Company Bankruptcy Prediction

Kushal Chaudhari, Bahram Khodabandehlouee, and Jacob Reichlin

Keywords: Bankruptcy prediction, Machine learning, Deep learning, Neural Network, Decision Tree, Naive Bayes

I. Abstract

Bankruptcy prediction has been an important issue in finance and management science, which attracts the attention of entrepreneurs, researchers and even governments for years. With the great development of modern information technology, it has evolved into using machine learning or deep learning algorithms to do the prediction, from the initial analysis of financial statements. In this paper, we will implement the machine learning or deep learning models on Company Bankruptcy Prediction dataset, including the classical machine learning Naive Bayes and Decision tree with ID3 algorithm models, as well as a Neural Networks (NN) model using deep learning techniques.

II. Background/Related Work

The prediction of bankruptcy in companies is a problem that has concerned entrepreneurs, researchers and even governments for years, since detecting early signs that a company is going to enter bankruptcy involuntarily and being able to save it from that process, can help reduce the economic losses of bankruptcy[1]. In other words, bankruptcy prediction is the problem of detecting financial distress in businesses which will lead to eventual bankruptcy. Bankruptcy prediction has been studied since at least 1930s. The early models of bankruptcy prediction employed univariate statistical models over financial ratios. The univariate models were followed by multi-variate statistical models such as the famous Altman Z-score model. The recent advances in the field of Machine learning have led to the adoption of Machine learning algorithms for bankruptcy prediction. Machine Learning methods are increasingly being used for bankruptcy prediction using financial ratios. A study by Barboza, Kimura and Altman found that Machine Learning models can outperform classical statistical models like multiple discriminant analysis (MDA) by a significant margin in bankruptcy prediction[2].

III. Theory

A. Naive Bayes Model

In this model, to calculate the probability of predicting each class for a given observation we use:

$$P(y|\mathbf{x}) = \frac{P(y)P(\mathbf{x}|y)}{P(\mathbf{x})} \approx \frac{P(y)\Pi_{j=1}^{D} P(x_j|y)}{P(\mathbf{x})}$$

The Probability Density Function is calculated by:

$$P(x|\mu,\sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

Where σ is the standard deviation and μ is the mean.

B. Decision Tree Model

Our second model uses the following greedy algorithm:

```
function DTL(examples, attributes, default) returns a decision tree
    if examples is empty then return default
    else if all examples have the same classification then return the classification
    else if attributes is empty then return MODE(examples)
    else
        best ← CHOOSE-ATTRIBUTE(attributes, examples)
        tree ← a new decision tree with root test best
        for each value v_i of best do
            examples_i ← {elements of examples with best = v_i}
            subtree ← DTL(examples_i, attributes − best, MODE(examples))
            add a branch to tree with label v_i and subtree subtree
        return tree
```

To determine the best attribute to use in each iteration, we calculate the entropy as follows:

$$H\left(P\left(v_1\right), ..., P\left(v_k\right)\right) = \sum_{i=1}^{K} \left(-P\left(v_i\right) log_K P\left(v_i\right)\right)$$

C. Neural Network Model

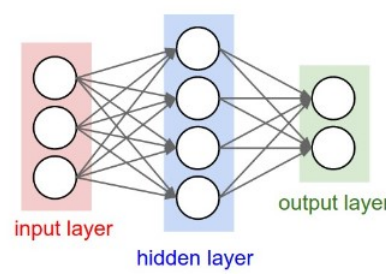
In our Neural Network Model we applied a Softmax activation function:

$$g\left(z\right) = \frac{e^z}{\sum_i e^{z_i}}$$

Additionally, we use a cross entropy objective function:

$$J = -\sum_{k=1}^{K} y_k ln(\hat{y}_k + \epsilon)$$

We implement two architectures with one hidden layer and one architecture without any hidden layer.



In the Fully Connected Layer (FCL), in addition to Softmax, we also use Tanh and Sigmoid activation functions.

Tanh Activation Function:

$$f\left(x\right) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

Sigmoid Activation Function:

$$f\left(x\right) = \frac{1}{1 + e^{-x}}$$

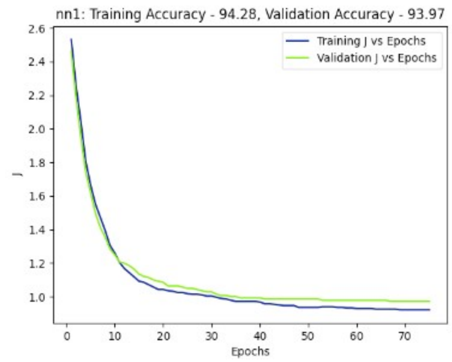
IV. Approach and Results

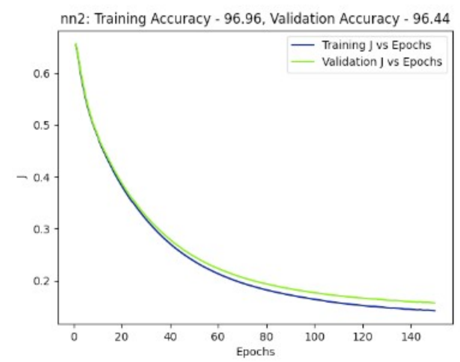
The data was sourced from a Kaggle dataset. The dataset in .csv format consists of 6819 observations across 95 features. The data is already standardized. The data was collected from Taiwan Economic Journal for the years 1999 to 2009. Company bankruptcy was defined based on the business regulations of the Taiwan Stock Exchange. It also has a column having the information if the company went bankrupt or not. If the company went bankrupt it was labeled as 1 if not it was labeled as 0. In all, the features-set matrix is a 2d array: X(6819,95), and the label-set is a vector: Y(6819,1). The problem essentially is a classification problem to classify the outcome(bankrupt or not) based on given financial ratios and numbers for a given company. The three chosen methods for evaluation are - Naive Bayes classification, Decision Tree ID3 algorithm for classification and a neural network approach for classification.

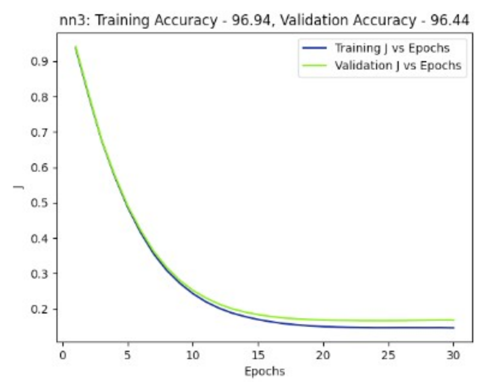
The neural network approach is accomplished by using modular classes known as ‘Layers’. In this, each layer that can be created serves an independent function. The modularity ensures that each layer has its own methods for forward propagation, updating weights and biases in case of fully connected layers, computation of gradient and backward propagation if needed by the layer. First layer chosen was a fully connected layer with size of (95,2) that takes in input data and uses a weighted sum to produce output for every node for the next layer. The Softmax layer was chosen as an activation function to facilitate a multi-class classification. And a cross-entropy layer is used to compute losses in the predictions. The dataset has been split into three parts, two for training and one for validation. The X and Y representations were also randomly shuffled before every forward and backward pass. The label vector Y is converted into one-hot encoding. Using different learning rates and epochs, 3 different architectures are run. Each architecture is a step of forward propagation followed by loss computation followed by a backward propagation. We keep track of losses per epoch to plot a Loss vs Epoch graph to confirm the evidence of learning for the models or any overfitting. Learning is evident from the fact that losses decrease as the model learns. Finally after all iterations, accuracy is computed using predictions on training as well as test set after all interactions and reported for the 3 neural network architectures in the table below.

Table 1
FCL: FullyConnected Layer, Sig: Sigmoid Activation function, CE: Cross Entropy objective function, SM: Softmax Activation function, Tanh: Tangent Hyperbolic Layer

Model	ARCHITECTURE	η	epochs	Training accuracy	Test accuracy
nn1	FCL(95,2) - SM - CE	0.005	75	94.28%	93.9%
nn2	FCL(95,10) - TANH - FCL(10,2) - SM - CE	0.005	150	96.96%	96.4%
nn3	FCL(95,20) - SIG - FCL(20,2) - SM - CE	0.01	30	96.94%	96.4%







Using the ID3 decision tree algorithm is one of the other methods employed to make a model. In a nutshell, the model tries to check how a particular input feature classifies the target variable in this case Y independently. To help with this we compute the information gain that helps us identify how well the data can be split wrt. each variable and helps us decide what should be the root node for the decision tree. The algorithm then performs a greedy search—goes over all input features and their unique values, calculates information gain for every combination, and saves the best split feature and threshold for every node. In this way, the tree is built recursively until a termination criterion is reached. Then predictions can be done by traversing through the nodes of the decision tree based on feature values. Lastly, we measure the accuracy of the prediction compared to the given and known output. Initially we implemented this with all the feature-set X(6819,95) and Y(6819,1). The runtime was more than 3-4 hrs due to which we decided to only use a set of 25 feature labels to train and test the model. The accuracy for this model was 96.7%.

The final classification approach used was the Naive Bayes

Classifier. This method assumes independence across all 95 features. The data is randomly shuffled and is split into training and test sets in a ratio 2:1. The labels are then used to separate data by class in the form of a dictionary. This is then used to create normal models for each of the 95 feature vectors. The normal models compute the mean, standard deviation and total observations in the given vector. This is then used to compute class prior probabilities for every feature. Then validation data is used to make predictions based on computed class prior probability summaries. The following table summarizes the model evaluation metrics for this method.

Table 2

Naive Bayes Model Evaluation	
Accuracy	92.6969%
Precision	0.2258
Recall	0.4321
F-Measure	0.2966

V. Conclusion and Future Scope

Of the three models we employ in our study, the model resulting in the highest accuracy measure was the Decision Tree model with an accuracy of roughly 96.7%. While being the most accurate of our models, the issue of computational efficiency and runtime hinders the practical application for the purposes of our study. The Naive Bayes model performed the weakest relative to our other results. For this reason along with the feature independence assumption, we do not believe this model to be our strongest. There is likely some level of correlation among features that contribute to whether a company will go bankrupt. The neural network models we use have a better potential in both accurately predicting bankruptcy, as well as further exploration into additional architectures. The use of One-hot encoding with a logistic regression model is a natural next step.

Even though the data is from businesses based in Taiwan, our models can be extrapolated in the future for different geographic locations since the financial conditions leading to bankruptcy largely would be the same across different geographies. Our model could be utilized by banks or investment firms to gauge bankruptcy risk in the face of an acquisition, or simply be used as an additional measure in an investment analysis.

References

[1] Yi Qu, Pei Quan, Minglong Lei, Yong Shi, " Review of bankruptcy prediction using machine learning and deep learning techniques," *7th International Conference on Information Technology and Quantitative Management (ITQM 2019)*

[2] Aditya Narvekar,* and Debashis Guha, " Bankruptcy prediction using machine learning and an application to the case of the COVID-19 recession" *Data Science in Finance and Economics - DSFE, 1(2): 180–195. DOI: 10.3934/DSFE.2021010*