*Project Report On*

**Prediction of Financial distress**

*Submitted by*

**V.Shireesh (16IT147)**

**A.Naveen kumar (16IT105)**

**B.Nishanth (16IT110)**

**I.Ravindra (16IT116)**

**VI SEM B.Tech (IT)**

*Under the guidance of*

**Nagamma Patil**

**Dept of IT, NITK Surathkal**

*in partial fulfillment for the award of the degree*

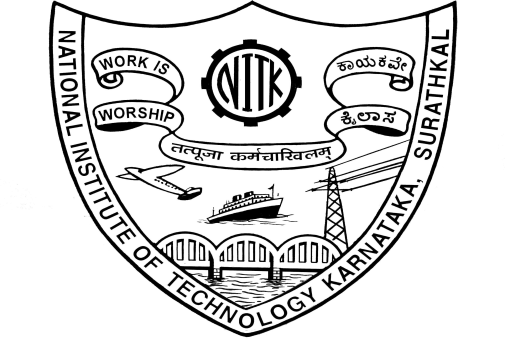
*of*

**Bachelor of Technology**

in

**Information Technology**

at



**Department of Information Technology**

**National Institute of Technology Karnataka, Surathkal**.

***April 2019***

**Acknowledgement:**

We have made this report file on the topic “Prediction of Financial Distress”. We have tried our best to elucidate all the relevant detail to the topic to be included in the report, while in the beginning we have tried to give a general view about this topic. Our efforts have ended on a successful note. We express my sincere gratitude to Nagamma patil ma’am, for giving me this opportunity to develop a project on Prediction of Financial Distress. Without this, it wouldn’t have been possible to develop a project in this domain .

**Declaration:**

*Name of the Student Register No. Signature with Date*

1. V.Shireesh 16IT147

2. A.Naveenkumar 16IT105

3. B.Nishanth 16IT110

4. I.Ravindra 16IT116

**Name of Project Guide:**

Signature of the Project Guide:

Place:

Date:

**Table of Contents:**

**Abstract :**

The deterioration in profitability of listed companies not only threatens the interests of the enterprise and internal staff, but also makes investors face significant financial loss. It is important to establish an effective early warning system for prediction of financial crisis for better corporate governance.

This paper studies the phenomenon of financial distress for some companies that received the label ‘special treatment’ over five years. The aim of ST is to warn the managers hand investors and also to use external mechanism to force the firm to enhance its operating performance .

Financial distress of a company usually refers to the situation that operating cash flow of a company cannot supersede the negative net assets of the firm. We use data mining techniques to build financial distress warning models based on 65 financial indicators by comparing these firms to a control group of firms.

We observe that the performance of neural networks is more accurate than other classifiers, such as decision trees and support vector machines, as well as an ensemble of multiple classifiers combined using majority voting.

This paper provides a suitable method for prediction of financial distress for listed companies.

**Introduction:**

Prediction of financial distress has been a topic of interest over the decades because of its great importance to listed companies, interested stakeholders and even the economy of a country.If the prediction of financial distress is reliable, managers of firms can initiate remedial measures to avoid deterioration before the crisis, and investors can grasp the profitability situation of the listed companies and adjust their investment strategies to reduce anticipated investment related losses.

However, the rapid development of the capital market and the integration of the global economy have increased the number of companies that suffer from financial distress over the years. In October 2007, the stock market in China crashed and wiped out more than two-thirds of its market value. Therefore, discovery of a suitable model for predicting the financial distress of listed Chinese companies is likely to be of great significance to global investors.

We will perform prediction based on Neural networks , Decision tree Calssifier , Support Vector Machines and Majority Voting. We compare those results and are described in results section.

**3.Proposed Methodology**

In the previous sections, we formally introduced the problem statement of bankruptcy prediction. In this section, we explain our step-by-step solution of how we achieved benchmark results for bankruptcy prediction .

Firstly, we introduce the Polish bankruptcy dataset and explain the details of the dataset like features, instances, data organization, etc. Next, we delve into data preprocessing steps, where we state the problems present with the data like missing data and data imbalance, and explain how we dealt with them. Next, we introduce the classification models we have considered and explain how we train our data using these models.

**3.1 Data:**

The dataset we have considered for addressing the bankruptcy prediction problem is the Polish bankruptcy data, a huge repository of freely accessible datasets for research and learning purposes intended for the Machine Learning/Data Science community. The bankrupt companies were analyzed in the period 2000-2012, while the still operating companies were evaluated from 2007 to 2013. The dataset is very apt for our research about bankruptcy prediction because it has highly useful econometric indicators as attributes (features) and comes with a huge number of samples of Polish companies that were analyzed in 5 different timeframes.

Based on the collected data five classification cases were distinguished, that depends on the forecasting period.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Data | Total Instances | Bankrupt instances | Non-Bankrupt  instances |
|  | 1st year | 7027 | 271 | 6756 |
| No.of | 2nd year | 10173 | 400 | 9773 |
| Instances | 3rd year | 10503 | 495 | 10008 |
|  | 4th year | 9792 | 515 | 9227 |
|  | 5th year | 5910 | 410 | 5500 |

Table 1: Summary of Bankruptcy dataset.

\*nth year : The data contains financial rates from nth year of the forecasting period and corresponding class label that indicates bankruptcy status after (6-n) years (0<n<=5).

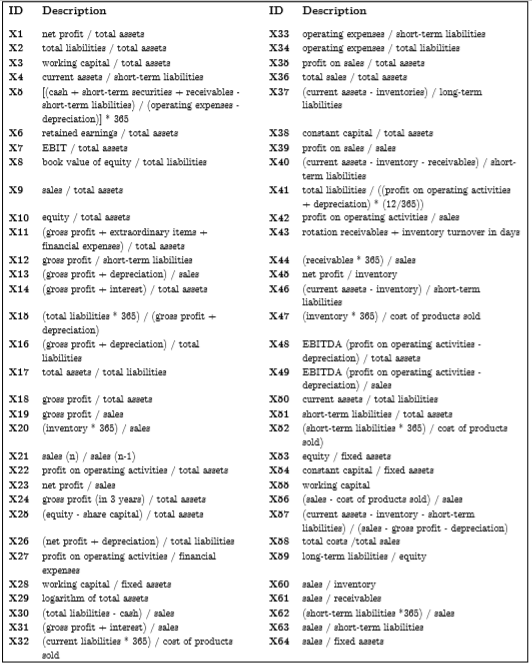


Table 2 :Summary of features in the bankruptcy data

Table 1 shows the total number of features and instances in the dataset, and the number of samples in each class (bankrupt or not-bankrupt) of all the 5 datasets. The features are explained in Table 2 above. As shown in the table, there are 64 features labelled X1 through X64, and each feature is a synthetic feature.

**3.2 Dataset Quality Assessment:**

Now we move on to assessing the quality of the dataset. As we have mentioned earlier, the dataset suffers from missing values and data imbalance.

**3.2.1 Missing Data:**

First, we look at some statistics of missing values.

|  |  |  |  |
| --- | --- | --- | --- |
| Dataset | Total instances | Instances with missing values | Instances that would remain if all rows with missing values were dropped |
| Year 1 | 7027 | 3833 | 3194 |
| Year 2 | 10173 | 6085 | 4088 |
| Year 3 | 10503 | 5618 | 4885 |
| Year 4 | 9792 | 5023 | 4769 |
| Year 5 | 5910 | 2879 | 3031 |

Table 3 : Assessing the missing values for all the dataset

**3.2.2 Dealing with Missing Data:**

Missing data causes 3 problems:

1. Missing data can introduce a substantial amount of bias.

2. Makes the handling and analysis of the data more difficult.

3. Create reductions in efficiency.

Dropping all the rows with missing values or Listwise deletion, introduces bias and affects representativeness of the results. The only viable alternative to Listwise deletion of missing data is Imputation. Imputation is the process of replacing missing data with substituted values and it preserves all the cases by replacing missing data with an estimated value, based on other available information. In our project we explored Mean imputation technique.

**3.2.3 Mean Imputation**

Mean imputation technique is the process of replacing any missing value in the data with the mean of that variable in context. In our dataset, we replaced a missing value of a feature, with the mean of the other non-missing values of that feature. Mean imputation attenuates any correlations involving the variable(s) that are imputed. This is because, in cases with imputation, there is guaranteed to be no relationship between the imputed variable and any other measured variables. Thus, mean imputation has some attractive properties for univariate analysis but becomes problematic for multivariate analysis. Hence we opted Mean Imputation as a baseline method. We achieved mean imputation using scikit-learn’s Imputer class.

**3.3. Dealing with Data Imbalance:**

Moving on to the other shortcoming of the bankruptcy dataset, we now explain how we dealt with the Data Imbalance. Data Imbalance can be treated with Oversampling and/or Undersampling. In data analysis, Oversampling and Undersampling are opposite and roughly equivalent techniques of dealing with Data Imbalance, where they adjust the class distribution of a data set . Oversampling is increasing the class distribution of the minority class label whereas Undersampling is decreasing the class distribution of the majority class label.

In our project, we explored Synthetic Minority Oversampling Technique or SMOTE.

**3.3.1 Synthetic Minority Oversampling Technique (SMOTE) :**

Synthetic Minority Oversampling Technique (SMOTE) is a widely used oversampling technique. To illustrate how this technique works consider some training data which has s samples, and f features in the feature space of the data. For simplicity, assume the features are continuous. As an example, let us consider a dataset of birds for clarity. The feature space for the minority class for which we want to oversample could be beak length, wingspan, and weight. To oversample, take a sample from the dataset, and consider its k nearest neighbors in the feature space. To create a synthetic data point, take the vector between one of those k neighbors, and the current data point. Multiply this vector by a random number x which lies between 0, and 1. Adding this to the current data point will create the new synthetic data point. SMOTE was implemented from the imbalancedlearn library.

**3.4 Data Modeling:**

In this section, we will look at the various classification models that we have considered for training on the bankruptcy datasets to achieve the task of coming up with a predictive model that would predict the bankruptcy status of a given (unseen) company with an appreciable accuracy.

We have considered the following 6 models:

1. Gaussian Naïve Bayes

2. Decision Tree

3.Support Vector machines

4. Random Forests

5.Multi layer perceptron

6.Extreme Gradient boosting

7.Majority Voting

**K-Fold Cross Validation:**

Since the Polish bankruptcy dataset does not have a separate ‘unlabeled’ test dataset, it is obvious that we need to split the training data to obtain a validation dataset (for each year’s data). If the split was done in a simple way, we end up with just one validation dataset and the inherent difference in the class label distributions for training and validation datasets would lead to poor performance of the model on the training and hence on validation sets.

Alternatively, in K-Fold Cross Validation, the training dataset is split into K bins. In each iteration (total K iterations), one bin is retained as a validation dataset and the other bins of data are used for training the model. The performance are noted for each validation set. After all the iterations, each of the bins will have served as validation dataset at least once (depending on K). The metrics are averaged over all the K iterations and the final metrics are output.

**3.4.1 Gaussian Naïve Bayes Classifier** :

Naive Bayes classifier is one of the supervised learning algorithms which is based on applying Bayes’ theorem with the “naive” assumption of independence between every pair of features. Given a class variable 𝑦 and a dependent feature vector 𝑥1 through 𝑥𝑛, Bayes’ theorem states the following relationship:



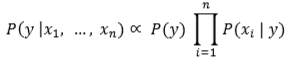
Using the naive independence assumption that:



for all 𝑖, this relationship is simplified to:



Since 𝑃(𝑥1, …, 𝑥𝑛) is constant given the input, we can use the following classification rule:



Gaussian Naïve Bayes implements the Gaussian Naive Bayes algorithm for classification. The likelihood of the features is assumed to be Gaussian:



**3.4.2 Decision Trees Classifier:**

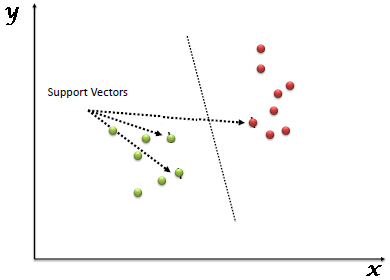
Decision Trees (DTs) are a non-parametric supervised learning method used for classification and regression. For our classification task, we create a model that predicts the value of a target variable (y = will a firm go bankrupt?) by learning simple decision rules inferred from the data features (𝑥1,𝑥2…. 𝑥64 - all the financial distress variables of a firm). While building decision tree, the data comes in records in the form:

(x, Y) = (𝑥1, 𝑥2, ………, 𝑥64, Y)

Our model considers all features and gives equal weights to each of them while looking for best split during construction of a decision tree. We have considered ‘*Gini*’ index as a measure the quality of a split.

**3.4.3 Support Vector Machines:**

More formally, a support-vector machine constructs a [hyperplane](https://en.wikipedia.org/wiki/Hyperplane) or set of hyperplanes in a [high-](https://en.wikipedia.org/wiki/High-dimensional_space) or infinite-dimensional space, which can be used for [classification](https://en.wikipedia.org/wiki/Statistical_classification), [regression](https://en.wikipedia.org/wiki/Regression_analysis), or other tasks like outliers detection. Intuitively, a good separation is achieved by the hyperplane that has the largest distance to the nearest training-data point of any class (so-called functional margin), since in general the larger the margin, the lower the [generalization error](https://en.wikipedia.org/wiki/Generalization_error) of the classifier.



**3.4.4 Random Forests Classifier:**

A random forest is a meta estimator that fits a number of decision tree classifiers on various sub-samples of the dataset and use averaging to improve the predictive accuracy and control over-fitting. In random forests, each tree in the ensemble is built from a sample drawn with replacement from the training set. Also, when splitting a node during the construction of the tree, the split that is chosen is no longer the best split among all features. Instead, the split that is picked is the best split among a random subset of the features.

As a result of this randomness, the bias of the forest usually slightly increases but, due to averaging, its variance also decreases, usually more than compensating for the increase in bias, hence yielding an overall better model. In our model, the number of estimators used are 5 and we have considered ‘Entropy’ as a measure of the quality of a split .

**3.4.5 Multi Layer Perceptron :**

A multilayer perceptron (MLP) is a class of [feedforward](https://en.wikipedia.org/wiki/Feedforward_neural_network) [artificial neural network](https://en.wikipedia.org/wiki/Artificial_neural_network). A MLP consists of at least three layers of nodes: an input layer, a hidden layer and an output layer. Except for the input nodes, each node is a neuron that uses a nonlinear [activation function](https://en.wikipedia.org/wiki/Activation_function). MLP utilizes a [supervised learning](https://en.wikipedia.org/wiki/Supervised_learning) technique called [backpropagation](https://en.wikipedia.org/wiki/Backpropagation) for training.

