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

**Contents**

[**1)** **Introduction** 3](#_Toc113259236)

[**2)** **Understanding data** 3](#_Toc113259237)

[**3)** **Data Pre-Processing** 4](#_Toc113259238)

[**3.1.** **Remove unwanted variables** 4](#_Toc113259239)

[**3.2.** **Missing values and treatment** 4](#_Toc113259240)

[**3.3.** **Features Selection** 4](#_Toc113259241)

[**4)** **Build models** 6](#_Toc113259242)

[**5)** **Inferences from the models** 7](#_Toc113259245)

[**6)** **Managerial Implications** 10](#_Toc113259251)

[**7)** **Appendix** 11](#_Toc113259253)

[**6.1**  **Data Dictionary** 11](#_Toc113259254)

[**6.2** **Glossary** 12](#_Toc113259255)

[**6.3** **Association between Shareholderquity\_code and the target variable** 15](#_Toc113259256)

[**6.4** **References** 15](#_Toc113259258)

**List of Tables and Figures**

[**Figure 1: Comparison chart of performance metrics of all models** 6](#_Toc113259142)

[**Table 1: Comparison table of performance metrics of all models** 6](#_Toc113259143)

[**Figure 2: Pruned Decision Tree** 7](#_Toc113259145)

[**Figure 3: Pruned Decision Rules** 9](#_Toc113259146)

[**Figure 4: Variable Importance Plot generated by Decision Tree** 9](#_Toc113259147)

[**Figure 5: Variable Importance Plot generated by Random Forest** 10](#_Toc113259148)

[**Figure 6: Variable Importance Plot generated by XGBoost** 10](#_Toc113259149)

[**Table 2: Threshold values of most important factors for class membership** 11](#_Toc113259152)

[**Table 3: Association between Shareholderquity\_code and the target variable** 15](#_Toc113259157)

# **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. K-Nearest Neighbour
5. Linear Discriminant Analysis
6. 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.
4. **Shareholderquity\_code –** Thiscolumn also indicates health of the firm and Positive or Negative equity.

0 = healthy firm +ve equity

1 = healthy firm -ve equity

2 = banktrupt firm +ve equity

3 = banktrupt firm -ve equity

We observed that the columns, Target (Dummy Coded: Healthy= 0; NPA= 1) & Shareholderquity\_code are associated after analysing the contingency table. Ref. Appendix 6.3

## **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.
* **Twenty variables are chosen by Decision Tree, Random Forest and LDA models**. They are ['Asset\_coverage', 'Cash\_ratio', 'Changeinsales\_Industry', 'Current\_ratio', 'EBIT\_Sales', 'Fixed Asset Turnover Ratio', 'Interest\_coverage', 'Inventory\_turnover', 'Ln\_GVA', 'Operating Cash Flow/Total Debt', 'Operating Cash Flow/Total Sales', 'Quick\_ratio', 'ROS(new)', 'Receivable\_turnover(new)', "Total shareholders' funds", 'YOY EBIT Growth Rate', 'Year Encoded', 'debt\_asset', 'debt\_equity', 'debt\_income']
* **Nine variables were selected by at least two of the above three models.** They are as follows:

['Cash\_ratio', 'Changeinsales\_Industry', 'debt\_equity', 'debt\_income', 'Interest\_coverage', 'Operating Cash Flow/Total Debt', 'Receivable\_turnover(new)', 'ROS(new)', "Total shareholders' funds"]

# **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 performance measures of all the models (Logistic Regression, Decision Tree, Random Forest, LDA, KNN and XGBoost) is given below:

|  |  |  |
| --- | --- | --- |
|  |  |  |
| **AUROC** | **Precision** | **Recall** |

# **Figure 1: Comparison chart of performance metrics of all models**

|  |  |  |  |
| --- | --- | --- | --- |
|  | Results of average of ten-fold cross validation | | |
| Model | AUROC | Precision | Recall |
| Logistic Regression | 0.671746 | 0.347486 | 0.242293 |
| CART | 0.751404 | 0.569493 | 0.578224 |
| RF | 0.935203 | **0.875267** | 0.515562 |
| LDA | 0.687096 | 0.592857 | 0.009062 |
| KNN | 0.687889 | 0.630005 | 0.274032 |
| XGB | **0.944597** | 0.829587 | **0.597032** |

# **Table 1: Comparison table of performance metrics of all models**

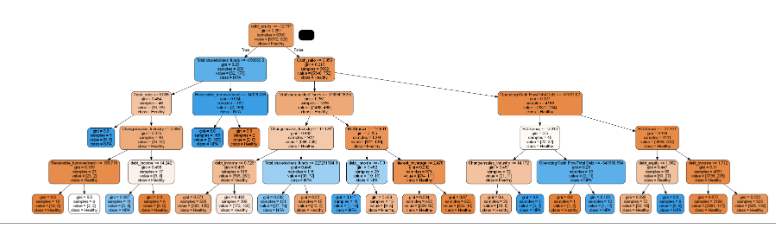
Even though XGBoost model gives a better AUROC and Recall than all the other models, we have chosen the Decision Tree (CART) model as our best model because of the fairly good accuracy in prediction, and its power of interpretation. Performance measures of the model, Random Forest is also good.

**For the Decision Tree (CART) model,**

* **AUROC is 0.71514.** Going by the thumb rule for AUROC, an AUROC of 0.70 – 0.80 is a good performance while 0.5 is considered as bad performance equivalent to random guessing. An AUROC of 0.75 would actually mean that let’s say we take two data points belonging to separate classes, Healthy and NPA, then there is 75% chance model would be able to segregate them correctly. This is acceptable.
* **Precision is 0.5695.** In other words, when it predicts a company becoming a NPA, it is correct approximately, 57% of the time. This is acceptable.
* **Recall is 0.5782.** In other words, it correctly identifies approximately, 58% of all NPAs. This is acceptable

# **Inferences from the models**

CART Tree



# **Figure 2: Pruned Decision Tree**

Decision Rules

if ( debt\_equity <= -12.757390975952148 ) {

if ( Total shareholders' funds <= -150650.5 ) {

if ( Cash\_ratio <= 0.005304956343024969 ) {

Class = NPA

} else {

if ( Changeinsales\_Industry <= 0.6953547894954681 ) {

if ( Receivable\_turnover(new) <= 160.7162322998047 ) {

Class = Solvant

} else {

Class = NPA

}

} else {

if ( debt\_income <= 14.242411136627197 ) {

Class = NPA

} else {

Class = Solvant

}

}

}

} else {

if ( Receivable\_turnover(new) <= 34026.326171875 ) {

Class = NPA

} else {

Class = Solvant

}

}

} else {

if ( Cash\_ratio <= 0.05487825535237789 ) {

if ( Total shareholders' funds <= 378359152.0 ) {

if ( Changeinsales\_Industry <= 14.128055095672607 ) {

if ( debt\_income <= 0.7259111702442169 ) {

Class = Solvant

} else {

Class = Solvant

}

} else {

if ( Total shareholders' funds <= 227251984.0 ) {

Class = NPA

} else {

Class = Solvant

}

}

} else {

if ( ROS(new) <= -14.960599899291992 ) {

if ( debt\_income <= -0.00016701318963896483 ) {

Class = NPA

} else {

Class = Solvant

}

} else {

if ( Interest\_coverage <= 2.478181481361389 ) {

Class = Solvant

} else {

Class = Solvant

}

}

}

} else {

if ( Operating Cash Flow/Total Debt <= -97937.01953125 ) {

if ( ROS(new) <= 0.013428250094875693 ) {

if ( Changeinsales\_Industry <= 14.172158241271973 ) {

Class = Solvant

} else {

Class = NPA

}

} else {

if ( Operating Cash Flow/Total Debt <= -341930.59375 ) {

Class = Solvant

} else {

Class = NPA

}

}

} else {

if ( ROS(new) <= -27.421380043029785 ) {

if ( debt\_equity <= 1.9619519114494324 ) {

Class = Solvant

} else {

Class = NPA

}

} else {

if ( debt\_income <= 1.7123816013336182 ) {

Class = Solvant

} else {

Class = Solvant

}

}

}

}

}

# **Figure 3: Pruned Decision Rules**

**We observe the following features are very important in predicting bankruptcy:**

|  |
| --- |
|  |
|  |

Decision trees (CART) offer importance scores based on the reduction in the criterion (gini) used to split points.

# **Figure 4: Variable Importance Plot generated by Decision Tree**

**From Random Forest Model**

|  |
| --- |
|  |
|  |

# **Figure 5: Variable Importance Plot generated by Random Forest**

**From XGBoost Model**

|  |
| --- |
|  |
|  |

# **Figure 6: Variable Importance Plot generated by XGBoost**

# **Managerial Implications**

The following three variables appear to be most important factors leading to NPA of a company as per all the three models (Decision Tree, Random Forest and XGBoost). They appear in the top 3 most important variables in the Variable importance plots and table.

1. debt\_equity
2. Total shareholders' funds
3. Cash\_ratio

A table of threshold values of the above variables for class membership – NPA is given below:

|  |  |
| --- | --- |
| **Important factor** | **Threshold value for membership in class, NPA** |
| debt\_equity | <= -12.75 |
| Total shareholders' funds | <= -150650 |
| Cash\_ratio | <= 0.0053 |

# **Table 2: Threshold values of most important factors for class membership**

From the above table, we infer that any company if these financial indicators fall below their thresholds, the company will become NPA.

# **Appendix**

## **6.1 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 |
|  | | |

## **6.2 Glossary**

**A) List of Machine Learning techniques used**

**1) Logistic Regression**

Logistic regression is one of the most popular Machine Learning algorithms in Supervised Learning techniques. Logistic regression is a classification technique that is used to calculate (or predict) the probability of a binary (yes/no) event occurring. In Logistic regression, instead of fitting a regression line, we fit an "S" shaped logistic function, sigmoid to predict.

The curve from the logistic function indicates the likelihood of whether the firm will go bankrupt or not. Logistic regression is a significant machine learning algorithm because it has the ability to provide probabilities and classify new data using continuous and discrete datasets.

Logistic regression can be used to classify the observations using different types of data and can easily determine the most effective variables used for the classification.

**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. Decision Tree is a supervised learning technique that can be used for both classification and regression problems, but mostly it is preferred for solving classification problems.

It is a tree-structured classifier, where internal nodes represent the features of a dataset, branches represent the decision rules and each leaf node represents the outcome.

In a Decision Tree, there are two nodes, Decision Node and Leaf Node. Decision Nodes are used to make any decision and have multiple branches, whereas Leaf Nodes are the output of those decisions and do not contain any further branches.

It is a graphical representation for getting all possible solutions to a problem / decision based on given conditions.

*\*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 popular machine learning algorithm that belongs to the supervised learning technique. It can be used for both Classification and Regression problems. It is based on the concept of ensemble learning, which is a process of combining multiple classifiers to solve a complex problem and to improve the performance of the model. It builds decision trees on different samples and takes their majority vote for classification and average in case of regression.

The greater number of trees in the forest leads to higher accuracy and prevents the problem of evalutaion of algorithms on training data is different from new or unseen data (overfitting).

**4) K-Nearest Neighbour**

In statistics, the K-Nearest Neighbors algorithm (k-NN) is a non-parametric supervised learning method first developed by Evelyn Fix and Joseph Hodges in 1951, and later expanded by Thomas Cover. It is 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.

A K-NN is a type of classification where the function is only approximated locally and all computation is deferred until function evaluation. It is also called a lazy learner algorithm.

**5) Linear Discriminant Analysis**

Linear Discriminant Analysis (LDA) or Normal Discriminant Analysis (NDA) or Discriminant Function Analysis (DFA) is a generalization of Fisher's linear discriminant, a method used in statistics and other fields, to find a linear combination of features that characterizes or separates two or more classes of objects or events.

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.

LDA is one of the commonly used dimensionality reduction techniques in machine learning to solve more than two-class classification problems.

**6) 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. It was designed for performance and speed. It is an implementation of gradient boosting machines created by **Tianqi Chen**.

XGBoost supports the three main forms of gradient boosting:

1. Gradient Boosting, known as Gradient Boosting Machine (GBM)
2. Stochastic Gradient Boosting, a boosting technique with sub-sampling at the row, column, and column per split levels
3. Regularized Gradient Boosting, which includes boosting with L1 and L2 regularization

**B) Model Performance Metrics**

1. **ROC and AUROC**

Receiver Operating Characteristic (ROC) curve is a plot of True Positive Rate and False Positive Rate values derived from different decision thresholds for a particular classification model.

AUC or AUROC is the area under the ROC curve and it characterizes the model performance.

1. **Precision**

Precision may be seen as the fraction of correct positive predictions made for all cases that are classed as positive.

1. **Recall**

The recall measures the model's ability to detect positive samples. The higher the recall, the more positive samples detected.

**C) Financial Metrics**

1. **debt\_equity**, a financial ratio indicating the weight of total debt and financial liabilities against shareholders' equity.
2. **Total shareholders' funds** is the shareholders' investment in the company by adding the share capital to all the retained profits of the company.
3. **Cash\_ratio**, a metric which evaluates a company's ability to repay its short-term liabilities with cash and cash equivalents.
4. **Interest\_coverage**, a metric that helps assess how easily an entity could pay the interests against the outstanding dues it has.

## **6.3 Association between Shareholderquity\_code and the target variable**

We removed null values from the data and then we prepared a contingency table for the variables, Target & Shareholderquity\_code.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Target** | **Shareholderquity\_code** | | | |
|  | banktrupt firm +ve equity | banktrupt firm -ve equity | Healthy firm +ve equity | Healthy firm -ve equity |
| **Healthy** | **0** | **0** | **3305** | **601** |
| **NPA** | **865** | **439** | **0** | **0** |
|  | | | | |

# **Table 3: Association between Shareholderquity\_code and the target variable**

We infer that knowing the value of the variable, Shareholderquity\_code gives us the information about the Target variable.

There is no case where the value of the variable, Shareholderquity\_code is:

1. bankrupt and (positive or negative) equity and the target variable is **Healthy**

**or**

1. Healthy and (positive or negative) equity and the target variable is **NPA**.

Therefore, we conclude that the variables Shareholderquity\_code and our Target variable, **Dummy Coded :Healthy= 0; NPA= 1** are completely dependent and

there is no need to perform Chi-square test for independence.

## **6.4 References**

1. <https://www.javatpoint.com/logistic-regression-in-machine-learning>
2. [https://www.javatpoint.com/machine-learning-decision-tree-classification-algorithm](https://www.javatpoint.com/machine-learning-decision-tree-classification-algorithm%20)
3. <https://www.javatpoint.com/machine-learning-random-forest-algorithm>
4. [K-Nearest Neighbor(KNN) Algorithm for Machine Learning - Javatpoint](https://www.javatpoint.com/k-nearest-neighbor-algorithm-for-machine-learning)
5. [Linear Discriminant Analysis (LDA) in Machine Learning - Javatpoint](https://www.javatpoint.com/linear-discriminant-analysis-in-machine-learning)
6. <https://www.nvidia.com/en-us/glossary/data-science/xgboost/>
7. <https://data-flair.training/blogs/xgboost-tutorial/>
8. https://deepai.org/machine-learning-glossary-and-terms/precision-and-recall#:~:text=Precision%20and%20recall%20are%20two%20numbers%20which%20together,fraction%20of%20retrieved%20instances%20among%20all%20relevant%20instances.