# Introduction:

Data Analysis is the statistics and probability to figure out trends in the data set. It helps in drilling down the information, to transform metrics, facts, and figures into initiatives for improvement. In this essay, we will explore the data using python and the supporting libraries.

# Data:

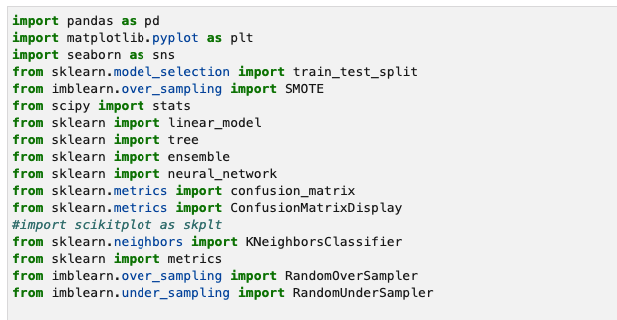
The data set we are going to use is the Taiwanese Bankruptcy Prediction Data Set publicly available in the UCI Machine earning Repository (<https://archive.ics.uci.edu/ml/datasets/Taiwanese+Bankruptcy+Prediction>)

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

# Exploratory Data Analysis:

We will explore the data sets and perform the exploratory data analysis in python.

First, we start by importing libraries in the Python notebook:



Once we import the libraries and load the csv files, we explore the shape of the dataframe:

Graphical user interface, application

Description automatically generated

Let’s take a look to the head of the dataset. For our screenshot, we are going to display first few columns of the data set.

Table

Description automatically generated

Is there any null value in the data frame?

Graphical user interface, text, application, chat or text message

Description automatically generated

Let’s find out the name of the columns in this data frame:

Graphical user interface, text, application

Description automatically generated

|  |  |  |
| --- | --- | --- |
| Bankrupt? | Operating Gross Margin | After-tax net Interest Rate |
| ROA(C) before interest and depreciation before interest | Realized Sales Gross Margin | Non-industry income and expenditure/revenue |
| ROA(A) before interest and % after tax | Operating Profit Rate | Continuous interest rate (after tax) |
| ROA(B) before interest and depreciation after tax | Pre-tax net Interest Rate | Operating Expense Rate |
| Research and development expense rate | Net Value Per Share (A) | Operating Profit Per Share (Yuan ¥) |
| Cash flow rate | Net Value Per Share (C) | Per Share Net profit before tax (Yuan ¥) |
| Interest-bearing debt interest rate | Persistent EPS in the Last Four Seasons | Realized Sales Gross Profit Growth Rate |
| Tax rate (A) | Cash Flow Per Share | Operating Profit Growth Rate |
| Net Value Per Share (B) | Revenue Per Share (Yuan ¥) | After-tax Net Profit Growth Rate |
| Regular Net Profit Growth Rate | Cash Reinvestment % | Debt ratio % |
| Continuous Net Profit Growth Rate | Current Ratio | Net worth/Assets |
| Total Asset Growth Rate | Quick Ratio | Long-term fund suitability ratio (A) |
| Net Value Growth Rate | Interest Expense Ratio | Borrowing dependency |
| Total Asset Return Growth Rate Ratio | Total debt/Total net worth | Contingent liabilities/Net worth |
| Operating profit/Paid-in capital | Average Collection Days | Operating profit per person |
| Net profit before tax/Paid-in capital | Inventory Turnover Rate (times) | Allocation rate per person |
| Inventory and accounts receivable/Net value | Fixed Assets Turnover Frequency | Working Capital to Total Assets |
| Total Asset Turnover | Net Worth Turnover Rate (times) | Quick Assets/Total Assets |
| Cash/Total Assets | Revenue per person | Current Assets/Total Assets |

For the sake of simplicity, some of the variable names have been dropped from the above table.

Let’s take a look to the column Bankrupt? Because this column is one of the column that we are going predict based on the other variables.

Chart

Description automatically generated

Since the dataset contains 96 variables and lot of the column we wont’ be using in our training, let’s make a new data frame using the variables that will be in our interest to predict the target variable.

Table

Description automatically generated

Now let’s look at the distribution of the data from our dataframe

Text

Description automatically generated

Chart

Description automatically generated

Chart, histogram

Description automatically generated

Chart

Description automatically generated

Chart, line chart

Description automatically generated

# Correlations:

The correlations between variable is very important to draw any conclusion regarding the relationship between data points. Looking at the row data, it’s really hard to find any relationship however, looking at the visual, it’s very obvious to establish a correlation. Let’s construct correlation matrix to observe the strength of relationships of each variable with bankruptc.

Icon

Description automatically generated with medium confidence

From the distribution and the heatmap, there are a lot of multicollinearity issues, skewed features and the data is imbalanced.

Let’s look at the relationship between some variables:

Does Net Income to Total Assets has any relationship with being bankrupt?

Chart

Description automatically generated

Does Retained Earnings to Total Assets has any relationship with being bankrupt?

Chart, histogram

Description automatically generated

Does Total Asset Growth Rate has any relationship with being bankrupt?

Chart, histogram

Description automatically generated

# Split Data into Train and Test Data set:

Now that we have explored the data, we want to come up with a model which should predict the target variable based on the data provided.. in our case the target variable is “Bankrupt?”

Using the sklearn library, we now divide the data set into training and test data set

Graphical user interface, text, application

Description automatically generated

# Random Oversampling and Undersampling for Imbalanced Classification

As we already seen that the data is imbalanced and skewness exists in the dataset, we need to address this issue. The skewness in training dataset can influence the modes leading some to ignore the minority class entirely. To address the this issue, we need to randomly resample the training dataset. The two approaches are as follows:

* Undersampling
* Oversampling

We will take advantage of the library imbalanced-learn available.

Graphical user interface, text, application, email

Description automatically generated

Now that we have two additional training set, let’s draw a baseline accuracy which will help us with our model evaluation

Graphical user interface, text, application

Description automatically generated

# Decision Tree Classifier:

Using our training datasets, let’s make Decision tree classifier:

Graphical user interface, text, application

Description automatically generated

In the above code, we essentially created three models. First one is the regular Decision tree classifier model which took our original training datasets. The second and third took the undersampling and oversampling training dataset respectively.

Now that we have the models, let’s evaluate with our test dataset:

Graphical user interface, text, application

Description automatically generated

Looks like the undersampling model performance is not near the baseline. Infact, none of the models was able to beat the base line in terms of their accuracy.