doi:10.3233/FAIA230838

Company Bankruptcy Prediction with Machine Learning Techniques

Caiwei SUN[1]
Data Science and Big Data Technology, Faculty of Statistics and Mathematics, Shandong University of Finance and Economics, Jinan, Shandong, 250220, China

Abstract: Bankruptcy prediction is an important economic problem. It is a crucial problem in finance, as successful prediction enables stakeholders to take early actions to reduce their economic losses. Machine learning can effectively predict a company's bankruptcy situation and provide timely reminders to the enterprise. In this study, a real-world dataset collected from the Taiwan market from 1999 to 2009 is used. The bankruptcy status is identified and set as the prediction target. An oversampling framework is used to balance the label distribution, and PCA is used for feature reduction. Four machine learning classifiers are implemented and compared, namely, support vector machine, random forest, AdaBoost and XGBoost. The numerical results demonstrate that the random forest classifier outperforms baselines in terms of the F1 score.

Keywords: Bankruptcy prediction, machine learning, random forest

1. Introduction

Bankruptcy prediction is a significant section in finance, and its importance can be reflected in the following three aspects. First, public policy makers of our government can take proactive steps to minimize the negative influence of bankruptcy. If our government finds the local bankruptcy rate to show an upward trend, it needs to think twice and come up with some changes and measures to deter this trend. Second, according to the company's quarterly or annual financial statements, managers can predict the bankruptcy rate of the company; therefore, they can quickly understand the company's operational status and make appropriate adjustments to avoid bankruptcy. Eventually, these related predictions not only allow investors to obtain more effective information and make more appropriate investments but can also reduce error rates and minimize avoidable losses.

Some bankruptcy prediction studies began in the 1930s. In the last decade, machine learning has undergone enormous development and made substantial progress, which has achieved good results. Through machine learning, we can use some state-of-the-art approaches to make predictions effectively on whether a company will go bankrupt and the probability of its bankruptcy. Machine learning has extensive applications in the financial and economic domains [1-4]. Machine learning can assist financial institutions in risk assessment, including credit risk, market risk, operational risk, and more.

[1] Corresponding author: Caiwei SUN, Data Science and Big Data Technology, Faculty of Statistics and Mathematics, Shandong University of Finance and Economics; e-mail: 2967318528@qq.com

Machine learning automatically identify potential risk factors and predict future risks. Linear, polynomial, and Radial Basis Function (RBF) kernels are used in SVM for risk assessment in [5]. The German Credit Data is a widely used benchmark dataset, and a novel approach to credit risk assessment is proposed using machine learning techniques. Different machine learning models have been used to make credit risk assessments.

Machine learning can help investors make better investment decisions. By analyzing historical data, machine learning algorithms can identify market trends and predict future stock price movements. Investors can use these predictions to develop better investment strategies. Two different machine learning algorithms are used and compared in [6]. Historical stock prices, trading volume, price-to-earnings ratio, and price-to-book ratio data are used. The study results suggest that machine learning techniques can be successfully applied to stock price prediction and can help investors make more informed investment decisions.

Financial institutions can use machine learning algorithms to detect fraudulent activities. Machine learning can analyze large volumes of transaction data and identify unusual transaction behaviors, thereby reducing the losses incurred by financial institutions. Various techniques are discussed in [7].

Machine learning can assist financial institutions in assessing the credit risks of loan applicants. By analyzing the historical data of borrowers, machine learning algorithms can predict their future repayment ability, helping financial institutions make better loan decisions. A credit scoring model in [8] is based on support vector machines (SVMs), which is a powerful machine learning algorithm with high prediction accuracy.

Quantitative trading is a method of using machine learning algorithms to execute trading decisions. Machine learning can analyze market data and predict future price movements. Traders can use these predictions to develop trading strategies and automatically execute trades. A reinforcement learning framework is realized in three instances in [9] with a convolutional neural network (CNN) [10], a basic recurrent neural network (RNN) [11, 12], and long short-term memory (LSTM) [13] for portfolio management.

In this paper, we collect data from the Taiwan market, which includes 95 financial variables. To solve the label imbalance problem, oversampling is used. PCA is used for feature reduction. We use and compare four machine learning classifiers in this study, namely, support vector machine, random forest, AdaBoost and XGBoost. Based on the experiments, it is concluded that the random forest classifier outperforms baselines in terms of the F1 score.

2. Related Work

In [14], based on the Korean bankruptcy dataset, the author first reviewed several approaches for handling the class imbalance problem. This paper found that ensemble-based models outperformed baselines in predicting bankruptcy on the Korean dataset in terms of the geometric mean and the area under the receiver operating characteristic (ROC) curve.

In [15], the authors use a Bayesian network model to predict the bankruptcy status of companies and employ the least absolute shrinkage selection operator (LASSO) method to select financial indicators as input features. The proposed method is applied to a dataset of 32,344 US companies from 1961 to 2018. The model proposed in this paper has better interpretability.

Based on a dataset from Taiwan companies, the authors in [16] use MLP for bankruptcy prediction, considering different settings. Different evaluation metrics are used to search for the optimal parameter settings and examine the differences between methods.

The MLPs are implemented using the open source Python module Scikit-learn. The authors employ the function MLPclassifier, an MLP classifier that uses the log-loss function. They create 12 different models, of which six use an MLP with one hidden layer and the other six use two hidden layers. To determine the proper number of neurons for our models M1–M6, they begin by calculating the accuracy of each model, starting with one neuron and then iteratively increasing the number up to 50. For models M12–M62, they calculate the accuracy for one neuron in each layer and then increase the number of neurons up to 20. In total, they compare 400 different architectures for every MLP with two hidden layers. For each model, the architecture with the highest accuracy is chosen. To train the different models, they set the batch size to 1, which means that weight adjustment occurs via online learning. They also set the parameter random state to 42 to randomly generate initial weight values, the parameter shuffle to true, and the initial learning rate of all models to 0.001.

A bankruptcy prediction model is proposed in [17] for small and medium-sized enterprises that uses transaction data and variables based on payment networks without requiring financial (accounting) data. To improve the interpretability of the model and increase the feature acquisition cost, the authors propose a two-stage multiobjective feature selection method and achieve similar performance with significantly less features.

In [18], classification is considered as a multiobjective optimization problem and constructs the predictive model as a whole. The MOCS algorithm, undersampling, oversampling, and no sampling methods are used in combination with static and dynamic classification models. The proposed algorithm is tested using a Russian data.

A combined approach is used in [19] using fuzzy set qualitative comparative analysis (fsQCA) and a convolutional neural network (CNN), where the CNN utilizes calibration variables from fuzzy sets to improve performance accuracy. The results show that the fuzzy convolutional neural network (FCNN) performs better than when using traditional methods.

Machine learning techniques are applied in [20] to a quarterly dataset covering the financial ratios of a large number of publicly listed companies in the United States from 1970 to 2019. The best model using XGBoost is used to predict the overall bankruptcy rate in the United States for the second half of 2020.

A Belgium financial dataset is used in [21] and XGBoost, SVM, and deep neural networks are applied. With the combination of return on assets, current ratio, and solvency ratio, bankruptcy can be predicted with a global accuracy of 82-83%.

Based on a large sample of 1824 American companies, it is shown in [22] that the use of the synthetic minority oversampling technique (SMOTE) to balance the training dataset significantly improves classification accuracy. The results demonstrate that combining SMOTE with cluster-based undersampling yields the best classification performance, with significant improvements in accuracy, particularly in terms of recall and AUC, proving the rationality of using synthetic sampling when training default prediction classifiers.

Different methods are used to solve the imbalance problem. Le (2022) [14] utilized oversampling methods including SMOTE, ADASYN, Borderline-SMOTE, SMOTE-Tomek, SMOTE-ENN, and CBoost, as well as cost-sensitive methods. Cao et al. (2022) [15] employed a Bayesian network model and the LASSO method. Brenes et al. (2022)

[16] used optimization algorithms to optimize neural networks. Kou et al. (2021) [17] used a two-stage multiobjective feature selection method and models including LDA, LR, SVM, DT, RF, XGB, and NN. Zelenkov and Volodarskiy (2021) [18] used SMOTE, ADASYN, RUS, Tomek, and AdaBoost. Jabeur and Serret (2023) [19] utilized fuzzy convolutional neural networks and the fsQCA method. Narvekar and Guha (2021) [20] used RF, SVM, and XGBoost, as well as Borderline-SVM, SMOTE, and ADASYN. Shetty et al. (2022) [21] employed XGBoost, SVM, and deep neural networks. Finally, Garcia (2022) [22] used the synthetic minority oversampling technique (SMOTE).

From the above studies, it can be seen that SMOTE and its variants are common oversampling methods that are widely used in various studies. Additionally, other oversampling methods such as ADASYN, RUS, Tomek, and Borderline-SMOTE have also been adopted in some studies. Furthermore, with the development of deep learning, models such as XGBoost, SVM, RF, and NN have also been widely applied in various studies. In addition, some studies have explored different methods and models, such as feature selection methods, Bayesian networks, and fuzzy convolutional neural networks. Therefore, it can be seen from these studies that different methods and models can complement each other and improve the effectiveness in addressing imbalanced data problems.

3. Methodologies

3.1 Machine Learning Classifiers

3.1.1 Decision Tree

A decision tree is a graphical representation of a decision-making process that involves mapping out all possible outcomes and the probabilities associated with them. It is a popular tool used in data mining and machine learning for both classification and regression analysis. The most commonly used algorithms for constructing decision trees are ID3, C4.5, and CART. The process of constructing a decision tree involves two steps. The first step is splitting the node; if we cannot determine the attribute of one node, we need to split it into two subnodes. The second step is selecting an appropriate threshold to minimize the training error. A decision tree requires supervised learning.

A decision tree has some merits and demerits, and one of its advantages is that it is easy to interpret and explain. However, decision trees can be prone to overfitting, especially when dealing with noisy or complex data, and may require pruning or other methods to reduce their complexity and improve their accuracy.

3.1.2 Bagging

The training processes of Bagging are as follows. Randomly select a set of m samples from a training dataset of size X, train a basic classifier C1 on these samples, assign a weight W1 to C1, and then repeat the process by randomly selecting another set of m samples. Finally, the classifiers are combined by linearly adding them with certain weights to form a strong classifier composed of the basic classifiers.

Bagging features include a parallel ensemble method in which each base model can be generated separately, independently, and without mutual influence. It mainly reduces variance and has no significant effect on bias. Therefore, it is suitable for models with high variance and low bias. In terms of bias, the bias after bagging is similar to that of a single model. In terms of variance, bagging can reduce variance.

It is possible to obtain the same data when choosing samples randomly. This will result in when sampling m samples each time, it is possible to draw the same data as before, which can lead to a poor performance of the strong classifier formed by a linear combination of classifiers.

3.1.3 Boosting

Boosting is a machine learning integration method that combines multiple weak learners into a single strong learner. Boosting sequentially trains each model, with each subsequent model focusing on samples that were misclassified by the previous model.

The characteristics of boosting include a sequential ensemble method, in which the base models generated in each iteration mainly improve the areas where the previous generation of base models performed poorly. By constantly iterating weak classifiers, bias can be reduced. Therefore, it is suitable for models with low variance and high bias.

3.1.4 Random Forest

Random forest is a bagging method that belongs to ensemble learning. Random forest is composed of multiple decision trees, and there is no correlation between different decision trees.

Assume a sample size of N and that each sample has M attributes. Based on this sample, constructing a random forest involves four steps. The first step is to randomly sample with replacement N times, each time only selecting one, which will form N samples. These samples are used to train a decision tree and serve as the root node samples. The second step is to randomly select m attributes from the M attributes when a decision tree needs to be split. The value of m should be smaller than M. A strategy is then used to select one of these m attributes as the classification attribute for the node. The third step is that each node in the decision tree formation process must be split according to step 2 until it can no longer be split. The fourth step is to build a large number of decision trees to form a random forest.

Random forest has many advantages, such as being able to handle high-dimensional datasets without the need for dimensionality reduction or feature selection. It can make a judgment about the importance of features and the interactions between different features. Random forest is not prone to overfitting, and its training speed is relatively fast, making it easy to implement in parallel. It can balance errors for imbalanced datasets and maintain accuracy even when a large portion of features are missing. However, random forest may experience overfitting in certain classification or regression problems with high levels of noise. Additionally, when applied to data with attributes of different values or properties with many possible values, the attribute weights produced by random forest may be unreliable.

3.1.5 AdaBoost

The core idea of AdaBoost is to combine multiple weak classifiers into a strong classifier. It iteratively weights the training data so that each weak classifier focuses more on the data that were misclassified by the previous classifiers. Each new weak classifier thus focuses on the data that previous classifiers failed to classify correctly, gradually improving the accuracy of the entire ensemble.

In AdaBoost, each weak classifier is trained on a specific set of features, and the outputs of all classifiers are combined using weighted voting to form the final prediction. AdaBoost updates the weights of each sample based on an exponential function of the error rate, making each classifier focus more on the misclassified data.

Compared to other ensemble learning algorithms, AdaBoost is simple to use, has good accuracy and stability and is less prone to overfitting. AdaBoost is suitable for various types of classification problems, such as binary classification, multiclass classification, and regression.

3.1.6. XGboost

The core idea of XGBoost is to train multiple weak classifiers (decision trees) iteratively and combine them into a strong classifier. During the training of each classifier, XGBoost performs feature preprocessing and data missing value processing on the dataset and uses second-order Taylor expansion and regularization methods to improve training speed and accuracy. In addition, XGBoost supports multithreading and distributed computing, making it suitable for handling large-scale and high-dimensional feature data.

XGBoost is widely used in various types of machine learning tasks, such as classification, regression, and ranking. It has been widely applied in competitions, recommendation systems, search engines, and advertising, among other practical applications.

3.2. Oversampling

Oversampling is one of the methods for handling class imbalance in an imbalanced dataset in machine learning. When the number of samples in one class is relatively small in a dataset, oversampling can balance the dataset by replicating or generating samples of that class, thereby improving the performance of machine learning algorithms.

Oversampling can be implemented using various methods, including random oversampling, synthetic minority oversampling technique (SMOTE), adaptive synthetic sampling (ADASYN), etc. Among them, SMOTE is the most commonly used method, which generates new samples by interpolating the minority class samples to expand the sample size of that class while avoiding the overfitting problem that may arise from simple duplicate samples. It should be noted that although oversampling can improve the prediction accuracy of the minority class, excessive use may lead to overfitting problems, and therefore, it needs to be used with caution.

The benefit of oversampling in machine learning is to improve the algorithm's ability to identify and predict minority classes. In imbalanced datasets, if the number of samples in the minority class is too small, machine learning algorithms may focus more on the majority class, resulting in a decrease in the ability to predict the minority class. Oversampling increases the number of samples in the minority class, allowing the algorithm to consider each class more evenly during training, thus improving the accuracy and generalization ability of predicting the minority class. Additionally, oversampling can reduce the classifier bias caused by imbalanced datasets, improving the performance and stability of the classifier.

3.3 Feature Dimension Reduction

PCA stands for principal component analysis, which is a popular technique used in statistics, data analysis, and machine learning for dimensionality reduction. Data dimensionality reduction has many benefits. On the one hand, it can reduce noise and redundant information in the data, thereby improving the efficiency and accuracy of algorithms. On the other hand, the reduced data can be more easily visualized, providing an intuitive representation of the structure and relationships within the data, making it easier to analyze and understand. Data dimensionality reduction can also improve data distribution, enabling the distribution of data in a lower-dimensional space, making it easier to perform clustering or classification operations, and improving the accuracy of machine learning algorithms.

In essence, PCA transforms high-dimensional data into a lower-dimensional representation by identifying the underlying patterns and correlations in the data. It does this by projecting the original data onto a new coordinate system, where the first axis captures the most variance in the data, followed by the second axis, and so on. These new axes, known as principal components, are orthogonal to each other and are arranged in descending order of importance.

PCA is often used in data preprocessing to reduce the number of features or variables in a dataset, which can help to improve the accuracy and efficiency of machine learning algorithms. It is also used in data visualization, as the first few principal components can be used to create scatter plots or heatmaps that provide insight into the structure and relationships within the data.

Overall, PCA is a powerful tool for exploratory data analysis and can help to simplify complex datasets while preserving the most important information.

4. Dataset Description

4.1 Data

We use data collected from the Taiwan Economic Journal for the years 1999 to 2009 [23] (https://www.kaggle.com/datasets/fedesoriano/company-bankruptcy-prediction). Company bankruptcy was defined based on the business regulations of the Taiwan Stock Exchange. The dataset has a total of 6819 rows of data, with no missing values, and the data types include integer and float.

There are several key indicators in a dataset. The first is total revenue, which refers to the total income generated by a business over a certain period of time and is often used to measure the size and revenue-generating activity of the business. The second is net profit, which represents the profitability and operational efficiency of a business by subtracting total expenses from total revenue over a certain period of time. The third is the balance sheet, which is an accounting report that outlines a business's assets, liabilities, and owner's equity. It can help investors evaluate the financial health of the business. The fourth is the cash flow statement, which is an accounting report that lists a business's cash inflows and outflows over a certain period of time. It can help investors evaluate the cash flow position of the business. The fifth is earnings per share (EPS), which represents the profit generated by a business per share of common stock. It is an important indicator for investors to assess a business's profitability and return on investment. Finally, financial ratios are a set of indicators used to evaluate a business's financial condition, including the debt-to-equity ratio, current ratio, quick ratio, gross profit margin, net profit margin, and others. These ratios can help investors understand a business's financial health, debt level, and profitability.

The output is 0 or 1, which indicates whether the company is bankrupt. Figure 1 shows the label distribution, and the prediction is a highly imbalanced classification problem.

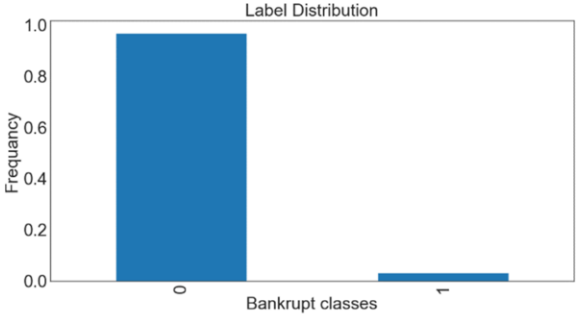


Figure 1. The distribution of labels.

As indicated in Figure 2, for bankrupt companies, their Net Income to Total Assets ratio, which is the ratio of net income to total assets, is significantly lower, indicating a poorer level of profitability and making them more susceptible to bankruptcy. As shown in Figure 3, in terms of the proportion of retained earnings to total assets, bankrupt companies also have a lower percentage.

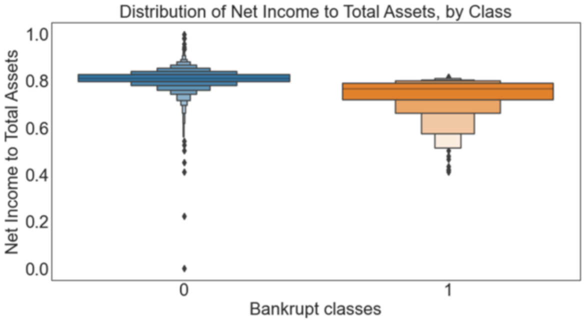


Figure 2. The distribution of net income to total assets ratio.

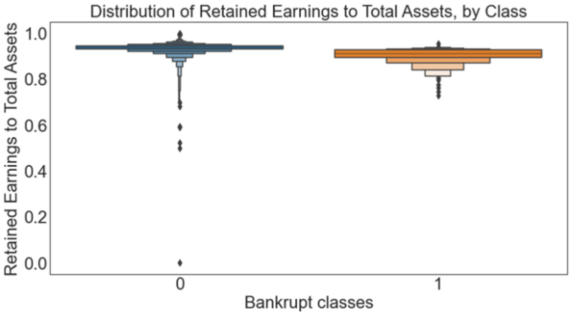


Figure 3. The distribution of retained earnings to total assets.

5. Discussion

5.1 Experiment settings

In this study, we use Python and toolkits such as scikit-learn, xgboost, and imblearn for data processing and prediction. For each model, different parameter combinations in Table 1 are tried on the training set, and then the best-performing model trained with the best parameter combination on the test set is evaluated.

Table 1. The value range for model parameters.

	Parameter	Value Range
Decision Tree	Max Tree Depth	5, 10, 15, 20, 25
Random Forest	Max Tree Depth	5, 10, 15, 20, 25
	Number of Estimators	20, 50, 100, 200
AdaBoost	Max Tree Depth	5, 10, 15, 20, 25
	Number of Estimators	20, 50, 100, 200
XGBoost	Max Tree Depth	5, 10, 15, 20, 25
	Number of Estimators	20, 50, 100, 200

5.2 Results

The accuracies and F1 scores for different models are shown in Table 2 and Table 3.

Table 2. The accuracies for different models.

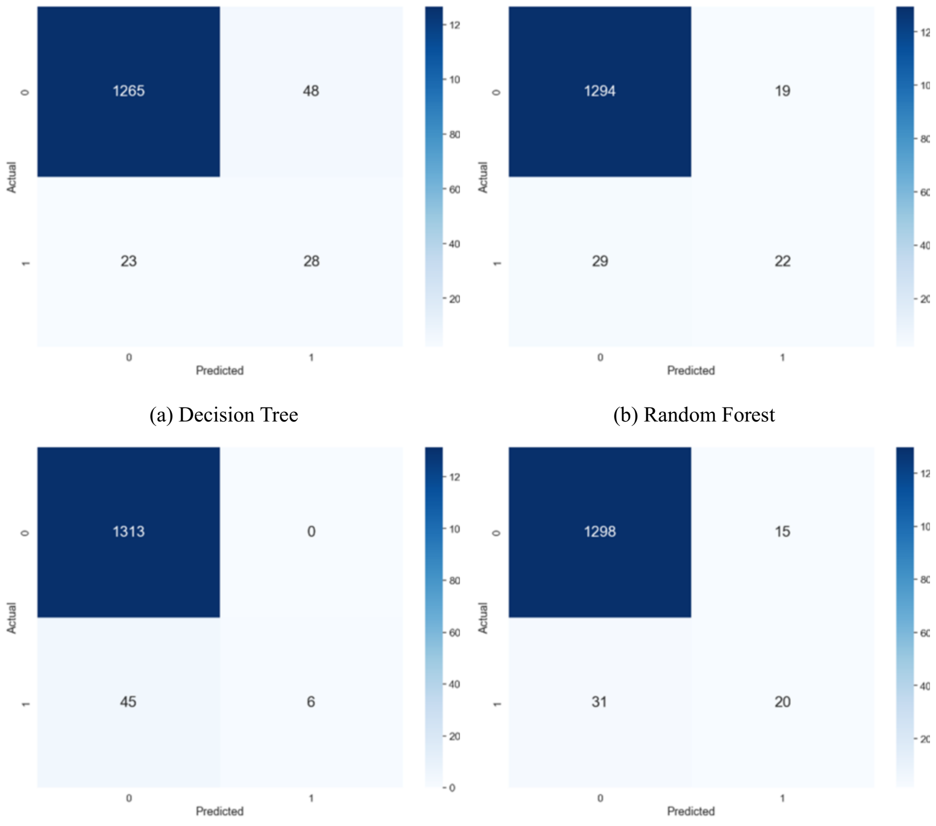
Model	Base	+Oversampling	+PCA		
			25	50	75
Decision Tree	0.960	0.948	0.964	0.962	0.957
Random Forest	**0.967**	0.965	0.966	0.965	0.965
AdaBoost	0.965	**0.967**	0.962	0.964	0.963
XGBoost	0.965	0.966	0.963	0.964	**0.967**

Table 3. F1 scores for different models.

Model	Base	+Oversampling	+PCA		
			25	50	75
Decision Tree	0.286	0.441	0.347	0.333	0.341
Random Forest	0.286	**0.478**	0.145	0.200	0.143
AdaBoost	0.200	0.211	0.133	0.169	0.107
XGBoost	0.333	0.465	0.242	0.246	0.328

5.3 Confusion Matrices

The confusion matrices for different models are shown in Figure 4.



(a) Decision Tree

(b) Random Forest

(c) AdaBoost

(d) XGBoost

Figure 4. The confusion matrices for different models.

5.4 Discussion

By analyzing the accuracies and F1 score of different models, we found that random forest performed the best. Adding oversampling was effective, as it improved the overall performance of the model in terms of the F1 score, despite reducing the accuracy. On the other hand, adding principal component analysis (PCA) for dimensionality reduction did not have much effect. We tried PCA with 25, 50, and 75 components to process our data, but the results showed no significant improvement in terms of accuracy and F1 scores.

As shown in Figure 5, for accuracy, as the number of input features increased from 25 to 75, the accuracy of decision trees and random forests decreased, while XGBoost's accuracy increased. When the number of input features increased from 75 to 95, random forests, decision trees, and AdaBoost showed an upward trend in accuracy. For AdaBoost, the accuracy increased when the number of input features increased from 25 to 50 and from 75 to 95 but decreased from 50 to 75.

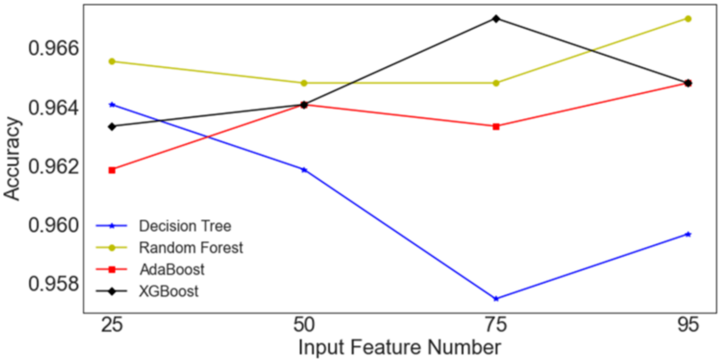


Figure 5. The influence of input feature number on model accuracy.

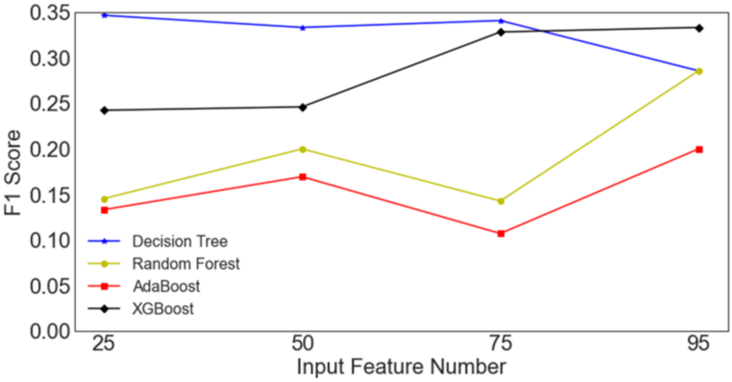


Figure 6. The influence of the number of input features on the model F1 score.

In Figure 6, for the F1 score, the overall trends of AdaBoost and random forest are similar. They both show an upward trend when the input features increase from 25 to 50 and from 75 to 95, but the amplitude of the random forest is larger. Both models show a decrease in the F1 score when the input features increase from 50 to 75. For XGBoost, the F1 score remains the same when the input features increase from 25 to 50 and then increases steadily as the input features increase from 50 to 95, with a significantly larger increase in the 50 to 75 range. For decision trees, the F1 score remains almost unchanged when the input features increase from 25 to 75 and then decreases when the input features increase from 75 to 95.

6. Conclusion and outlook

In this study, we considered the imbalanced bankrupt prediction problem with a real-world Taiwan dataset. We employ some models, including decision tree, random forest, AdaBoost, and XGBoost. We found that among the four models for predicting bankruptcy based on initial data, the RF model performed the best with an accuracy of 0.967. However, from the perspective of the F1 score, none of the four models performed well. After oversampling the data and retesting the models' prediction levels, the AdaBoost model had the highest accuracy, while the RF model had the largest F1 score. Next, we used PCA to perform dimensionality reduction on the data, retaining the number of principal components required to preserve 25%, 50%, and 75% of the variance in the dataset. The results showed that when 75% of the variance in the dataset was preserved, the XGBoost model had the highest accuracy, with a precision of 0.967.

Currently, although the accuracy of the model's bankruptcy prediction is relatively high, the best F1 score value is only 0.478, which is still not ideal. In the future, we can collect a larger dataset to train the model or use more advanced models to predict bankruptcy issues, e.g., graph-based neural networks [24, 25].

References

[1] Jiang, W. (2021). Applications of deep learning in stock market prediction: recent progress. Expert Systems with Applications, 184, 115537.

[2] Wang, Q., & Jiang, W. (2022). Beat Wash-Sale Tax with Multigraph Convolutional Neural Networks Based Trading Strategy. Security and Communication Networks, 2022.

[3] Jiang, W. (2022). Loan Default Prediction with Deep Learning and Muddling Label Regularization. IEICE TRANSACTIONS on Information and Systems, 105(7), 1340-1342.

[4] Li, Z., & Jiang, W. (2020, December). Evaluation of Machine Learning Techniques for Stock Market Movement Prediction. In 2020 Management Science Informatization and Economic Innovation Development Conference (MSIEID) (pp. 248-253). IEEE.

[5] Putri, N. H., Fatekurohman, M., & Tirta, I. M. (2021, March). Credit risk analysis using support vector machines algorithm. In Journal of Physics: Conference Series (Vol. 1836, No. 1, p. 012039). IOP Publishing.

[6] Karthikeyan, C., Anandan, P., Prabha, R., Mohan, D., & Babu, V. (2021, January). Predicting Stock Prices Using Machine Learning Techniques. In 2021 6th International Conference on Inventive Computation Technologies (ICICT) (pp. 1-5). IEEE.

[7] Ahmed, M., Mahmood, A. N., & Islam, M. R. (2016). A survey of anomaly detection techniques in financial domain. Future Generation Computer Systems, 55, 278-288.

[8] Huang, C. L., Chen, M. C., & Wang, C. J. (2007). Credit scoring with a data mining approach based on support vector machines. Expert systems with applications, 33(4), 847-856.

[9] Jiang, Z., Xu, D., & Liang, J. (2017). A deep reinforcement learning framework for the financial portfolio management problem. arXiv preprint arXiv:1706.10059.

[10] Jiang, W., & Zhang, L. (2018). Geospatial data to images: A deep-learning framework for traffic forecasting. Tsinghua Science and Technology, 24(1), 52-64.

[11] Jiang, W. (2022). Internet traffic prediction with deep neural networks. Internet Technology Letters, 5(2), e314.

[12] Jiang, W. (2022). Internet traffic matrix prediction with convolutional LSTM neural network. Internet Technology Letters, 5(2), e322.

[13] Jiang, W. (2022). Deep learning based short-term load forecasting incorporating calendar and weather information. Internet Technology Letters, 5(4), e383.

[14] Le, T. (2022). A comprehensive survey of imbalanced learning methods for bankruptcy prediction. IET Communications, 16(5), 433-441.

[15] Cao, Y., Liu, X., Zhai, J., & Hua, S. (2022). A two-stage Bayesian network model for corporate bankruptcy prediction. International Journal of Finance & Economics, 27(1), 455-472.

[16] Brenes, R. F., Johannssen, A., & Chukhrova, N. (2022). An intelligent bankruptcy prediction model using a multilayer perceptron. Intelligent Systems with Applications, 200136.

[17] Kou, G., Xu, Y., Peng, Y., Shen, F., Chen, Y., Chang, K., & Kou, S. (2021). Bankruptcy prediction for SMEs using transactional data and two-stage multiobjective feature selection. Decision Support Systems, 140, 113429.

[18] Zelenkov, Y., & Volodarskiy, N. (2021). Bankruptcy prediction on the base of the unbalanced data using multi-objective selection of classifiers. Expert Systems with Applications, 185, 115559.

[19] Jabeur, S. B., & Serret, V. (2023). Bankruptcy prediction using fuzzy convolutional neural networks. Research in International Business and Finance, 64, 101844.

[20] Narvekar, A., & Guha, D. (2021). Bankruptcy prediction using machine learning and an application to the case of the COVID-19 recession. Data Science in Finance and Economics,

[21] Shetty, S., Musa, M., & Brédart, X. (2022). Bankruptcy Prediction Using Machine Learning Techniques. Journal of Risk and Financial Management, 15(1), 35.

[22] Garcia, J. (2022). Bankruptcy prediction using synthetic sampling. Machine Learning with Applications, 9, 100343.

[23] Liang, D., Lu, C. C., Tsai, C. F., & Shih, G. A. (2016).) Financial ratios and corporate governance indicators in bankruptcy prediction: A comprehensive study. European journal of operational research, 252(2), 561-572.

[24] Jiang, W., & Luo, J. (2022). Graph neural network for traffic forecasting: A survey. Expert Systems with Applications, 117921.

[25] Jiang, W. (2022). Graph-based deep learning for communication networks: A survey. Computer Communications, 185, 40-54.