greatlearning
Power Ahead

# FINANCE & RISK ANALYTICS

## (Milestone 1)

Mohamed Rifaz Ali K S

PGP-DSBA Online

May’ 22

# Contents

# List of Figures

# Problem: Probability at default

## Introduction

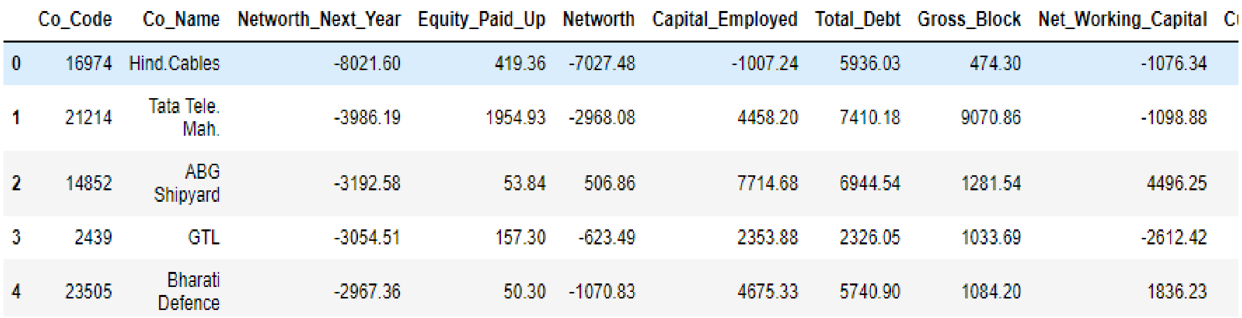
Businesses or companies can fall prey to default if they are not able to keep up their debt obligations. Defaults will lead to a lower credit rating for the company which in turn reduces its chances of getting credit in the future and may have to pay higher interests on existing debts as well as any new obligations. From an investor's point of view, he would want to invest in a company if it is capable of handling its financial obligations, can grow quickly, and is able to manage the growth scale.

A balance sheet is a financial statement of a company that provides a snapshot of what a company owns, owes, and the amount invested by the shareholders. Thus, it is an important tool that helps evaluate the performance of a business.

Data that is available includes information from the financial statement of the companies for the previous year (2015). Also, information about the Net worth of the company in the following year (2016) is provided which can be used to drive the labeled field.

## About Data

- The number of rows (observations) is 3586 **and t**he number of columns (variables) is 67**.**
- There are very few missing values present in the entire dataset. By default, **numerical &** categorical columns properly mapped**.**
- **Default value can be created using** Networth Next Year **variable.**
- **Total cell size is** 240262 **and the total missing value is 118.** It is just 0.05% of the missing values present in the dataset.
- **For the Inventory velocity days variable, only** 3% of the total is missing**.**
- **There is one bad data in the Inventory velocity day column and it is in negative. This is definitely data entry issue** -199 can be **substituted** to 199**.**



| | Co_Code | Co_Name | Networth_Next_Year | Equity_Paid_Up | Networth | Capital_Employed | Total_Debt | Gross_Block | Net_Working_Capital | C |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 16974 | Hind.Cables | -8021.60 | 419.36 | -7027.48 | -1007.24 | 5936.03 | 474.30 | -1076.34 | |
| 1 | 21214 | Tata Tele. Mah. | -3986.19 | 1954.93 | -2968.08 | 4458.20 | 7410.18 | 9070.86 | -1098.88 | |
| 2 | 14852 | ABG Shipyard | -3192.58 | 53.84 | 506.86 | 7714.68 | 6944.54 | 1281.54 | 4496.25 | |
| 3 | 2439 | GTL | -3054.51 | 157.30 | -623.49 | 2353.88 | 2326.05 | 1033.69 | -2612.42 | |
| 4 | 23505 | Bharati Defence | -2967.36 | 50.30 | -1070.83 | 4675.33 | 5740.90 | 1084.20 | 1836.23 | |

*Figure 1. Sample Company Finance Dataset*

## 1. Outlier Treatment

Almost, all the variables contain outliers.

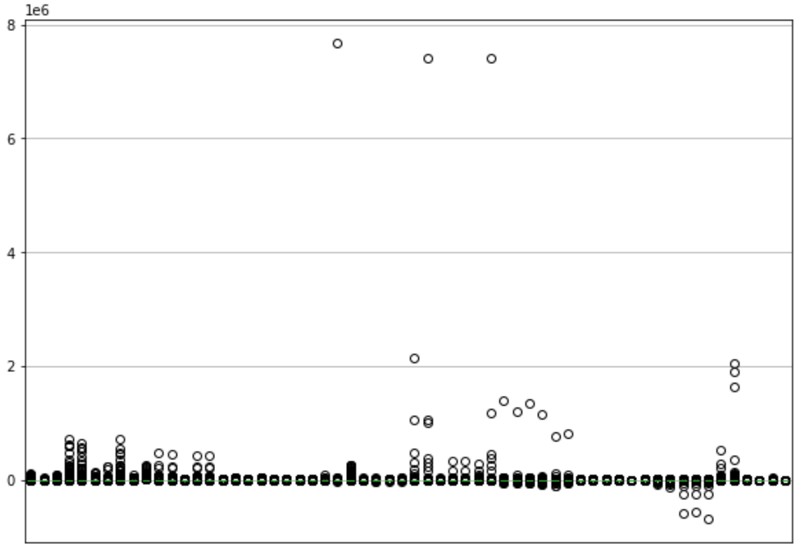


*Figure 2. Boxplot to check the outliers*

- Lets use the capping method to treat the outliers.
- Out of 3585 records, there are just 388(11% of the) records with default status. Data is highly imbalanced.
- We have 42449 outliers in total (i.e., 18% of the data's are outliers), of which 19% (~ 73 in number) are with a default status 1. As such we have only 388 default instances in our data therefore dropping these 73 records. i.e., 19% will not be a good option for this dataset.
- Before treating the outliers, let's include only the significant columns and remove the unwanted columns from the dataframe.
- The variables such as Revenue_earnings_in_forex, Revenue_expenses_in_forex, Capital_expenses_in_forex, ROG_Revenue_earnings_in_forex_perc, ROG_Revenue_expenses_in_forex_perc, ROG_Market_Capitalisation_perc do not have much variablility in the values, their 25%, 50% and 75% are just constant as 0. It might not be useful for our model building approach. We can keep Book Value (Unit Curr) column and drop Book Value (Adj.) (Unit Curr) column because both the columns are mimic to each other. Let's drop these redundant variables from the dataframe.

- For the remaining numerical columns, let’s calculate the Inter Quartile range between 0.25 and 0.75 and treat the lower and upper range range values.

  *Q1,Q3=np.percentile(col,[25,75])*

  *IQR=Q3-Q1*

  *lower_range= Q1-(1.5 * IQR)*

  *upper_range= Q3+(1.5 * IQR)*

- After treating the outliers, boxplots of the numerical variables looks neat.

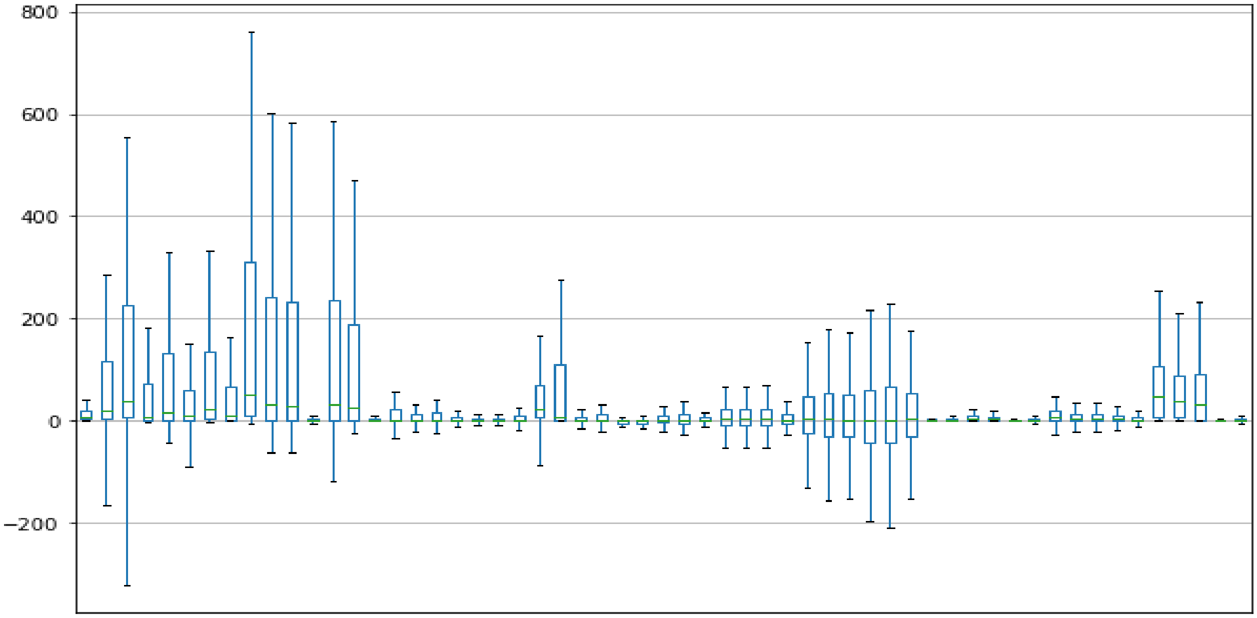


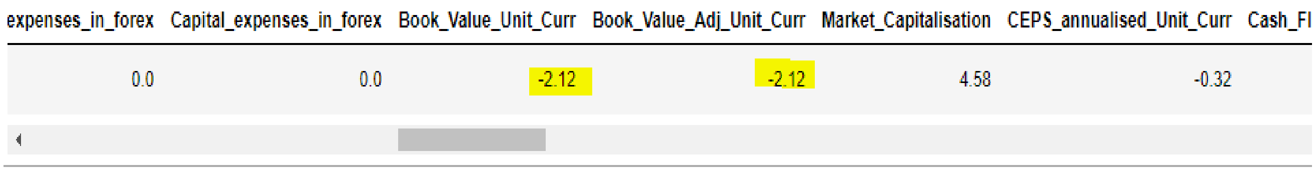
*Figure 3. Boxplot after outlier treatments*

## 2. Missing Value Treatment

- There are few columns with just 1 missing value in it.
- There are very few missing values in the entire dataset and present only in few columns.
- Total cell size is 240195 and Total missing values are 118. It is just 0.05% of the missing values present in the entire dataset.
- There is one row contains 11 missing values in it. We can identify and drop the row. The company named ‘**G M Breweries (Co_code=3240)**’ missed to provide the much financial information.

- After dropping the ‘G M Breweries’ company data, it looks much better now.

- Only 2 columns have missing values now. The columns are **Inventory_Velocity_Day** and **Book_Value_Adj_Unit_Curr.**
- **Inventory_Velocity_Day** – 103 missing values. Approximately, 3% of this column contains missing entries.
- Inventory_Velocity_Day has the data entry issue with the value -199, can be replaced with 199 days.
- Upon reviewing, **31%** of the company needs 0 days to convert the inventory into sales. So, let's impute the missing entries with 0 days using *fillna(value=0)* method.
- **Book_Value_Adj_Unit_Curr** – 4 missing values. It is just 0.1% of this column contains missing entries. On further analyzing the data, the column '*Book Value (Unit Curr)*' and '*Book Value (Adj.) (Unit Curr)*' contains the same values. We can drop the '*Book Value (Adj.) (Unit Curr)*' from the dataframe. Because 2 columns with the same data do not add any values to the model building. However, using the *fillna()* method, we copied the missing values from the *'Book Value (Unit Curr)'* column(*where the data is present*).
- Ensure to check for one record, we checked for **company code=495**, the value -2.12 is imputed in missing *Book Value (Adj.) (Unit Curr) column.*



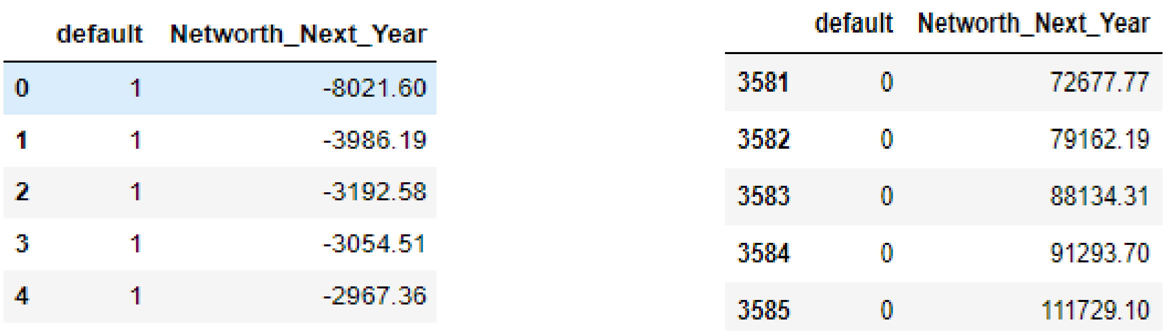
| expenses_in_forex | Capital_expenses_in_forex | Book_Value_Unit_Curr | Book_Value_Adj_Unit_Curr | Market_Capitalisation | CEPS_annualised_Unit_Curr | Cash_Fl |
|---|---|---|---|---|---|---|
| 0.0 | 0.0 | -2.12 | -2.12 | 4.58 | -0.32 | |

*Figure 4. After imputation of Book Value (Adj.) (Unit Curr)*

- Now, there is no missing entry present in the entire dataset.

## 3. Transform Target variable into 0 and 1

Create a dependent variable named 'default' that should take the value of 1 when '*Networth_Next_Year*' column value is **negative** & **0** when '*Networth_Next_Year'* value is positive. It is achieved using the *numpy* where function.



| | default | Networth_Next_Year |
|---|---|---|
| 0 | 1 | -8021.60 |
| 1 | 1 | -3986.19 |
| 2 | 1 | -3192.58 |
| 3 | 1 | -3054.51 |
| 4 | 1 | -2967.36 |

| | default | Networth_Next_Year |
|---|---|---|
| 3581 | 0 | 72677.77 |
| 3582 | 0 | 79162.19 |
| 3583 | 0 | 88134.31 |
| 3584 | 0 | 91293.70 |
| 3585 | 0 | 111729.10 |

*Figure 5. Default variable created using Networth Next Year column*

4. Univariate (4 marks) & Bivariate ( 6marks) analysis with proper interpretation. (You may choose to include only those variables which were significant in the model building)(10 marks)

**Univariate Analysis**

**Default Value counts**

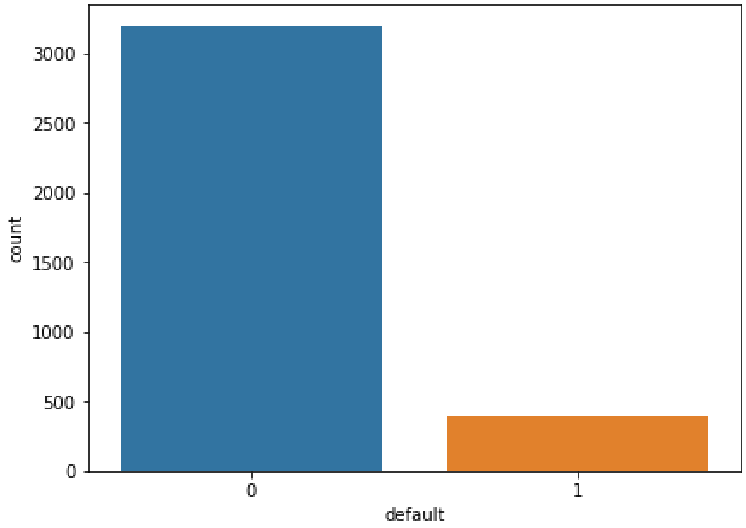
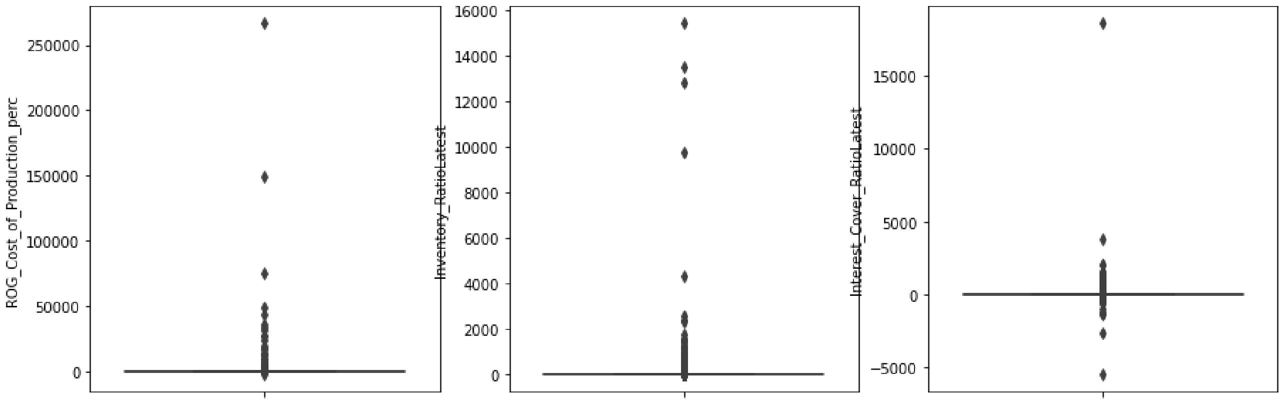


*Figure 6. Default value counts*

- Company fall under the ‘Default’ category is very minimal compared to the non-defaulters.
- Non-Defaulter company count is 3197 and Defaulter company count is 388.

**Boxplot for most important variables**



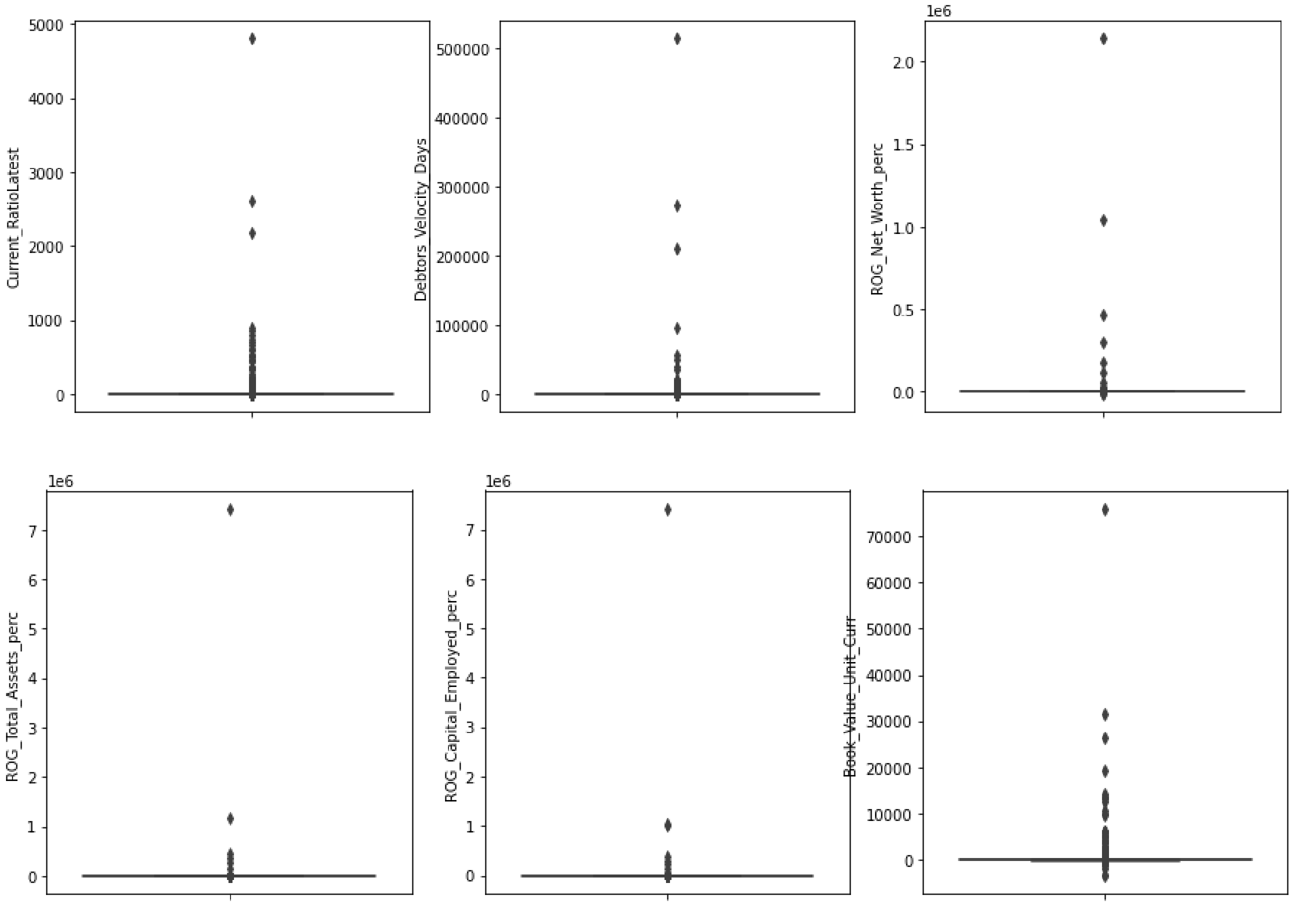


*Figure 7. Boxplot for significant variables*

- Almost all the variable contains abundant outliers.
- Variables such as *ROG_Cost_of_Production_perc,Inventory_RatioLatest, Current_RatioLatest, Debtors_Velocity_Days, ROG_Total_Assets_perc, ROG_Capital_Employed_perc, Book_Value_Unit_Curr* contains largely the upper range values.
- Interest_Cover_RatioLatest contains both the upper & lower range outliers.
- ROG_Net_Worth_perc, Book_Value_Unit_Curr contains the largely the upper outliers and minimal lower outliers.

## Histogram for most important variables

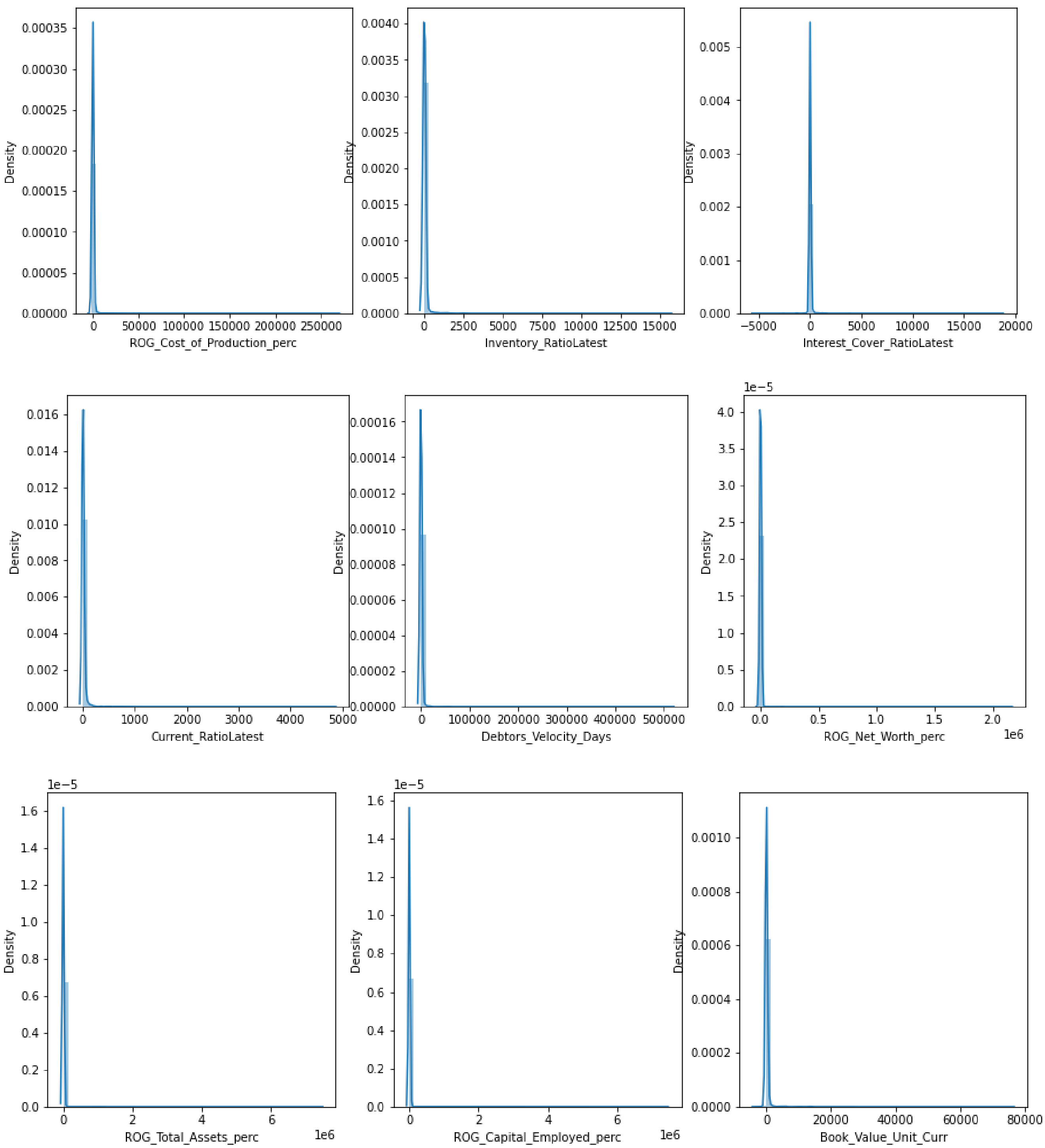


*Figure 8. Histogram for significant variables*

- Skewness for ROG_Cost_of_Production_perc is:  37.26
- Skewness for Inventory_RatioLatest is:  27.0
- Skewness for Current_RatioLatest is:  31.25
- Skewness for Interest_Cover_RatioLatest is:  40.82
- Skewness for Debtors_Velocity_Days is:  38.66
- Skewness for ROG_Net_Worth_perc is:  44.83
- Skewness for ROG_Total_Assets_perc is:  57.3
- Skewness for ROG_Capital_Employed_perc is:  56.43
- Skewness for Book_Value_Unit_Curr is:  32.98

- All the variables are positively skewed and the data's are clustered around the left tail of the distribution while the right tail of the distribution is longer.

**<u>Bivariate Analysis</u>**

**<u>Default Vs Book Value Unit Curr</u>**

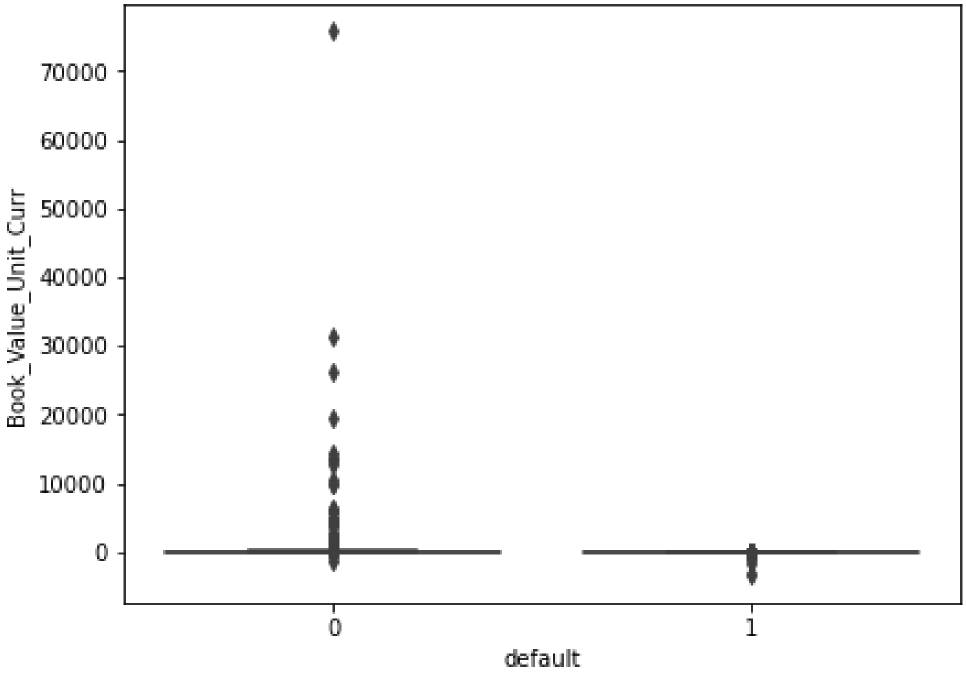


*Figure 9. Default Vs Book Value Unit Curr*

- Net asset value of defaulter's values is inclined towards negative side

**Default Vs Net Sales**

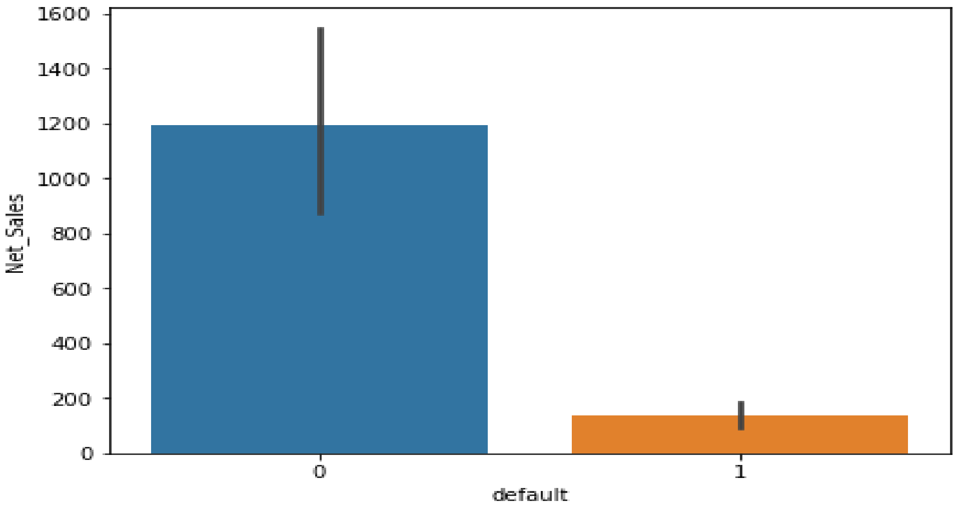


*Figure 10. Default Vs Net Sales*

- Defaulters seem to have very less net sales after subtracting the gross sales with returns, allowances and discounts.

**Default Vs Inventory Velocity Days**

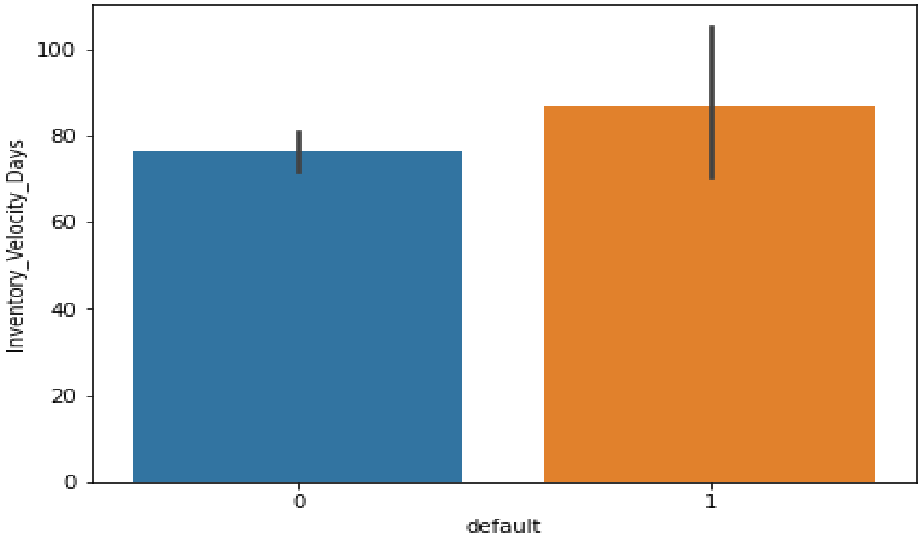


*Figure 11. Default Vs Inventory Velocity Days*

- The average number of days, Company needs to turn its inventory into sales for Defaulters is high compared to the non-defaulters.

**Heat Map**

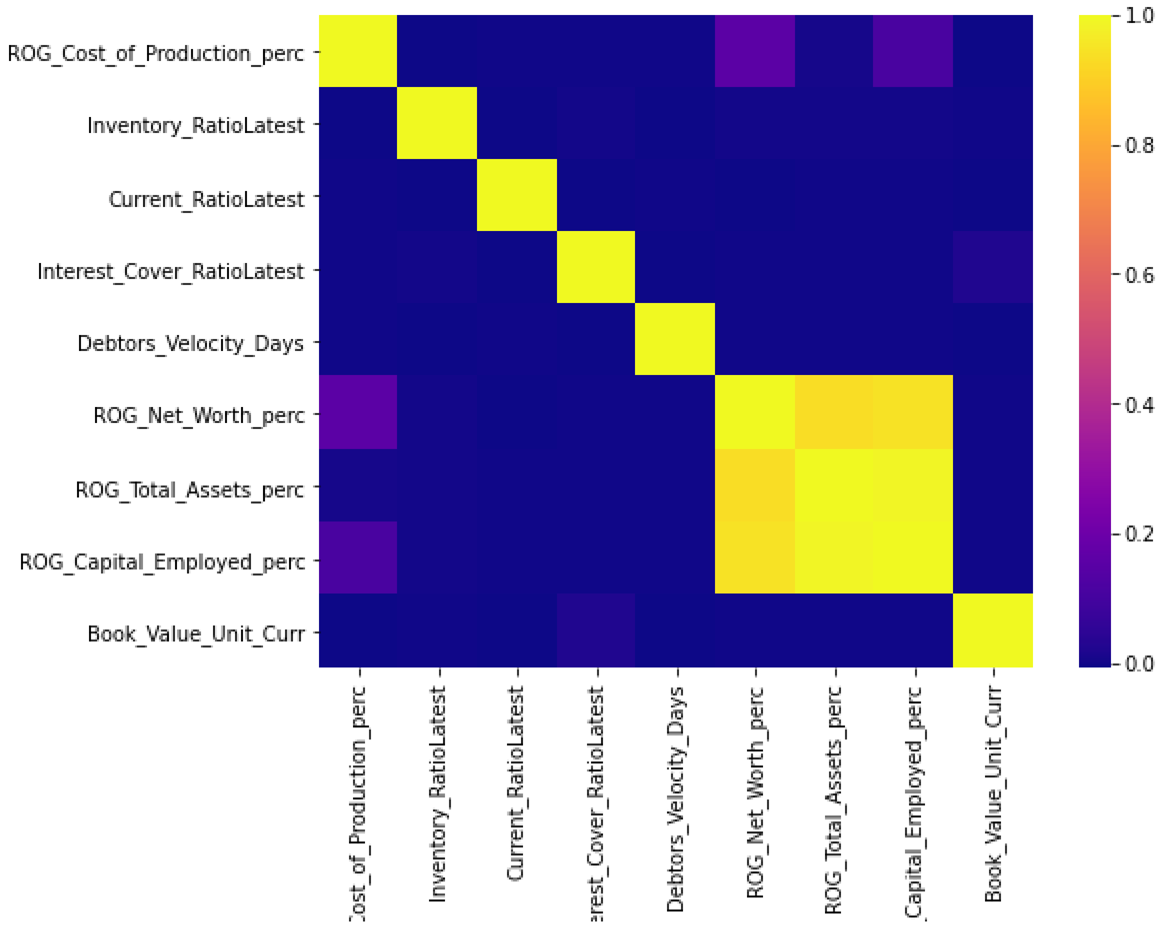


*Figure 12. Heat Map for Significant Variables*

- From the correlation heat map for the significant variables, we still see that there is high correlation on *ROG_Net_Worth_perc, ROG_Total_Assets_perc, ROG_Capital_Employed_perc* variables however the variation inflation factor values of these attributes below 5. So, let's keep it in the model building.
- The variables *ROG_Net_Worth_perc* and *ROG_Capital_Employed_perc* are very weakly correlated with *ROG_Cost_of_Production_perc* .
- Other variables such as *Inventory_RatioLatest, Interest_Cover_RatioLatest, Current_RatioLatest, Debtors_Velocity_Days, Book_Value_Unit_Curr* are not correlated with any other models which is the good sign of including these predictors in the model building.

## 5. Train Test Split

We see that proportions of dependent values (*default variable*) are highly imbalanced. i.e., 0 with 89% and 1 with 11%

Using *train_test_split()* method from sklearn library, split the data into train & test dataset in a ratio of 67:33 & used random_state =42 with the *stratify= Company['default']* parameters.

This stratify parameter makes a split so that the proportion of values in the sample produced will be the same as the proportion of values provided to parameter stratify

## 6. Build Logistic Regression Model (using statsmodel library) on most important variables on Train Dataset and choose the optimum cutoff. Also showcase your model building approach.

**Model Building using Logistic Regression for 'Probability at default'**

The equation of the Logistic Regression by which we predict the corresponding probabilities and then go on predict a discrete target variable is

$$\mathbf{y} = \frac{1}{1 - e^{-z}}$$

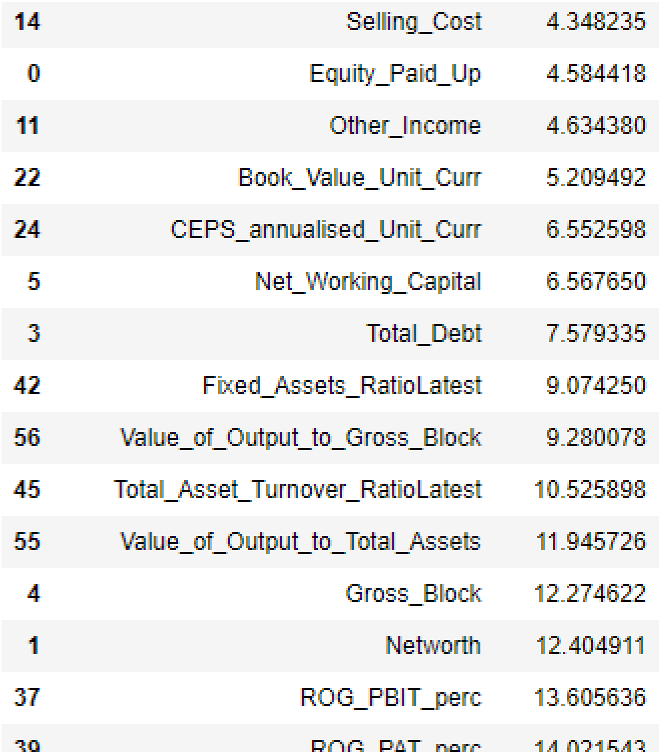
**Note: z =** $\beta_0$ **+**$\sum_{i=1}^{n}(\beta_i X_1)$

Using the statsmodels library, creating logistic regression equation & storing it in f_1

*model = SM.logit(formula=’Dependent Variable ~ Σ Independent Variables (k)’ data = ‘Data Frame containing the required values’).fit()*

Before starting model building, let’s look at the problem of multi-collinearity. Multi-collinearity occurs when two or more independent variables are highly correlated with one another in a regression model.

Using the VIF library, let’s calculate the VIF values for all the variables.



| | | |
|---|---|---|
| 14 | Selling_Cost | 4.348235 |
| 0 | Equity_Paid_Up | 4.584418 |
| 11 | Other_Income | 4.634380 |
| 22 | Book_Value_Unit_Curr | 5.209492 |
| 24 | CEPS_annualised_Unit_Curr | 6.552598 |
| 5 | Net_Working_Capital | 6.567650 |
| 3 | Total_Debt | 7.579335 |
| 42 | Fixed_Assets_RatioLatest | 9.074250 |
| 56 | Value_of_Output_to_Gross_Block | 9.280078 |
| 45 | Total_Asset_Turnover_RatioLatest | 10.525898 |
| 55 | Value_of_Output_to_Total_Assets | 11.945726 |
| 4 | Gross_Block | 12.274622 |
| 1 | Networth | 12.404911 |
| 37 | ROG_PBIT_perc | 13.605636 |
| 39 | ROG_PAT_perc | 14.021543 |

*Figure 13. VIF Table for all the Variables*

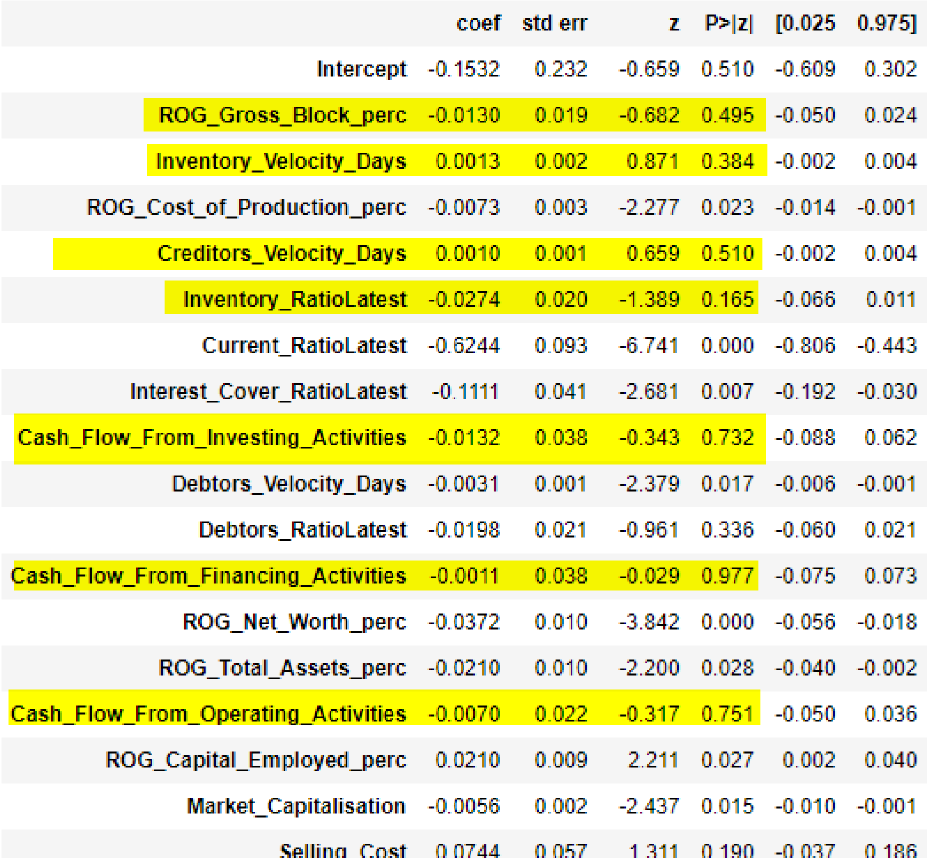
We see that the value of VIF is high for many variables. Here, we may drop variables with VIF more than 5 (very high correlation) & build our model.

Below independent variables are with the VIF values below 5.

*f_1 = 'default ~ ROG_Gross_Block_perc+Inventory_Velocity_Days+ROG_Cost_of_Production_perc+Creditors_Velocity_Days+Inventory_RatioLatest+Current_RatioLatest+Interest_Cover_RatioLatest+Cash_Flow_From_Investing_Activities+Debtors_Velocity_Days+Debtors_RatioLatest+Cash_Flow_From_Financing_Activities+ROG_Net_Worth_perc+ROG_Total_Assets_perc+Cash_Flow_From_Operating_Activities+ROG_Capital_Employed_perc+Market_Capitalisation+Selling_Cost+Equity_Paid_Up+Other_Income+Book_Value_Unit_Curr'*

Let’s train the model with the above variables using the train data.

The model summary is shown below



| | coef | std err | z | P>\|z\| | [0.025 | 0.975] |
|---|---|---|---|---|---|---|
| Intercept | -0.1532 | 0.232 | -0.659 | 0.510 | -0.609 | 0.302 |
| ROG_Gross_Block_perc | -0.0130 | 0.019 | -0.682 | 0.495 | -0.050 | 0.024 |
| Inventory_Velocity_Days | 0.0013 | 0.002 | 0.871 | 0.384 | -0.002 | 0.004 |
| ROG_Cost_of_Production_perc | -0.0073 | 0.003 | -2.277 | 0.023 | -0.014 | -0.001 |
| Creditors_Velocity_Days | 0.0010 | 0.001 | 0.659 | 0.510 | -0.002 | 0.004 |
| Inventory_RatioLatest | -0.0274 | 0.020 | -1.389 | 0.165 | -0.066 | 0.011 |
| Current_RatioLatest | -0.6244 | 0.093 | -6.741 | 0.000 | -0.806 | -0.443 |
| Interest_Cover_RatioLatest | -0.1111 | 0.041 | -2.681 | 0.007 | -0.192 | -0.030 |
| Cash_Flow_From_Investing_Activities | -0.0132 | 0.038 | -0.343 | 0.732 | -0.088 | 0.062 |
| Debtors_Velocity_Days | -0.0031 | 0.001 | -2.379 | 0.017 | -0.006 | -0.001 |
| Debtors_RatioLatest | -0.0198 | 0.021 | -0.961 | 0.336 | -0.060 | 0.021 |
| Cash_Flow_From_Financing_Activities | -0.0011 | 0.038 | -0.029 | 0.977 | -0.075 | 0.073 |
| ROG_Net_Worth_perc | -0.0372 | 0.010 | -3.842 | 0.000 | -0.056 | -0.018 |
| ROG_Total_Assets_perc | -0.0210 | 0.010 | -2.200 | 0.028 | -0.040 | -0.002 |
| Cash_Flow_From_Operating_Activities | -0.0070 | 0.022 | -0.317 | 0.751 | -0.050 | 0.036 |
| ROG_Capital_Employed_perc | 0.0210 | 0.009 | 2.211 | 0.027 | 0.002 | 0.040 |
| Market_Capitalisation | -0.0056 | 0.002 | -2.437 | 0.015 | -0.010 | -0.001 |
| Selling_Cost | 0.0744 | 0.057 | 1.311 | 0.190 | -0.037 | 0.186 |

*Figure 14. Model Summary for all the variables*

Noticed, the few columns of probability values are above 0.05.

We can see that few variables are insignificant & may not be useful to discriminate the cases of default.

Let us look at the adjusted pseudo R-square value

| | |
|---|---|
| R – Square | 0.6385 |
| Adj. R-Square | 0.6142 |

Adjusted pseudo R-square seems to be lower than Pseudo R-square value which means there are insignificant variables present in the model. Lets try & remove variables whose p value is greater than 0.05 & rebuild our model.

Feature selection is the process of tuning down the number of predictor variables used by the models you build.

For example, when faced with two models with the same or nearly the same score, but with the latter model using more variables, your immediate instinct should be to choose the one with fewer variables. That model is simpler to train, simpler to understand, easier to run, and less time consuming.

Let’s start with feature selection methods & validate them back using manual feature selection using backward elimination approach:

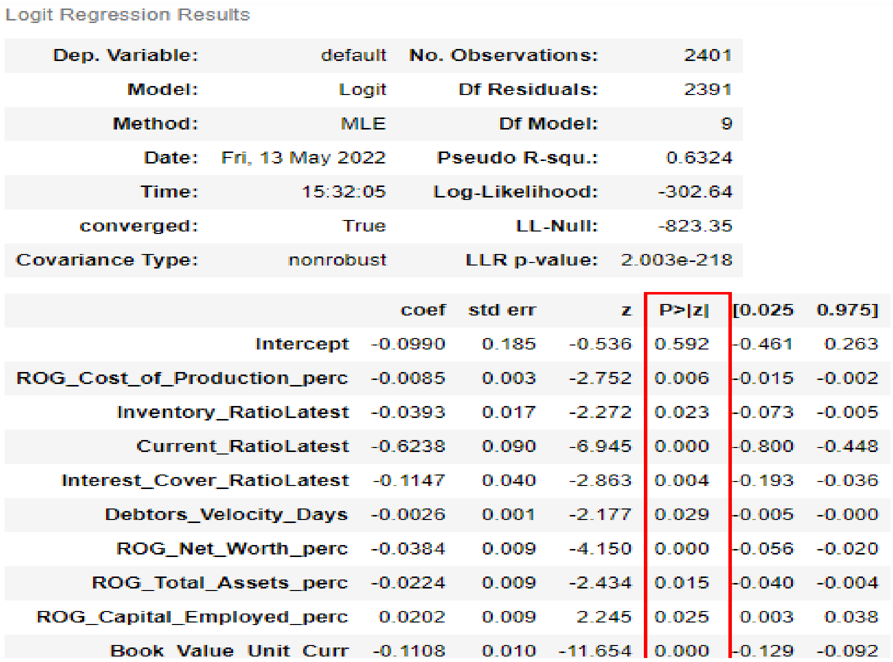
The variable *Cash flow from Financial Activities* has the highest p-value (0.977), therefore we need to eliminate it and rerun the model.

After going through this exercise, removing the variables has the p-values > 0.05 one after the other.

Finally, we have the model trained with below variables.

*f_1 = 'default ~ ROG_Cost_of_Production_perc+Inventory_RatioLatest+Current_RatioLatest+Interest_Cover_RatioLatest+Debtors_Velocity_Days+ROG_Net_Worth_perc+ROG_Total_Assets_perc+ROG_Capital_Employed_perc+Book_Value_Unit_Curr'*

The new efficient model summary is shown below



Logit Regression Results

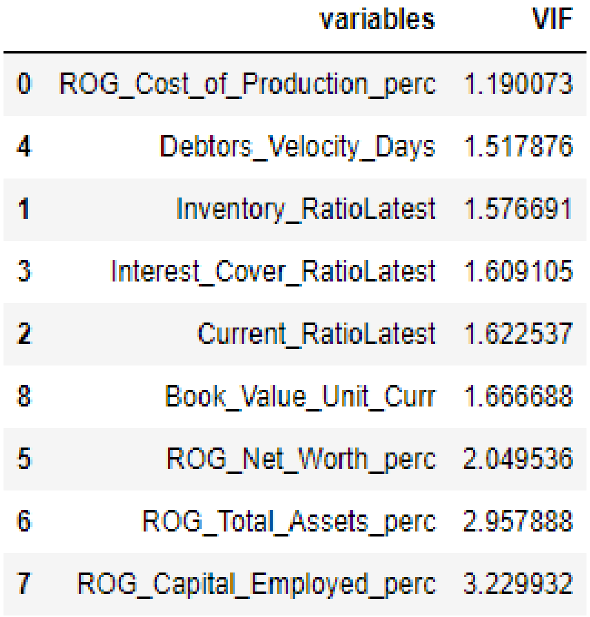
| | | | |
|---|---|---|---|
| Dep. Variable: | default | No. Observations: | 2401 |
| Model: | Logit | Df Residuals: | 2391 |
| Method: | MLE | Df Model: | 9 |
| Date: | Fri, 13 May 2022 | Pseudo R-squ.: | 0.6324 |
| Time: | 15:32:05 | Log-Likelihood: | -302.64 |
| converged: | True | LL-Null: | -823.35 |
| Covariance Type: | nonrobust | LLR p-value: | 2.003e-218 |

| | coef | std err | z | P>\|z\| | [0.025 | 0.975] |
|---|---|---|---|---|---|---|
| Intercept | -0.0990 | 0.185 | -0.536 | 0.592 | -0.461 | 0.263 |
| ROG_Cost_of_Production_perc | -0.0085 | 0.003 | -2.752 | 0.006 | -0.015 | -0.002 |
| Inventory_RatioLatest | -0.0393 | 0.017 | -2.272 | 0.023 | -0.073 | -0.005 |
| Current_RatioLatest | -0.6238 | 0.090 | -6.945 | 0.000 | -0.800 | -0.448 |
| Interest_Cover_RatioLatest | -0.1147 | 0.040 | -2.863 | 0.004 | -0.193 | -0.036 |
| Debtors_Velocity_Days | -0.0026 | 0.001 | -2.177 | 0.029 | -0.005 | -0.000 |
| ROG_Net_Worth_perc | -0.0384 | 0.009 | -4.150 | 0.000 | -0.056 | -0.020 |
| ROG_Total_Assets_perc | -0.0224 | 0.009 | -2.434 | 0.015 | -0.040 | -0.004 |
| ROG_Capital_Employed_perc | 0.0202 | 0.009 | 2.245 | 0.025 | 0.003 | 0.038 |
| Book_Value_Unit_Curr | -0.1108 | 0.010 | -11.654 | 0.000 | -0.129 | -0.092 |

*Figure 15. Model summary for specific significant variables*

We can see that all the variables p-value < 0.05 & the variables are significant & may be useful to discriminate cases of default.

Let us also check the multi-collinearity of the model using Variance Inflation Factor (VIF) for the predictor variables



| | variables | VIF |
|---|---|---|
| 0 | ROG_Cost_of_Production_perc | 1.190073 |
| 4 | Debtors_Velocity_Days | 1.517876 |
| 1 | Inventory_RatioLatest | 1.576691 |
| 3 | Interest_Cover_RatioLatest | 1.609105 |
| 2 | Current_RatioLatest | 1.622537 |
| 8 | Book_Value_Unit_Curr | 1.666688 |
| 5 | ROG_Net_Worth_perc | 2.049536 |
| 6 | ROG_Total_Assets_perc | 2.957888 |
| 7 | ROG_Capital_Employed_perc | 3.229932 |

*Figure 16. VIF Table for Significant variables*

We can see that multi-collinearity still exists but let’s not drop them as VIFs are not very high.

Let us look at the adjusted pseudo R-square value

| | |
|---|---|
| R – Square | 0.6324 |
| Adj. R-Square | 0.6215 |

We see that adjusted R square is now close to R square, thus suggesting lesser insignificant variables present 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. Before making the prediction, let’s decide the optimum cutoff value.

## Optimum Cutoff

Let’s plot the graph between ‘default’ & ‘predicted train default’ data.

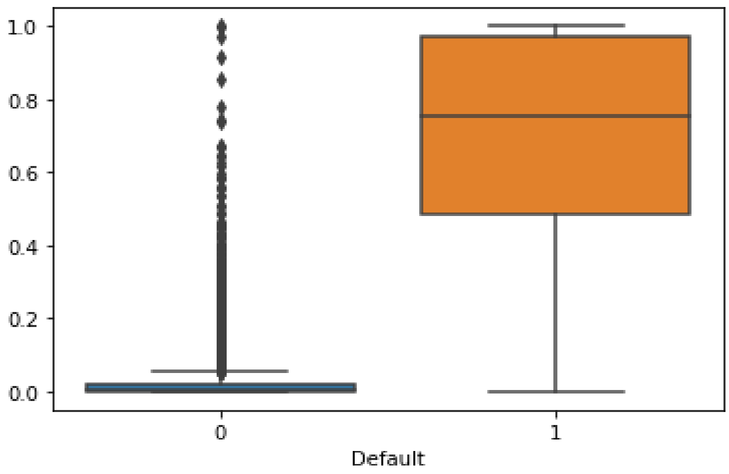


*Figure 17. Default Vs Predicted Train default Boxplot*

From the above boxplot, we need to decide on one such value of a cut-off which will give us the most reasonable descriptive power of the model.

Let us take a cut-off of 0.06 and check.

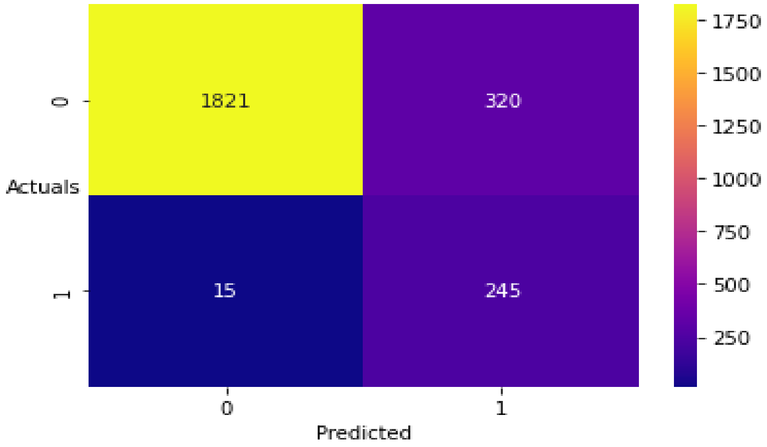
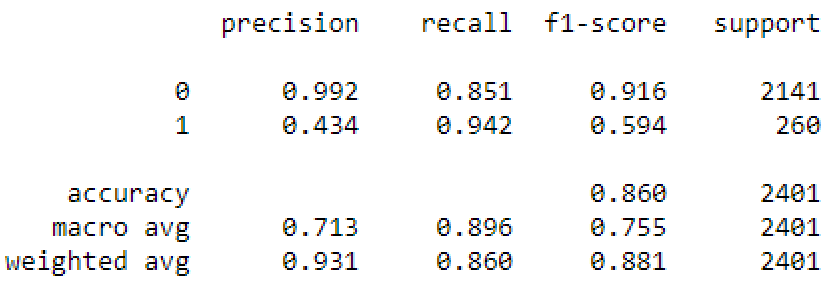


*Figure 18. Confusion Matrix for Training set with 0.06 cut-off*

- True Negative: 1821 False Positives: 320
- False Negatives: 15 True Positives: 245

Let us now go ahead and print the classification report of training data to check the various other parameters using the 0.06 as the cut-off



```
              precision    recall  f1-score   support

           0      0.992     0.851     0.916      2141
           1      0.434     0.942     0.594       260

    accuracy                          0.860      2401
   macro avg      0.713     0.896     0.755      2401
weighted avg      0.931     0.860     0.881      2401
```

*Figure 19.Classification Report for Training Data with 0.06 cut-off*

## 7. Validate the Model on Test Dataset and state the performance matrices. Also state interpretation from the model

For the testing set using the 0.06 as the cutoff

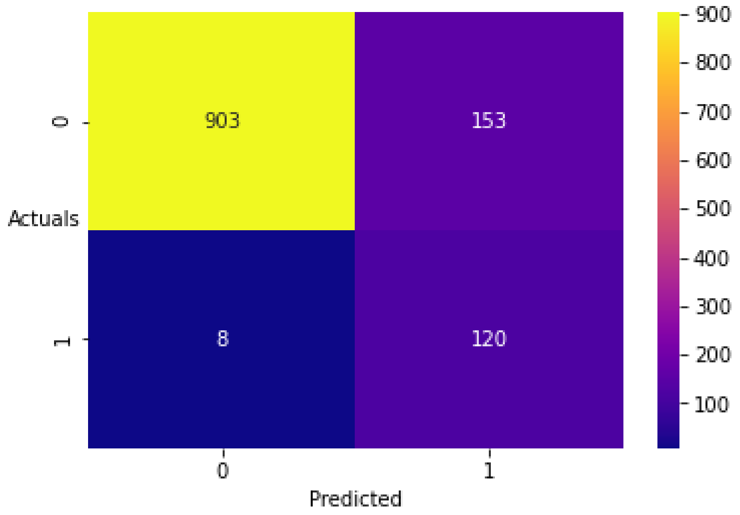
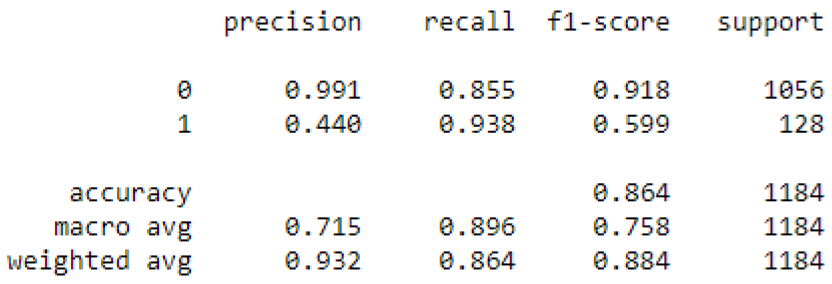


*Figure 20. Confusion Matrix for Testing Set with 0.06 cut-off*

- True Negative: 903 False Positives: 153
- False Negatives: 8 True Positives: 120

Let us now go ahead and print the classification report for test record to check the various other parameters using 0.06 as the cut-off



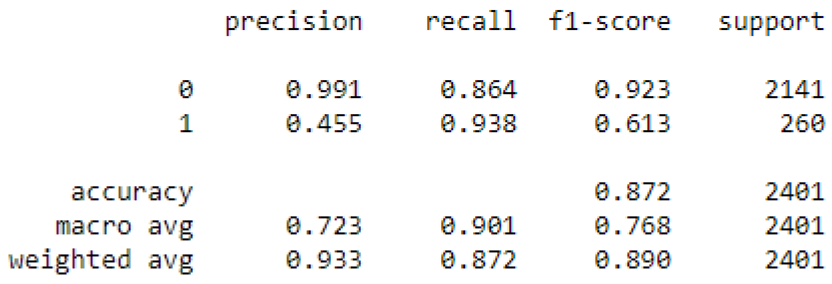
```
              precision    recall  f1-score   support

           0      0.991     0.855     0.918      1056
           1      0.440     0.938     0.599       128

    accuracy                          0.864      1184
   macro avg      0.715     0.896     0.758      1184
weighted avg      0.932     0.864     0.884      1184
```

*Figure 21. Classification Report for Test Data with 0.06 cut-off*

- As observed above, accuracy of the model.  i.e. %overall correct predictions is 86%
- Sensitivity of the model is 94% i.e. 94% of those defaulted were correctly identified as defaulters by the model

**Let us take a cut-off of 0.07 and check if our predictions have improved**

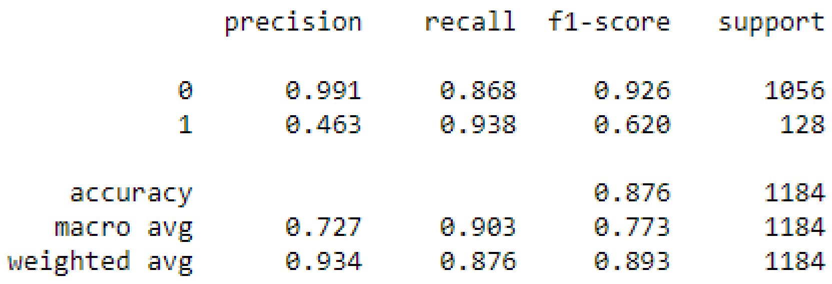
**Training Set**



```
              precision    recall  f1-score   support

           0      0.991     0.864     0.923      2141
           1      0.455     0.938     0.613       260

    accuracy                          0.872      2401
   macro avg      0.723     0.901     0.768      2401
weighted avg      0.933     0.872     0.890      2401
```

*Figure 22. Classification Report for Training Data with 0.07 cut-off*

**Testing Set**



```
              precision    recall  f1-score   support

           0      0.991     0.868     0.926      1056
           1      0.463     0.938     0.620       128

    accuracy                          0.876      1184
   macro avg      0.727     0.903     0.773      1184
weighted avg      0.934     0.876     0.893      1184
```

*Figure 23. Classification Report for Test Data with 0.07 cut-off*

| Threshold Value | 0.06 As the cut-off for prediction (%) | | 0.07 As the cut-off for prediction (%) | |
|---|---|---|---|---|
| | Accuracy | Recall | Accuracy | Recall |
| Training Record | 86 | 94.2 | **87.2** | **93.8** |
| Testing Record | 86.4 | 93.8 | **87.6** | **93.8** |

When we choose 0.07 as the cut-off, our model's accuracy of the model i.e. %overall correct predictions has increased from 86% to 87% and the sensitivity of the model becomes constant for both training & testing record as 93.8%. **So, 0.07 is the optimum cut-off for our model.**

**Performance Metrics for 0.07 cut-off**

When we need to check or visualize the performance of the classification problem, we use the AUC **(Area Under The Curve**) ROC (**Receiver Operating Characteristics**) curve. It is one of the most important evaluation metrics for checking any classification model's performance. AUC - ROC curve is a performance measurement for the classification problems at various threshold settings. ROC is a probability curve and AUC represents the degree or measure of separability. It tells how much the model is capable of distinguishing between classes. Higher the AUC, the better the model is at predicting 0 classes as 0 and 1 classes as 1. Always, aim for high AUC score. We calculate Accuracy, confusion matrix, ROC Curve, AUC score and classification report for both training and test set for all the classification models built.

**ROC Curve for Training Record**

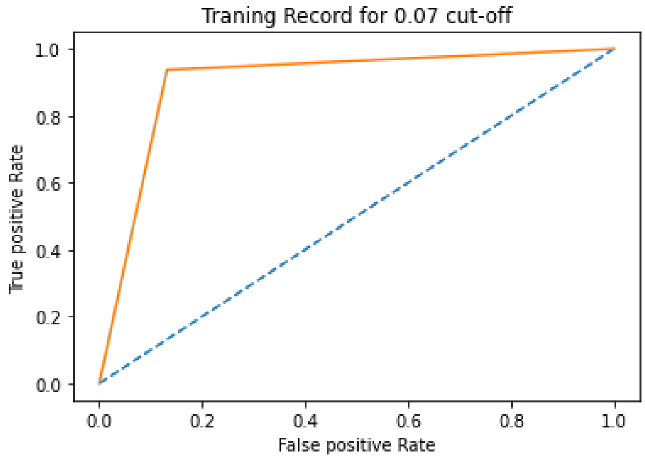


*Figure 24. ROC Curve for Training Record with 0.07 cut-off*

AUC Score for training record is 90.1%

**ROC Curve for Testing Record**

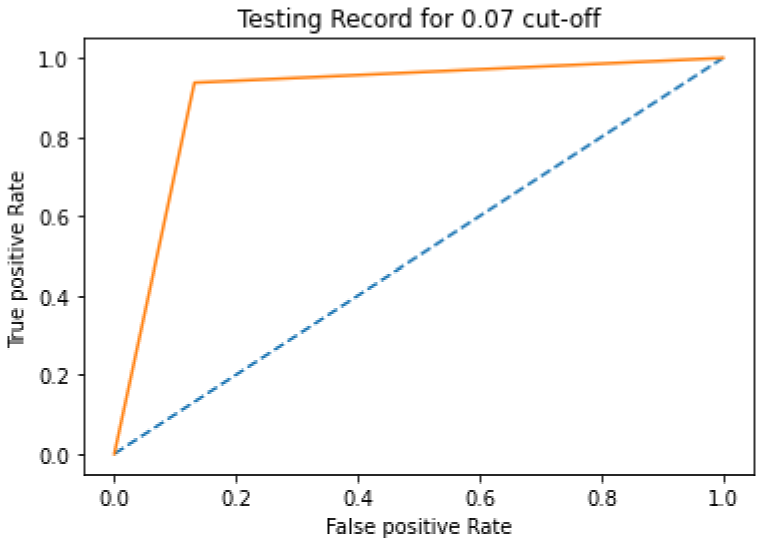


*Figure 25. ROC Curve for Testing Record with 0.07 cut-off*

AUC Record for testing record is 90.3%.

# THE END!