Overall, the advantage on predicting the good customers over decision trees could cover its lower accuracy of the prediction of the bad ones, so adaboost is a greater method than decision trees.

3.3.Bagging

Bagging [12] is also an effective and popular ensemble method in machine learning. In the progress of bagging iterations, new training sets are generated by sampling with replacement. This kind of sample is known as a bootstrap samples. Decision trees are trained with these bootstrap samples, and the predictions of these trees are combined by voting or averaging for a consensus prediction. Although bagging very easy to implement, it can not only improve the stability and accuracy, but also reduces variance and helps to avoid overfitting [13].

Similar to adaboost, we also need to determine how many trees we should use to build bagging model. We compute the cross validation (cv) error of bagging models with different number of trees, and the results are plotted in Figure 5. By finding the minimum value of the graph, we also decided to use 20 trees to accomplish the bagging model for a fair comparison with the adaboost model. The results of the prediction accuracies are shown in Table 5.

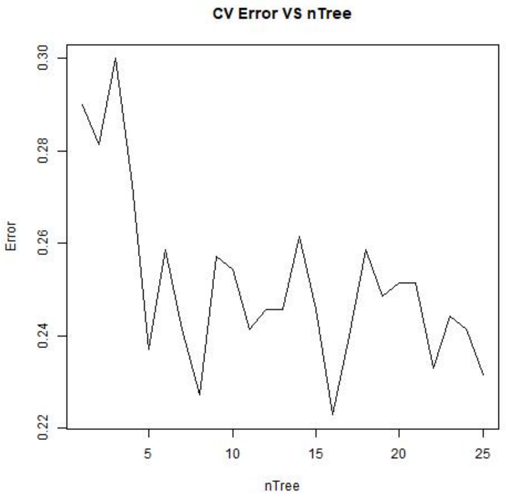


Figure 5: The cv error of bagging models with different number of trees.

Table 5: The accuracy of bagging model on the training data and testing set.

Training data			Testing data		
	Bad	Good		Bad	Good
Bad	0.88276	0.117241	Bad	0.66667	0.33333
Good	0.13874	0.861261	Good	0.25	0.75

As shown in Table 5, when it comes to the prediction of test data, the bad customers are still difficult to predict comparing to the good customers. This phenomenon is very similar to the performances of the models built before, and we also try to give more weight to the bad customers to solve the problem as we did in weighted tree model. Consequently, we added the same weight 2.5:1 to the bad customers. Again, we compute the cv error of weighted bagging models with different number of trees, and the results are shown in Figure 6. We also set the optimal number of trees as 20, and the prediction accuracies are shown in Table 6. The results in Table 6 show a significant improvement in the prediction of bad customers just like the weighted tree model. For both types of customers, the method of bagging achieves a prediction accuracy more than 70%, which easily beat the adaboost model, let alone decision trees.

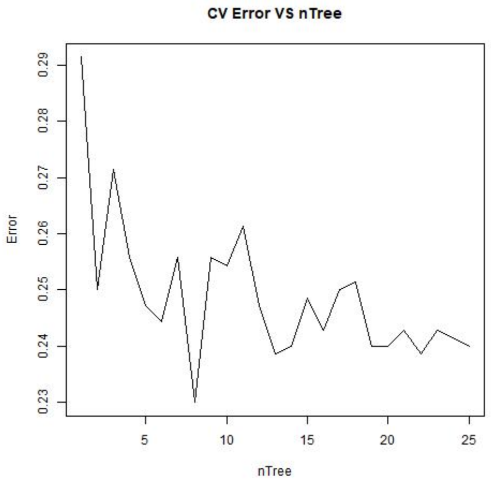


Figure 6: The cv error of weighted bagging models with different number of trees.

Table 6: The accuracy of weighted bagging model on the training data and testing set.

Training data			Testing data		
	Bad	Good		Bad	Good
Bad	0.96032	0.039683	Bad	0.70732	0.29268
Good	0.14634	0.853659	Good	0.25483	0.74517

3.4.Random Forest

A random forest classifier [14, 15] is a specific type of bootstrap aggregating (bagging). Random forest is just slightly different from bagging: they select both a random subset of the features ("feature bagging") and a random subset of samples in the learning process. Intuitively, the stronger predictors for the dependent variable, more times these features would be selected in the feature bagging progress. Similar to bagging, the bootstrapping procedure achieves better performance because it decreases the variance of the model without increasing the bias [20]. Moreover, random forests can avoid the overfitting to the training date set [21].

Similar to the CV error in the bagging model, the out-of-bag (OOB) error of random forest are commonly used to measure the prediction ability of random forests. To determine the optimal numbers of trees for the whole model, the OOB error for random forest with different number of trees are calculated and the results are plotted in Figure 6. For the same purpose of avoiding overfitting, we chose to generate 200 trees to establish the random forest model, which is reasonably accurate enough. The prediction error of corresponding model on the training data and testing data are shown in Table 7.

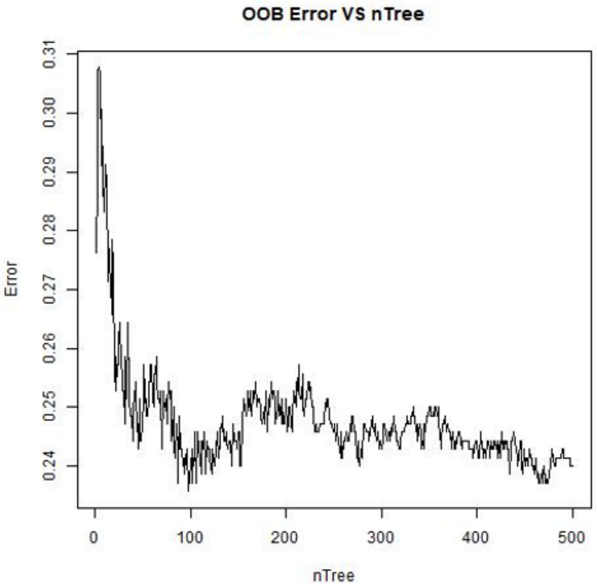


Figure 7: The OOB error of random forest with different number of trees.

Table 7: The accuracy of random forest on the training data and testing set.

Training data			Testing data		
	Bad	Good		Bad	Good
Bad	1	0	Bad	0.32632	0.67368
Good	0	1	Good	0.06829	0.93171

Obviously, the similar problem of ineffective prediction of the bad customers appears again. We utilized, as always, the method of weighting the bad customers to boost the prediction of the bad ones. A new random forest model with a weight of 2.5:1 on bad customers was built, and the new model showing OOB error vs the number of trees is plotted in Figure 7. The optimal number of trees is also set as 200, and the prediction error of this weighted random forest are shown in Table 8. The sacrifice of the accuracy of the prediction of the good ones again resulted in a more precise model of predicting the bad ones, bringing it to more than 60%. However, it still seems like an accuracy too low to the results of bagging model, which generates more than 70% correct predictions for both sorts of customers, whereas the adaboost model and decision trees would not overcome the random forest model.

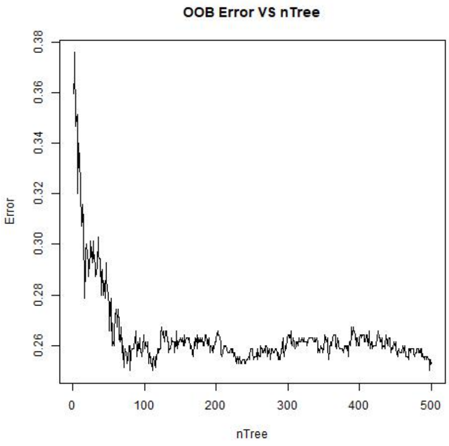


Figure 8: The OOB error of weighted random forest with different number of trees.

Table 8: The accuracy of weighted random forest on the training data and testing set.

Training data			Testing data		
	Bad	Good		Bad	Good
Bad	0.97561	0.02439	Bad	0.63158	0.36842
Good	0.09091	0.909091	Good	0.22439	0.77561

4 Summaries and Discussions

We established 4 models by using 4 different methods (Decision trees, adaboost, bagging, and random forest) to find a way to predict if a customer is trustable or not to lend loan to. When building the models, we met a number of problems such as low efficiency and bad accuracy. We solved the problems through using a more advanced model like bagging to decision trees, which effectively improved the situation. However, the most common puzzle we met was the imbalance between the predictions of good customers and bad customers--the prediction of good customers always tended to be more precise than the prediction of bad ones. The main reason was that the size of the bad samples was too small for building a good model. We eventually solved this problem by add more weights on the prediction of the bad ones, which was almost equivalent to adding more bad samples, so it helps the balance the accuracy of the predictions of both types of customers.

The project we were doing is extremely significant to the economy, because a huge amount of economic loss caused by false recognition to customers can be avoided if we can predict if a borrower is trustable. Moreover, banks can lend this money to the good customers who are really in need, so the customers become capable of making more profit. Eventually, the economy of the entire society would be prominently promoted. However, our models are not mature enough to achieve this image yet due to their insufficient accuracy. Fortunately, there are some directions for us to develop. For instance, we could advance our models by integrating them together, which would theoretically boost the accuracy of the models. With this direction, hopefully we can successful create a model that is able to promote our economy on a huge extent in the future**.**

References

1. T Michalski, E Gołebiowska. Taxonomy methods in credit risk evaluation[J]. *International Advances in Economic Research*, **2(4)**:409-412 (1996).
2. N Dardac. Credit Institutions Management Evaluation using Quantitative Methods[J]. *Theoretical & Applied Economics*, **2(497)**: 35-40(2006).
3. E Brynjolfsson, A Mcafee. Big Data's Management Revolution[J]. *Harvard Business Review*, **90(10)**:60 (2012).
4. Mitchell. Machine Learning[M]. *China Machine Press ;McGraw-Hill Education (Asia)*, (2003).
5. N.S Altman, "An introduction to kernel and nearest-neighbor nonparametric regression". *The American Statistician.* **46 (3)**: 175–185 (1992).
6. Rish, Irina, An empirical study of the naive Bayes classifier. *IJCAI Workshop on Empirical Methods in AI* (2001).
7. "Artificial Neural Networks as Models of Neural Information Processing | Frontiers Research Topic". Retrieved 2018-02-20.
8. J R Quinlan, "Simplifying decision trees". *International Journal of Man-Machine Studies*. **27 (3)**: 221(1987).
9. R. Quinlan, "Learning efficient classification procedures", *Machine Learning: an artificial intelligence approach*, p. 463-482(1983).
10. P E Utgoff, Incremental induction of decision trees. *Machine learning*, **4(2)**, 161-186(1989).
11. Y Freund;E R Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting". *Journal of Computer and System Sciences*. **55**: 119(1997).
12. L Breiman, "Bagging predictors". *Machine Learning*. **24 (2)**: 123–140(1996).
13. Shinde, Amit, A Sahu, D Apley, and G Runger. "Preimages for Variation Patterns from Kernel PCA and Bagging." *IIE Transactions*, **Vol.46**, Iss.5(2014).
14. Ho, T Kam, Random Decision Forests. Proceedings of the 3rd International Conference on Document Analysis and Recognition, Montreal, QC, pp. 278–282(1995).
15. Ho T Kam, "The Random Subspace Method for Constructing Decision Forests". *IEEE Transactions on Pattern Analysis and Machine Intelligence*. **20 (8)**: 832–844(1998).
16. L Rokach; O Maimon, Data mining with decision trees: theory and applications. *World Scientific Pub Co Inc* (2008).
17. L Breiman; J H Friedman.; R A Olshen; C J Stone, Classification and regression trees. Monterey, *CA: Wadsworth & Brooks/Cole Advanced Books & Software*(1987).
18. J Gareth; D Witten; T Hastie; R Tibshirani, An Introduction to Statistical Learning. *New York: Springer.* p. 315(2015).
19. D Opitz.; R Maclin, "Popular ensemble methods: An empirical study". *Journal of Artificial Intelligence Research*.(1999).
20. L Breiman, "Random Forests". *Machine Learning*. **45 (1)**: 5–32(2001).
21. T Hastie; R Tibshirani; J Friedman, The Elements of Statistical Learning (2nd ed.). *Springer*(2008).