Predicting Bankruptcy of Polish Companies

Mohammed (Ali) BENCHEKROUN | Daniel CHUNG | Muhammad (Ahmad) HUSSAIN

Problem Definition

Our problem focuses on bankruptcy prediction for Polish companies and in the process we seek to identify and understand in depth financial indicators that can indicate risk of bankruptcy. Predicting the likelihood of bankruptcy is a key decision in the lending process where banks seek to protect against bad loans. Accurate prediction models can have two transformational effects: the first is to enable access to financial capital where it is required and in the process, accelerate the process of economic growth. The second is to ensure more financial stability for banks where they remain insulated against high-risk lending which has been a key feature in economic downturns of the past.

Our focus is on analyzing a comprehensive list of financial metrics encompassing profitability, liquidity and other financial ratios drawn from the companies' reported bank statements to discern whether certain metrics have an outsized influence on a company's bankruptcy odds.

Dataset

The dataset is about bankruptcy prediction of Polish companies. The data was collected from Emerging Markets Information Service (EMIS, [Web Link]),[1] which is a database containing information on emerging markets around the world. The bankrupt companies were analyzed in the period 2000-2012, while the still operating companies were evaluated from 2007 to 2013. The dataset contains the financial data for 10503 companies with 64 financial attributes measured for each. The target variable encodes whether the company went bankrupt or not after 3 years of follow-up.

Data Preprocessing

Our original dataset was a challenge in terms of drawing sound analysis from. We identified three broad issues across the dataset which was an issue for running prediction models: missing data, heavy outliers and class imbalance.

Our data have many missing values: if we choose to remove all lines containing at least one missing value, we would have lost at least 50% of the dataset. In order to treat this, we utilized a k-means imputation algorithm where we assign the missing value for a particular dimension the mean value of its nearest neighbors. In terms of initial EDA, we realized that a significant number of our columns had abnormally long tails which was caused by leverage points (extreme points in the predictor space) that existed both in directions (minimum and maximum). This had the possibility of skewing the regression coefficients and hence we chose to treat for this by incorporating a winsorization method. By utilizing this, we chose to set winsorize values that were beyond the middle 98% of the data and in doing so, we restored the data probability distribution to resemble a normal curve.

Class imbalance was another area of interest for us since the number of bankruptcy cases (4.71%) were far fewer than the number of non-bankruptcy cases (95.29%). This would potentially be an issue in terms of training our data models. To ensure our models were well-trained, we ran three different variants of each model. The first model was with the original dataset. The second was with an oversampled version where we generated synthetic instances of the minority class to ensure we had a sufficient number of data points with positive class. The third case was generated through undersampling where we decrease the size of the majority class. Alongside this, we ensured that our train/test split accounted for the class imbalance by ensuring we have the same proportion of positive class in both the training and testing set.

Model Approaches

Logistic Regression:

We first tried logistic regression both because it was the simplest model but also because it has interpretable coefficients. In total, we fit 12 logistic classification classifiers. This was to evaluate model performance on all 3 variations of the data, namely the original, undersampled, and oversampled data.

For each of these 3 data variations we evaluated 4 potential logistic regression models. The first was an initial model with no sparsification and no hyperparameter tuning. After assessing the results of the initial model, we evaluated which features were significant ($p<=0.05$) and created a sparse, second model trained only on those significant features. We cross-validated the third logistic regression model using random grid search, using 3 folds and 100 iterations (random seed 42), using the best hyperparameters found from random gridsearch. This cross-validated model was not sparsified, but we created a fourth logistic regression model that was both sparsified and cross-validated.

To find a classification threshold for each model, we plotted recall, precision, and accuracy and chose the threshold that maximized recall while preserving accuracy above 0.5. For each of the 3 data cases, we selected the best of the 4 logistic regression classifiers and reported the results from that model.

The coefficients of these models offered our first insights into bankruptcy classification. From the cross-validated and sparsified model trained on oversampled data, we learned sales/fixed assets ratio has the greatest positive coefficient while sales/inventory ratio has the lowest. A high sales/fixed assets ratio thus increases the predicted probability of bankruptcy and is something lenders should watch carefully.

CART:

We next tried CART in the hope of gaining insights from tree structures. Here, we fit a total of 6 CART models. As before, we considered 3 data scenarios, namely the original data, undersampled data, and oversampled data. For each of these 3 scenarios, we evaluated 2 potential CART models. The first was an initial model with no hyperparameter tuning, and the second was a model with cross-validated hyperparameters. As before, we found
the best hyperparameters via random gridsearch with 3 folds and 100 iterations. For each of the 3 data cases, we selected the best of the 2 CART classifiers and reported the results from that model.

The advantage of CART, of course, is interpretability, and as can be seen in figure 2, we determined that gross profit (in 3 years) / total assets is the most important variable that CART considers when making a bankruptcy classification.

Random Forest:

We train Random Forest with the 3 versions of our data, after optimizing hyperparameters. Here, performance tradeoffs were less severe, so we maximized recall while ensuring an accuracy above 70%.

Results are significantly better for training and testing AUC, which are almost the same for the 3 versions of the data and always around 0.80 (figure 1). However, undersampling and oversampling allow for better recall with the same accuracy requirement. This comes at the cost of interpretability. We did attempt to recover feature significance using shapley values, but this proved prohibitively memory-intensive.

XGBoost:

XGBoost is less interpretable, with many hyperparameters. Because it is easy to overfit, we optimize these hyperparameters using grid search and cross-validation. We trained it for the 3 variants of our datasets. This model boasts the strongest results, with test AUC above 0.90, and even 0.95 with oversampled data.

Optimal Classification Trees:

Optimal Classification Trees are developed by IAI (InterpretableAI) and aim to find the best splits globally, as opposed to the greedy approach adopted by CART and other ensemble methods when they build trees. We refer to Professor Berstimas's Machine Learning textbook[2] for the Mixed-Integer Optimization formulation and the local search procedures used to build the tree.

OCT shines in interpretability, as we have a single tree for which we can interpret the splits, while ensuring better performance than CART. Here, however, OCT doesn't reach the performance of XGBoost and Neural Networks, although they outperform logistic regression and CART. This may indicate decision trees with splits on one variable cannot reach a satisfactory performance, even if we optimize them.

OCT-H, the variant of OCT that allows hyperplane splits, is likely to perform as well as XGBoost and Neural Networks. In fact, neural networks and OCT-H are in theory equivalent: we can transform a trained NN into a tree with hyperplane splits (see Bertsimas & Dunn book). However, training proved prohibitively time-intensive and we couldn't run this algorithm.

Neural Networks:

To further explore our predictive power, we decided to utilize Feed Forward Neural Networks. Neural networks are state of the art deep learning models that mimic the human brain in terms of decision making by utilizing hidden layers of perceptrons. To tune our neural network we experiment with various numbers of layers, number of nodes per layer, activation functions in the hidden and output layer, as well as the learning rate, batch size, and number of epochs.

After extensive parameter testing, we built a neural network with three hidden layers with the ReLu activation function. In terms of class weights, we account for class imbalance by making the weights for each label inversely proportional to the number of entries in each class and in doing so, ensure that equal weightages are assigned to both labels. This makes sure that our model is not skewed towards the majority class. Interpretability is again sacrificed, as Shapley values for feature significance again require too much RAM.

Comparative Analysis

The overall classification quality and recall of our 6 approaches using both undersampling and oversampling can be seen in comparison below.

AUC	Logistic Regression	OCT	CART	Neural Network	Random Forest	XGBoost
Original Data	0.53	0.64	0.63	0.75	0.80	0.94
Undersampled	0.60	0.64	0.70	0.73	0.79	0.90
Oversampled	0.58	0.63	0.73	0.75	0.80	0.95

Figure 1: AUC comparison of all approaches on original, undersampled, and oversampled data

Recall	Logistic Regression	OCT	CART	Neural Network	Random Forest	XGBoost
Original Data	0.61	0.67	0.30	0.68	0.58	0.65
Undersampled	0.65	0.68	0.66	0.73	0.71	0.67
Oversampled	0.58	0.70	0.67	0.64	0.72	0.72

Figure 2: Recall comparison of all approaches on original, undersampled, and oversampled data

We prioritized recall in our modeling process because it is worse to lend to a company that goes bankrupt than to not lend money to a company that performs fine. False negatives, in other words, are more costly than false positives, meaning recall was paramount. Here the price of interpretability is visible as neural nets, random forests, and XGBoost generally outperformed OCT and CART in recall.

Key Findings

We see a wide range in performance across different classification methods with CART and OCT underperforming compared to XGBoost which performed the best in our sample. Although XGBoost does deliver excellent results across our testing set, it remains to be seen whether these models to generalize to other bankruptcy data across economies is an interesting area of research. Another factor that we have not included in the analysis is exogenously introduced vulnerability that is common across economic downturns. Periods of recessions empirically correlate with bankruptcies but we lack the data on macroeconomic indicators to model any such relationship.

In terms of feature importance, we assess it across all of the predictive models with the exception of neural networks. We identify common financial metrics such as quick ratio *(liquid current assets/current liabilities)* as a metric chosen by random forests, XGBoost and OCT's. CART and logistic regression also focus on *sales and profitability ratios* which is interesting to see. The *focus on liquidity* though is common across our models which makes intuitive sense given that a liquidity crunch (resulting in no money for business operating activities) is the predecessor for declaring bankruptcy.

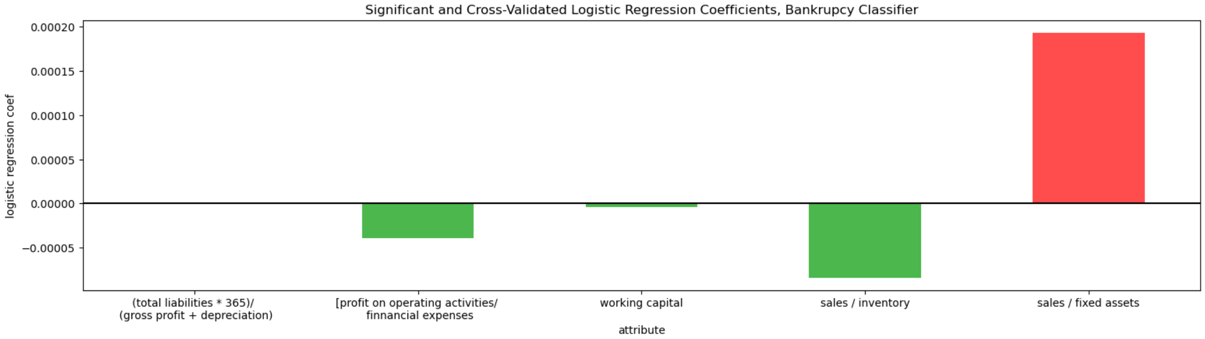
We also observe a tradeoff across more sensitive metrics such as recall and precision which may be of great interest to banks in the loan issuance process. Naturally, as a bank, we seek to minimize false negatives (defaults our model fails to predict). This comes at a cost in precision, whereby we increase the number of false positives which means we reject a large number of companies, erroneously predicting they will go bankrupt. This raises an important financial cost for banks where less lending will lead to smaller revenues and hence any model needs to be assessed through an economic cost-benefit analysis.

In terms of implementing such models, we anticipate industry resistance to black box techniques predicting defaults. This is natural given the ethical challenges around access and fairness, whereby these models may simply perpetuate the structural barriers erected by the current financial system through which less mature/ infant companies may be given reduced access to capital.

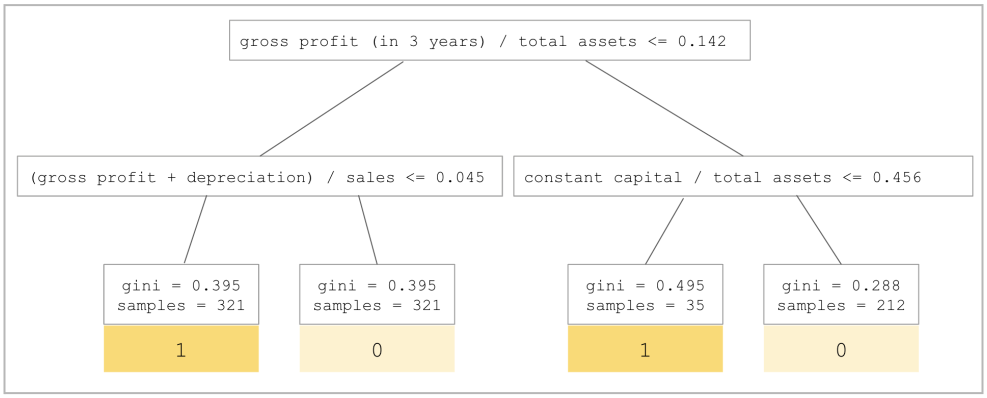
Thus to conclude, while the models shown above exhibit promise in terms of predictive power, it remains to see whether they can adapt to practical real world challenges that accompany implementation.

Appendix

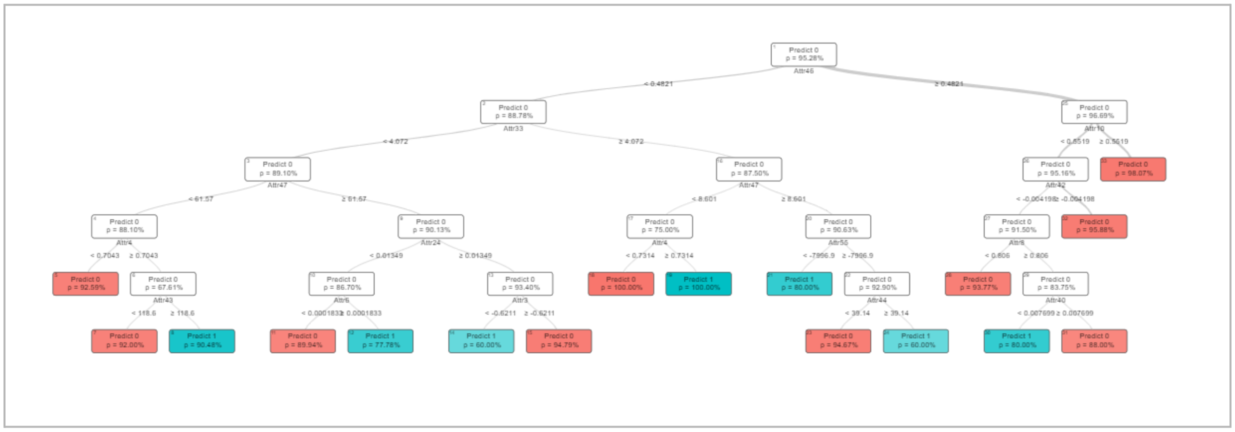
1. Logistic Regression Coefficients, (oversampled, cross-validated, sparsified)



2. CART Tree (undersampled, cross-validated)


gross profit (in 3 years) / total assets <= 0.142
(gross profit + depreciation) / sales <= 0.045
constant capital / total assets <= 0.456
gini = 0.395
samples = 321
1
gini = 0.395
samples = 321
0
gini = 0.495
samples = 35
1
gini = 0.288
samples = 212
0

3. Optimal Classification Tree



References

1. Dataset: https://archive.ics.uci.edu/ml/datasets/polish+companies+bankruptcy+data
2. *Machine Learning under a Modern Optimization Lens* (Bertsimas & Dunn)