Columbia-ACTU-5841-Project

Final Group Project for ACTU 5841 - Data Science in Finance And Insurance

By: Dennis Goldenberg, Anh Vu Lieu, Tianyi Xu, Esteban Gutierrez, Samuel Ma

Our Dataset:

For the final project for data science for finance and insurance we decided to work on a dataset to predict bankruptcy in companies based on different financial ratios that describe the financial health of a company. The definition of bankruptcy was the one provided by the Taiwan Stock Exchange, we obtained the dataset from Kaggle, the publisher of the original dataset is the National Central University of Taiwan.

The dataset was composed of 96 columns, one of them was if the company was either bankrupt or not defined by 1 (bankrupt) and 0 (non-bankrupt). The other 95 features were different financial ratios related to the return on different measures, other macro-economic factors, and the different margins that a company may have at different operating levels.

Here is an example of some of the features to get an idea of what they are. First, we have the net income to total assets, which tells us how much revenue a company has generated divided by the assets that helped brin that income. Second, Net worth to assets which tells us how much the company is worth divided by its assets. Third, Gross profit to sales, this feature is the income after subtracting the costs of the products sold divided by the total revenue. Also called Gross profit margin. Then, there is the debt ratio that measures how much a company is leveraged and is calculated by dividing total liabilities to total assets. And the last example is the ROA, which is the return on assets that a company may have either before or after applying taxes.

We choose to make our project on bankruptcy because we believe that bankruptcy in a business is a significant risk not only to the individual enterprise that is going under, but also to the broader global economy. Our approach focuses on a comprehensive analysis of various Key Performance indicators that are indicative of a company’s financial health to predict if the company has the possibility of going bankrupt. When this problem goes unaddressed, these indicators can lead to a company’s downfall and have major consequences in many people’s lives.

The core objective with this project is to develop a robust predictive model capable of accurately forecasting the likelihood of a company facing bankruptcy. This tool will be instrumental in enabling businesses to make informed and timely decisions, potentially averting financial crisis and contributing to overall economic stability.

Data Preprocessing - Feature Relationships and Feature Selection

In the realm of financial analysis, preprocessing data is a crucial step in unveiling meaningful insights and building robust predictive models. The dataset contains information on 6819 companies and their 96 attributes including one that indicating bankruptcy and 95 financial features. After we read the data file into DataFrame “bdata”, we dropped the variable named “ Net Income Flag” due to its zero variance, as all companies in our dataset have a negative net income for the last two years. Subsequently, we computed a correlation matrix, laying bare the interrelationships between different financial variables. This matrix is then visualized as a heatmap, leveraging Matplotlib, providing an intuitive representation of variable correlations.

The correlation matrix heatmap shows most financial features have little correlation with the target variable “Bankrupt?”. Aiming to filter our pertinent features, our group defined the threshold to be ± 0.1 and identified variables with correlation to bankruptcy larger than 0.1 and smaller than -0.1. There are 32 variables considered relevant based on the threshold, and we then created a list of these variables, denoted as “important_variables”. Additionally, the dataset is partitioned into feature variables (X) and the target variable (y), setting the stage for subsequent modeling endeavors.

Upon scrutinizing the correlation matrix heatmap, we realized multiple selected features exhibit high intercorrelation. For example, features such as "Net Income to Total Assets," "ROA(A) before interest and % after tax," "ROA(B) before interest and depreciation after tax," and "ROA(C) before interest and depreciation before interest" displayed a discernible red hue, indicative of their strong correlation. This observation underscores an opportunity for further dimensionality reduction in our dataset, and we decided to pursue additional preprocessing methods.

Data Preprocessing - PCA and Imbalance Sampling

Principal Component Analysis

PCA simplifies the data set by reducing dimensionality while retaining the variation information in the original data set as much as possible. Given that our original dataset had 95 features and the features selected through the correlation matrix still had 32, PCA is useful.

PCA first accounts for correlation by creating perpendicular components, then it reduces dimensionality by taking components accounting for most variance. We use PCA on the original dataset (containing 95 features) and the dataset after feature selection (containing 32 features). The result shows that selection of important features helps to reduce the dimensionality of the data, with just 12 components being able to explain 95% of the variance of the 32 highly correlated features compared to 52 components explaining 95% of the variance of all features prior to the selection. Therefore, we use the 12 principal components obtained in the graph on the right for the next modeling work.

Imbalanced Sampling

In addition to having too many features, our dataset suffers from severe data imbalance. In our data set, 6599 companies do not go bankrupt, and only 220 companies go bankrupt. That is to say, nearly 97% of the values in our target variable (binary variable) are 0, and 3% are 1. This results in the model tending to predict 0 for most samples because doing so is statistically more likely to be correct. However, as our main goal is to identify companies that are likely to go bankrupt, such a tendency is dangerous.

To solve this problem, we use the popular method in oversampling, SMOTE. The main purpose is to increase the weight of the minority class, thereby alleviating the problem of misprediction.

SMOTE creates new samples by finding the K nearest neighbors of each minority class sample, and then interpolating between these neighbors. The feature value of the newly generated data will be between each minority class sample and its neighbor samples. In this way, we increase the number of “1” in the target variable to the same number as the number of “0”. Note that we applied this imbalanced sampling on only the training dataset, as 20% of the dataset was allocated towards testing our models. This feature selected, pre-processed, and balanced data is finally ready for us to implement our selected statistical machine learning models.

Methods we Implemented - Descriptions and Performance

K-Nearest Neighbors

KNN is a supervised machine learning algorithm used for classification and regression. It groups observations based on their similarities to a specified number of neighbors. The number of neighbors, denoted as 'k,' is an important hyperparameter. For this dataset, we experimented with different values of 'k' using cross-validation and observed the performance of each model to identify the one that performs the best. We then used the optimal 'k' value to train the final model and evaluated it with the test set. According to the results, setting 'k' to 2 yielded the best performance with an accuracy of 0.94, the highest among the models we considered. However, the model performed poorly in predicting the true label for the positive class, a critical aspect for solving our problem. The confusion matrix reveals that this model correctly predicts the true label for class 1 only slightly more than half of the time (50.98%). We will revisit and compare the results from different models in the last section.

Linear and Quadratic Discriminant Analysis

Discriminant Analysis is a supervised machine learning classification technique used to identify a combination of features that best separates classes in a dataset, assuming that these features follow a normal distribution. We employ two different types: For Linear Discriminant Analysis, we assume that features share the same covariance matrix. For Quadratic Discriminant Analysis, we assume that features have different covariance matrices. Upon examining the distribution of values in the features, we observe that most of them exhibit a normal distribution. However, we do not perform the models on the original dataset because the data has undergone processing with PCA, making interpretation impossible. Linear Discriminant Analysis (LDA) achieves a prediction accuracy of 0.84, while Quadratic Discriminant Analysis (QDA) performs better with an accuracy of 0.96. Despite its lower overall performance, LDA excels in predicting the true label for class 1 (90.20% compared to 37.25% for QDA).

Logistic Regression

Logistic Regression is a statistical technique used in the field of data analysis to predict the probability of a binary outcome. It is particularly useful in situations where the outcome can only take two values, meaning there are only two possibilities. We employ the GridSearchCV method from Scikit-Learn to select the appropriate hyperparameters for optimal performance in our model. The best-performing hyperparameters are: {'C': 1, 'penalty': 'l1', 'solver': 'saga'}, and the highest prediction accuracy achieved is 0.85. The model accurately predicts the true label for class 1 with an accuracy of 82.35%.

Naive Bayes

Naive Bayes is a supervised learning algorithm used for classification and regression tasks. The method assumes that, for given values of the class variable, the features are independent of each other. As we have seen earlier, the features in our dataset are strongly correlated, so using this method on the original dataset results in much lower accuracy. The model performs significantly better after processing the data with PCA and oversampling. We choose Gaussian Naive Bayes because the other methods are not applicable to our dataset. The test accuracy is 0.926, which is average among our models. However, this method is very poor at predicting the true label for class 1, with only a 29% accuracy rate.

Decision Trees

Decision Trees are supervised learning methods used for classification and regression. In contrast to other methods, we run the Decision Tree (DT) model using the original dataset. We employ the DecisionTreeClassifier with the following parameters: max_depth=10, min_samples_leaf=4, min_samples_split=10. The result on the test set is an accuracy of 0.897, which is not as good as some of the other models we have introduced. However, when we use the RandomForestClassifier to enhance accuracy by reducing overfitting, the overall accuracy reaches 0.95. The ability to predict the true label for class 1 remains the same at 56.86%.

Final Analysis - Conisderations for further study

Many of our models had high accuracy in bankruptcy prediction. However, as previously mentioned, our dataset was incredibly imbalanced; 96.77% of companies were not bankrupt, so a naive classifier that just picked 0 every single time would be able to achieve about 95% accuracy. Therefore, a more notable result is the True Positive rate, or the probability of predicting bankruptcy given that a company's true outcome is 1, or bankrupt. This is for 2 reasons; first, a false negative, or failing to predict bankruptcy, is more dangerous financially than a false positive. Second, a high true positive rate indicates that the model is not overfit to all of the 0s in our dataset. From this perspective, the models with the highest True Positive rate were Linear Discriminant Analysis, and logistic regression. Linear Discriminant Analysis likely worked well due to the separability emphasized by our minority oversampling method and the interpolation between existing data points for bankrupt companies it introduced. Logistic Regression likely worked well due to the regularization on the coefficients disallowing overfitting on the mostly non-bankrupt dataset.

Our models likely could have been improved by a more sophisticated minority oversampling method. We simply balanced out the dataset by oversampling from the minority The article from which we derived the idea recommended not just oversampling the minority class but also undersampling the majority class; that way, the extra variance from all of the new data points is limited. Each model was also only trained on a subset of our features, and the subset was chosen by the magnitude of the correlation coefficient, which only measures linear relationships; this assumes a linear relationship between the financial ratios and bankruptcy, which may not be uniformly the case across all of our features; a better heuristic on feature selection may be necessary. Finally, though beyond the scope of this particular class, the utilization of deep learning - such as Neural Networks and Convolutional Neural Networks - with its many parameters could have helped deal with the large number of features relative to our simple binary classification task. Further Consideration and study should be completed along all 3 avenues - minority oversampling, feature selection intuition and deep learning implementation.