Company Bankruptcy Prediction

Shantanu Houzwala

Data science trainee, AlmaBetter, Bangalore

Abstract:

Prediction of bankruptcy is a phenomenon of increasing interest to firms who stand to lose money because on unpaid debts. Since computers can store huge dataset pertaining to bankruptcy making accurate predictions from them before hand is becoming important.

The data were collected from the Taiwan Economic Journal for the years 1999 to 2009. Company bankruptcy was defined based on the business regulations of the Taiwan Stock Exchange.

Problem Statement:

The main objective of this project is to use various classification algorithms on bankruptcy dataset to predict bankruptcies with satisfying accuracies long before the actual event.

Dataset Information:

- Number of instances: 6819
- Number of attributes: 96

Understanding Data:

Updated column names and description to make the data easier to understand (Y = Output feature, X = Input features)

Y - Bankrupt? Class label 1 : Yes , 0: No

X1 - ROA(C) before interest and depreciation before interest: Return On Total Assets (C)

X2 - ROA(A) before interest and % after tax: Return On Total Assets(A)

X3 - ROA(B) before interest and depreciation after tax: Return On Total Assets(B)

X4 - Operating Gross Margin: Gross Profit /Net Sales

X5 - Realized Sales Gross Margin: Realized Gross Profit/Net Sales

X6 - Operating Profit Rate: Operating Income/Net Sales

X7 - Pre-tax net Interest Rate: Pre-Tax Income/Net Sales

X8 - After-tax net Interest Rate: Net Income/Net Sales

X9 - Non-industry income and expenditure/revenue: Net Non-operating Income Ratio

X10 - Continuous interest rate (after tax): Net Income-Exclude Disposal Gain or Loss/Net Sales

X11 - Operating Expense Rate: Operating Expenses/Net Sales

X12 - Research and development expense rate: (Research and Development Expenses)/Net Sales

X13 - Cash flow rate: Cash Flow from Operating/Current Liabilities

X14 - Interest-bearing debt interest rate: Interest-bearing Debt/Equity

X15 - Tax rate (A): Effective Tax Rate

X16 - Net Value Per Share (B): Book Value Per Share(B)

X17 - Net Value Per Share (A): Book Value Per Share(A)

X18 - Net Value Per Share (C): Book Value Per Share(C)

X19 - Persistent EPS in the Last Four Seasons: EPS-Net Income

X20 - Cash Flow Per Share

X21 - Revenue Per Share (Yuan ¥): Sales Per Share

X22 - Operating Profit Per Share (Yuan ¥): Operating Income Per Share

X23 - Per Share Net profit before tax (Yuan ¥): Pretax Income Per Share

X24 - Realized Sales Gross Profit Growth Rate

X25 - Operating Profit Growth Rate: Operating Income Growth

X26 - After-tax Net Profit Growth Rate: Net Income Growth

X27 - Regular Net Profit Growth Rate: Continuing Operating Income after Tax Growth

X28 - Continuous Net Profit Growth Rate: Net Income-Excluding Disposal Gain or Loss Growth

X29 - Total Asset Growth Rate: Total Asset Growth

X30 - Net Value Growth Rate: Total Equity Growth

X31 - Total Asset Return Growth Rate Ratio: Return on Total Asset Growth

X32 - Cash Reinvestment %: Cash Reinvestment Ratio

X33 - Current Ratio

X34 - Quick Ratio: Acid Test

X35 - Interest Expense Ratio: Interest Expenses/Total Revenue

X36 - Total debt/Total net worth: Total Liability/Equity Ratio

X37 - Debt ratio %: Liability/Total Assets

X38 - Net worth/Assets: Equity/Total Assets

X39 - Long-term fund suitability ratio (A): (Long-term Liability+Equity)/Fixed Assets

X40 - Borrowing dependency: Cost of Interest-bearing Debt

X41 - Contingent liabilities/Net worth: Co ntingent Liability/Equity

X42 - Operating profit/Paid-in capital: Operating Income/Capital

X43 - Net profit before tax/Paid-in capital: Pretax Income/Capital

X44 - Inventory and accounts receivable/N et value: (Inventory+Accounts Receivables)/Equity

X45 - Total Asset Turnover

X46 - Accounts Receivable Turnover

X47 - Average Collection Days: Days Rec eivable Outstanding

X48 - Inventory Turnover Rate (times)

X49 - Fixed Assets Turnover Frequency

X50 - Net Worth Turnover Rate (times): E quity Turnover

X51 - Revenue per person: Sales Per Empl oyee

X52 - Operating profit per person: Operati on Income Per Employee

X53 - Allocation rate per person: Fixed As sets Per Employee

X54 - Working Capital to Total Assets

X55 - Quick Assets/Total Assets

X56 - Current Assets/Total Assets

X57 - Cash/Total Assets

X58 - Quick Assets/Current Liability

X59 - Cash/Current Liability

X60 - Current Liability to Assets

X61 - Operating Funds to Liability

X62 - Inventory/Working Capital

X63 - Inventory/Current Liability

X64 - Current Liabilities/Liability

X65 - Working Capital/Equity

X66 - Current Liabilities/Equity

X67 - Long-term Liability to Current Assets

X68 - Retained Earnings to Total Assets

X69 - Total income/Total expense

X70 - Total expense/Assets

X71 - Current Asset Turnover Rate: Curre nt Assets to Sales

X72 - Quick Asset Turnover Rate: Quick Assets to Sales

X73 - Working capitcal Turnover Rate: W orking Capital to Sales

X74 - Cash Turnover Rate: Cash to Sales

X75 - Cash Flow to Sales

X76 - Fixed Assets to Assets

X77 - Current Liability to Liability

X78 - Current Liability to Equity

X79 - Equity to Long-term Liability

X80 - Cash Flow to Total Assets

X81 - Cash Flow to Liability

X82 - CFO to Assets

X83 - Cash Flow to Equity

X84 - Current Liability to Current Assets

X85 - Liability-
Assets Flag: 1 if Total Liability exceeds T
otal Assets, 0 otherwise

X86 - Net Income to Total Assets

X87 - Total assets to GNP price

X88 - No-credit Interval

X89 - Gross Profit to Sales

X90 - Net Income to Stockholder's Equity

X91 - Liability to Equity

X92 - Degree of Financial Leverage (DFL)

X93 - Interest Coverage Ratio (Interest ex
pense to EBIT)

X94 - Net Income Flag: 1 if Net Income is
Negative for the last two years, 0 otherwis
e

X95 - Equity to Liability

➢ Steps Involved:

- **Data Collection**

To proceed with the problem, first we will load our dataset that is given in .csv file format into a dataframe. Mount the drive and load the csv file into a dataframe.

- **Splitting the Data in two categories:**

The records are observed to be highly imbalanced. Thus it is necessary to consider balancing the dataset through "Upsampling or Downsampling" techniques.

Through df.info(), we observed that we have a majority of "float64" data. The categorical data is distinguished as binary 1 and 0, thus stored as "int64". We separate the numeric and categoric data to analyze our dataset.

- **Exploratory Data Analysis**

After loading the dataset, I looked for duplicate values and null values in the dataset. There were none. So, we performed EDA by comparing our target variable that is “Bankrupt?” with other independent variables. This process helped us figuring out various aspects and relationships among the target and the independent variables. It gave us a better idea of which feature behaves in which manner compared to the target variable.

- **Data Modelling**

The dataset is highly imbalanced. Thus, be fore training the model, we need to deal wi th this data. For this we need to take following steps:

- Split the dataset into training and testi ng sets (80% - 20%). We preserve the 20% testing set for the final evaluation
- Through "Stratified K Fold CrossVali dation" we will now distribute the 80 % training set into further training and testing splits.
- Since we are dealing with over 50 feat ures, we use "Randomized Search Cro ss Validation" as this technique proves to perform better with many features.

- **Fitting different Models**

For Model fitting, I tried various classification algorithms like:

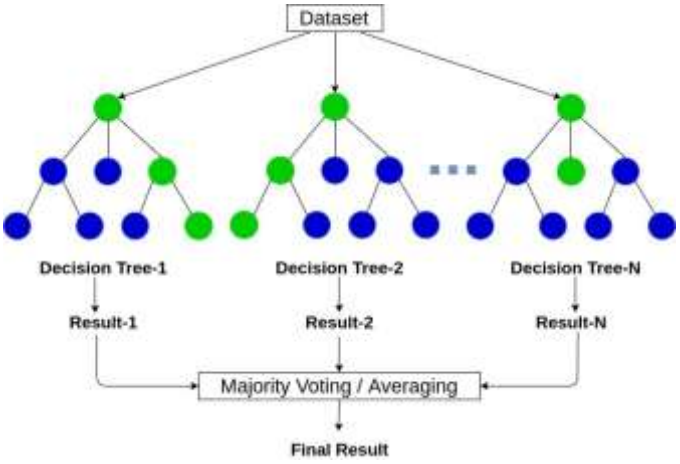
1. Random Forest Classifier
2. K Nearest Neighbour
3. Support Vector Classifier
4. Logistic Regressor
5. Decision Tree Classifier

- **Feature Selection**

I used “SelectKBest” function to select features that can add more value to the target variable.

➢ Algorithms:

1. Random Forest Classifier:



The Random forest or Random Decision Forest is a supervised Machine learning algorithm used for classification, regression, and other tasks using decision trees.

The Random forest classifier creates a set of decision trees from a randomly selected subset of the training set. It is basically a set of decision trees (DT) from a randomly selected subset of the training set and then It collects the votes from different decision trees to decide the final prediction.

When we implement this model, we get the following scores:

Model Score	Precision	Recall	F1 score	ROC-AUC score
98.9%	0.81	0.92	0.86	0.96

2. K Nearest Neighbor

K-nearest neighbors (KNN) is a type of supervised learning algorithm used for both regression and classification. KNN tries to predict the correct class for the test data by calculating the distance between the test data and all the training points. Then select the K number of points which is closest to the test data. The KNN algorithm calculates the probability of the test data belonging to the classes of 'K' training data and class which holds the highest probability will be selected.

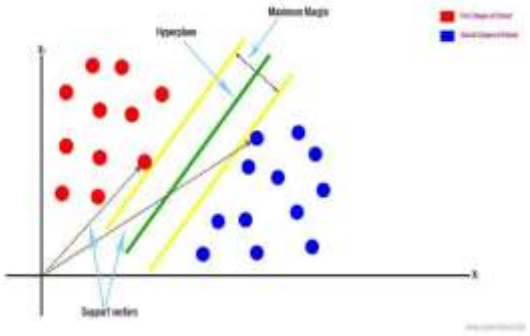
$$d(\mathbf{p},\mathbf{q}) = d(\mathbf{q},\mathbf{p}) = \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2 + \cdots + (q_n - p_n)^2}$$

$$= \sqrt{\sum_{i=1}^{n} (q_i - p_i)^2}.$$

When we implement this model, we get the following scores:

Model Score	Precision	Recall	F1 score	ROC-AUC score
98.02%	0.67	0.94	0.78	0.96

3. Support Vector Classifier



The objective of Support Vector Classifier 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. If the number of input features is two, then the hyperplane is just a line. If the number of input features is three, then the hyperplane becomes a 2-D plane. It becomes difficult to imagine when the number of features exceeds three. When we implement this model, we get the following scores:

Model Score	Precision	Recall	F1 score	ROC-AUC score
94.43%	0.40	0.94	0.56	0.94

4. Logistic Regression:

Logistic regression is basically a supervised classification algorithm. In a classification problem, the target variable (or output), y, can take only discrete values for a given set of features (or inputs), X. Logistic regression becomes a classification technique only when a decision threshold is brought into the picture. The setting of the threshold value is a very important aspect of Logistic regression and is dependent on the classification problem itself. When we implement this model, we get the following scores:

$$P = \frac{e^{a+bX}}{1+e^{a+bX}}$$

Model Score	Precision	Recall	F1 score	ROC-AUC score
88.86%	0.23	0.82	0.36	0.86

5. Decision Tree Classifier

Decision Tree is a Supervised Machine Learning Algorithm that uses a set of rules to make decisions, similarly to how humans make decisions. One way to think of a Machine Learning classification algorithm is that it is built to make decisions. The intuition behind Decision Trees is that you use the dataset features to create **yes/no** questions and continually split the dataset until you isolate all data points belonging to each

$$IG(D_p, f) = I(D_p) - \sum_{j=1}^{m} \frac{N_j}{N} I(D_j)$$

f: feature split on
D_p: dataset of the parent node
D_j: dataset of the jth child node
I: impurity criterion
N: total number of samples
N_j: number of samples at jth child node

class. With this process we are organizing the data in a tree structure. When we implement this model, we get the following scores:

Model Score	Precision	Recall	F1 score	ROC-AUC score
84.68%	0.18	0.90	0.31	0.87

- **Feature Selection using SelectKBest:**

  Scikit-learn API provides SelectKBest class for extracting best features of given dataset. The SelectKBest method selects the features according to the k highest score. Selecting best features is important process when we prepare a large dataset for training. It helps us to eliminate less important part of the data and reduce a training time. I applied this function on Random Forest Classifier and K Nearest Neighbor Classifier Algorithms but the results were not satisfactory.

- **Conclusion:**

  We have reached the end of our exercise:

  Starting with loading the data so far, we have done EDA, identifying null and duplicate values, finding correlation between target variables and other variables, model building and feature selection.

  After testing our models, we have found that Random Forest Classifier and K Nearest Neighbor Algorithms gives the highest accuracy rate of 98% and highest f-score. Thus, we can use these models to train our data.

References-

1. Towards Data Science
2. GeeksforGeeks
3. Analytics Vidhya