**A report on Bankruptcy Prediction by Machine Learning techniques**

**Table of Contents**

[**1)** **Introduction** 4](#_Toc107121836)

[**2)** **Understanding data** 4](#_Toc107121837)

[**3)** **Data Pre-Processing** 5](#_Toc107121838)

[**3.1.** **Remove unwanted variables** 5](#_Toc107121839)

[**3.2.** **Missing values and treatment** 5](#_Toc107121840)

[**3.3.** **Features Selection** 5](#_Toc107121841)

[**4)** **Data Exploration** 6](#_Toc107121842)

[**5)** **Build models** 6](#_Toc107121843)

[**6)** **Inferences from the best model** 6](#_Toc107121844)

[**7)** **Appendix** 7](#_Toc107121845)

[**a)** **Data Dictionary** 7](#_Toc107121846)

[**b)** **List of Machine Learning techniques used** 8](#_Toc107121847)

# **Introduction**

Bankruptcy prediction is the art of predicting bankruptcy and various measures of financial distress of public firms for the benefit of creditors and investors to evaluate the likelihood that a firm may go bankrupt.

The history of bankruptcy prediction includes application of numerous statistical tools which gradually became available, and involves deepening appreciation of various pitfalls in early analyses.

Bankruptcy prediction has been a subject of formal analysis since at least 1932, when FitzPatrick published a study of 20 pairs of firms, one failed and one surviving, matched by date, size and industry, in The Certified Public Accountant.

Various techniques such as Survival methods, Multiple Discriminant Analysis, Z Score, Neural Network models and other network models and other sophisticated models have been tested on bankruptcy prediction.

In this paper, we have used the following Machine Learning Techniques:

1. Logistic Regression
2. Decision Tree (CART)
3. Random Forest
4. Support Vector Machine
5. K-Nearest Neighbour
6. Linear Discriminant Analysis
7. XGBoost

You may refer to the appendix for description of each technique.

Ref:

<https://en-academic.com/dic.nsf/enwiki/11068795>

<https://en.wikipedia.org/wiki/FitzPatrick_1932>

# **Understanding data**

The given data file contains 9000 observations and 34 columns. Target variable is “Dummy Coded: Healthy= 0; NPA= 1”. For details, you may refer to the Data Dictionary in the Appendix section.

# **Data Pre-Processing**

## **Remove unwanted variables**

We observe that the following columns are not required.

1. **Row** representing the company code
2. **Company**\_**name** -- We are more interested in knowing the characteristics of the company going bankrupt rather than the company name
3. **Year** - We already have the column Year Encoded representing the year. Hence Year is removed.

## **Missing values and treatment**

1. Missing values for each column
2. Detecting and handling missing values in the correct way is important, as they can impact the results of the analysis. We have observed that 29 columns have missing values ranging from 1% to 45.9%. We could not remove the missing values as it will result in information loss.
3. They cannot be imputed with general ways of using mean, mode, or median which ignores the inherent relationship among data and also it can pollute the data. We observe that on a few occasions, data is missing in a dataset and is related to the other features and hence they can be predicted using other feature values. Imputing by prediction of missing values is superior to other techniques since the inherent relationship among data is not ignored.
4. We are imputing missing numeric values using the IterativeImputer class in sklearn.  
     
   Ref: <https://www.numpyninja.com/post/mice-and-knn-missing-value-imputations-through-python>

## **Features Selection**

* Recursive Feature Elimination, or RFE for short, is a popular feature selection algorithm in a dataset that is more or more relevant in predicting the target variable. RFE applies a backward selection process to find the best combination of features. This is done as follows:

1. Builds a model based on all features and calculates the importance of each feature in the model.
2. It ranks the features and removes the feature(s) with the least importance iteratively based on model evaluation metrics such as accuracy ratio.  
     
   Ref. <https://towardsdatascience.com/effective-feature-selection-recursive-feature-elimination-using-r-148ff998e4f7>

* We have used Decision Tree (CART) , Random Forest and LDA techniques to identify the most important variables influencing the target variable.
* **Nineteen variables are chosen by Decision Tree, Random Forest and LDA models**. They are ['Asset\_coverage', 'Cash\_ratio', 'EBIT\_Sales', 'Fixed Asset Turnover Ratio', 'Interest\_coverage',  'Inventory\_turnover', 'Ln\_GVA', "Operating Cash Flow/Shareholder's Equity", 'Operating Cash Flow/Total Debt',  'Operating Cash Flow/Total Sales', 'Quick\_ratio', 'ROS(new)', 'Shareholderquity\_code',  "Total shareholders' funds", 'YOY EBIT Growth Rate', 'YOY Sales Growth Rate', 'Year Encoded', 'debt\_asset', 'debt\_equity']
* **Nine variables were selected by at least two of the above three models. They are**["Cash\_ratio", "Interest\_coverage", "Inventory\_turnover", "Operating Cash Flow/Total Debt", "Operating Cash Flow/Total Sales", "Shareholderquity\_code", "Total shareholders' funds", "YOY EBIT Growth Rate","debt\_equity"]

# **Data Exploration**

# **Build models**

We used Logistic Regression, Decision Tree (CART), Random Forest, Support Vector Machine (SVM), K-Nearest Neighbour (KNN), Linear Discriminant Analysis (LDA) and XGBoost techniques to build models.

The comparison chart for all the models for both training and test datasets is given below:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Model** | **Dataset** | **Precision** | **Recall** | **F1 ratio** | **AUROC** | **Accuracy Ratio** |
| Logistic Regression | Training | 0.25 | 0.28 | 0.27 | 0.57 | 0.77 |
|  | Test | 0.24 | 0.3 | 0.27 | 0.57 | 0.76 |
| **CART** | **Training** | **1** | **1** | **1** | **1** | **1** |
|  | **Test** | **0.99** | **1** | **1** | **1** | **1** |
| **RANDOM FOREST** | **Training** | **1** | **1** | **1** | **1** | **1** |
|  | **Test** | **0.99** | **1** | **1** | **1** | **1** |
| SVM | Training | 0 | 0 | 0 | 0.5 | 0.85 |
|  | Test | 0 | 0 | 0 | 0.5 | 0.85 |
| LDA | Training | 0.4 | 0.01 | 0.01 | 0.5 | 0.85 |
|  | Test | 0.29 | 0.01 | 0.01 | 0.5 | 0.85 |
| KNN | Training | 0.76 | 0.31 | 0.44 | 0.65 | 0.88 |
|  | Test | 0.52 | 0.24 | 0.33 | 0.6 | 0.86 |
| **XGBOOST** | **Training** | **1** | **1** | **1** | **1** | **1** |
|  | **Test** | **0.99** | **1** | **1** | **1** | **1** |
| **CONSENSUS OF 3 MODELS (CART, RF, XGBOOST)** | **Training** | **1** | **1** | **1** | **1** | **1** |
|  | **Test** | **0.99** | **1** | **1** | **1** | **1** |

We have chosen the Decision Tree (CART) model as the best model because of the accuracy in prediction, and its power of interpretation as the performance of other models such as Logistic Regression, SVM, KNN and LDA gave very poor performance (esp. Recall, which **measures the model's ability to detect positive samples** varying from 0 to 33%) in predicting the target variable and of no use to the stakeholders.

**Performance measures of the best model, Decision Tree (CART) model – Training dataset**

|  |  |
| --- | --- |
|  | **Class Precision Recall F1-score**  **Healthy 1.00 1.00 1.00**  **NPA 1.00 1.00 1.00**  **Accuracy Ratio: 1.00**  **Area Under RoC: 1.00**  **Number of Observations: 6300** |
|  |  |

**Performance measures of the best model, Decision Tree (CART) model – Test dataset**

|  |  |
| --- | --- |
|  | **Class Precision Recall F1-score**  **Healthy 1.00 1.00 1.00**  **NPA 0.99 1.00 1.00**  **Accuracy Ratio: 1.00**  **Area Under RoC: 1.00**  **Number of Observations: 2700** |
|  |  |

We observe that the performance of the CART model for both training and test dataset are comparable and hence there is no model overfit.

# **Inferences from the best model**

# **Appendix**

## **Data Dictionary**

|  |  |  |
| --- | --- | --- |
| **S. No** | **Variable** | **Variable Definition** |
| 1 | Row | Company code |
| 2 | Year | year |
| 3 | Company\_name | Company Name |
| 4 | Year Encoded | year; 0 means latest year |
| 5 | Dummy Coded :Healthy= 0; NPA= 1 | NPA = 1 and 0= Healthy, Target Variable |
| 6 | Asset\_turnover | Total income/ Total assets |
| 7 | Receivable\_turnover(new) | Net sales/Total assets |
| 8 | Inventory\_turnover | COGS/ Total inventories |
| 9 | Cash\_ratio | Cash and cash balance/ Total Current liabilities |
| 10 | Quick\_ratio | Cash and Cash Equivalents + Receivables + Marketable securities/ Total Current liabilities |
| 11 | Current\_ratio | Current asset/current liabilities |
| 12 | ROA(new) | Net income/Total assets |
| 13 | ROE(new) | Net income/Shareholder's equity |
| 14 | ROS(new) | Net income/Total sales |
| 15 | ROI(new) | Net income/Total investment |
| 16 | debt\_asset | Total debt/Total asset |
| 17 | debt\_equity | Total debt/ Total equity |
| 18 | debt\_income | Total debt / EBIT |
| 19 | Interest\_coverage | EBITDA/Interest |
| 20 | Asset\_coverage | Total asset - (CA-CL) / (Total debt) |
| 21 | EBIT\_Sales | EBIT/Total sales |
| 22 | Sales\_CE | Sales/Total capital employed |
| 23 | ROCE\_CE | (EBIT/Sales) \* (Sales/CE) |
| 24 | Changeinsales\_Industry | Sales (current year)- Sales (Previous year)/ Sales (current year) |
| 25 | Grossvaluedadded | Grossvaluedadded/Total grossvaluedadded |
| 26 | Ln\_GVA | Ln (Gross value added) |
| 27 | Operating Cash Flow/Total Sales | Operating cash flow/Total sales |
| 28 | Operating Cash Flow/Total Debt | Operating cash flow/Total debt |
| 29 | Operating Cash Flow/Shareholder's Equity | Operating cash flow /Total equity |
| 30 | Fixed Asset Turnover Ratio | Total income/ Fixed asset |
| 31 | YOY Sales Growth Rate | Y-O-Y Sales Growth rate |
| 32 | YOY EBIT Growth Rate | Y-O-Y EBIT Growth rate |
| 33 | Total shareholders' funds | Total shareholder's equity |
| 34 | Shareholderquity\_code | Dummy variable  0 = healthy firm +ve equity  1 = healthy firm -ve equity  2 = banktrupt firm +ve equity  3 = banktrupt firm -ve equity |
|  | | |

## **List of Machine Learning techniques used**

**1) Logistic Regression**

Logistic regression is a classification technique that is used to calculate (or predict) the probability of a binary (yes/no) event occurring.

**2) CART or Decision Tree:**

Classification And Regression Trees (CART) or Decision Tree algorithm is a classification algorithm for building a decision tree based on Gini's impurity index\* as splitting criterion.

*\*Gini index or Gini impurity index measures the degree or probability of a particular variable being wrongly classified when it is randomly chosen.*

**3) Random Forest**

Random forest is a Supervised Machine Learning Algorithm that is used widely in Classification and Regression problems. It builds decision trees on different samples and takes their majority vote for classification and average in case of regression.

**4) Support Vector Machine**

Support Vector Machine (SVM) is a supervised machine learning algorithm that is used for both classification and regression. The objective of SVM algorithm is to find a hyperplane in an N-dimensional space that distinctly classifies the data points. The dimension of the hyperplane depends upon the number of features.

**5) K-Nearest Neighbour**

K-Nearest Neighbour is one of the simplest Machine Learning algorithms based on Supervised Learning technique used for both classification and regression problems. K-NN algorithm assumes the similarity between the new case/data and available cases and put the new case into the category that is most similar to the available categories.

**6) Linear Discriminant Analysis**

This is a supervised classification technique. It is used for modelling differences in groups i.e. separating two or more classes by finding a linear combination of features that characterizes or separates two or more classes of objects or events. The resulting combination may be used as a linear classifier.

**7) XGBOOST**

XGBoost stands for “Extreme Gradient Boosting”. Gradient boosting is a supervised learning algorithm, which attempts to accurately predict a target variable by combining the estimates of a set of simpler, weaker models.

It implements Machine Learning algorithms under the Gradient Boosting framework. It provides a parallel tree boosting to solve many data science problems in a fast and accurate way.