Spring 1-26-2020

Developing and improving risk models using machine-learning based algorithms

Yan Wang
ywang63@students.kennesaw.edu

Sherry Ni
sni@kennesaw.edu

Recommended Citation
Yan Wang, Xuelei Sherry Ni. Developing and improving risk models using machine-learning based algorithms. Proceeding of the 2019 ACM Southeast Conference, pp. 281-282, 2019.

Developing and Improving Risk Models using Machine-learning Based Algorithms

Yan Wang
Kennesaw State University
Kennesaw, GA, USA
ywang63@students.kennesaw.edu

Xuelei Sherry Ni
Kennesaw State University
Kennesaw, GA, USA
sni@kennesaw.edu

ABSTRACT

The objective of this study is to develop a good risk model for classifying business delinquency by simultaneously exploring several machine learning based methods including regularization, hyper-parameter optimization, and model ensembling algorithms. The rationale under the analyses is firstly to obtain good base binary classifiers (include Logistic Regression (*LR*), K-Nearest Neighbors (*KNN*), Decision Tree (*DT*), and Artificial Neural Networks (*ANN*)) via regularization and appropriate settings of hyper-parameters. Then two model ensembling algorithms including bagging and boosting are performed on the good base classifiers for further model improvement. The models are evaluated using accuracy, Area Under the Receiver Operating Characteristic Curve (AUC of ROC), recall, and F1 score via repeating 10-fold cross-validation 10 times. The results show the optimal base classifiers along with the hyper-parameter settings are *LR* without regularization, *KNN* by using 9 nearest neighbors, *DT* by setting the maximum level of the tree to be 7, and *ANN* with three hidden layers. Bagging on *KNN* with K valued 9 is the optimal model we can get for risk classification as it reaches the average accuracy, AUC, recall, and F1 score valued 0.90, 0.93, 0.82, and 0.89, respectively.

CCS CONCEPTS

• **Computer systems organization → Machine learning; Modeling**.

KEYWORDS

Improve Risk Model, Machine Learning

ACM Reference Format:
Yan Wang and Xuelei Sherry Ni. 2019. Developing and Improving Risk Models using Machine-learning Based Algorithms. In *2019 ACM Southeast Conference (ACMSE 2019), April 18–20, 2019, Kennesaw, GA, USA.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3299815.3314478

1 PROBLEM AND MOTIVATION

Many studies have demonstrated that the performance of risk models can be improved by using many machine-learning based methods including regularization, hyper-parameter optimization, and ensembling algorithms [4]. In this study, we aim to develop a good risk model for classifying business delinquency by jointly and comprehensively exploring the effect of the above-mentioned model-improving algorithms.

2 BACKGROUND AND RELATED WORK

Logistic Regression (*LR*) is a widely used technique for binary classification because of its strong interpretability and competitive performance. Moreover, regularized *LR*, which leads to a substantial decrease in variance and prediction error, outperforms *LR* in some studies. Two commonly used versions of *LR* include $L1$-regularized *LR* and $L2$-regularized *LR*, with the former version penalizes the $L1$ norm of the coefficients while the latter version penalizes the $L2$ norm of the coefficients [1]. Decision Tree (*DT*) is another widely acceptable binary classification technique but different settings of its hyper-parameters could largely affect its performance [2]. Similarly, in an Artificial Neural Network (*ANN*) and k-Nearest Neighbor (*KNN*), careful tuning of the hyper-parameters can improve their performance [6] [8]. Moreover, model ensembling is another effective way to improve the model performance [7] [9].

3 APPROACH

The dataset used in this study contains the financial information of 9500 US companies in 2014 with the delinquent rate valued 49.69%. We randomly select 80% of the data as the training set and use the rest as the testing set. The dimensionality of the features is reduced to 100 via hierarchical variable clustering in the data preprocessing stage. Four base classifiers including *KNN*, *LR*, *DT* and *ANN* are developed by using regularization and hyper-parameter optimization algorithms. To be specific, *LR* is regularized by using both $L1$ and $L2$-regularization. In *KNN*, the hyper-parameter 'K', denoting the number of nearest neighbors, is tuned by taking a series of values ranging from 3 to 13. In *DT*, we tuned 'max_depth', which denotes the maximum level of the tree structure, by using different values ranging from 5 to 15. In *ANN*, the hyper-parameter 'layer_size', representing the number of hidden layers with units in each layer, is tuned by taking a series of values of '50', '50_25', '50_25_13', and '50_25_13_6'. For example, the value of '50_25_13' means there are three hidden layers in the *ANN* and the number of units in each layer is 50, 25, and 13, respectively. Area Under the Receiver Operating Characteristic Curve (AUC of ROC), recall, and F1 score are used as evaluation metrics based on testing set by repeating 10-fold cross validation 10 times [3] [5]. After building the base classifiers as accurate as possible, two model ensembling techniques (including bagging and boosting) are performed to examine whether the performance can be further improved or not.

ACMSE 2019, April 18–20, 2019, Kennesaw, GA, USA

ACM ISBN 978-1-4503-6251-1/19/04.
https://doi.org/10.1145/3299815.3314478

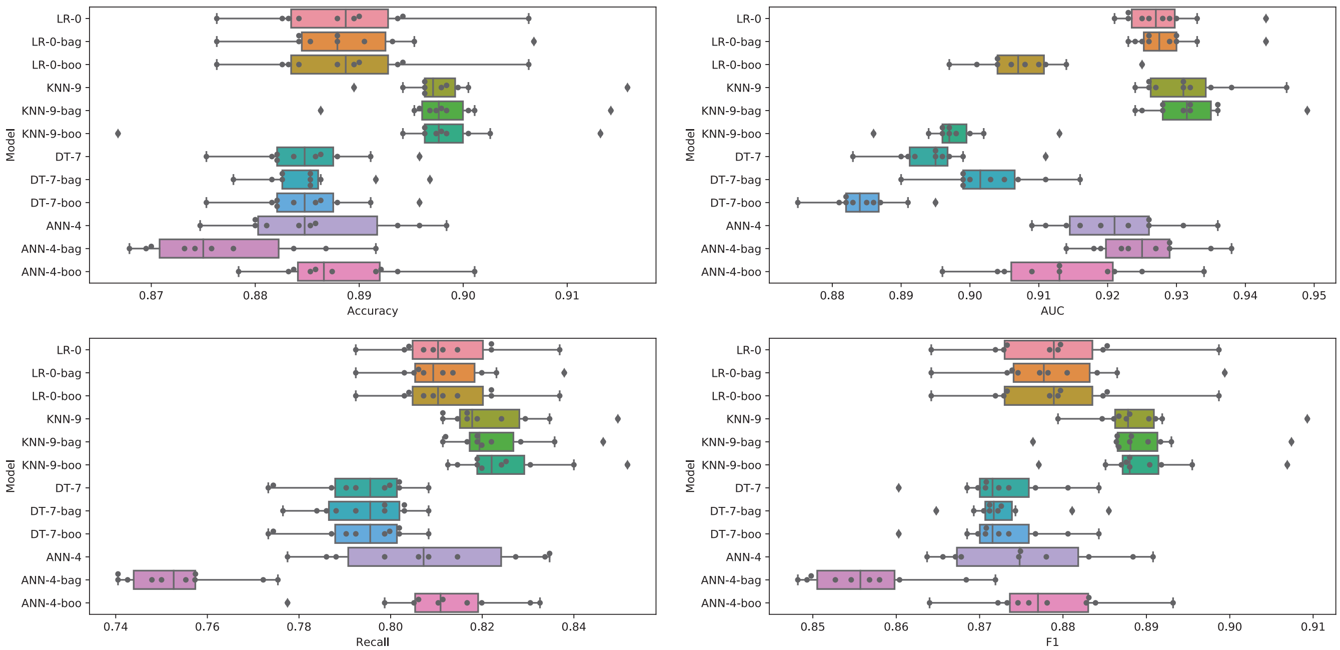

Figure 1: Average Performance of the Best Base Classifiers and their Ensemble Models from 10 Times 10-fold Cross Validation

4 RESULTS

The best base classifiers along with their hyper-parameter setting are *LR*-0 (i.e, *LR* without regularization), *KNN*-9 (i.e., *K* valued 9), *DT*-7 (i.e., 'max_depth' valued 7), and *ANN*-4 (i.e., contains three hidden layers and having units valued 50, 25, and 13 respectively in each layer). Figure 1 shows the result of the best classifiers as well as their ensemble models. The post-fix bag and boo denote bagging and boosting, respectively. In *LR*, bagging and boosting cannot significantly improve the model performance with respect to accuracy, recall, and F1 score. Quite unexpectedly, boosting on *LR* even hurt the AUC by a large extent. In *KNN*, both bagging and boosting are beneficial when considering accuracy, recall, and F1 score measures. On the contrary, boosting on *KNN* decrease AUC significantly. Bagging on *DT* outperforms base *DT* as it produces significantly higher AUC and marginally higher accuracy. However, boosting significantly decrease AUC of *DT*. Compared with base *ANN*, boosting on *ANN* is beneficial in terms of accuracy, recall, and F1 score while it hurt AUC significantly. Bagging on *ANN* can significantly decrease accuracy, recall, and F1 score, indicating that *ANN* is not a good base classifier to be bagged on. By comparing all the aforementioned results, we conclude that *KNN*-9-bag (i.e., bagging on the base classifier *KNN* with *K* valued 9) is the optimal risk model in our study with average accuracy, AUC, recall, and F1 score valued 0.90, 0.93, 0.82, and 0.89, respectively.

5 CONCLUSIONS AND CONTRIBUTIONS

In this paper, we aim at developing and improving the risk modeling via the widely used machine-learning based algorithms including regularization, hyper-parameter optimization, and ensembling simultaneously. The results show that the optimal hyper-parameter settings for the base classifiers are *LR* without regularization, *KNN* by using 9 nearest neighbors, *DT* by setting the level of the tree to be 7, and *ANN* with three hidden layers. The optimal model we get for classifying business delinquency is through bagging on *KNN* with *K* valued 9, which reaches the average accuracy, AUC, recall, and F1 score valued 0.90, 0.93, 0.82, and 0.89, respectively. Although different conclusions may be obtained because of various dataset used in the future, the study methodology provided by us is a good reference for studies that aiming to improve risk modeling using machine-learning based algorithms.

REFERENCES

[1] G. Fort and S. Lambert-Lacroix. 2004. Classification using Partial Least Squares with Penalized Logistic Regression. *Bioinformatics* 21, 7 (2004), pp. 1104–1111.
[2] E. Hazan, A. Klivans, and Y. Yuan. 2017. Hyperparameter Optimization: A Spectral Approach. *arXiv preprint arXiv:1706.00764* (2017).
[3] M. Hossin and MN. Sulaiman. 2015. A Review on Evaluation Metrics for Data Classification Evaluations. *International Journal of Data Mining & Knowledge Management Process* 5, 2 (2015), 1.
[4] K. Ozcift and A. Gulten. 2011. Classifier Ensemble Construction with Rotation Forest to Improve Medical Diagnosis Performance of Machine Learning Algorithms. *Computer methods and programs in biomedicine* 104, 3 (2011), pp. 443–451.
[5] T. Saito and M. Rehmsmeier. 2015. The Precision-recall Plot is More Informative than the ROC Plot when Evaluating Binary Classifiers on Imbalanced Datasets. *PloS one* 10, 3 (2015), e0118432.
[6] J. Wang, P. Neskovic, and N. Cooper. 2006. Neighborhood Size Selection in the K-nearest-neighbor Rule using Statistical Confidence. *Pattern Recognition* 39, 3 (2006), pp. 417–423.
[7] Y. Wang, S. Ni, and J. Priestley. 2019. Improving Risk Modeling via Feature Selection, Hyper-parameter Adjusting, and Model Ensembling. *Global Journal of Economics and Finance* 3, 1 (2019), pp. 30–47.
[8] Y. Wang, S. Ni, and B. Stone. 2018. A Two-stage Hybrid Model by using Artificial Neural Networks as Feature Construction Algorithms. *International Journal of Data Mining & Knowledge Management Process (IJDKP)* (2018).
[9] C. Zhang and Y. Ma. 2012. *Ensemble Machine Learning: Methods and Applications*. Springer.