MTU
Ollscoil Teicneolaíochta na Mumhan
Munster Technological University

Munster Technological University

APPLIED MACHINE LEARNING

Assignment 2

Student name: Jin Xiubin

Student ID: R00241682

Professor: Haithem Afli

Date: 29/04/2024

Table of Contents

Company Bankruptcy Prediction using Machine Learning Models

Abstract

This paper conducts a systematic analysis of the dataset named "Company Bankruptcy Prediction" from Taiwan, dealing with a binary classification problem to find the best model for predicting factors leading to bank closures and whether a bank will close (0/1). The dataset is divided into training, validation, and test sets. We explored and compared the performance of KNN, Random Forest, SVM, and Gradient Boosting on the dataset and used ensemble algorithms such as BaggingClassifier and AdaClassifier to evaluate performance enhancements for KNN and SVM. Initial analysis included using box plots and histplots to observe data distribution and identify outliers. Correlation analysis was conducted to examine the relationship between features and the target variable. Techniques like PCA (Principal Component Analysis) and VIF (Variance Inflation Factor) were employed to reduce redundant features, comparing their performance and accuracy scores across the four models. Before modelling, the SMOTE technique was used to balance the dataset, and StandardScaler was employed for data normalization. Grid search cross-validation was utilized to tune 3-4 main hyperparameters impacting the model. The model performance was assessed using confusion matrices, classification reports, accuracy scores, and ROC AUC. GBC showed high accuracy scores of 96.26% on both the training and test sets but had an ROC AUC close to random guessing. KNN had an accuracy score of 91.57% with a superior ROC AUC. Given the multitude of features, potential optimization strategies during the feature engineering stage were also discussed to avoid underfitting or overfitting issues. Finally, the report examined the feature importance of the optimal model and saved the best model and the most impactful factors for regulators and policymakers to understand and predict the risks and factors of bank closures.

1. Introduction

1.1. The dataset

The dataset was found in kaggle, sourced from the Taiwan Economic Journal spanning the years 1999 to 2009. Kaggle link: https://www.kaggle.com/datasets/fedesoriano/company-bankruptcy-prediction. Source: Deron Liang and Chih-Fong Tsai, deronliang '@' gmail.com; cftsai '@' mgt.ncu.edu.tw, National Central University, Taiwan. The data was obtained from UCI Machine Learning Repository: https://archive.ics.uci.edu/ml/datasets/Taiwanese+Bankruptcy+Prediction.

Attribute Information: Updated column names and description to make the data easier to understand (Bankrupt? = Target Variable, The rest = Input features), this dataset has 95 columns and 6819 rows including 2 integer features and 93 float features.

1.2. The problem

In the "Company Bankruptcy Prediction - Advanced Classification Techniques" project, our objective is to construct a reliable binary classification model capable of accurately predicting

whether a company will face bankruptcy. The dataset was found in Kaggle, sourced from the Taiwan Economic Journal spanning the years 1999 to 2009, defines bankruptcy based on the regulations of the Taiwan Stock Exchange. This endeavour holds significant importance in the financial sector, empowering investors, creditors, and stakeholders with valuable insights for financial decision-making. The dataset encompasses a diverse array of attributes pertaining to company financials and performance, presenting a comprehensive pool of features for developing a robust predictive model for bankruptcy risk assessment.

1.3. The goal

Our main goal is to develop a highly accurate categorization model that, using a variety of financial data, can forecast corporate bankruptcy. Our model attempts to find trends and indicators of financial distress by using a large dataset from the Taiwan Economic Journal that contains financial performance and other important indicators of enterprises from 1999 to 2009. Reaching this will greatly advance the area of financial risk assessment by giving creditors, investors, and other parties concerned in keeping an eye on and controlling the financial health of their companies an invaluable instrument. This project aims to improve strategic decision-making and strengthen preventative measures against financial failures of companies.

1.4. Objectives

- Thoroughly explore and understand the dataset
- Gain insights into the factors that influence bankruptcy
- Evaluate various Classification algorithms and select the most promising candidates
- Fine-tune model's hyperparameters to optimize performance

2. Background

The closing of banks presents a substantial risk to the stability of the global financial system, and the recent failures of Credit Suisse and Silicon Valley Bank have intensified worries in international markets.[1] In order to tackle this problem, academics have devised multiple prediction models to analyse and pre-empt the risks of bankruptcy. The current models utilised consist of machine learning algorithms such as Random Forest, Support Vector Machine (SVM), K-Nearest Neighbours (KNN), and Gradient Boosting. These models have the ability to analyse financial and operational features of banks, evaluate massive amounts of data, and identify crucial aspects that may result in bank failures. For example, if Credit Suisse and Silicon Valley Bank had utilised these sophisticated analytical methods prior, they might have detected early indications of a financial disaster.[2]In addition, ensemble learning techniques such as Bagging and Boosting are employed to augment the predictive capability of individual models. This is achieved by amalgamating the predictions of numerous models in order to enhance overall accuracy and resilience. These advanced analytical techniques offer substantial assistance to financial regulators and policymakers by enhancing their ability to accurately forecast and mitigate the risks associated with bank closures. This ultimately ensures the general stability of the financial system.[3]

3. Research

3.1. Synthetic Minority Over-sampling Technique (SMOTE)

Usage and Current State:

SMOTE is a widely used technique for addressing class imbalance in various predictive modeling challenges, including fraud detection, medical diagnoses, and bankruptcy prediction.[4] It generates synthetic samples from the minority class instead of creating copies, thereby contributing to a more generalized model without overfitting issues associated with simple oversampling.

Working Principle:

SMOTE works by selecting samples that are close in the feature space, drawing a line between the samples in this space, and generating new samples along this line. Specifically, it involves randomly selecting a point from the minority class, calculating the k-nearest neighbours for this point, and then adding synthetic points between the chosen point and its neighbours. In my dataset, which comprises a total of 6819 rows, there is a severe imbalance between the classes of the target variable; class 0 has 6599 instances while class 1 has only 220 instances. Given the extreme imbalance, the strategy I have adopted is to use oversampling to increase the number of minority class samples, thus equalizing the counts between the two classes to balance the dataset. This approach considers the equality of classes and the correlation between features and the target variable.[5]

Literature Review:

Chawla, N.V., et al. presented the Synthetic Minority Over-sampling Technique (SMOTE) in their influential paper titled "SMOTE: Synthetic Minority Over-sampling Technique," published in the Journal of Artificial Intelligence Research, volume 16 (2002), pages 321-357. They showcased the efficacy of the method in enhancing the performance of the classifier on many binary classification datasets. In their publication titled "SMOTE for Learning from Imbalanced Data: Progress and Challenges, Marking the 15-year Anniversary," Fernandez, A., et al. conducted a review of the advancements and difficulties encountered since the introduction of SMOTE. The article was published in the Journal of Artificial Intelligence Research, volume 61 (2018), pages 863-905. SMOTE has been utilised in several domains such as financial fraud detection, medical diagnostics, image recognition, and bioinformatics. SMOTE, a technique used in credit card fraud detection, is beneficial for generating additional samples of illegal transactions. This enables the model to better understand the patterns and features associated with fraudulent behaviour, thereby improving the accuracy of fraud detection.[4-5]

3.2. Cross-Validation

Usage and Current State:

Cross-validation is a robust method for assessing the performance of predictive models, particularly in scenarios where model selection and error estimation are crucial, such as in finance and healthcare. It is essential for avoiding overfitting and ensuring that a model

generalizes well to new data. It is extensively used in parameter tuning and comparing the performance of different machine learning models.[6]

Working Principle:

Cross-validation entails dividing the data into many folds, with a portion of each fold allocated for training the model and the remaining piece for validating the model. This method is iterated numerous times, with each iteration involving distinct data partitions. The outcomes of each iteration are averaged to approximate the overall performance of the model. K-fold cross-validation is the prevailing technique in which the data is partitioned into k smaller sets or folds. Compared to simple train-test splits, the advantage of cross-validation lies in its ability to traverse the entire dataset, allowing every sample in the dataset the opportunity to be both trained and tested. This significantly enhances the model's ability to learn from the data.[7]

Literature Review:

Kohavi, R. in "A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection," IJCAI International Joint Conference on Artificial Intelligence (1995): 1137-1145, explicates the method's effectiveness in model selection. Additionally, Varma, S., and Simon, R., in "Bias in Error Estimation When Using Cross-validation for Model Selection," BMC Bioinformatics 7, 91 (2006), discuss potential biases in error estimation using cross-validation, providing insights into its limitations and best practices.[6-7]

3.3. Conclusion

Both SMOTE and cross-validation are essential in developing predictive models for binary classification problems, such as bankruptcy prediction. SMOTE mitigates the issue of class imbalance by improving the algorithm's capacity to learn from the minority class. Meanwhile, cross-validation guarantees the model's dependability and ability to apply to various unseen datasets. By incorporating these strategies into the bankruptcy prediction methodology, the model's resilience and precision are improved, resulting in a potent instrument for evaluating financial risk and making informed decisions.[8]

4. Methodology

4.1. Pre-processing steps

In the preprocessing steps, I first conducted statistical analysis to examine the dataset size, missing values, outliers, duplicates, distribution of data, and the standard deviation of features. Subsequently, I engaged in data exploration, creating a correlation matrix, box plots, histograms, and visualizations of the category counts of the target variable. As most features had correlations with the target variable below 1, I identified 10 features with correlations greater than 0.1, which are:

- Debt ratio %
- Borrowing dependency
- Current Liability to Assets
- Current Liabilities/Equity

- Total expense/Assets
- Current Liability to Equity
- Equity to Long-term Liability
- Current Liability to Current Assets
- Liability-Assets Flag
- Liability to Equity

I also addressed the outliers corresponding to these features. Next, I performed data slicing to define X and Y. Given the extensive number of features (96), I applied PCA analysis to retain 15 principal component features. The entire X was divided into x_train, x_test, and x_val, and the entire Y into y_train, y_test, and y_val. The shapes are x_train: (4091, 15) and y_train: (4091,). I used the SMOTE technique to address the imbalance in the dataset, resulting in a balanced dataset of 7926 rows and 15 columns, with equal counts of categories 0 and 1 at 3963 each. Finally, given that I observed some skewness in the data distribution and inconsistent feature value ranges during the initial visualization phase, I utilized the StandardScaler technique to standardize the data.
In addition to using PCA techniques, I also performed a Variance Inflation Factor (VIF) analysis towards the end of my Python code. This method helps reduce feature redundancy by decreasing multicollinearity among the features. Generally, a VIF value greater than 10 indicates that a feature has strong collinearity with other features. Except for a few extreme cases, more than 70 variables had a VIF under 10. Therefore, I set the threshold at 2, retaining 38 features with a VIF less than 2. Using these features, I re-sliced the data and proceeded with model predictions. Unfortunately, the final accuracy scores and ROC AUC were very low, as shown in the figure 1.

Possible reasons for this may include misjudgment of collinearity: VIF may incorrectly identify some features as collinear, resulting in the removal of useful features for the model. [9] Inappropriateness of the data: some features may not be linearly correlated and are subsequently removed. [10]The impact of model selection: certain features may exhibit different importance before and after VIF processing, thereby affecting the performance of

the model.[11] Additionally, severe class imbalance in the dataset may also lead to misjudgements in analysing collinearity using VIF.

4.2. Model selection

Because my dataset deals with a binary classification problem, this paper primarily employs classification models such as KNN, random forest, SVM, and gradient boosting, as well as two ensemble algorithms, bagging and Adaboosting, applied to KNN and SVM, to enhance model performance. The K-nearest neighbours (KNN) technique quantifies the similarity between samples by evaluating the distance between their attributes. During the prediction phase, the method computes the distance between a new input sample and each sample in the training set, and then identifies the K nearest neighbours. Subsequently, it ascertains the classification of the new sample by considering the class labels of these neighbouring samples.[12] The underlying concept behind random forest is ensemble learning, where the predictions of numerous decision trees are combined to enhance the overall performance of the model. Bootstrap is utilised to generate each tree by employing distinct training data and feature subsets. This approach effectively mitigates overfitting and enhances the model's capacity to generalise.[13] Support Vector Machines (SVM) operate by maximising the classification boundary, which involves identifying the most ideal hyperplane to effectively divide samples belonging to distinct classes. During the training process, Support Vector Machines (SVM) optimise the classification boundary by maximising the distance between the support vectors and the hyperplane. This leads to the creation of an ideal decision border. [14]The fundamental concept behind gradient boosting is to enhance the model's performance by iteratively adjusting the residuals. During each iteration, the model updates the learning rate using gradient descent in order to prioritise samples that were previously mis predicted by the models. In the end, the forecasts from all models are merged to create the ultimate ensemble model.[15]

4.3. Hyper-parameter optimization

I have listed some commonly used hyperparameters in four models. Please see figure 2. I am using grid search cross-validation technique for parameter tuning. For each model, it traverses all the hyperparameters I have selected for that model. Through a certain number of cross-validation iterations, it iteratively tests hyperparameter combinations until the best model is obtained.

5. Evaluation

The performances of GBC, RF, and SVM on the validation set are the same, with an accuracy of 96.99%. However, there are differences in the performance of the three models in cross-validation. The gradient boosting model has the highest accuracy at 94.71%, while the random forest model has the lowest accuracy at 87.29%. The accuracy of the KNN model is 93.75%. Apart from a slight improvement in the accuracy of the KNN model, the accuracies of the other three models have decreased during cross-validation. Please refer to Figure 3. Grid search cross-validation technique was used to optimize the main hyperparameters of the four models. Please refer to Figure 4. By trying different combinations of hyperparameters to determine the best model, we can see that after parameter tuning, the performance of the other three models has significantly improved except for the KNN model. However, overall, the ROCAUC values of the four models are not ideal. The ROCAUC values of GBC, RF, and

SVM are all 0.5, indicating that the model's prediction results are close to random guessing. The ROCAUC value of KNN is 0.5046, slightly better, but the prediction results are also close to random guessing. Please refer to Figure 5. The confusion matrix of KNN shows that the model correctly predicted 1248 negative samples, 65 negative samples were incorrectly predicted as positive, 48 positive samples were incorrectly predicted as negative, and 3 positive samples were correctly predicted as positive. The confusion matrix parameters of the other three models are the same, indicating that 1313 negative samples were correctly predicted as negative, no negative samples were incorrectly predicted as positive, 51 positive samples were incorrectly predicted as negative, and no positive samples were correctly predicted as positive. I think the reason for this situation may be the severe imbalance of the samples. Even though I have adopted oversampling techniques to increase the minority class samples, the models still cannot capture the changing trends of both positive and negative classes well. Since random forest and GBC are already ensemble algorithms themselves, I tried using ensemble algorithms for SVM and KNN to optimize model performance. We can see that after optimization, the accuracy score of SVM remains unchanged at 96.26%, while the accuracy score of KNN has increased from 91.57% to 91.71%. This indicates that on our dataset, there is limited room for optimization in the predictions from models with the best parameters obtained through one-time grid search cross-validation. Therefore, using ensemble algorithms on top of this may not be meaningful.
I saved the important features that influence the performance of the best models, GBC and KNN. Please see figure 6 and 7. The relatively significant features influencing whether a bank goes bankrupt or not are: Operating gross margin, POA(B) before interest and depreciation after tax.

6. Conclusion

Data Insights

Whether a bank goes bankrupt or not is significantly influenced by Operating gross margin, POA(B) before interest and depreciation after tax. Operating expense rate, cash flow rate and ROA(A) before interest and after tax also play a role in determining the market value.

Model Choice

Among the four classifiers - KNN, SVM, RandomForestClassifier, and GradientBoostingClassifier - we observed commendable performances. Among them, the fine-tuned GradientBoostingClassifier emerged as the strongest performer. It achieved a remarkably good Accuracy score of 94.71% on validation dataset, and 96.26% of accuracy score on the test dataset. These metrics indicate that the GradientBoostingClassifier produced the most accurate and reliable predictions compared to the other regressors. Considering of ROCAUC KNN is more advantage among four models.

Future Work

- Eliminate outlier values that do not contribute to the model's performance
- Generate additional features to enhance the model's predictive ROCAUC
- Discard unnecessary features, retaining only those that are significant
- Fine-tune the top-performing model to achieve improved results
- Trying more techniques to address the issue of imbalanced datasets

References

[1] T. Adrian and H. S. Shin, "Liquidity and Leverage," *Journal of Financial Intermediation*, vol. 19, no. 3, pp. 418-437, 2010.

[2] I-C. Yeh, C-H. Lien, and Y-C. Tsai, "Supervised Machine Learning Techniques for Credit Risk Modeling: A Study of the Predictive Performance of BRT, SVM and ANN," *Expert Systems with Applications*, vol. 38, no. 5, pp. 6047-6056, May 2011.

[3] C. Borio, "The Financial Turmoil of 2007-?: A Preliminary Assessment and Some Policy Considerations," BIS Working Paper No. 251, 2008.

[4] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic Minority Over-sampling Technique*," Journal of Artificial Intelligence Research*, vol. 16, pp. 321-357, 2002.

[5] A. Fernandez, S. Garcia, F. Herrera, and N. V. Chawla, "SMOTE for Learning from Imbalanced Data: Progress and Challenges, Marking the 15-year Anniversary," *Journal of Artificial Intelligence Research*, vol. 61, pp. 863-905, 2018.

[6] R. Kohavi, "A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection," in Proc. *14th International Joint Conference on Artificial Intelligence,* Montreal, Quebec, 1995, pp. 1137-1145.

[7] S. Varma and R. Simon, "Bias in Error Estimation When Using Cross-validation for Model Selection," *BMC Bioinformatics*, vol. 7, no. 91, 2006.

[8] J. Brownlee, "Mastering Machine Learning Algorithms: Expert Techniques to Implement Popular Machine Learning Algorithms and Fine-Tune Your Models," *Packt Publishing*, 2018.

[9] James, Gareth, et al. "An Introduction to Statistical Learning: with Applications in R." Springer, 2013.

[10] Harrell, Frank E. Jr. "Regression Modeling Strategies: With Applications to Linear Models, Logistic and Ordinal Regression, and Survival Analysis." Springer, 2015.

[11] Hastie, Trevor, et al. "The Elements of Statistical Learning: Data Mining, Inference, and Prediction." Springer, 2009.

[12] T. Hastie, R. Tibshirani, and J. Friedman, "The Elements of Statistical Learning: Data Mining, Inference, and Prediction," Springer Science & Business Media, 2009.

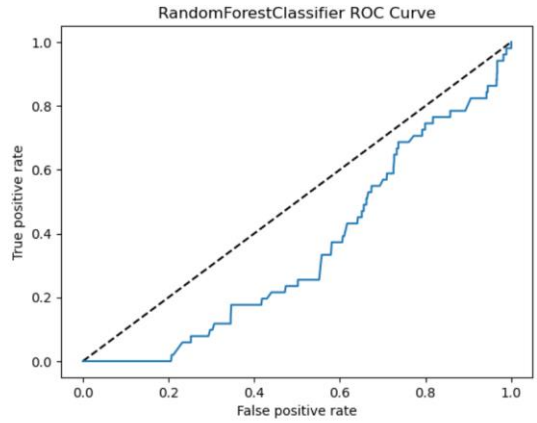
[13] L. Breiman, "Random forests," Machine Learning, vol. 45, no. 1, pp. 5-32, 2001.

[14] C. Cortes and V. Vapnik, "Support-vector networks," Machine Learning, vol. 20, no. 3, pp. 273-297, 1995.

[15] J. H. Friedman, "Greedy function approximation: A gradient boosting machine," Annals of Statistics, pp. 1189-1232, 2001.

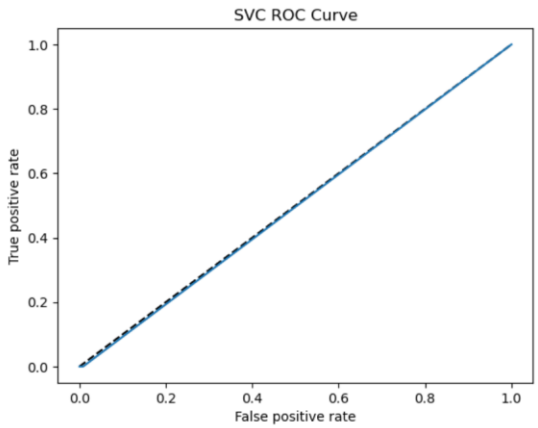
Appendix

Fig. 1 ROCAUC plot using VIF Method



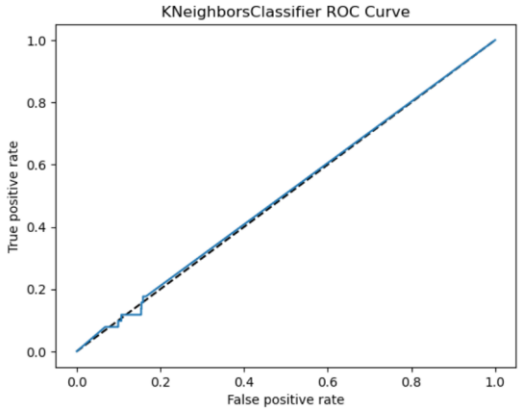
Validation Accuracy (RandomForestClassifier): 0.5535190615835777
Test Accuracy (RandomForestClassifier): 0.5586510263929618
AUC Score (RandomForestClassifier): 0.3844063139345609
Confusion Matrix:
[[752 561]
 [41 10]]
Classification Report:

	precision	recall	f1-score	support
0	0.95	0.57	0.71	1313
1	0.02	0.20	0.03	51
accuracy			0.56	1364
macro avg	0.48	0.38	0.37	1364
weighted avg	0.91	0.56	0.69	1364



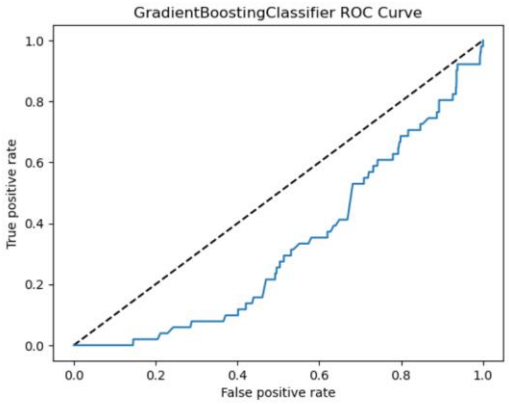
Validation Accuracy (SVC): 0.9640762463343109
Test Accuracy (SVC): 0.9545454545454546
AUC Score (SVC): 0.49581111957349583
Confusion Matrix:
[[1302   11]
 [51   0]]
Classification Report:

	precision	recall	f1-score	support
0	0.96	0.99	0.98	1313
1	0.00	0.00	0.00	51
accuracy			0.95	1364
macro avg	0.48	0.50	0.49	1364
weighted avg	0.93	0.95	0.94	1364



Validation Accuracy (KNeighborsClassifier): 0.8870967741935484
Test Accuracy (KNeighborsClassifier): 0.8687683284457478
AUC Score (KNeighborsClassifier): 0.49837223541358666
Confusion Matrix:
[[1180 133]
 [46   5]]
Classification Report:

	precision	recall	f1-score	support
0	0.96	0.90	0.93	1313
1	0.04	0.10	0.05	51
accuracy			0.87	1364
macro avg	0.50	0.50	0.49	1364
weighted avg	0.93	0.87	0.90	1364



Validation Accuracy (GradientBoostingClassifier): 0.3944281524926686
Test Accuracy (GradientBoostingClassifier): 0.4252199413489736
AUC Score (GradientBoostingClassifier): 0.38106118304138104
Confusion Matrix:
[[563 750]
 [34 17]]
Classification Report:

	precision	recall	f1-score	support
0	0.94	0.43	0.59	1313
1	0.02	0.33	0.04	51
accuracy			0.43	1364
macro avg	0.48	0.38	0.32	1364
weighted avg	0.91	0.43	0.57	1364

Fig. 2 Common hyperparameters for 4 models

Model	Hyperparameter	Description
Support Vector Machine (SVM)	**C**	Regularization parameter. Larger values decrease the regularization strength, potentially leading to overfitting.
	kernel	Choice of kernel function, such as linear, poly, rbf, sigmoid.
	gamma	Coefficient for the kernel (for poly, rbf, sigmoid). Controls the influence range of a single training sample.
	degree	Degree of the polynomial kernel function (when using poly).
Decision Tree	**max_depth**	Maximum depth of the tree. Restricts tree growth to prevent overfitting.
	min_samples_split	Minimum number of samples required to split an internal node.
	min_samples_leaf	Minimum number of samples required at a leaf node.
	max_features	Number of features to consider when looking for the best split.
Random Forest	**n_estimators**	Number of trees in the forest.
	max_depth	Maximum depth of each tree.
	min_samples_split	Minimum number of samples required to split an internal node.
	max_features	Maximum number of features considered for splitting a node.
Gradient Boosting Machine (GBM)	**learning_rate**	Learning rate, affecting the contribution of each tree. Lower rates require more trees to model all relationships.
	n_estimators	Number of boosting stages to perform.
	max_depth	Maximum depth of an individual regression tree.
	min_samples_split	Minimum number of samples required to split an internal node.

Fig. 3 Summary of 4 Models

	Model	Validation Accuracy	Average Training Accuracy (CV)	Model Accuracy After Tuning	ROC_AUC Score	Ensemble Method	Accuracy After Ensemble
0	Gradient Boosting Classifier	0.9699	0.9471	0.9626	0.5000	Boosting(Itself)	No
1	Random Forest Classifier	0.9699	0.8729	0.9626	0.5000	Bagging(Itself)	No
2	KNeighbors Classifier	0.9289	0.9375	0.9157	0.5046	Bagging	0.9171
3	SVM Classifier	0.9699	0.9215	0.9626	0.5000	Boosting	0.9626

Fig. 4 The Value of Hyperparameters for tuning using GridSearchCV

Model	Hyperparameters	Values
GBC	learning_rate	[0.1, 0.5, 1.0]
	n_estimators	[50, 100, 200]
	max_depth	[3, 5, 7]
	max_features	[3, 5, 7]
RF	n_estimators	[50, 100, 200]
	max_depth	[3, 5, 7]
	max_features	[3, 5, 7]
KNN	n_neighbors	[3, 5, 7, 9, 11]
	weights	['uniform', 'distance']
	algorithm	['auto', 'ball_tree', 'kd_tree', 'brute']
	p	[1, 2]
SVM	C	[0.1, 1, 10, 100]
	kernel	['linear', 'rbf', 'poly', 'sigmoid']
	gamma	['scale', 'auto']

Fig. 5 Summary of Confusion Matrix of 4 Models

Bagging(KNN) auc score: 0.5046592894583576
Confusion Matrix:
[[1248 65]
[48 3]]
Classification Report:

	precision	recall	f1-score	support
0	0.96	0.95	0.96	1313
1	0.04	0.06	0.05	51
accuracy			0.92	1364
macro avg	0.50	0.50	0.50	1364
weighted avg	0.93	0.92	0.92	1364

AdaBoost(SVM) auc score: 0.5
Confusion Matrix:
[[1313 0]
[51 0]]
Classification Report:

	precision	recall	f1-score	support
0	0.96	1.00	0.98	1313
1	0.00	0.00	0.00	51
accuracy			0.96	1364
macro avg	0.48	0.50	0.49	1364
weighted avg	0.93	0.96	0.94	1364

RF auc score: 0.5
Confusion Matrix:
[[1313 0]
[51 0]]
Classification Report:

	precision	recall	f1-score	support
0	0.96	1.00	0.98	1313
1	0.00	0.00	0.00	51
accuracy			0.96	1364
macro avg	0.48	0.50	0.49	1364
weighted avg	0.93	0.96	0.94	1364

GBC Auc score: 0.5
Confusion Matrix:
[[1313 0]
[51 0]]
Classification Report:

	precision	recall	f1-score	support
0	0.96	1.00	0.98	1313
1	0.00	0.00	0.00	51
accuracy			0.96	1364
macro avg	0.48	0.50	0.49	1364
weighted avg	0.93	0.96	0.94	1364

Fig. 6 Features Importance of Model GBC

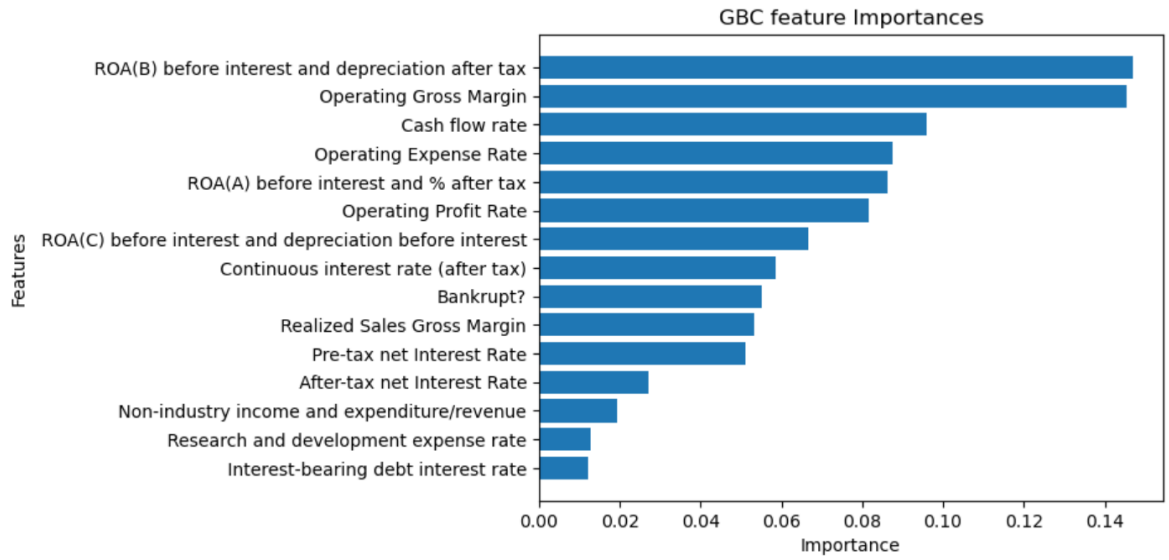


Fig. 7 Features Importance of Model KNN