WILFRID LAURIER UNIVERSITY

ST 694 - STATISTICAL LEARNING PROJECT: APPLICATION STREAM

An Analysis of Financial Performance Indicators in Bankruptcy Prediction with Machine Learning

---

Authors
Lorenzo CAREY JR.
Austin SAMMON
Wayne SHEN

Instructor
Dr. Devan BECKER

Abstract

In this analysis, we will discover the best machine learning method to approximate the bankruptcy of the companies in Taiwan from 1999 to 2019. With feature engineering, we deduced the number of predictors to 86 features to estimate non-bankrupt (0) and bankrupt (1) as the response variable. Due to the skewness towards the non-bankruptcy present in the dataset, 10-fold cross-validation was applied to each method considered in this study to ensure the error rate is minimized. Our results suggested that both K-nearest neighbours and extreme gradient boosted tree methods are the preferable methods to estimate the bankruptcy of 6819 companies in Taiwan. The dataset was collected from the University of California Irvine Repository of Machine Learning Databases, and the code of this analysis can be found at the following,

- `https://github.com/swaynes/Statistical-Learning-Project---Bankruptcy-Prediction-with-Machine-Learning.git`
- `https://github.com/a-sammon/An-Analysis-of-Financial-Performance-Indicators-in-Bankruptcy-Prediction-with-Machine-Learning.git`
- `https://github.com/BahamianQuant/Machine-Learning-Code.git`

Contents

Introduction

In the first and second quarters of 2023, the banking industry experienced a wave of uncertainty due to the crisis of large banks such as Silicon Valley Bank and Credit Suisse. There is significant systemic risk amongst large banks in the financial sector and consequently, has a domino effect on entities exposed to this risk; in a worst-case scenario, this may result in devastating bank runs that could be a severe detriment to the global economy. To mitigate some of this risk, banks are mandated, by the Bassel regulations, to allocate capital proportional to the risk due to their lending activities; this is known as Economic Capital. When a large company or several smaller companies declares bankruptcy due to solvency issues, this critically impacts a bank's ability to continue its lending activities. Thus, an effective predictive model for defaults would allow banks to allocate capital properly for these worst cases events and also make more informed lending decisions. As such, our goal is to produce a model that accurately predicts whether a client defaults based on their financials.

1.1 Data

This analysis was conducted using data from the Taiwan Economic Journal over a period of 20 years (1999 -2009) [1]. In this data set, bankruptcy was defined according to Taiwan Stock Exchange business regulations [2]. The data set includes 220 cases that resulted in company bankruptcies and 6599 non-bankruptcy cases, so a clear class imbalance exists in the dataset. In addition to the bankruptcy indicator, 95 other features were recorded, comprising company financials, all of which can be found in Table 4.1. There was little need for any data wrangling and cleaning since the data was collected and organized in a tidy way. We used a 90-10 split

for the training and validation set, respectively. We also include a histogram showing the various different distributions of the features in our model. We observe that there are several varieties of distributions in the data with some being approximately Gaussian while others are severally skewed. Identifying the distributions inside the model can provide insight into how the data should be scaled because different modelling techniques handle heterogeneous feature distributions. For example, there are several assumptions in linear regression that may not hold for the latter, while Random Forest are generally very robust to various distributions. Similarly, a majority of the features in the data set are ratios that range between 0 and 1 while others range from 0 to the hundreds of millions. Subsequently, we the standardize the training set then, using those parameters, scaling the validation and remove the Net Income Flag feature since it added no additional information (Variance of 0).

1.2 Feature Selection and Engineering

Feature Engineering is a critical aspect of model development and is multi-faceted, however, for the purpose of this analysis, we employed minimal feature selection using a combination of statistical techniques such as correlation matrices and expert domain knowledge to remove some features. Firstly, we explored the distribution of the target variable, "Bankruptcy".

There is an extreme imbalance in the target data set since approximately 97% of clients did not default while only 3% of them defaulted. This skewness will likely cause severe bias as the models trained on these data sets will be biased toward predicting non-bankruptcy.

1.2.1 Treatments for Class Imbalances and Feature Selection/Engineering

These methods are possible treatments for the issues of class skewness and feature selection problems in our analysis, but they are not implemented in the model. There are two primary techniques used to address class imbalances: one of them is undersampling, where observations in the minority class are duplicated synthetically. Of these methods, we may have opted for oversampling to mitigate the effect of this imbalance since it performs well with larger data sets and correspondingly large majority

classes.

Similarly, some possible useful methods of engineering and selection that were not included in this paper are Principal Component Analysis (PCA) and LASSO or Ridge Regression. The prior, a mean of feature engineering, PCA, decomposes the features and transforms them in such a way that they are: uncorrelated; and preserves the information the original features captured while reducing the number of features. However, with LASSO, a means of feature selection, the method selects features that explain a significant amount of variance in the response target and penalizes terms that are not strongly correlated, shrinking their coefficients to 0.

1.2.2 Inherent Feature Correlations

Due to the nature of the features used in the model and how correlations are estimated, we expect there to be a significant amount of correlated variables. The features used in the model are financial indicators and general balance sheet items for companies. Often, these statistics are linear combinations of each other. For example, the features Net.Asset.Value.Per.Share A, is strongly correlated with the features Net.Asset.Value.Per.Share B and Net.Asset.Value.Per.Share C. However, these features are simply the net asset value of the company divided by the number of shares in class A, B and C. Similarly, the features Net.Asset.Value and Debt Ratio are strongly negatively correlated because the Debt Ratio is the Debt divided by the Net Asset Value. These correlations are endemic in the data set and thus, a thorough analysis of the features along with expert judgement is required to properly prune the features. Nevertheless, a standard correlation function was used to identify features with correlations over 0.8. Then, these are examined and some features are removed based on knowledge of financial statements.

Methodology

2.1 Model Selection

In our analysis, we use a suite of models for this classification problem and outline delineate the rationale for them in this section. Logistic regression is an immediate first choice because it's well suited for the binary nature of the response data and allows for reasonably well prediction accuracy. KNN is also a reasonable choice due to its simplistic implementation and arbitrary decision boundary. The dataset was analyzed using support vector machines because they have a very high predictive capability as well as their ability to produce non-linear decision boundaries that can take on almost any shape. Besides its ability to perform extremely well when strong features and/or a large number of features are present, random forest is also considered since it tends to perform well in general with seldom tuning. The last model considered for this analysis is a boosted tree, which is included because it is also robust to strongly correlated features and with the proper tuning has been shown to outperform random forest.

2.2 K-fold Cross-Validation

Before we apply the method of choice, we split the data into training and validation sets as shown in Table 4.2. We will apply different algorithms with 10-fold cross-validation in order to find the best model with the best error rate to find the best model for predicting bankruptcy.

Cross-Validation (CV) randomly splits the training data into a training set for fitting with the respective algorithm and a test set to calculate the mean square error (MSE) for a quantitative response variable or the error

rate (Err) from misclassification for a qualitative response variable. Then, the CV algorithm averages the value as shown below,

$$CV_{(K)} = \frac{1}{K} \cdot \sum_{i=1}^{K} MSE_{(i)} \tag{2.1}$$

$$CV_{(K)} = \frac{1}{K} \cdot \sum_{i=1}^{K} Err_{(i)} \tag{2.2}$$

K represents the number of sets or folds that the training data will be split into and how many times the algorithm will try to refit the training sets and test sets. For K = n where n is the number of observations, it is called Leave-One-Out Cross-Validation (LOOCV) by repeating the process of cross-validation n times. Since there are 6168 observations in our training data, LOOCV will be computationally intensive and not feasible for our model building. Thus, we will use 10-fold Cross-Validation where $K = 10$.

2.3 Logistic Regression

With binary response variable of non-bankrupt (0) and bankrupt (1), logistic regression is the first method of choice as it predicts our classification through linear regression. That is for i = 1, . . . , n observations

$$y_i = \frac{e^{\eta_i}}{1 + e^{\eta_i}} \tag{2.3}$$

Where

$$\eta_i = \beta_0 + \beta_1 \cdot x_{1i} + \cdots + \beta_{95} \cdot x_{95i} \tag{2.4}$$

Since the interpretability of the coefficients in logistic regression (β_j) is straightforward, logistic regression is mainly used when the goal is explainability and to gain insight into the relationships between the features and the response. Nevertheless, it also holds as a baseline for the prediction of our data.

2.4 K-Nearest Neighbours

K-nearest neighbours (KNN) is considered one of the simplest non-parametric methods for predicting outcomes based on input data. For each prediction point x_0, it takes the K nearest data points in the training set to estimate the response based on the average responses of said data points.

$$\hat{f}(x_0) = \frac{1}{K} \cdot \sum_{x_i \in \aleph_0} y_i \tag{2.5}$$

For classifiers, similar to CV, Knn estimates the response based on the most vote of said data points.

$$\Pr(Y = j | X = x_i) = \frac{1}{K} \cdot \sum_{i \in \aleph_0} I(y_i = j) \tag{2.6}$$

2.5 Decision Trees

A decision tree is a regression on the partitioned feature space of the training data with each terminal node, or ending branch, being the mean of the observation, and thus, the trees will have low bias but high variance. Therefore, we can use bagging, or bootstrap aggregating, trees to reduce variance without affecting bias. That is,

$$\hat{f}_{bag}(x) = \frac{1}{B} \cdot \sum_{b=1}^{B} \hat{f}^{*b}(x) \tag{2.7}$$

2.5.1 Boosted Tree

A boosted tree is created by combining these many decision trees in a slightly different way, as opposed to fitting a single decision tree the trees are grown sequentially learning a little bit each time. Each time a new tree is grown, information from previously grown trees is used, slowly improving each successive tree. So for b= 1, . . . , B we continually update $\hat{f}$ and the residuals as in (2.8) to achieve the final boosted model as in (2.9)

$$\hat{f}(x) \leftarrow \hat{f}(x) + \lambda \cdot \hat{f}^b(x) \qquad r_i \leftarrow r_i - \lambda \cdot \hat{f}^b(x_i) \tag{2.8}$$

$$\hat{f}(x) = \sum_{b=1}^{B} \lambda \cdot \hat{f}^b(x) \tag{2.9}$$

Extreme gradient boosting is an extension of gradient boosting decision trees that increase computation speed and model performance. Gradient boosting makes use of a gradient descent optimization algorithm such that each new tree attempts to minimize (2.10) until a minimum is achieved. With extreme gradient boosting trees are built using the same algorithm but make use of the Newton-Raphson method to calculate the second order gradient providing more information about where the minimum is, as well as regularization techniques which help to prevent overfitting. These additions in turn increase computational speed and model performance.

$$R(\theta) = \frac{1}{2} \sum_{i=1}^{n} (y_i - f_\theta(x_i))^2 \tag{2.10}$$

2.5.2 Random Forest

If there exist strong features in the data, when building a large number of trees most trees would look similar and thus be highly correlated. Random forests are an extension of bagged trees that help to decorrelate the trees. At each split in random forest only a random subset $m \simeq \sqrt{p}$ is considered where p is the total number of features, meaning that on average $\frac{p-m}{p}$ won't consider the strong features.

2.6 Support Vector Machine

Based on maximum marginal classifiers, this method extends support vector classifiers. In this method a hyperplane is introduced that doesn't necessarily separate the two classes perfectly and allows for misclassifications of some of the observations, these are called support vectors. It then becomes an optimization problem trying to maximize the width of the margin M such that

$$y_i(x_i^T\beta) \geq M(1-\epsilon_i) \text{ for } \epsilon_i \geq 0;\ \sum_{i=1}^{n}\epsilon_i \leq C \tag{2.11}$$

With C controlling the amount of violation to the margin called the cost, and ϵ the slack variables allowing for misclassification. Support vector machines (SVM) allow for non-linear boundaries between the classes through the use of kernels to enlarge the feature space. The support vector classifier can be represented as (2.12) with K being the function representing the kernel. Kernels considered in this study include radial and sigmoidal in (2.13) respectively.

$$f(x) = \beta_0 + \sum_{i\in S}\alpha_i K(x_i, x_{i'}) \tag{2.12}$$

$$\begin{aligned} \text{Radial Kernel: } K(x_i, x_{i'}) &= \exp(-\gamma\sum_{j=1}^{p}(x_{ij}-x_{i'j})^2) \\ \text{Sigmoidal Kernel: } K(x_i, x_{i'}) &= \tanh(\alpha(x_i \cdot x_{i'}) + \beta) \end{aligned} \tag{2.13}$$

Models Comparison

3.1 Quantify Comparison

In order to measure the prediction capability of each model, we will classify bankrupt as negative, and non-bankrupt as positive,

- Accuracy: Measured by the number of correct predictions over total predictions

$$\text{Accuracy} = \frac{\text{True Positive} + \text{True Negative}}{\text{Total Number of Observations}} \tag{3.1}$$

- Sensitivity: Calculated by the number of predicted non-bankrupts over total non-bankrupts

$$\text{Sensitivity} = \frac{\text{True Positive}}{\text{Total Number of Actual Positives}} \tag{3.2}$$

- Specificity: Calculated by the number of predicted bankrupts over total bankrupts.

$$\text{Specificity} = \frac{\text{True Negative}}{\text{Total Number of Actual Negatives}} \tag{3.3}$$

3.2 Result

We construct our suite of models using the optimal tuning parameters, where applicable, obtained via cross-validation, and display their performance in the below table. .

Table 3.1: Methods Comparison

Methods	**Accuracy**	**Senstivity**	**Specificity**
Logistic Regression	0.9677	0.9984	0.0909
KNN Classifier	0.9723	0.9984	0.2272
Random Forest	0.9708	0.9984	0.1818
Extreme Gradient Boosting Tree	0.9723	0.9952	0.3181
Support Vector Machine (Radial)	0.9662	1	0
Support Vector Machine (Sigmoidal)	0.9662	1	0

3.3 Discussion

From Table 3.1, it is evident that the class imbalances are having a significant impact on more than half the models in our suite as those models

affected are predicting all observations as bankrupt. The Logistic regression, K-nearest neighbours, random forest, and boosted tree models excel in accuracy and sensitivity but not as well in specificity performing relatively well despite the class imbalance. However, the sigmoidal and radial support vector machines perform reasonably well in accuracy, but the sensitivity and specificity are entirely impacted by the imbalances. It is difficult to visualize SVM results, especially in higher dimensions. Though, visualizing a two dimensional case, the hyperplane generated by the SVM will be skewed towards the minority class. Moreover, the imbalances also influence the support vectors in the margin, causing an imbalance between negative and positive support vectors. This results in the SVM performing poorly on classifying minority class observations as discussed by Núñez [8].

Interestingly, the logistic model performs surprisingly well, relatively, on the validation set, particularly its specificity. Because of the weak representation of the majority class, it was expected that the logistic regression would not have enough information about their patterns to approximate its distribution properly. Even more so, the kNN classifier performed relatively well determining actual defaults and there is some theoretical basis to explain kNN's performance. The kNN classifier is robust to influences from the actual magnitude of classes in the response variable and consequently won't be biased to any class regardless of its size. This is a possibility of why this classifier had a significantly higher specificity compared to other models.

The most distinct aspect of the results are how big of an impact the imbalances have on the predictive ability of the models. However, the is likely because of the manner in how each model makes it classification; this highlights how, for some problems, different models are inherently more robust than others.

3.4 Conclusion

One of the most peculiar insights from this analysis is how certain models are inherently superior performers for a certain set of problems. However, it invokes an endeavor to improve the performance of weaker performing models on certain problems. A possible treatment of this may be examining the loss functions in the weaker models. For example, for the

class imbalance issue in our data set, models using a linear loss function like Classification Error will not be able to able to detect subtle changes in the minority class. However, convex loss functions such as Cross Entropy Loss are very sensitive to small changes in the response probability when the feature is close to zero. Thus, exchanging the linear loss function for a more convex one, may improve the sensitivity of the under-fit models.

Although the extreme gradient boosted tree and kNN classifier had the best performance in our analysis, we would not recommend them for use in risk management decisions. These models only produce a true non-default approximately $\frac{1}{4}$ of the time. Due to the repercussions insufficient economic capital caused by the models low specificity, we would recommend maximizing the important metrics from the from Table 3.1. This may be achieved through the use of feature engineering like PCA or selection methods like LASSO combined with a refined domain knowledge.

After such, these models may be further validated and if their is a reasonable increase in the relevant performance metrics, it may generate insight and accurate predictions of bankruptcy amongst companies in a financial institution's loan portfolio.

Bibliography

[1] Liang D, Tsai CF (2020). *Taiwanese Bankruptcy Prediction Data Set*. UCI Repository of Machine Learning Databases. https://archive.ics.uci.edu/ml/datasets/Taiwanese+Bankruptcy+Prediction.

[2] Taiwan Stock Exchange. (2022). *Operating Rules of the Taiwanese Stock Exchange Corporation.* https://twse-regulation.twse.com.tw/ENG/EN/law/DAT0201.aspx?FLCODE=FL007304.

[3] James G, Witten D, Hastie T, Tibshirani R. (2013). *An Introduction to Statistical Learning with Applications in R*. New York, Springer Science and Business Media.

[4] Wittmann F. M. (2016). *Visualization of SVM Kernels Linear, RBF, Poly and Sigmoid on Python* https://gist.github.com/WittmannF/60680723ed8dd0cb993051a7448f7805.

[5] Nvidia. (2023). *XGBoost*. Nvidia Corporation. https://www.nvidia.com/en-us/glossary/data-science/xgboost/

[6] Shah A. (2022). *XGBoost (Extreme Gradient Boosting) in Machine Learning*. Medium. https://medium.com/@jwbtmf/xgboost-extreme-gradient-boosting-in-machine-learning-3427b937b35c

[7] Keany E. (2020). *What makes "XGBoost" so Extreme?*. Medium. https://medium.com/analytics-vidhya/what-makes-xgboost-so-extreme-e1544a4433bb

[8] Núñez H, Gonzalez-Abril L, Angulo C. (2017). *Improving SVM Classification on Imbalanced Datasets by Introducing a New Bias*. Journal of Classification. pp. 427–443.

Appendices

4.1 Supplementary Tables and Figures

Table 4.1: Financial Ratios and Corporate Governance Indicators

ROA(C) before interest and depreciation before interest	ROA(A) before interest and % after tax	ROA(B) before interest and depreciation after tax
Operating Gross Margin	Realized Sales Gross Margin	Operating Profit Rate
Pre-tax net Interest Rate	After-tax net Interest Rate	Non-industry income and expenditure/revenue
Continuous interest rate (after tax)	Operating Expense Rate	Research and development expense rate
Cash flow rate	Interest-bearing debt interest rate	Tax rate (A)
Net Value Per Share (B)	Net Value Per Share (A)	Net Value Per Share (C)
Persistent EPS in the Last Four Seasons	Cash Flow Per Share	Revenue Per Share
Operating Profit Per Share	Per Share Net profit before tax	Realized Sales Gross Profit Growth Rate
Operating Profit Growth Rate	After-tax Net Profit Growth Rate	Regular Net Profit Growth Rate
Continuous Net Profit Growth Rate	Total Asset Growth Rate	Net Value Growth Rate
Total Asset Return Growth Rate Ratio	Cash Reinvestment %	Current Ratio
Quick Ratio	Interest Expense Ratio	Total debt/Total net worth
Debt ratio %	Net worth/Assets	Long-term fund suitability ratio (A)
Borrowing dependency	Contingent liabilities/Net worth	Operating profit/Paid-in capital
Net profit before tax/Paid-in capital	Inventory and accounts receivable/Net value	Total Asset Turnover
Accounts Receivable Turnover	Average Collection Days	Inventory Turnover Rate (times)
Fixed Assets Turnover Frequency	Net Worth Turnover Rate (times)	Revenue per person
Operating profit per person	Allocation rate per person	Working Capital to Total Assets
Quick Assets/Total Assets	Current Assets/Total Assets	Cash/Total Assets
Quick Assets/Current Liability	Cash/Current Liability	Current Liability to Assets
Operating Funds to Liability	Inventory/Working Capital	Inventory/Current Liability
Current Liabilities/Liability	Working Capital/Equity	Current Liabilities/Equity
Long-term Liability to Current Assets	Retained Earnings to Total Assets	Total income/Total expense
Total expense/Assets	Current Asset Turnover Rate	Quick Asset Turnover Rate
Working capitcal Turnover Rate	Cash Turnover Rate	Cash Flow to Sales
Fixed Assets to Assets	Current Liability to Liability	Current Liability to Equity
Equity to Long-term Liability	Cash Flow to Total Assets	Cash Flow to Liability
CFO to Assets	Cash Flow to Equity	Current Liability to Current Assets
Liability-Assets Flag	Net Income to Total Assets	Total assets to GNP price
No-credit Interval	Gross Profit to Sales	Net Income to Stockholder's Equity
Liability to Equity	Degree of Financial Leverage (DFL)	Interest Coverage Ratio (Interest expense to EBIT)
Net Income Flag	Equity to Liability	

Table 4.2: Class Breakdown of Data Split

	Training Set	Validation Set	Overall Dataset
Bankrupt	198	22	220
Non-Bankrupt	5970	629	6599
Total	6168	651	6819

Delineation of work

Throughout the process of conducting this study, each group member contributed equally to the writing of the report, the construction of the presentation and the formulation of the code. Wayne was responsible for all loops present in the code.