CORPORATE BANKRUPTCY PREDICTION USING MACHINE LEARNING TECHNIQUES

Anna Elizabeth Thambi

Introduction

- Estimating the risk of corporate bankruptcies is of large importance to creditors and investors. For this reason bankruptcy prediction constitutes an important area of research. In recent years artificial intelligence and machine learning methods have achieved promising results in corporate bankruptcy prediction settings.
- Therefore, in this study, three machine learning algorithms, namely random forest, gradient boosting and an artificial neural network were used to predict corporate bankruptcies.
- It is shown that a very good predictive performance can be achieved with the machine learning models. The reason for the impressive predictive performance is analyzed and it is found that the missing values in the data set play an important role.
- It is observed that prediction models with surprisingly good performance could be achieved from only information about the missing values of the data and with the financial information excluded.

- The intention of the study is to illustrate and investigate how machine learning can be exploited in the field of economics.
- During this study corporate bankruptcy prediction using machine learning methods have been studied
- Bankruptcy prediction is another field of economics which is compatible with machine learning. To a bank that lends money to small companies the possibility to use the model to draw statistical significant conclusions about the underlying behaviour is of limited interest. On the other hand, having a model with good predictive power is of highest importance.
- The purpose of the bankruptcy prediction is to assess the financial condition of a company and its future perspectives within the context of long-term operation on the market. As with any other machine learning problems, prediction depends heavily on availability of data to train it accurately.
- For forecasting we rely on data with economic indicators curated by domain experts and apply machine learning methodologies learned in the course and evaluate their accuracy and performance when applied to real world problems of corporate accounting exclusively focused towards predicting corporate bankruptcy.

Dataset and Features – from Paper

1. Data Gathering :

To evaluate the performance and accuracy of various techniques we are relying on dataset [2] which is hosted on UCI Machine Learning Repository. The dataset is about bankruptcy prediction of Polish companies in manufacturing sector. The motivation in choosing this repository was because since 2004 Poland saw many manufacturing sector going bankrupt. The data was collected from Emerging Markets Information Service (EMIS), 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 research sample consists of bankrupt and still operating companies (imbalanced sample). Finally, data has the 64 financial indicators to be analyzed as features.

2. Data Preparation :

The data that is available is in classified in 5 cases depending on forecasting period. First year forecasting period data had economic indicator for bankruptcy status after 5 years for 7,027 instances (financial statements) where 271(3.8%) represents bankrupted companies, 6,756(96%) firms that did not bankrupt in the forecasting period. Second year indicated bankruptcy status after 4 years for 10173 instances where 400 (3.9%) represents bankrupted companies, 9,773 (96%) firms that did not bankrupt in the forecasting period. Similarly, for third year, 10,503 instances, 495(4.7%) represents bankrupted companies, 10,008(95%) firms that did not bankrupt in the forecasting period. For fourth year, 9,792 instances, 515(5%) represents bankrupted companies, 9,277(95%) firms that did not bankrupt in the forecasting period. Finally, for fifth year of the forecasting period and corresponding class label that indicates bankruptcy status after 1 year. The data contains 5,910 instances, 410(6%) represents bankrupted companies, 5500(94%) firms that did not bankrupt in the forecasting period. I am considering evaluating on third year of forecasting data to maximize the number of training examples, percentage of positive and negative instances.

3.Training and Test Data :

They choose and split the third-year forecasting data randomly into 80% for training and hold out 20% data for testing. Each test data instance represents 64 economic indicators and bankruptcy status after three years. In training data, we have total 8,402 instances(firms), out of which 384(4.5%) represents bankrupted companies, 8,018 (95.5%) firms that did not bankrupt in the forecasting period. In test data, we have 2,101 instances(firms), 111(5.2%) represents bankrupted companies, 1,990(94.7%) firms that did not bankrupt in the forecasting period.

Figure 1 Training Data Split

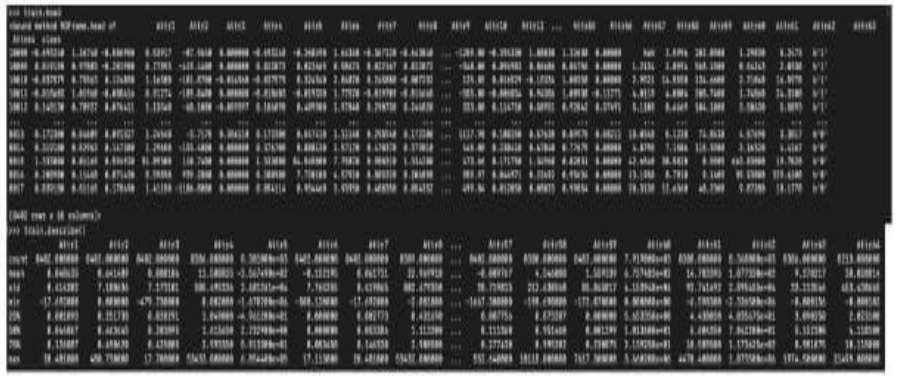


Figure 2 Training Data Details

4. Missing Data :

We saw that our dataset had lots of missing values across features. Missing data can introduce a substantial amount of bias in our learning during training. It can also impact the accuracy and efficiency of our models. Dropping such data is likewise harmful for the same reason as above. We considered using Mean, Median, Nearest Neighbors and Multivariate Imputation. We choose to use Mean Imputation for our problem as it reduces the correlations involving the feature variable been imputated. We achieved mean imputation using Imputer class from scikitlearn's library.

Model Implementation

The data set is typically dived into a training and a test set. The training set is used by the model to discover and unveil hidden patterns and relationships in the data. The test set is used to measure the strength and utility of the trained model.

A loss or cost function is typically defined and the training could then be seen as a optimization problem. The idea is that the cost function quantifies the error that the model makes in its predictions of the desired output.

To avoid overfitting a validation set is used in the training and test sets. Then the errors of predictions on both the training and validation set can be monitored during training.

a. Logistic Regression

Logistic Regression Logistic regression is a linear model of classification. We use L2 regularization to avoid overfitting the data and introducing high variance in our model. In this model, the probabilities describing the possible outcomes are modelled using a logistic function also called as sigmoid function.

$$p(y \mid x; \theta) = (h_\theta(x))^y \, (1 - h_\theta(x))^{1-y}$$

The cost function mentioned on the scikit-learn documentation with regularization is as below,

$$\min_{w,c} \frac{1}{2} w^T w + C \sum_{i=1}^{n} \log(\exp(-y_i(X_i^T w + c)) + 1).$$

b. Support Vector Machine (SVM)

A support vector machine is a supervised classification model that constructs a hyper-plane or set of hyper-planes in a high dimensional space, which can be used for classification tasks. Intuitively, a good separation is achieved by the hyper-plane that has the largest distance to the nearest training data points of any class (called functional margin), since in general the larger the margin the lower the generalization error of the classifier.

The advantages of support vector machines are that it is effective in high dimensional spaces, effective in cases where number of dimensions is greater than the number of samples. It also uses a subset of training points in the decision function (called support vectors), so it is also memory efficient due to kernels. Our cost function is as below,

$$= \sum_{i=1}^{n} \alpha_i y^{(i)} \langle x^{(i)}, x \rangle + b.$$

c. Neural Networks

We use Neural Network, which is part of supervised classification model using a multi-layer perceptron (MLP) algorithm that trains using backpropagation. We used L2 regularization term which helps in avoiding overfitting by penalizing weights with large magnitudes and stochastic Gradient descent as a way to achieve optimum value.

Adam Gradient descent though performs better, did not perform well for us. Our model had 5 neurons in 2 hidden layers. We simply used logistic function with Square Error loss function.

d. Naïve Bayes

Naive Bayes methods are a set of supervised learning algorithms based on applying Bayes' theorem with the "naive" assumption of conditional independence between every pair of features given the value of the class variable. Bayes' theorem states the following relationship, given class variable and dependent feature vector

$$P(y \mid x_1, \cdots, x_n) = \frac{P(y)P(x_1, \ldots x_n \mid y)}{P(x_1, \cdots, x_n)}$$

They used Naïve Bayes classifier based on Bernoulli Model, they experimented with different model, but they were not able to perform the prediction with high accuracy, For instance Gaussian based model was resulting in negative score no matter how they fine tune it.

e. Extreme Gradient Boosting

According to Wikipedia, Extreme Gradient Boosting is based on the principle of gradient boosting framework. Gradient boosting produces a prediction model in the form of an ensemble of weak prediction models typically decision tree. It builds the model in a stage-wise fashion like other boosting methods do, and it generalizes them by allowing optimization of an arbitrary differentiable loss function. XGBoost uses a more regularized model formalization to control overfitting for better performance. They used a depth of 3 with squared error objective function and gblinear booster (linear) with L2 regularization to avoid overfitting.

f. Light Gradient Boosting Machine

Light Gradient Boosting Machine is gradient boosting framework that uses tree-based learning algorithm and optimizes using histogram-based algorithms for performance efficiency. The difference is LGBM grows vertically as oppose to horizontally. In other words, it grows leaf vise while other grow level wise. Similar as above we use L2 regularization as it fits better due to binary problem nature.

Results – Sourse from Paper

All the methods were L2 regularized and had a learning rate of 0.001. They found that any other learning rate was causing delay in converging. Most of the parameters we will be using are explained later.

1. Cross Validation :

We perform K Fold cross validation for evaluation the performance of our models where K=5. Our Observations are graphically displayed in Figure 3.a and Figure 3.b.

2. Metrics :

Primary metrics used to evaluate the performance of models are Accuracy Score, Log Loss, Fit Times score and confusion matrix over both test and training data set. Accuracy Score and Log Loss are defined as below,

$$\texttt{accuracy}(y, \hat{y}) = \frac{1}{n_{\text{samples}}} \sum_{i=0}^{n_{\text{samples}}-1} 1(\hat{y}_i = y_i) \qquad L_{\log}(y, p) = -\log \Pr(y|p) = -(y \log(p) + (1-y) \log(1-p))$$

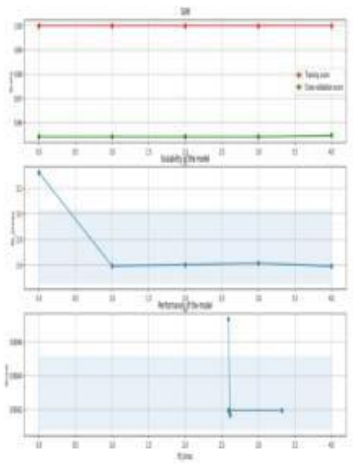
For instance, Confusion matrix for SVM was [[1990 0] [111 0]]

3. Data Scaling :

We choose not to standardize the features because for our observation of data we deduced that our data was not standard normally distributed. Forcing to remove mean and scaling to unit variance was resulting in performance drop.

Based on our observation SVM performed better than any other model in terms of maximum accuracy and minimum loss. SVM to be performing better for this scale of data and features was expected as we learned in class since it uses vectors aka subset of data to operate. Same can be said for performance for Logistic Regression and Neural Network. It's understandable that Naïve Bayes did not perform so well because of model assumptions.

The unexpected observation was for Gradient Boosting, which needs further inspection to elevate the performance. Future work should entail starting with replicating recent successes in academia research with synthetic feature generation to such data problem



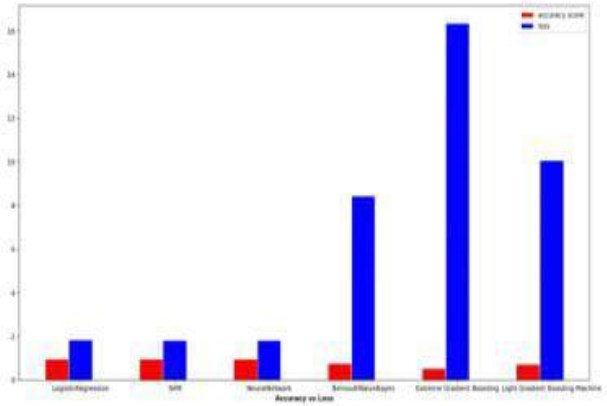


Figure 3.a: Learning Curve Metrics like Scalability, Efficiency and Correctness for SVM
Figure 3.b: Comparison of Model Performance under evaluation.

Classifier	Test Accuracy	Test Log Loss	Train Accuracy	Train Time
LogisticRegression	0.946692	1.841192	0.952392	1.01
SVM	0.947168	1.824752	1.000000	2.7
NeuralNetwork	0.947168	1.824752	0.954297	0.8
NaiveBayes	0.755831	8.433488	0.765413	0.006
ExtremeGradientBoosting	0.526416	16.357391	1.000000	1.8
LightGradientBoostingMachine	0.708710	10.061028	1.000000	0.7

Table 1 Comparison of Performance of models under evaluation

Conclusion & Future Work

The project evaluated the approaches for the problem of predicting the bankruptcy basing on the financial factors. They considered the financial condition of Polish companies from 2007 to 2012 for (still operating).

To solve the stated classification problem, they applied various models. The results gained by the SVM, Logistic and Neural Network were significantly better than the results gained by Boosting.

Furthermore,we should explore other methods like synthetic feature generation for better performance and try to propose a novel solution.