

**Corporate** **Bankruptcy**

Prediction

Using

**Machine** **Learning** **Techniques**

ADMN5006

Arnab Barua

Luke Laylo

Jad Naitali

Varun Nallamothu

Sreehari Ramadas

Ghita Tazi

**Abstract**

Since the last few decades, corporate bankruptcy has been of interest to researchers around the world and a source of worry for business stakeholders including management, investors, etc. Due to the numerous elements that contribute to bankruptcy, it is not enough to rely on a single predictive model; instead, the difficulty is in identifying only the most important factors. The high-class imbalance in the data is another significant barrier that impairs the performance of the model. There are many techniques that have been studied in the past, including Decision Trees, SVM, Neural Networks, and others, which use various pre-processing techniques.

**Introduction**

One important source of concern shared by creditors, investors, employees, and management is bankruptcy. It is a word used to describe a business that lacks the necessary operational funding to run the enterprise and is unable to make good on its obligations to its creditors. It is a challenging scenario to be in because the business comes to a standstill, leaving the staff and suppliers/customers high and dry. These days, increased competition and unpredictability in the global economy could push huge companies into bankruptcy. The most recent instance is the abrupt bankruptcy of the British travel company Thomas Cook, which resulted in the layoff of 21,000 individuals and the loss of millions of dollars in investment funds. It is therefore hard to overestimate the harm done in terms of the monetary loss. Corporate bankruptcy causes recession and has a detrimental effect on a nation's economy (Bernanke; 2015). The capacity to predict bankruptcy is crucial for making sound economic decisions since all businesses, no matter how big or small, affect investors, residents of the community, and business owners, which has an impact on both national and international policymakers. It can be crucial for a creditor to verify the financial indicators and identify the likelihood of bankruptcy because the corporate sector's performance heavily influences a nation's economy. Such choices may have an impact on the economic growth of a nation in addition to the development of a firm. A quick and reliable bankruptcy model is necessary, as seen by the current global financial crisis that various nations have experienced. The two main methods used worldwide to foresee a company's potential insolvency are the structural approach where the firm's characteristics and interest rates are thoroughly studied to estimate the likelihood of default. Balcaen and Ooghe (2006) provided a detailed analysis of statistical methods, and Kumar and Ravi (2007) investigated the use of intelligent strategies in addition to statistical methods to forecast the bankruptcy rate.

Although it is important to predict bankruptcy, there is currently a need to use the proper features to enhance prediction performance and cut down on computational time and expense. Second, there is a tonne of businesses that are not in bankruptcy, compared to the small number of businesses that do go bankrupt, albeit those few are sufficient to disrupt a number of industries and the economy as a whole.

The pre-processing and data transformation step, which is actually one of the most crucial stages of a data mining process, has been the focus of this study. The goal is to combine an acceptable feature selection strategy with a resampling technique, and then feed the changed data to a few machine learning models to compare their results.

**Related Work**

The issue of class imbalance is among the most difficult aspects of solving the bankruptcy prediction problem. As is well known, there are tens of thousands of stable, successful non-bankrupt enterprises operating all over the world, but there are very few bankrupt businesses. Despite being few in number, these failing businesses have the potential to cause financial catastrophe among lenders and investors as well as economic damage for the entire nation. There are a number of strategies that have also been investigated by researchers in past studies that can be used to address the problem of class imbalance in machine learning models. We will gain a general knowledge of how well these strategies work in the bankruptcy prediction problem from the literature in this section.

**Dealing with Class Imbalance**

In a study by Le et al. (2019), the authors used two approaches to deal with the problem of class imbalance. On the Korean bankruptcy dataset, the authors have developed a hybrid strategy employing cost-sensitive learning and oversampling techniques. First, an ideal performance on the validation set is calculated using an oversampling module and an optimal balancing ratio. Second, the CBoost method is employed as a learning model for bankruptcy prediction that is cost sensitive. In the dataset, there were 120048 non-bankrupt enterprises and 307 bankrupt firms, resulting in a balancing ratio of 0.0026. In our dataset, there are 92314 non-bankrupt companies and 558 companies that are bankrupt, which results in a balancing ratio of 0.006.

**Data Exploration**

Using visuals is one of the quickest ways to study and comprehend the data. When it comes to understanding the data's structure, the distribution of its values, and the existence of any correlations within the dataset, the findings of visual data exploration can be quite effective. Therefore, the findings of our dataset are as below:

* Identifying the datatype of all the variables:

Text, table

Description automatically generated

* Checking the presence of null values in the dataset:

Diagram

Description automatically generated with medium confidence

Table

Description automatically generated

* Correlation between the data frame

**Table

Description automatically generated**

Chart

Description automatically generated

* Plotting bar-graph to determine the distribution of bankrupt and non-bankrupt cases:

Chart

Description automatically generated

* Tackling Imbalanced Data

One of the frequent issues with classification modelling is the existence of uneven class distribution (Elrahman and Abraham; 2013). This indicates that there is a significant disparity in the number of observations made by one class and the other class. Because most machine learning algorithms are built to function best when both classes are equally balanced, this is a difficult problem. The prediction model that would be created in the event of a class imbalance could prove to be biased and inaccurate.

This might make it more likely that the minority class will be incorrectly classified.

To deal with class imbalance, we employed Synthetic Minority Oversampling Technique or simply SMOTE. SMOTE helps to tackle class imbalance by randomly increasing minority classes by replicating them.

**Modelling Approach**

This step in the machine learning process is regarded as crucial and necessary. The proposed models are to be put into practice after the data preparation procedure, which included feature selection and resampling. The effects of preprocessing methods on the various models can then be assessed and contrasted. Information on the model that was used, and this part discusses the model's operation.

* **Random Forest:**

An ensemble of several decision trees, known as a random forest model, is frequently applied to classification issues. It builds each tree using methods like feature randomness and bagging, resulting in an uncorrelated forest of trees (Khoshgoftaar; 2007). Every tree is supported by a distinct random sample.

This group of trees' prediction performance is more precise than that of any single tree. A few of the characteristics that make it a good fit for the selected dataset are the model's quick training speed, outlier resistance, and capacity for handling unbalanced data.

* **Naïve Bayes Classifier Algorithm:**

Based on the Bayes theorem, the Naive Bayes classifier categorises each value as independent of every other value. It enables us to forecast a class or category using probability utilising a given set of features.

Despite its simplicity, the classifier performs admirably and is frequently used because it outperforms more complex classification techniques.

* **Linear Support Vector Classifier (SVC):**

With a high number of data, the Linear Support Vector Classifier (SVC) approach performs well. It uses a linear kernel function to perform classification. When compared to the SVC model, the Linear SVC adds more parameters including the loss function and penalty normalization, which applies "L1" or "L2." Because linear SVC is based on the kernel linear technique, the kernel method cannot be modified.

**Implementation**

The detailed implementation of the suggested models for predicting corporate bankruptcy is described in this section. Additionally, it outlines the procedure used to resample the dataset and choose the most crucial attributes. Python was used for the entire implementation process, and Jupyter Notebook was used as the Integrated Development Environment (IDE). Python was selected for the implementation phase because it is simple to use, has a large online support community, and is also regarded as one of the top languages for code readability. Because there is a vibrant Python community, there are many tools available for managing unbalanced data and preparing data, making Python a popular choice for machine learning projects.

To begin with, we had to design and develop a machine learning model for bankruptcy prediction 2 years into the future using the dataset provided to us. The column “BK” in the data provided denotes whether the company goes bankrupt (indicated by 1) 2 years in the future, or not (indicated by 0). - Some of the input variables included are “Assets Growth”, “Sales Growth”, “Earnings-per-share (EPS)”, “Return-on-equity (ROE)”, etc.

Our goals were to understand the data that was provided to us and create a 50/50 sub-data frame ratio of “Bankrupt” and “Non-Bankrupt”.

Determine the classifiers that we are going to use and decide which one has the higher accuracy and UAC and understand the mistakes made with imbalanced datasets.

We imported the libraries that are going to help us solve the dataset. We, then, imported the data source into the notebook and saw the basic statistics of the report.

We witnessed the dataset consisted of 13 columns ('EPS', 'Liquidity', 'Profitability', 'Productivity', 'Leverage Ratio', 'Asset Turnover', 'Operational Margin', 'Return on Equity’, 'Market Book Ratio', 'Assets Growth', 'Sales Growth', 'Employee Growth', 'BK') and 92872 rows. To check our data validity; we needed to check if there are missing values in the table, check for uniqueness, check for normality, and also checked for extreme outliers in the dataset that could distort the algorithm.

We then checked for missing values in our dataset and got to know there were 12 columns that had missing values. If the columns had more than 50% missing values, we would have dropped the columns but looking at the percentage, we didn’t need to drop any of the columns. Following this, we checked for unique values in the columns and looking at the dataset all columns had more than 2 distinctions therefore, there wasn’t a need for dropping any column.

Delete Rows which have more than 60% missing values. Out of 92872 rows, we deleted 64 rows that had more than 7 columns that had no value in them. By seeing the distributions, we can have an idea of how skewed these features are, we can also see further distributions of the other features. Furthermore, we added missing values using KNNImputer with n\_neighbours = 5 as a result there were no missing values remaining in our dataset. After that, we split our data into training and testing data. We then split the data into training and testing classes. Then we chose three models to work with, which were Random Forest, Linear SVC and Naïve Bayes. We ran the three models and compared their outputs for the first round. The performance of models has been compared later in this literature.

We then performed dimensionality reduction to see if the performance improved. But we implied applying dimensionality reduction didn’t help in this dataset so we dropped that idea. Instead, we got a sub-sample of the dataset. This is because we saw that the original dataframe was heavily imbalanced. Using the original dataset would cause:

**Overfitting**:

* Our classification models will assume that in most cases there are no frauds! What we want for our model is to be certain when fraud occurs.

**Wrong Correlations**:

* by having an imbalanced dataframe we are not able to see the true correlations between the class and features.

Therefore, we applied SMOTE (Synthetic Minority Oversampling Technique). Smote is one of the more commonly used oversampling techniques to deal with the imbalance problem. It helps to balance the class distribution by randomly increasing minority classes by replicating them. It further helps in:

* **Location of the synthetic points**: SMOTE picks the distance between the closest neighbors of the minority class, in between these distances it creates synthetic points.
* **Final Effect**: More information is retained since we didn't have to delete any rows unlike in random undersampling.
* **Accuracy || Time Tradeoff**: Although it is likely that SMOTE will be more accurate than random under-sampling, it will take more time to train since no rows are eliminated as previously stated.

**We applied RobustScaler.**

It scales features using statistics that are robust to outliers. This method removes the median and scales the data in the range between 1st the quartile and 3rd quartile. i.e., in between 25th quantile and 75th quantile range. This range is also called an Interquartile range.

The median and the interquartile range are then stored so that it could be used upon future data using the transform method. If outliers are present in the dataset, then the median and the interquartile range provide better results and outperform the sample mean and variance.

RobustScaler uses the interquartile range so that it is robust to outliers.

#### **Confusion Matrix:**

* Positive/Negative: Type of Class (BK) ["No", "Yes"] True/False: Correctly or Incorrectly classified by the model.
* True Negatives (Top-Left Square): This is the number of correctly classifications of the "No" (No Bankruptcy Detected) class
* False Negatives (Top-Right Square): This is the number of incorrectly classifications of the "No"(No Bankruptcy Detected) class.
* False Positives (Bottom-Left Square): This is the number of incorrectly classifications of the "Yes" (Bankruptcy Detected) class
* True Positives (Bottom-Right Square): This is the number of correctly classifications of the "Yes" (Bankruptcy Detected) class.

Chart, treemap chart

Description automatically generated

**Model Comparison (Round 1):**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Accuracy** | **Precision** | **Recall** |
| **Random Forest** | **99%** | **0.50** | **0.00** |
| **Naïve Bayes** | **6%** | **0.01** | **0.95** |
| **Linear SVC** | **99%** | **0.0** | **0.0** |

**Model Comparison (Round 2):**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Accuracy** | **Precision** | **Recall** |
| **Random Forest** | **97%** | **0.97** | **1.00** |
| **Naïve Bayes** | **52%** | **0.51** | **0.97** |
| **Linear SVC** | **78%** | **0.82** | **0.71** |

**Note:** Accuracy is a good measure of model performance if the dataset we are dealing with doesn’t have class imbalance whereas, if the classes are imbalanced, it is recommended to look for precision and recall.

**Evaluation**

We concluded that after our final data processing, the performance comprising of accuracy, precision and recall improved significantly for many if not all.

For Random Forest:

The accuracy and precision kind of remained almost the same, however, the recall improved from 0.00 to 1 (which is great).

For Naïve Bayes:

The accuracy, now, bumped up to 52% and the precision increased from 0.01 to 0.51.

For Linear SVC:

The accuracy saw a downfall yet the precision and recall improved.

Based on this performance table, it is quite simple to determine the fact that Random Forest is the best model (among the three models compared) for this kind of dataset.

**Conclusion & Future Work**

In order to get the best forecast performance, many strategies have been used, as can be seen from the prior literature. The topic of bankruptcy prediction has garnered interest in recent decades. The difficult element, as mentioned, is choosing the best financial factors that play a significant role in a company's bankruptcy. The issue of class disparity is another barrier to this investigation.

In conclusion, we get to the result that among the three models compared, Random Forest was the best fit for our dataset.

These methods can be investigated further using alternative classifiers and sets of financial attributes. In this study, we focused only on the dataset presented to us for our assignment. One may afterwards investigate bankruptcy information for any other dataset in the financial setting.