Group P31 Final Project Midway Report

Karthik K Jayakumar **Naga Jahnavi Kommareddy** **Elizabeth Lin**

1 Introduction

Financial stability and solvency are essential considerations for businesses and investors alike. In an era of dynamic economic shifts, understanding the factors that lead to corporate bankruptcy has become a critical facet of financial management. This project focuses on the task of bankruptcy prediction using financial data, specifically from Taiwanese firms, with the goal of developing a robust classification model capable of predicting whether a company is at risk of bankruptcy

Our dataset, Taiwanese Bankruptcy Prediction [1], downloaded from the UCI Machine Learning Repository, includes data from the years 1999 to 2009 collected from the Taiwan Economic Journal. It consists of 6819 rows of individual company data and 96 columns, which include 95 attributes and 1 column for the outcome of bankruptcy. The attributes are various financial indicators such as *operating gross margin, cash flow rate, inventory turnover rate*, etc.

2 Method

2.1 Synthetic minority oversampling technique

Imbalanced datasets have long been the subject of much research as they can introduce error [3, 5]. Two popular methods to overcome this are (1) oversampling: the minority class is oversampled, therefore increasing their numbers and (2) undersampling: the majority class is undersampled, in order to reduce their numbers to almost that of the minority class. Methods such as random oversampling and undersampling are not ideal as it produces ties in the data [6]. Synthetic minority oversampling technique (SMOTE) is a method that has shown success [2]. SMOTE examines the minority data points and generates *synthetic* samples based on similarities in the feature space.

SMOTE works as follows: For each minority class instance that exists, its k nearest neighbors (of the same class) are selected (using Euclidean distance) and one of those neighbors are selected at random. A new synthetic instance belonging to the minority class is then created using the following equation:

$$S = x + u \cdot (x^R - x)$$

Where x is an instance of the minority class, x^R is the randomly selected neighbor of x, from its k nearest neighbors, u is a random number such that $0 \leq u \leq 1$, and S is the synthetically created minority class instance. $x^R - x$ is the distance vector between x^R and x, and the newly created instance is a point between the original neighbors.

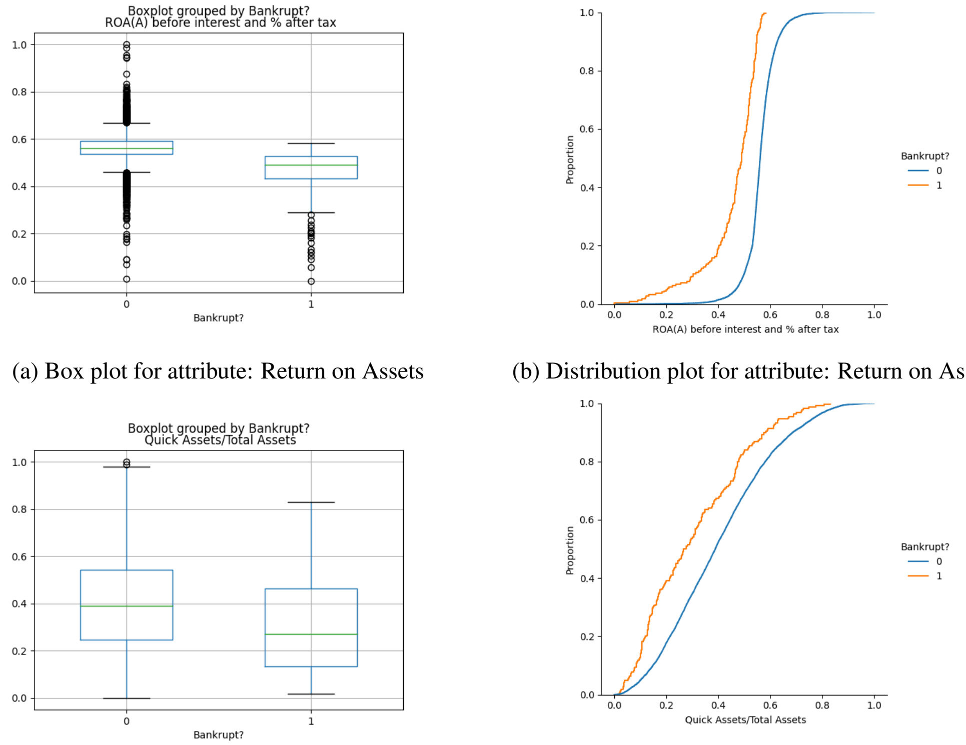
This way, class imbalances can be dealt with without introducing a bias during model training.

2.2 Mutual Information

MI quantifies the shared information between attributes (X) and the target variable (Y), allowing us to filter out less informative features. MI is valuable for feature selection because it effectively captures non-linear relationships, accommodates diverse feature types, doesn't assume linearity, and remains robust to feature scaling. By using MI, we can identify and retain the most relevant attributes, while reducing dimensionality. Mutual information is calculated using the below formula:

$$MI(X, Y) = \sum \sum P(x, y) \cdot log2(P(x, y)/(P(x) \cdot P(y)))$$

37th Conference on Neural Information Processing Systems (NeurIPS 2023).



(a) Box plot for attribute: Return on Assets

(b) Distribution plot for attribute: Return on Assets

(c) Box plot for attribute: Quick Assets / Total Assets

(d) Distribution plot for attribute: Quick Assets / Total Assets

Figure 1: Graphs from exploratory data analysis

2.3 Cross Validation

Cross validation provides a robust estimate of the model's performance by repeatedly splitting the data into different training and testing subsets. This reduces the impact of a single random split and gives a more stable evaluation. It also ensures that a substantial portion of data is used for both training and validation.

2.4 Random Forest

Random Forest is an ensemble learning method that combines multiple decision trees, first propose by Ho in 1995 [4]. Each decision tree in the forest is constructed using a subset of the training data and a random subset of features. Decision tree classifiers are straightforward and intuitively appealing but may overfit data when the model becomes complex. Random decision forests was proposed to overcome this limitation, it aggregates the predictions of individual trees to make a final prediction.

3 Experiment Setup

We plan on adding a diagram that details our experiment setup. Our GitHub repository has the code for our experiment `https://github.ncsu.edu/etlin/engr-ALDA-Fall2023-H25/tree/main/project`

3.1 Exploratory Data Analysis

The first step to understanding data is exploratory data analysis. First we check for missing values and duplicates, which are not existent in our dataset. Next we calculate the *mean, standard deviation, minimum, maximum, Q1, median, Q3* for each attribute, the calculated values are in `project/explor-`

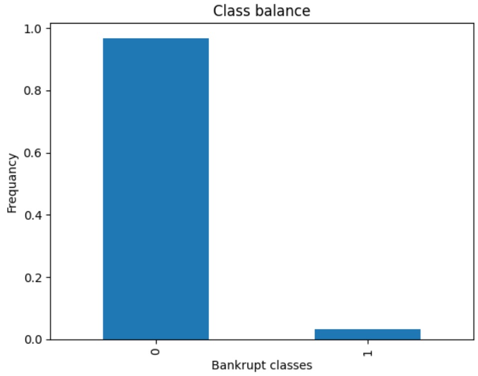


Figure 2: Relative frequency of bankruptcy outcome

`atory-data-analysis/df-descrite.csv` in our GitHub repository. To better visualize the data, we generated box plots and distribution plots for each attribute.

As seen from the figures 1, some attributes have different distributions when we group them by bankruptcy outcome. From the graphs, we can infer that companies with lower *ROA* and *Quick Assets / Total Assets* ratios are more likely to go bankrupt.

We also analyzed the correlation between various attributes, which can be helpful for dimension reduction as we have 95 attributes. The complete heatmap for correlation between the attributes is very large, therefore we did not include it in this report. The heatmap can be found in the file `project/exploratory-data-analysis/figures/heatmap.png` in our GitHub repository. Attribute pairs such as *(operating gross margin, realized sales gross margin)* or *(pre-tax net interest rate, after-tax net interest rate)* have a correlation of nearly 1.

Additionally, we addressed class imbalance by calculating the frequency of each bankruptcy outcome, as displayed in Figure 2. The severe data imbalance toward ’not bankrupt’ highlights the necessity of implementing sampling methods, such as SMOTE [2], to ensure balanced training data.

3.2 Feature Selection

We choose Mutual Information (MI) for feature selection in a dataset with 96 attributes to mitigate the curse of dimensionality. The `mutual_info_classif` function from scikit-learn was employed to compute feature scores, which measure the importance of each attribute in predicting the target feature. As shown in figure 3, a visual representation of these scores revealed that some attributes had very low scores, and a few even scored zero, indicating their limited contribution to the target prediction. To streamline the dataset and enhance model efficiency, a threshold of 0.005 was established, and attributes with scores below this value were removed. As a result, the initial set of 96 attributes was reduced to a more manageable 65, focusing on the most informative features for the predictive model.

3.3 Resampling

Through exploratory data analysis, we can see that there is a heavy class imbalance between the bankrupt data points and the not bankrupt data points. Of the 6819 records that we have, there are only 220 records for bankrupt companies whereas there are 6599 records for not bankrupt companies. Only 3.22% of the total dataset contains records of bankrupt companies. Therefore, there is an inherent bias, where just predicting that a company will not go bankrupt will have an accuracy of 96.77%.

Resampling methods such as *RandomOverSampler, SMOTE, ADASYN,RandomUnderSampler, NearMiss*, and *ClusterCentroids* were implemented in the code. SMOTE (Synthetic Minority Over-sampling Technique) was chosen as the initial resampling method because of its established effectiveness in generating synthetic examples for the minority class, addressing class imbalance, and improving the model’s ability to handle imbalanced data.

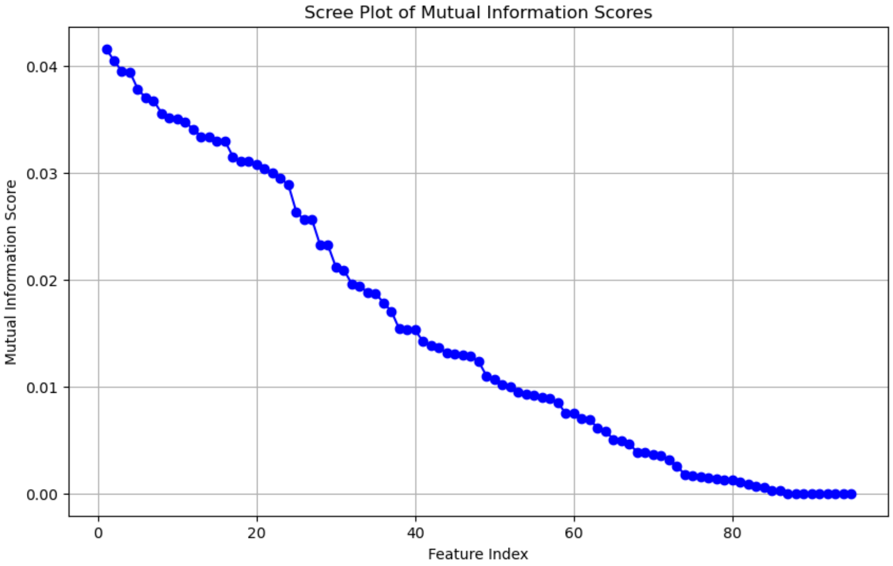


Figure 3: Scree plot of mutual information scores

Table 1: Preliminary results from running random forest classifier

	Precision	**Recall**	**f1-score**	**Support**
Bankruptcy = 0	1.00	0.96	0.98	6599
Bankruptcy = 1	0.97	1.00	0.98	6599
accuracy			0.98	13198
macro avg	0.98	0.98	0.98	13198
weighted avg	0.98	0.98	0.98	13198

3.4 Model Building and Cross Validation

To ensure a reliable evaluation, we implemented a 13-fold cross-validation strategy with data shuffling and a fixed random seed for reproducibility. This means our dataset is split into 13 subsets, and the model is trained and tested 13 times, ensuring thorough validation. The `cross_val_predict` function is then employed to generate predictions for each data point in our dataset. This allows us to assess the model's performance more comprehensively, as we obtain predictions for all data points across multiple cross-validation folds. The random forest classifier is used to predict the bankruptcy outcome.

4 Preliminary Results

We ran our random forest classifier using 13-fold cross validation and achieved a f1-score of 0.98. Our results are shown in table 1.

It can be concluded that the random forest classifier has high precision and recall for both classes (0 and 1). This indicates that the model is good at correctly identifying both classes with an overall high accuracy of 98%. The f1-score, which combines precision and recall, is also high at 0.98, reflecting strong model performance. The macro and weighted averages of these metrics confirm the model's effectiveness in making accurate predictions for the dataset with balanced support for both classes.

5 Conclusion

Thus far, we have applied sampling methods on the data and implemented it on a classifier. Between now and the final due date, we plan to apply other classifiers and methods and better understand what features contributed most.

Our future steps include:

1. Applying various classification algorithms, including ensemble methods, and conducting hyperparameter tuning to enhance model performance.
2. Assessing model performance using different implemented sampling techniques.
3. Exploring additional feature selection and extraction techniques such as PCA to further optimize model performance.

References

[1] Taiwanese Bankruptcy Prediction. UCI Machine Learning Repository, 2020. DOI: https://doi.org/10.24432/C5004D.

[2] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357, 2002.

[3] Haibo He and Edwardo A Garcia. Learning from imbalanced data. *IEEE Transactions on knowledge and data engineering*, 21(9):1263–1284, 2009.

[4] Tin Kam Ho. Random decision forests. In *Proceedings of 3rd international conference on document analysis and recognition*, volume 1, pages 278–282. IEEE, 1995.

[5] Robert C Holte, Liane Acker, Bruce W Porter, et al. Concept learning and the problem of small disjuncts. In *IJCAI*, volume 89, pages 813–818, 1989.

[6] David Mease, Abraham J Wyner, and Andreas Buja. Boosted classification trees and class probability/quantile estimation. *Journal of Machine Learning Research*, 8(3), 2007.