**Finance and Risk Analytics**

# Project Report

**Part 1 - Credit Risk**

The dataset contains information on default payments & company details of Companies in India. You are requested to create a Credit Risk Model, using the data provided. Please use the logistic regression framework to develop the credit default model.

* 1. **Exploratory Data Analysis**

We import all the required libraries

We import the data using the pd.read function. We also update the column names by updating the replacing blanks with “\_” to make it easier while referring to individual columns

We use the df.head function to see if the data has been imported properly. Further we look at the summary of the data, using the df.info and df. describe function

df.info function gives us the data types of each variable and number of non-null values. df. describe tells us the 5 point summary of the continuous variables

We also use the shape function to see the dimension of the data i.e. the number of rows and columns. Our data set has 3586 rows (observations) and 24 columns (variables)

1. Outlier Treatment

To treat the outliers we first plot the boxplot and see which variables have outliers.

A screenshot of text

Description automatically generated

From the boxplot we can observe that all the variables have outliers. In order to identify the outliers we use the IQR method wherein we will use flooring an capping to replace all values greater than 1.5\*IQR and lesser than 1.5\*IQR with the Q3 and Q1 values respectively. We could choose to delete the outliers had there been very few in number however, since there are many values as outliers we replace them in order to have enough information to build an effective model

To do this we define a function to identify the lower range and upper range and then using the for loop we replace the values in all the columns with the Q1 and Q3 values as applicable.

After treating the outliers we again plot a boxplot to see whether any outliers are still available in the dataset

A screenshot of a social media post

Description automatically generated

We can observe from the above boxplot that only Company Code variable has outliers. Since the variable is a nominal variable we have not treated the outliers in this case.

1. Missing Value Treatment

To identify the missing values we use the isnull().sum() function. This shows us the null values present in the each column.

A screenshot of text

Description automatically generated

From the output we can see that we have some null values in the variables – Current Assets, PBIT, Book Value Adj Unit Curr, EPS Annualised Adjusted Unit Curr, APATM % Latest, Creditors Velocity Days, Total liabilities and current ratio

From sklearn we import Simple Imputer to impute these missing values. We use the median strategy as our data had many outliers. Even though we have treated the outliers, the mean may skew our data and therefore we will use median.

After imputing we run the isnull function once again to verify whether all null values have been updated

A screenshot of text

Description automatically generated

As we can see there are no more null values in the dataset

1. New Variables Creation

As per the problem statement we need to create 3 new variables in our dataset –

* Return on Total Asset = Profit Before Tax(PBT)/ Total Assets
* Profit Margin = Profit Before Tax(PBT)/Total Sales or Revenue
* Debt to Equity Ratio = Total Liabilities / Total Equity

We create new calculated fields in our dataset by adding new columns by calculating the values on the basis of our variables

Our data set then transforms from 24 columns to 27 columns

1. Transform target variables into 0 and 1

We use the variable net worth next year to build our target variable “default” . We replace the values of instance where network next year is below 0 with 1 – Will default and the remaining with 0 – Will not default

0 – Company will not default

1 – Company will default

This gives us our target variable with 0 and 1 values

We also check the proportion of 1s and 0s in the dataset.

The overall default rate is 10.8%

1. Perform Univariate and Bivariate analysis (Including heatmap)

For univariate analysis we plot the histogram for a few variable to see the spread of the data

First is our target variable – Net worth next year –

A picture containing drawing

Description automatically generated

We can observe that the majority of the data lies around 0 value. It is not an even spread hence we cannot say that it is normally distributed

Similar we have plotted histograms for some other variables

A picture containing drawing

Description automatically generated A picture containing drawing

Description automatically generated A picture containing drawing

Description automatically generated

A screenshot of a cell phone

Description automatically generated A close up of text on a white background

Description automatically generated

From the above variable we can see that apart from PBT as perc of total income all variables are highly right skewed. PBT as perc of income seems more normally distributed

For bivariate analysis we looked at the relation of a few variable with the target variable – Networth next year

A screenshot of a cell phone

Description automatically generated A screenshot of a cell phone

Description automatically generated A screenshot of a cell phone

Description automatically generated A close up of a map

Description automatically generated

For bivariate analysis we have looked at 4 variables – Sales, PBT as perc of total income, Retained earnings and Return on Total asset

As is apparent from the scatter plot, there is no linear relationship between the target variables and the independent variable. Only in Return on Total Assets we can see that the irrespective of the net worth the return on total asset is almost always 0.

We can do a deeper dive between variable using correlation. We plot a heatmap to identify if there is high correlation between any of the variables resulting in multicollinearity. If multicollinearity is present in the data the same has to be dealt with before building the model so that it does not affect the accuracy.

On plotting the heatmap we observe that there are many variable highly correlated. (Kindly refer the appendix for the heatmap)

The highest correlation value is of 0.99 between the following the variables –

1. CP and PBDT
2. PAT and PBT

There are many other variables with very high correlation and the same will be dealt with while building the model

1. Perform Train Test Split

To build a model we split our data into train and test data. We build the model on the train data and then test it using the test data to verify the accuracy.

We populate a new variable X with all the dependent variable. We drop the target variable – default, Company code as it is not relevant field, Networh Next year as the target variable is calculated on the basis of this and Profit on Margin. Profit on margin was a new variable that was calculated however it has infinity values which hampers the model building and therefore it has been dropped

We carry out the train and test split of 67% and 33% respectively.

* 1. **Modelling**

Before building a model, since we have many variables, we check for the multicollinearity between the variables using VIF ( variance inflation factor)

We import the relevant library and define a function to calculate the VIF for each variable in the dataframe (X)

We calculate the VIF values in and eliminate the ones which have high multicollinearity (value of greater than 5). Out of the 24 variables, 13 variables have a value of VIF greater than 5 which means there is high collinearity. Therefore we will not consider these variables while building our model

a) Build a Logistic Regression Model on most important variables on Train Dataset

We then create the logistic regression and store it in the variable model. Then we fit the data on the train set

The results of our model on train data are shown below:

**A screenshot of a cell phone

Description automatically generated**

We can see that few variables are insignificant and may not be useful for cases of default

We also look at the adjusted pseudo – r square value (it is a measure of how well variables of the model explain the probability of default) we have a value of 0.57

We then look at the adjusted pseudo r square value (it has been adjusted for the number of predictors in the model). We have a value of 0.559

Adjusted pseudo R-square seems to be lower than Pseudo R-square value which means there are insignificant variables present in the model.

We then try & remove variables whose p value is greater than 0.05 & rebuild our model. The summary is as below

A screenshot of a cell phone

Description automatically generated

We can see all the variables are significant

Pseudo R-square value -0.56

Adjusted pseudo r-square value- 0.558

We see that adjusted R sq. is now close to R sq., thus suggesting lesser insignificant variables in the model

We also notice that current model has no insignificant variables and can be used for prediction purposes.

Let’s test the prediction of this model on train and test dataset

Prediction on the data

We first check the distribution plot of the logit function values. We use the sns.distplot function to plot this

A close up of a logo

Description automatically generated

We then predict the probability values for the X train data

We then plot a boxplot for the default variable and see the spread of the values. Plot of actual default vs predicted default

A picture containing clock

Description automatically generated

Based on the boxplot we take a cut-off value of 0.07 for probability of default and use this to see the predicted classes

b) State the accuracy, specificity, and sensitivity of the model based upon the optimized cut-off value

We plot the confusion matrix to check the accuracy and other performance measures for the training set with a cut-off value of 0.07

A screenshot of a cell phone

Description automatically generated

Sensitivity -94% of those defaulted were correctly identified as defaulters by the model

Specificity - 83% of those not defaulted were correctly identified as non-defaulters by the model

Accuracy -Overall 84% of correct predictions to total predictions were made by the model

We also check the performance measures using the a cut-off value of 0.08

A screenshot of a cell phone

Description automatically generated

Sensitivity -93% of those defaulted were correctly identified as defaulters by the model

Specificity - 84% of those not defaulted were correctly identified as non-defaulters by the model

Accuracy -Overall 85% of correct predictions to total predictions were made by the model

b) State the accuracy, specificity, and sensitivity of the model based upon the optimized cut-off value

We can see that 0.08 is the optimized cut-off value. Therefore –

Sensitivity -93% of those defaulted were correctly identified as defaulters by the model

Specificity - 84% of those not defaulted were correctly identified as non-defaulters by the model

Accuracy -Overall 85% of correct predictions to total predictions were made by the model

**1.3 Model Validation**

1. Validate the Model on Test Dataset and state the performance matrices

We use the logistic regression with a cut-off of 0.08 to validate the model on the test data

The accuracy of the model is 84.5%

A screenshot of a cell phone

Description automatically generated

Sensitivity -96% of those defaulted were correctly identified as defaulters by the model

Specificity - 83% of those not defaulted were correctly identified as non-defaulters by the model

Accuracy -Overall 84.5% of correct predictions to total predictions were made by the model

Classification report

A screenshot of a cell phone

Description automatically generated

1. Find out the Altman Score on test and train dataset

The Z-score model or Altman score model is based on five key financial ratios, and it relies on the information contained in the 10-K report. It increases the model’s accuracy when measuring the financial health of a company and its probability of going bankrupt.

The Altman’s Z-score formula is written as follows:

​ζ = 1.2A + 1.4B + 3.3C + 0.6D + 1.0E

Where:

- Zeta (ζ) is the Altman’s Z-score

- A is the Working Capital/Total Assets ratio

- B is the Retained Earnings/Total Assets ratio

- C is the Earnings Before Interest and Tax/Total Assets ratio

- D is the Market Value of Equity/Total Liabilities ratio

- E is the Total Sales/Total Assets ratio

We calculate the Altman score by calculating the values for A B C D and E in the formula and substituting them in the equation

We then use the standard threshold of 1.8 and use it to predict classes

We then build the confusion matrix

A screenshot of a cell phone

Description automatically generated

Sensitivity -77% of those defaulted were correctly identified as defaulters by the model

Specificity - 72% of those not defaulted were correctly identified as non-defaulters by the model

Accuracy -Overall 73% of correct predictions to total predictions were made by the model

1. Compare Altman’s Score with the Logistic Regression Model

On comparison of the Altman model and the Logistic Regression Model we can observe that the Logistic model has a higher accuracy as well as sensitivity and Specificity score which shows us that the Logistic model is the better model to predict the probability of default for a company. The accuracy of the model can be further improved if we consider more independent variables in the model.

However in we had an imbalanced dataset with high number of outliers. After treating the outliers the essence of the dataset could have changed. Perhaps if we had followed another approach of treating the outliers the result may have been different.

To conclude in this case, the logistic regression model is more accurate with an efficiency of 84.5% on the test data as opposed to accuracy of 73% for the Altman Z Score Test.

**Part 2: Market Risk**

The dataset contains 6 years of information(weekly stock information) on the stock prices of 10 different Indian Stocks. Calculate the mean and standard deviation on the stock returns and share insights.

We import the data using the pd.read function. We also update the column names by updating the replacing blanks with “\_” to make it easier while referring to individual columns

We use the df.head function to see if the data has been imported properly.

Further we look at the summary of the data, using the df.info and df. describe function

df.info function gives us the data types of each variable and number of non-null values. We have one object type variable (Date) and the remaining are all integers with no null values.

df. describe tells us the 5 point summary of the continuous variables

We also use the shape function to see the dimension of the data i.e. the number of rows and columns. Our data set has 314 rows (observations) and 11 columns (variables)

**2.1 Draw Stock Price Chart for any 2 variables**

In order to draw the Stock Price Chart we will use the scatter plot function. We are essentially plotting the data and the stock price – which shows the us the variation of a stock price over a period of 6 years. We can identify how the stock behaves. We first convert the date field which has data type object to date time data type and then plot the corresponding stock price values

We have plotted the Stock Price Chart for 2 companies – SAIL and Shree Cement

**SAIL**

A screenshot of a cell phone

Description automatically generated

From this graph we can observe that the stock prices of SAIL seem to follow a cyclical trend. There is no linear relationship. From 2014 to 2016 we saw a fall in the stock price followed by an increase from 2016 to middle of 2018 followed by decline till 2021

**Shree Cement**

A screenshot of a social media post

Description automatically generated

In contrast with the previous graph we can observe that the stock prices of Shree Cement share a linear and positive relationship with time. We can observe an increasing trend of Shree Cement stock prices over time showing that as time has passed shree cement stock prices have increased

**2.2 Calculate Returns**

Returns means the amount the investor will receive for the amount they have invested

We take log of price to reduce the noise factor and standardize the returns of the price we have

We assign new variable - **stock returns**. For this variable we drop date variable as the data is time dependent and take difference between stock values. We then take the head of this variable and can view the logarithmic return values for the first 5 rows

A screenshot of a cell phone

Description automatically generated

**2.3 Calculate Stock Means and Standard Deviation**

In order to understand the risk measure of our returns we have to understand the mean returns as well as the volatility of these returns

We calculate the stock mean using the .mean() function

A screenshot of a cell phone

Description automatically generated

We can observe that Mahindra, SAIL, SUN Pharma, Jindal Steel, Idea Vodafone and Jet Airways have negative returns whereas, Infosys, Indian Hotel, Axis Bank and Shree Cement have positive returns

Shree Cement has the highest return whereas Jet Airways has the lowest returns

We calculate the volatility of the stock using standard deviation. The standard deviation is calculated using the .std() function. It shows us that for even a small change in any external economic factor affecting the market, how much it will affect the stock price

A picture containing bird

Description automatically generated

We can observe that the Idea Vodafone is the most volatile stock and Infosys stock has the least volatility.

**2.4 Draw a plot of Stock Means vs. Standard Deviation and share insights**

We plot a scatter plot of the stock means and stock standard deviation to see the relationship between the return and volatility (risk) of the stock. Higher the level of Riskiness higher the level of returns

A screenshot of a social media post

Description automatically generated

Stock with a lower mean & higher standard deviation do not play a role in a portfolio that has competing stock with more returns & less risk.

Thus for the data we have here, we are only left few stocks:

- One with highest return and lowest risk &

- One with lowest risk and highest return

Based on the Returns and risk perspective, we can see that the best stocks to opt for would be Shree Cement and Infosys as they offer the highest returns and lowest risk in this dataset. If one wants to include more stocks in their portfolio we could also include Indian Hotel and Axis Bank in their portfolio as the returns are positive and risk is also not very high however, the return is quite low with respect to the risk therefore the investor must weigh the pros and cons before investing in these two stocks