FINANCIAL RISK ANALYTICS BUSINESS REPORT

THAKUR ARUN SINGH

DECEMBER 2021

This Business Report shall provide detailed explanation of how we approached each problem given in the assignment. It shall also provide relative resolution and explanation with regards to the problems

CONTENTS

Problem 1:

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 Networth of the company in the following year (2016) is provided which can be used to drive the labeled field.

Explanation of data fields available in Data Dictionary, 'Credit Default Data Dictionary.xlsx'

Exploratory data analysis

Dataset has 67 variables of which 63 are of float data type, 3 are integer type and 1 is object type.

The head of the dataset is as below:

	Co_Code	Co_Name	Networth Next Year	Equity Paid Up	Networth	Capital Employed	Total Debt	Gross Block	Net Working Capital	Current Assets	...	PBIDTM (%) [Latest]	PBITM (%) [Latest]	PBDTM (%) [Latest]	CPM (%) [Latest]	APATM (%) [Latest]	Debt Velo (Da
0	16974	Hind.Cables	-8021.60	419.36	-7027.48	-1007.24	5936.03	474.30	-1076.34	40.50	...	0.00	0.00	0.00	0.00	0.00	
1	21214	Tata Tele. Mah.	-3986.19	1954.93	-2968.08	4458.20	7410.18	9070.86	-1098.88	486.86	...	-10.30	-39.74	-57.74	-57.74	-87.18	
2	14852	ABG Shipyard	-3192.58	53.84	506.86	7714.68	6944.54	1281.54	4496.25	9097.64	...	-5279.14	-5516.98	-7780.25	-7723.67	-7961.51	
3	2439	GTL	-3054.51	157.30	-623.49	2353.88	2326.05	1033.69	-2612.42	1034.12	...	-3.33	-7.21	-48.13	-47.70	-51.58	
4	23505	Bharati Defence	-2967.36	50.30	-1070.83	4675.33	5740.90	1084.20	1836.23	4685.81	...	-295.55	-400.55	-845.88	379.79	274.79	3

5 rows × 67 columns

The data has 3586 Rows and 67 Columns. No duplicate data is present in the data set.

```
df.isna().sum().sum()
```

```
0
```

The data has 3586 Rows and 67 Columns. No duplicate data is present in the data set.

We dropped unrequired columns like Co_Code and Co_Name since they do not add value to the analysis.

Descriptive statistics / 5 point summary is shown below.

	Co_Code	Networth Next Year	Equity Paid Up	Networth	Capital Employed	Total Debt	Gross Block	Net Working Capital	Current Assets	Current Liabilities and Provisions	...
count	3586.000000	3586.000000	3586.000000	3586.000000	3586.000000	3586.000000	3586.000000	3586.000000	3586.000000	3586.000000	...
mean	16065.388734	725.045251	62.966584	649.746299	2799.611054	1994.823779	594.178829	410.809665	1960.349172	391.992078	...
std	19776.817379	4769.681004	778.761744	4091.988792	26975.135385	23652.842746	4871.547802	6301.218546	22577.570829	2675.001631	...
min	4.000000	-8021.600000	0.000000	-7027.480000	-1824.750000	-0.720000	-41.190000	-13162.420000	-0.910000	-0.230000	...
25%	3029.250000	3.985000	3.750000	3.892500	7.602500	0.030000	0.570000	0.942500	4.000000	0.732500	...
50%	6077.500000	19.015000	8.290000	18.580000	39.090000	7.490000	15.870000	10.145000	24.540000	9.225000	...
75%	24269.500000	123.802500	19.517500	117.297500	226.605000	72.350000	131.895000	61.175000	135.277500	65.650000	...
max	72493.000000	111729.100000	42263.460000	81657.350000	714001.250000	652823.810000	128477.590000	223257.560000	721166.000000	83232.980000	...

8 rows × 66 columns

Current iabilities and rovisions	...	PBIDTM (%) [Latest]	PBITM (%) [Latest]	PBDTM (%) [Latest]	CPM (%) [Latest]	APATM (%) [Latest]	Debtors Velocity (Days)	Creditors Velocity (Days)	Inventory Velocity (Days)	Value of Output/Total Assets	Value c Output/Gros Bloc
6.000000	...	3585.000000	3585.000000	3585.000000	3585.000000	3585.000000	3586.000000	3.586000e+03	3483.000000	3586.000000	3586.00000
1.992078	...	-51.162890	-109.213414	-311.570357	-307.005632	-365.056187	603.894032	2.057855e+03	79.644559	0.819757	61.88454
5.001631	...	1795.131025	3057.635870	10921.592639	10676.149629	12500.051387	10636.759580	5.416948e+04	137.847792	1.201400	976.82435
-0.230000	...	-78870.450000	-141600.000000	-590500.000000	-572000.000000	-688600.000000	0.000000	0.000000e+00	-199.000000	-0.330000	-61.00000
0.732500	...	0.000000	0.000000	0.000000	0.000000	0.000000	8.000000	8.000000e+00	0.000000	0.070000	0.27000
9.225000	...	8.070000	5.230000	4.690000	3.890000	1.590000	49.000000	3.900000e+01	35.000000	0.480000	1.53000
5.650000	...	18.990000	14.290000	14.110000	11.390000	7.410000	106.000000	8.900000e+01	96.000000	1.160000	4.91000
2.980000	...	19233.330000	19195.700000	15640.000000	15640.000000	15266.670000	514721.000000	2.034145e+06	996.000000	17.630000	43404.00000

The values of mean, standard deviation, minimum and maximum, 25th, 50th and 75th percentile are mentioned in the above tables.

Next we checked for null values.

```
Inventory Velocity (Days)           103
Book Value (Adj.) (Unit Curr)         4
Inventory Ratio[Latest]               1
Interest Cover Ratio[Latest]          1
Current Ratio[Latest]                 1
                                   ...
Value of Output/Total Assets          0
Cash Flow From Operating Activities   0
CEPS (annualised) (Unit Curr)         0
Market Capitalisation                 0
Co_Code                               0
Length: 67, dtype: int64
```

Further details on missing values is covered under 1.2

PROBLEM 1.1

Outlier Treatment.

Resolution:

Describing the data:

- First we import all the necessary libraries in Python, and then import the data file which is ‘Company_Data2015-1’. Once we import the file we confirm whether the data has been uploaded correctly or not using ‘head’ function. Using this function we can view the data and all the columns and headers whether they are aligning correctly or not.

- Then using the ‘shape’ function we can understand how many row and columns are there in our data set.

- To check the data type of all the columns and also to check the null values, ‘info’ function. Has been used.

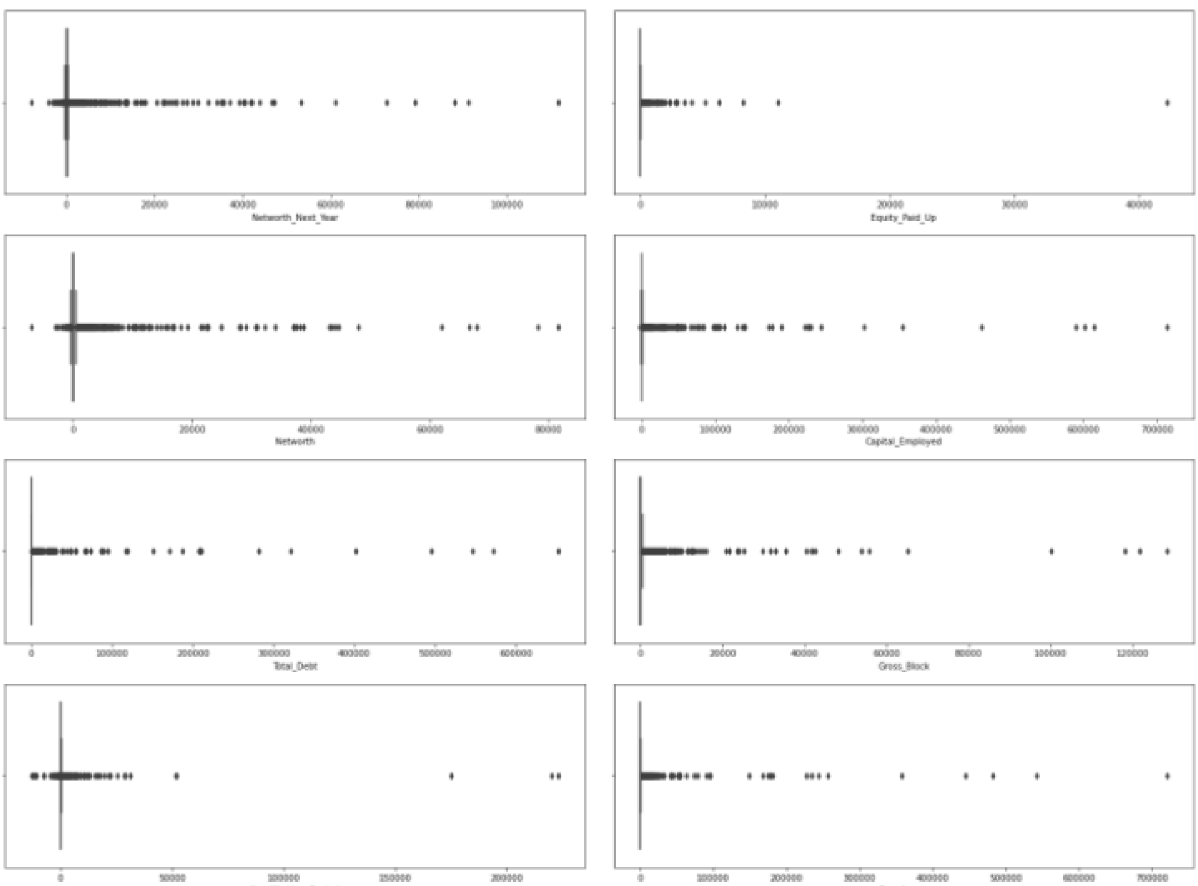
- To see the detail description of the data such as, Count, Mean, Median, Min, Max, Standard Deviations etc,

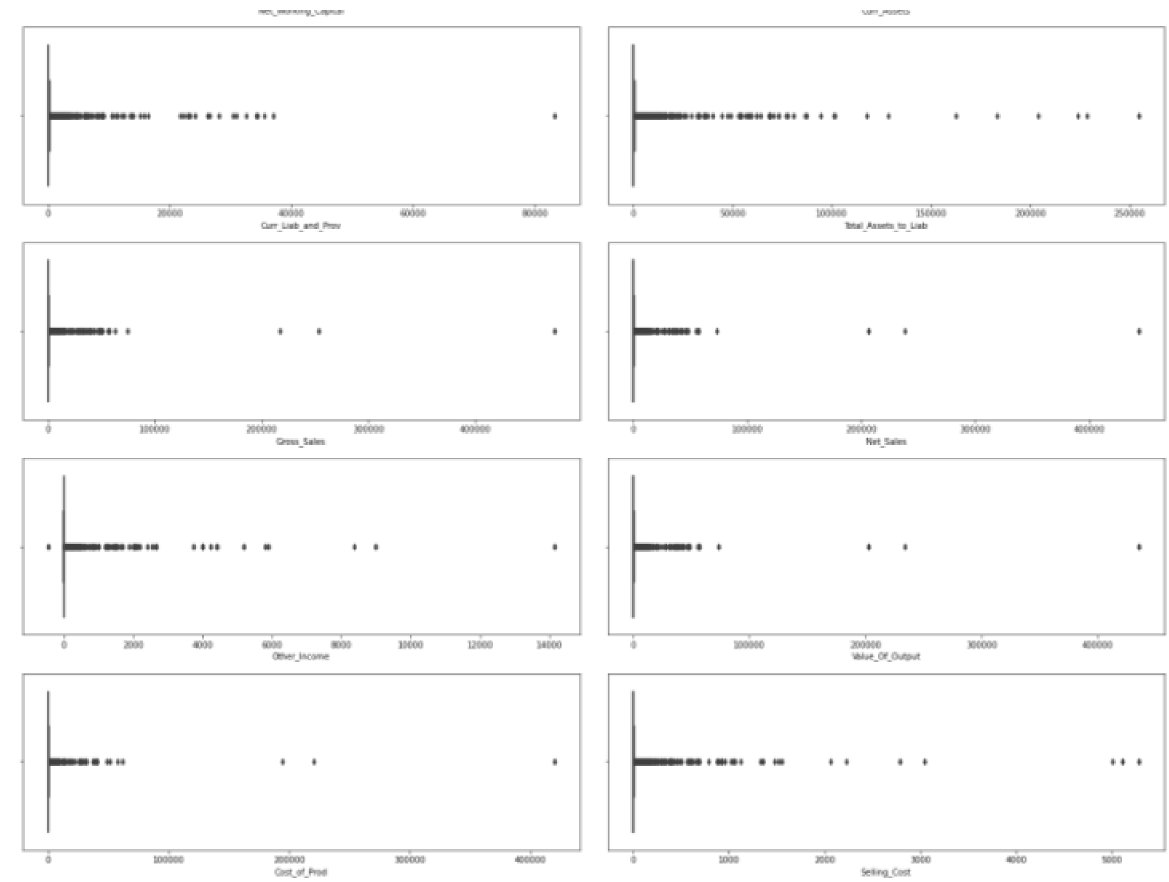
- Using the ‘isnull’ function, one can understand if there are any null values in the data set. And we do not have any null values in the existing data set.

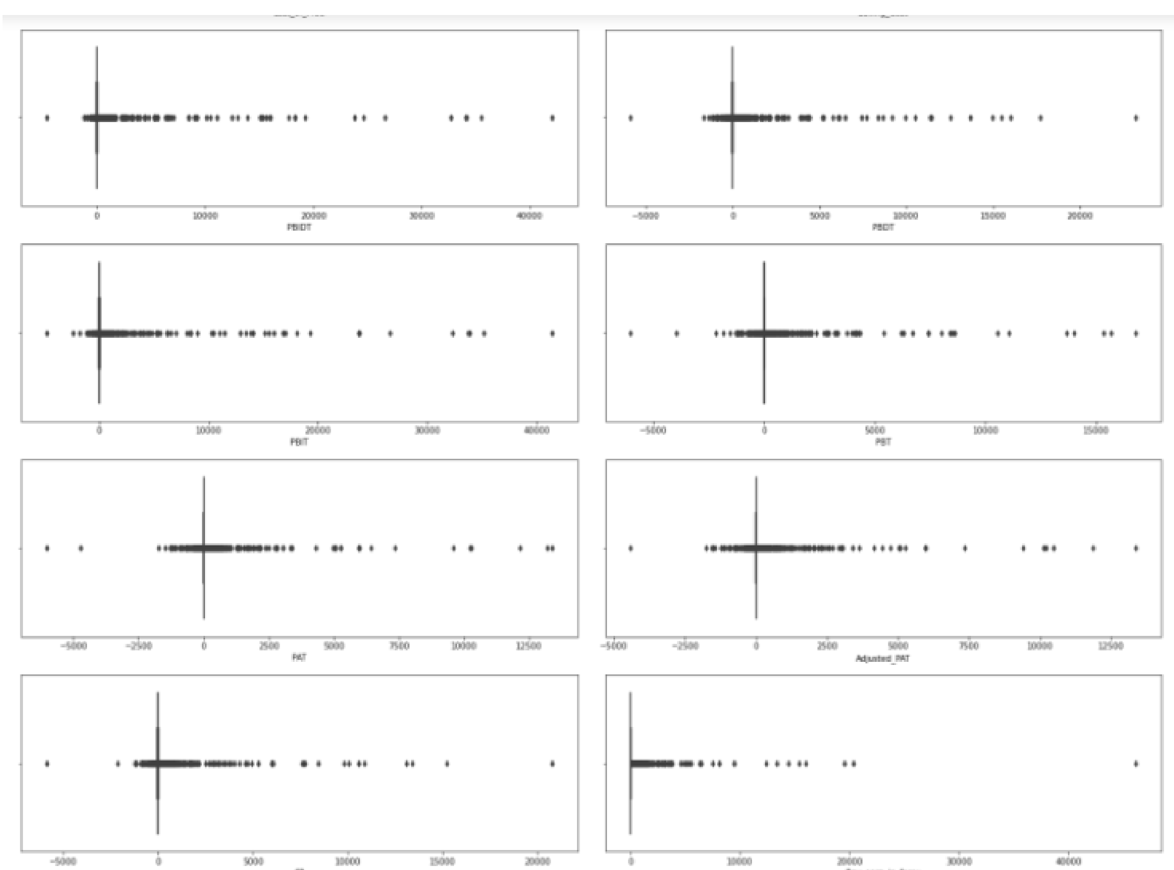
- Using the ‘dups’ function we check for the duplicates and there were no duplicate values.

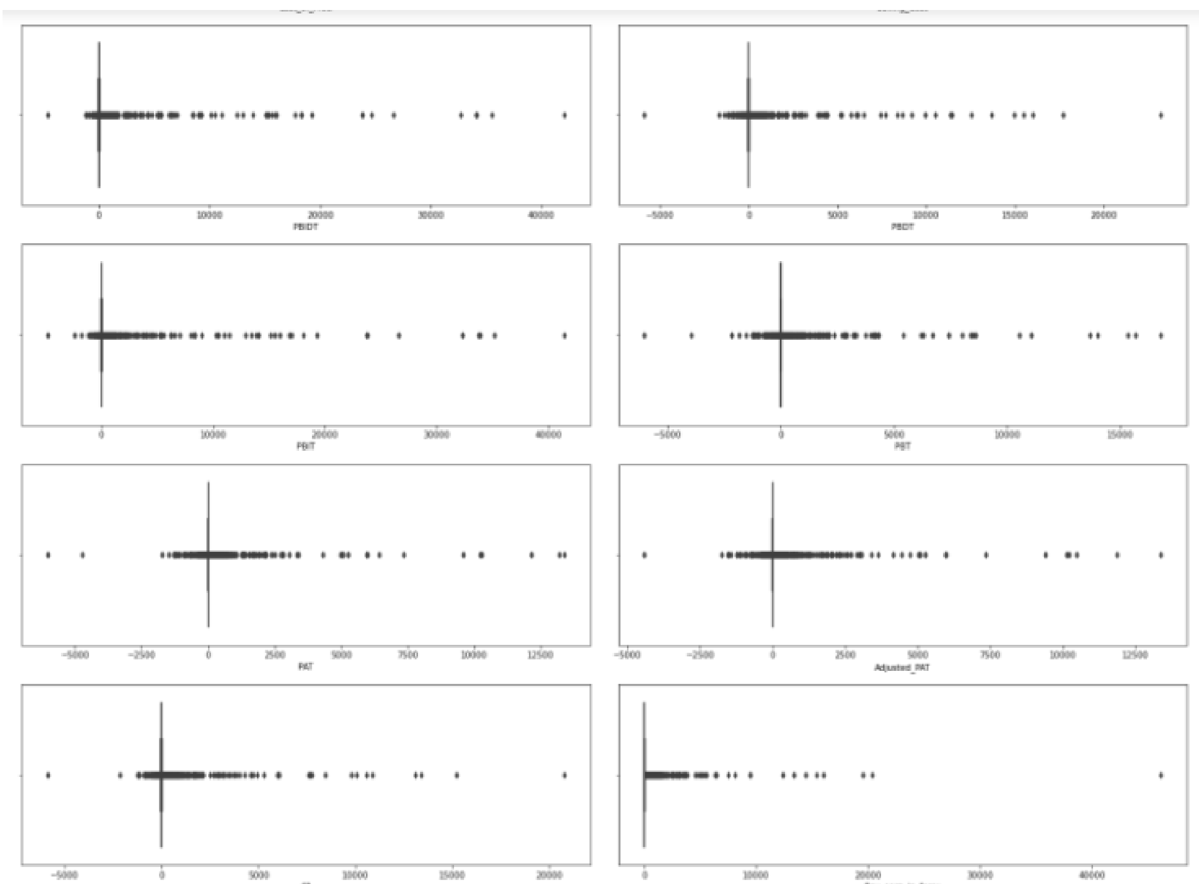
- We also identified the unique values in categorical data.

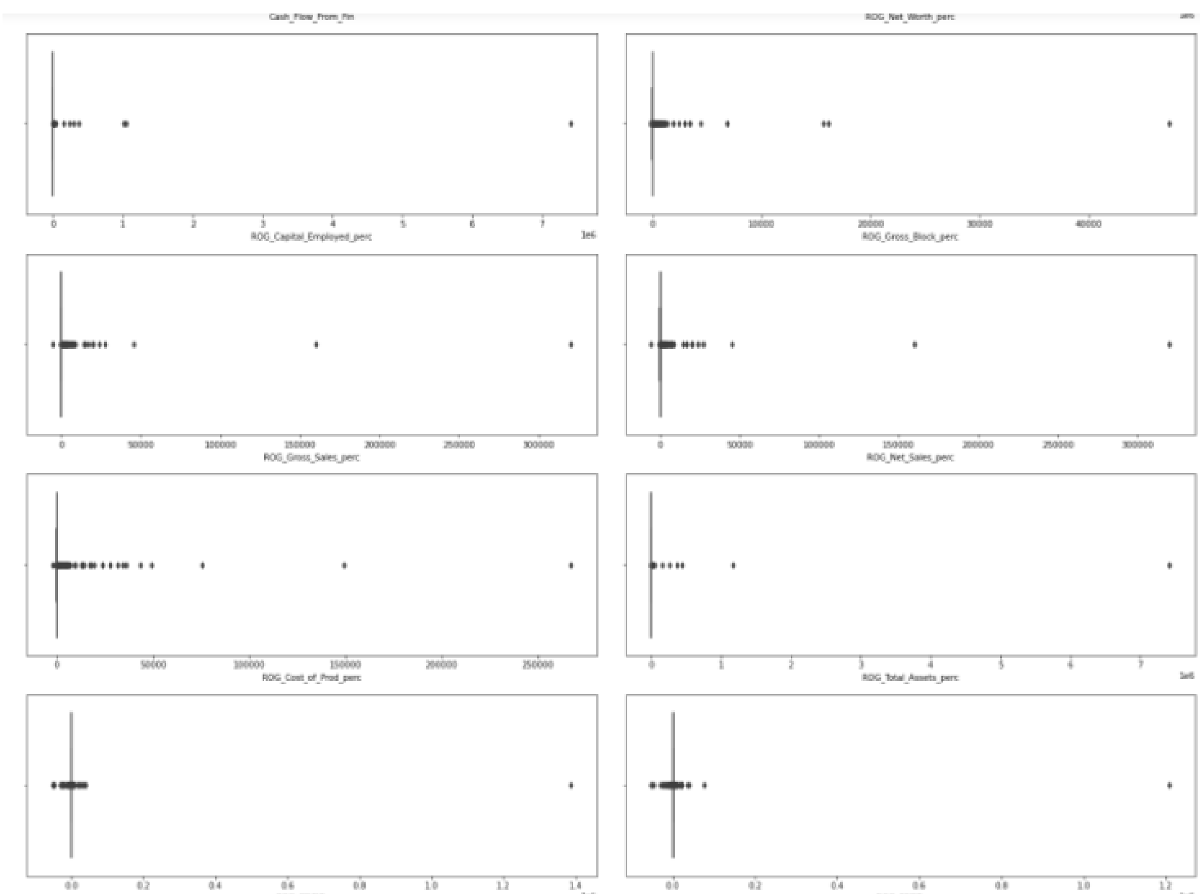
We used 3 times the IQR range as the criteria to determine the outliers. Our analysis gave significant chunk of outliers in the data. Below are boxplots which were plotted to analyze this data.

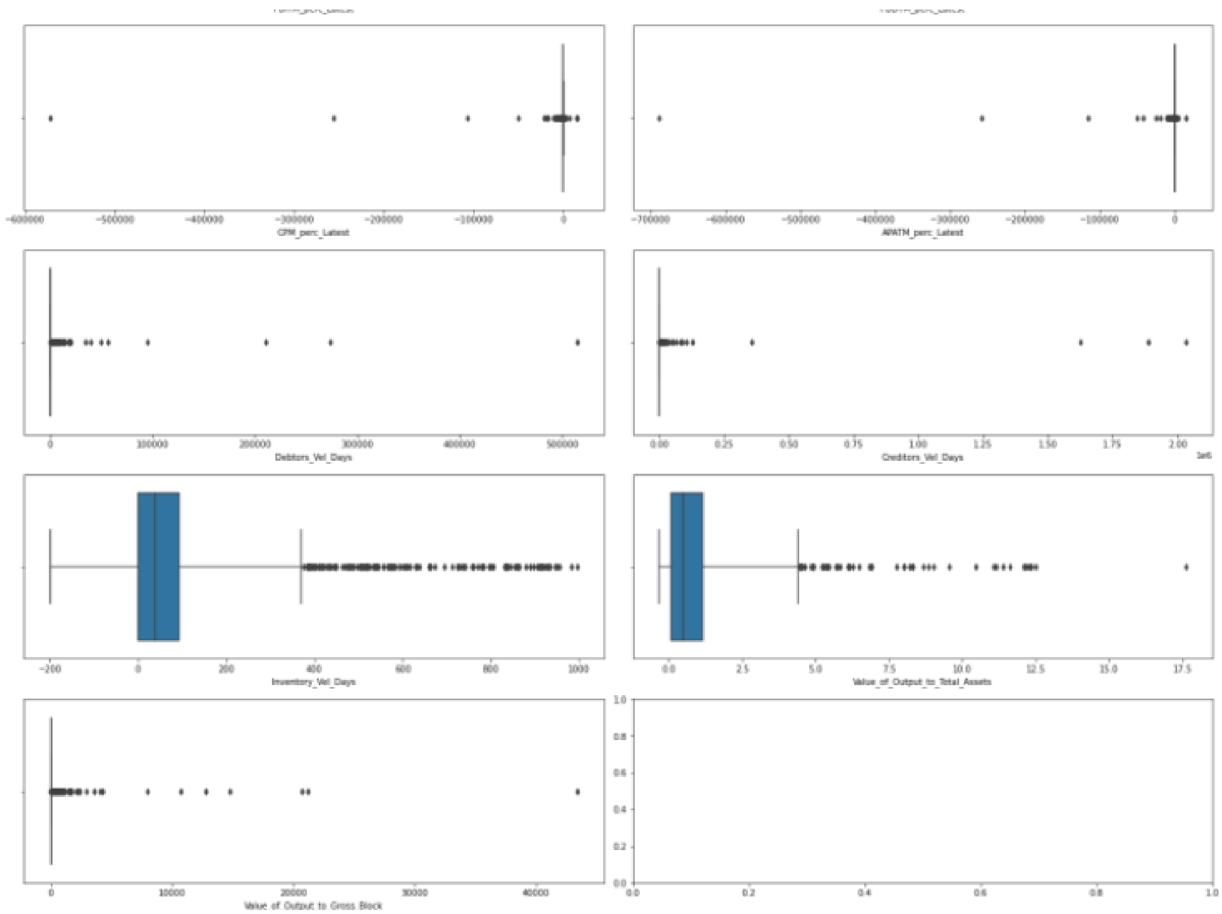



Curr_Liab_and_Prov
Total_Assets_to_Liab
Gross_Sales
Net_Sales
Other_Income
Value_Of_Output
Cost_of_Prod
Selling_Cost


PBIDT
PBDT
PBIT
PBT
PAT
Adjusted_PAT


PBIDT
PBDT
PBIT
PBT
PAT
Adjusted_PAT


Cash_Flow_From_Fin
ROG_Net_Worth_perc
ROG_Capital_Employed_perc
ROG_Gross_Block_perc
ROG_Gross_Sales_perc
ROG_Net_Sales_perc
ROG_Cost_of_Prod_perc
ROG_Total_Assets_perc
ROG_PBIDT_perc
ROG_PBDT_perc



OUTLIER TREATMENT

Significant number of outliers were present for almost all the variables. We captured the actual percentage of data which was above and below the third and first quintiles respectively.

Data above third quintile.

ROG_Rev_exp_in_forex_perc	22.926557
Capital_exp_in_forex	19.380061
ROG_Rev_earn_in_forex_perc	17.900028
Rev_earn_in_forex	17.648701
Rev_exp_in_forex	16.727171
PAT	14.884111
Market_Capitalisation	14.744485
PBT	14.688634
Adjusted_PAT	14.632784
CP	14.325607
PBDT	14.074281
PBIT	13.683329
PBIDT	13.655404
Selling_Cost	13.292376
Other_Income	13.236526
Cash_Flow_From_Opr	13.068975
Networth_Next_Year	13.041050
Total_Debt	12.901424
Networth	12.566322
Capital_Employed	12.538397
Curr_Ratio_Latest	12.510472
Curr_Liab_and_Prov	12.259145
Curr_Assets	12.007819
Total_Assets_to_Liab	11.979894
Value_Of_Output	11.616867
Gross_Sales	11.616867

Data below first quintile.

ROG_Rev_exp_in_forex_perc	22.172577
ROG_Rev_earn_in_forex_perc	18.877409
Cash_Flow_From_Inv	14.800335
Cash_Flow_From_Fin	13.739179
APATM_perc_Latest	11.309690
CPM_perc_Latest	8.209997
PBDTM_perc_Latest	7.595644
PBITM_perc_Latest	7.176766
ROG_Gross_Block_perc	7.092991
Adjusted_PAT	6.785814
PAT	6.646188
ROG_Net_Worth_perc	6.450712
ROG_PBT_perc	6.087685
PBT	6.059760
ROG_PAT_perc	5.836359
ROG_CP_perc	5.585032
ROG_PBDT_perc	5.473331
PBIDTM_perc_Latest	5.110304
ROG_PBIT_perc	4.691427
ROG_PBIDT_perc	4.300475
Interest_Cover_Ratio_Latest	3.797822
PBDT	3.769897
CP	3.630271
Cash_Flow_From_Opr	3.546495
ROG_Capital_Employed_perc	2.624965
ROG_Gross_Sales_perc	2.261938

Since the number of outliers are too large in number to be treated, as treated such large number of records would mean changing the essence of the data. Also given the fact that this is a financial data and the outliers might very well reflect the information which is genuine in nature. Since there is data captured for small, medium as well as large companies.

Hence we decided against treating the outliers in this data set.

PROBLEM 1.2

Missing Value Treatment

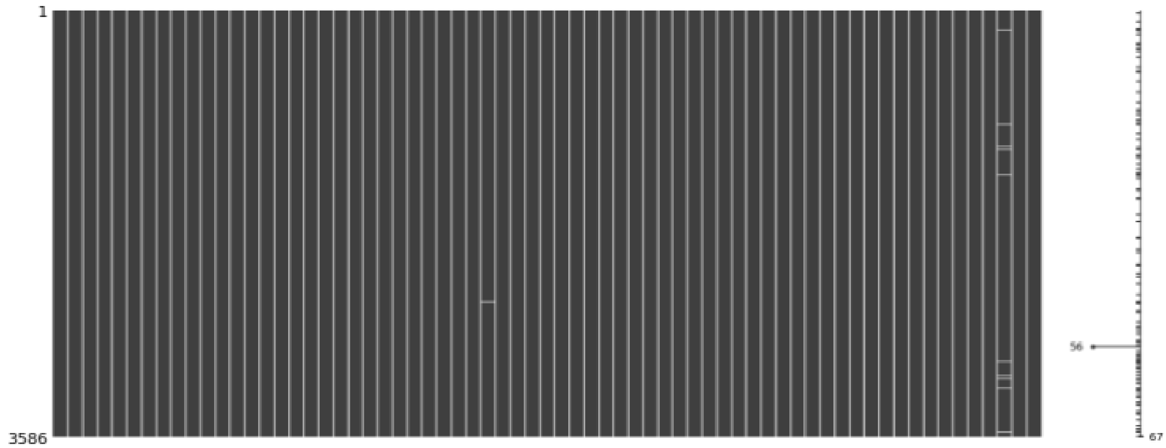
Resolution:

Given the size of the data set i.e. 3586 rows, there were not many missing values to start with. There were a total of 118 missing records observed in the entire data.

Snapshot from missingno library has been published below for reference.

```
In [14]: import missingno as msno
         msno.matrix(df)

Out[14]: <AxesSubplot:>
```



Null values were present in many columns, however significant number was present in "Inventory_Vel_Days" column. This is the one which we treated.

Records with missing value in "Inventory_Vel_Days" column were imputed with the average value.

After this imputation, there were another 15 rows with missing data, however this number was too small to warrant any additional efforts. Hence we dropped these rows the purpose of the analysis.

No more missing values were present after treatment.

```
df.isna().sum().sum()

0
```

PROBLEM 1.3

Transform Target variable into 0 and 1

Resolution:

A new dependent variable named "Default" was created based on the criteria given in the project notes.

<u>Criteria</u> -

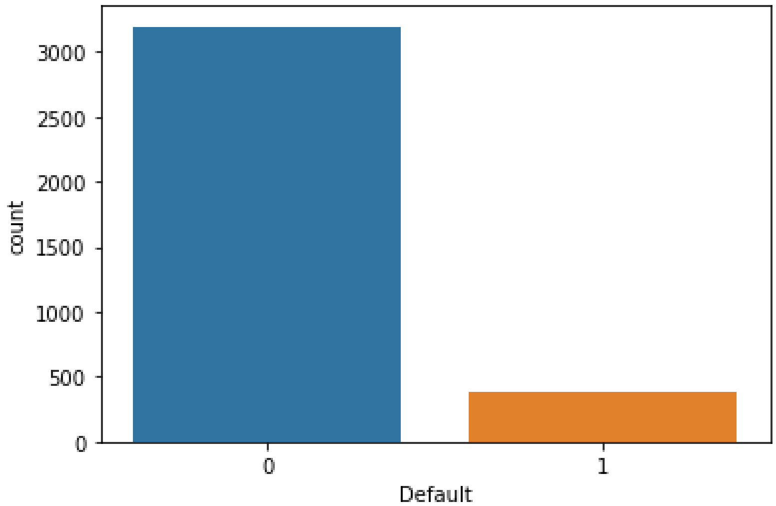
1 - If the Net Worth Next Year is negative for the company 0 - If the Net Worth Next Year is positive for the company

Making use of np.where function to achieve this.

Making the dependent Variable

In [48]:
```
df['Default']=np.where(df['Networth_Next_Year']<0,1,0)
```

After generating the dependent column, we checked for the split of data based on this dependent variable. Below is a bar plot showing the same.



In [57]:
```
df_new['Default'].value_counts()
```

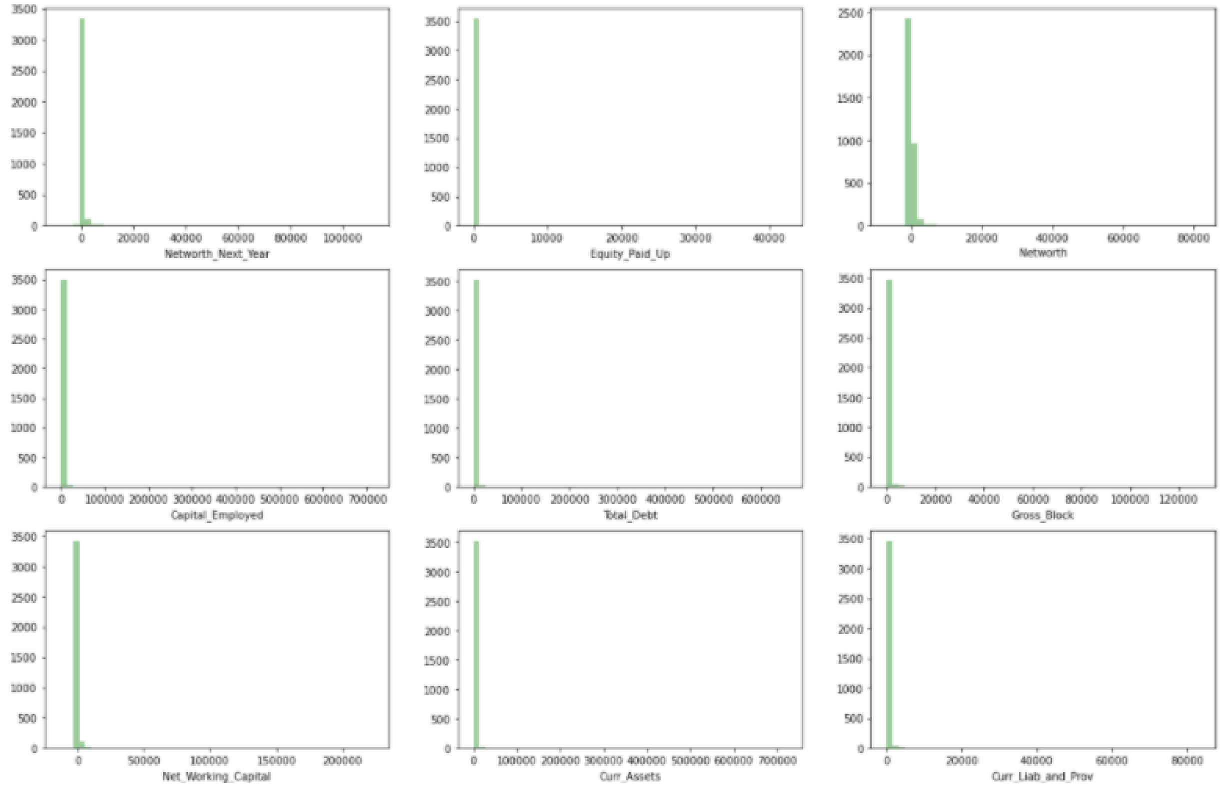
Out[57]:
```
0    3195
1     386
Name: Default, dtype: int64
```

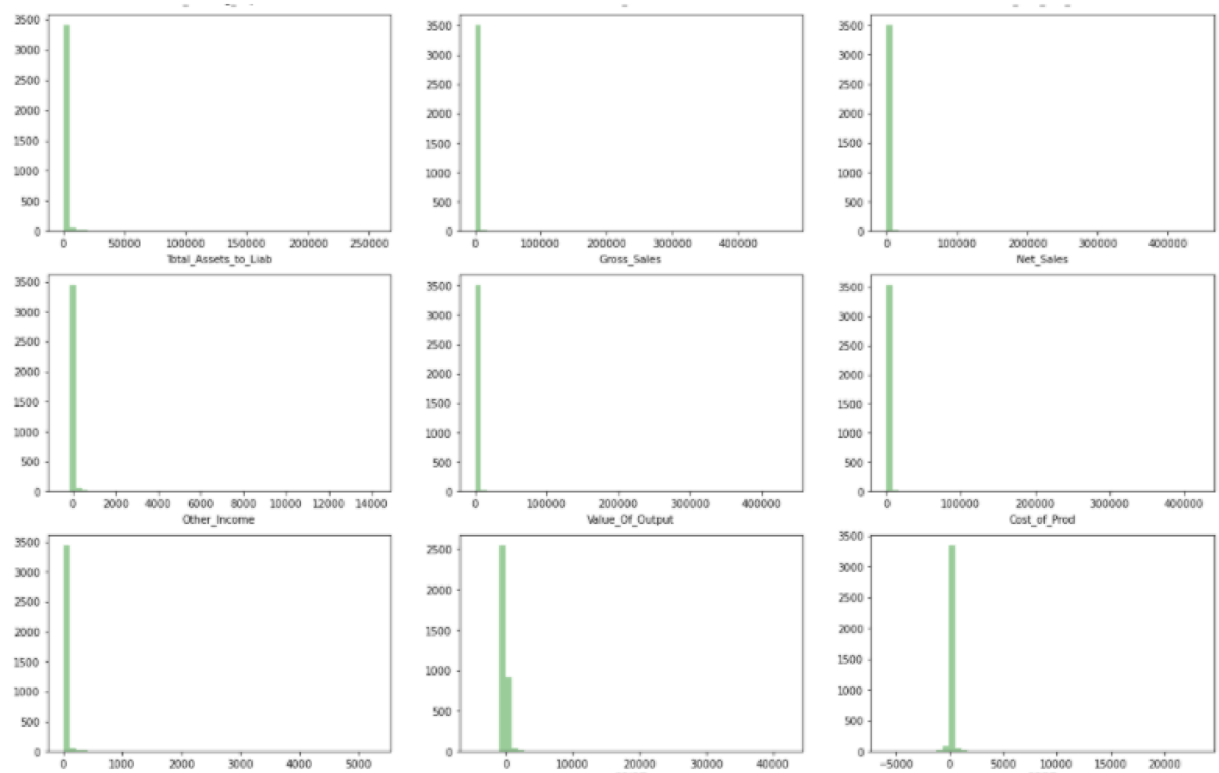
PROBLEM 1.4

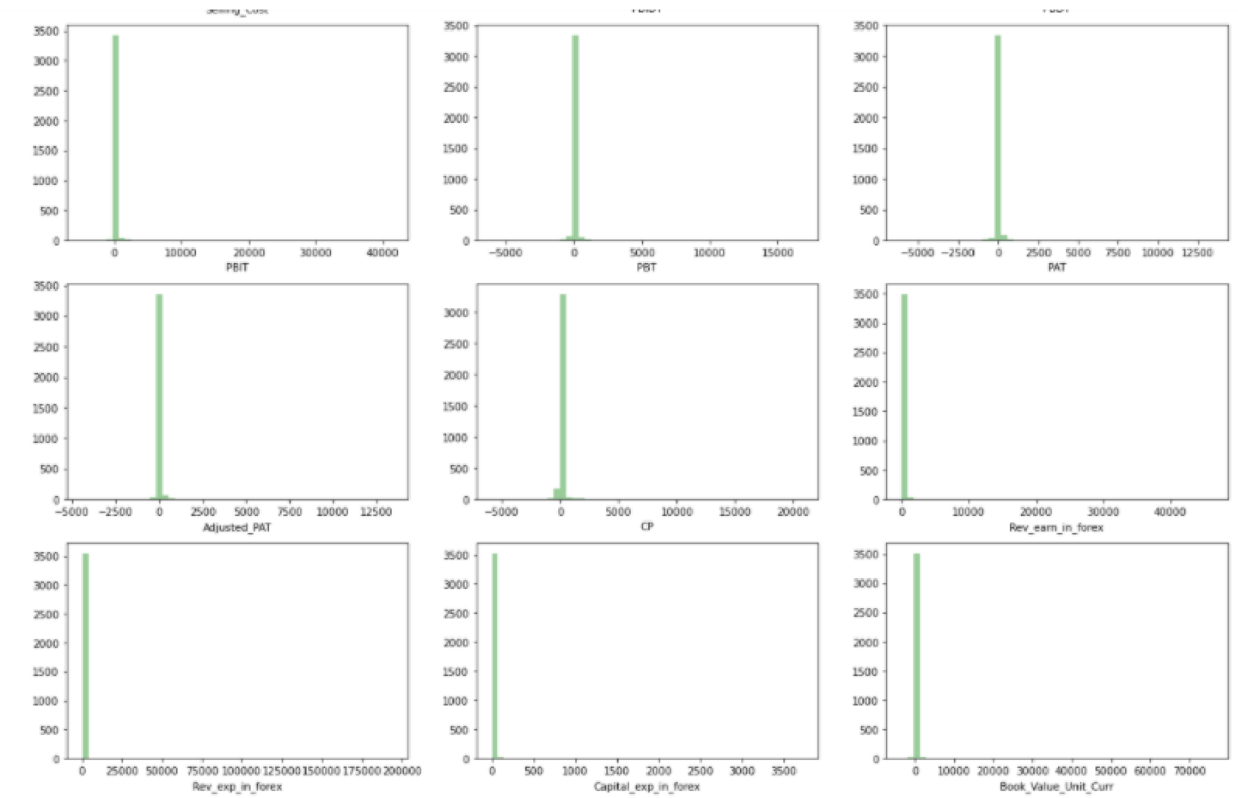
Univariate & Bivariate analysis with proper interpretation.

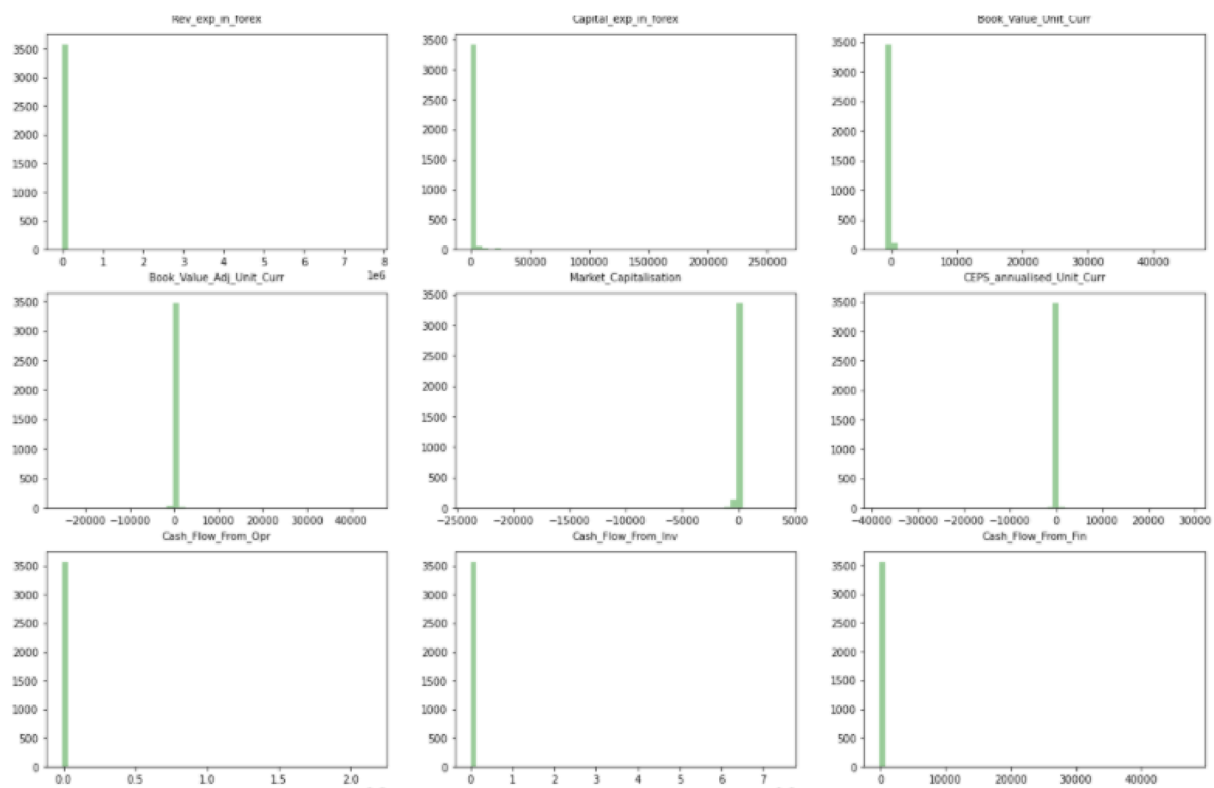
Resolution:

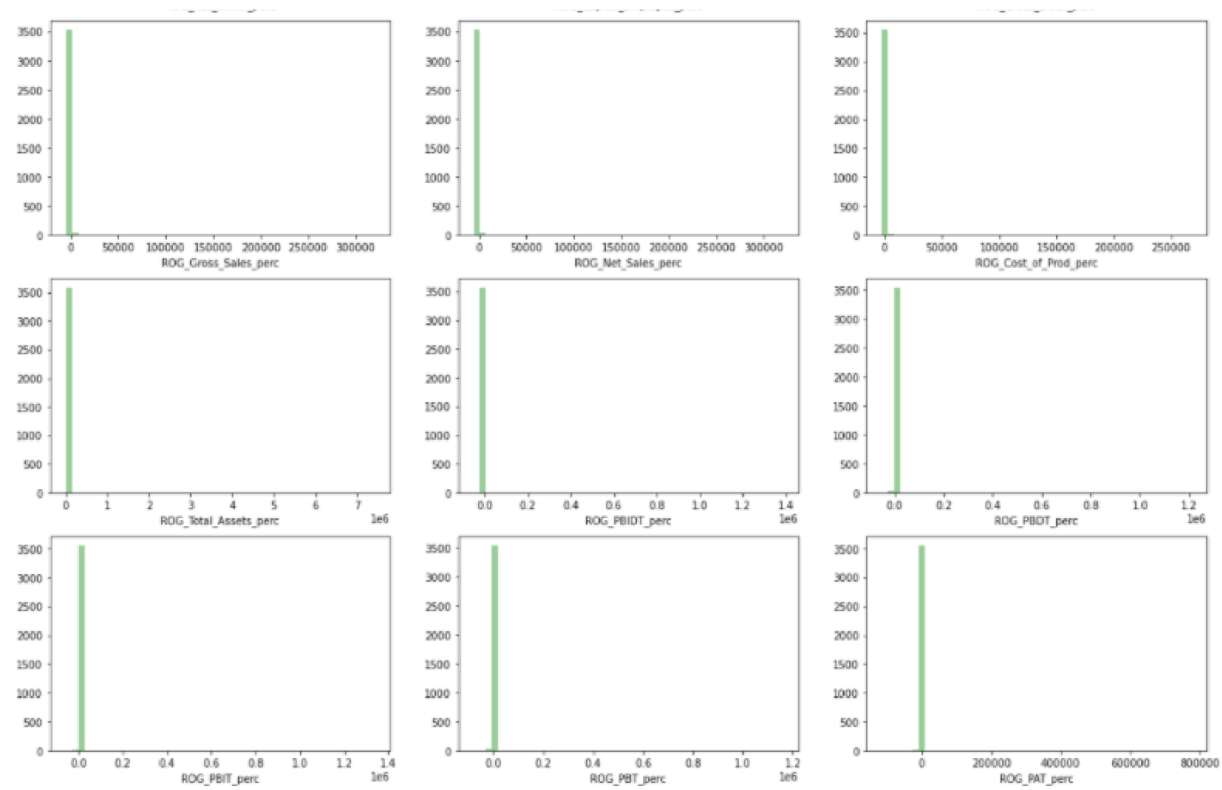
Distplot were plotted for all the variables to analyze the distribution of all the variables.

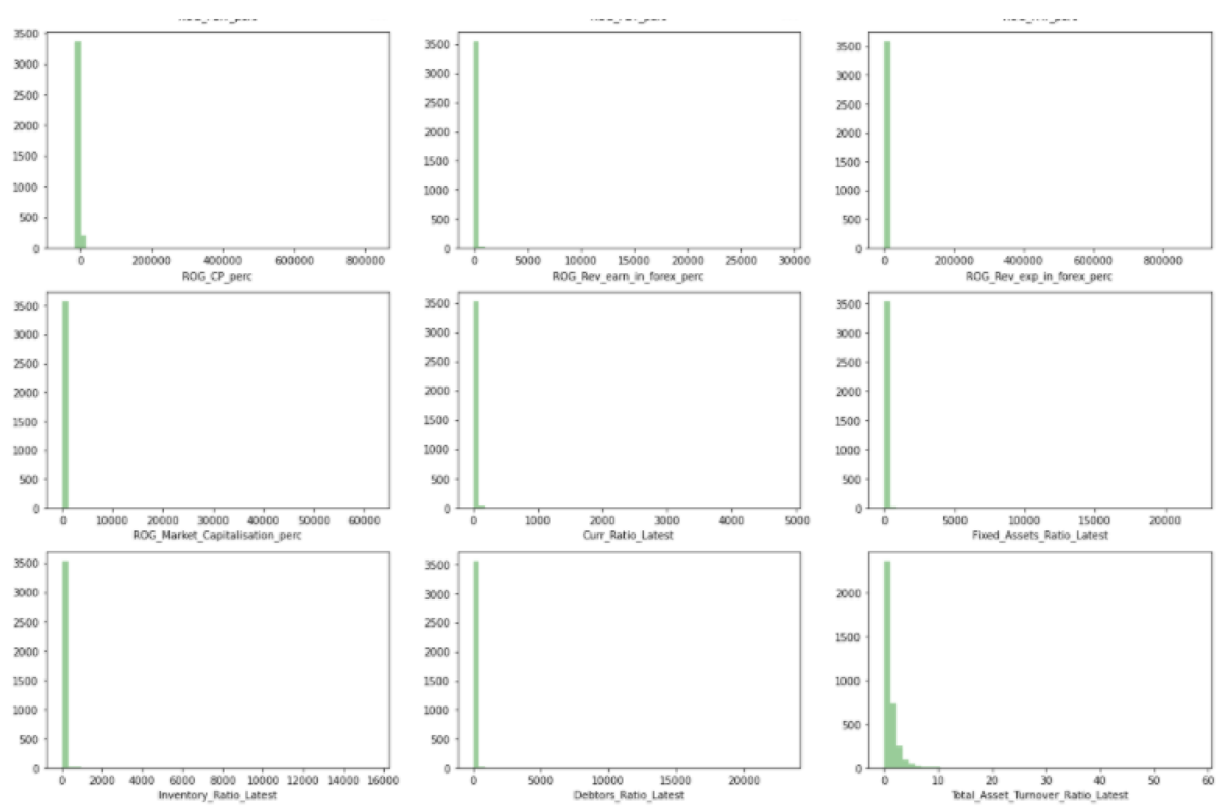

Networth_Next_Year
Equity_Paid_Up
Networth
Capital_Employed
Total_Debt
Gross_Block
Net_Working_Capital
Curr_Assets
Curr_Liab_and_Prov

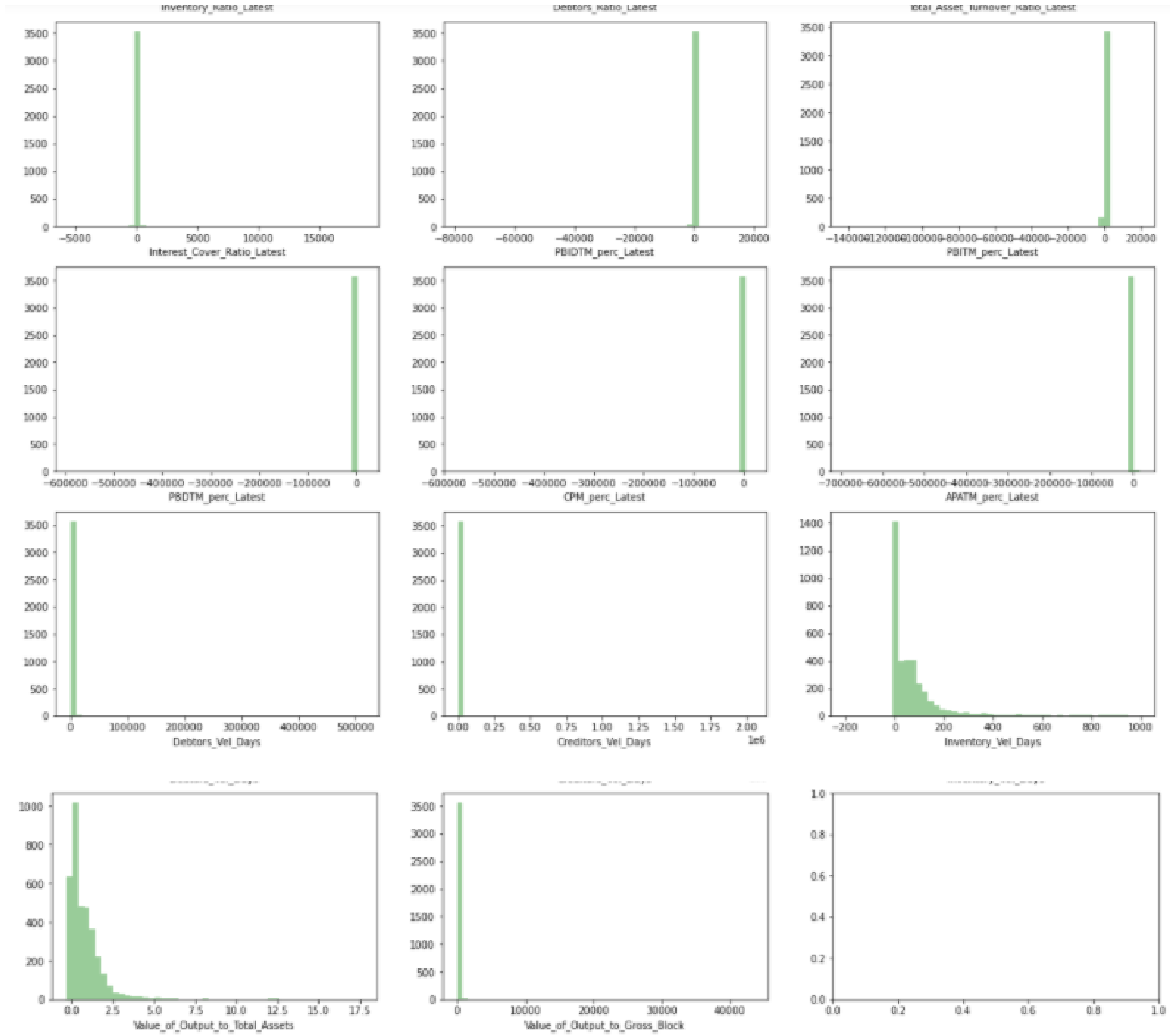

Total_Assets_to_Liab
Gross_Sales
Net_Sales
Other_Income
Value_Of_Output
Cost_of_Prod
Selling_Cost
PBIDT
PBDT


PBIT
PBT
PAT
Adjusted_PAT
CP
Rev_earn_in_forex
Rev_exp_in_forex
Capital_exp_in_forex
Book_Value_Unit_Curr


Rev_exp_in_forex
Capital_exp_in_forex
Book_Value_Unit_Curr
Book_Value_Adj_Unit_Curr
Market_Capitalisation
CEPS_annualised_Unit_Curr
Cash_Flow_From_Opr
Cash_Flow_From_Inv
Cash_Flow_From_Fin
ROG_Net_Worth_perc
ROG_Capital_Employed_perc
ROG_Gross_Block_perc


ROG_Gross_Sales_perc
ROG_Net_Sales_perc
ROG_Cost_of_Prod_perc
ROG_Total_Assets_perc
ROG_PBIDT_perc
ROG_PBDT_perc
ROG_PBIT_perc
ROG_PBT_perc
ROG_PAT_perc


ROG_CP_perc
ROG_Rev_earn_in_forex_perc
ROG_Rev_exp_in_forex_perc
ROG_Market_Capitalisation_perc
Curr_Ratio_Latest
Fixed_Assets_Ratio_Latest
Inventory_Ratio_Latest
Debtors_Ratio_Latest
Total_Asset_Turnover_Ratio_Latest



None of the variables show perfect normal distribution. Few of the variables have skewness in data. There are no duplicate values.

Skewness was observed in almost all teh variables. Most of the variables were right skewed while a few were also found to be left skewed.

	Skewness
Book_Value_Adj_Unit_Curr	59.835459
ROG_PBIT_perc	58.884442
ROG_PBIDT_perc	58.839674
ROG_PBDT_perc	58.366940
ROG_PBT_perc	57.292161
ROG_Market_Capitalisation_perc	57.290651
ROG_Total_Assets_perc	57.264557
ROG_Rev_exp_in_forex_perc	56.767771
ROG_CP_perc	56.749344
ROG_Capital_Employed_perc	56.397091
ROG_PAT_perc	52.606895
CEPS_annualised_Unit_Curr	48.499662
Equity_Paid_Up	45.897050
ROG_Net_Sales_perc	45.373973
ROG_Gross_Sales_perc	45.373047
ROG_Gross_Block_perc	44.839948
ROG_Net_Worth_perc	44.800706
Interest_Cover_Ratio_Latest	40.801193
Debtors_Vel_Days	38.633905
ROG_Cost_of_Prod_perc	37.243494
Debtors_Ratio_Latest	35.236850
Rev_exp_in_forex	34.817286
Cost_of_Prod	34.564594
Creditors_Vel_Days	34.097478
Book_Value_Unit_Curr	32.961635
Value_of_Output_to_Gross_Block	31.976222
Gross_Sales	31.538417
Curr_Ratio_Latest	31.264765
Net_Sales	31.063594
ROG_Rev_earn_in_forex_perc	31.030547
Value_Of_Output	30.790974
Net_Working_Capital	30.559236
Capital_exp_in_forex	27.591003
Inventory_Ratio_Latest	26.987799
Rev_earn_in_forex	24.159618
Fixed_Assets_Ratio_Latest	24.109628
Curr_Assets	20.764954
Total_Debt	19.404026
Selling_Cost	18.865968
Other_Income	18.792561
Gross_Block	18.515689
Capital_Employed	18.061056
Curr_Liab_and_Prov	15.280793
Market_Capitalisation	14.381041
CP	14.335178
PBIT	13.999627
Adjusted_PAT	13.865880
PBDT	13.545492
Total_Assets_to_Liab	13.358549
PBIDT	13.169771
PBT	13.118280
PAT	13.058587
Networth_Next_Year	13.032120
Networth	11.730548
Total_Asset_Turnover_Ratio_Latest	10.354365
Cash_Flow_From_Opr	6.630075
Value_of_Output_to_Total_Assets	4.708705
Inventory_Vel_Days	3.547576
Cash_Flow_From_Fin	1.702510
Cash_Flow_From_Inv	-21.550029
PBIDTM_perc_Latest	-30.914273
PBITM_perc_Latest	-35.977753
CPM_perc_Latest	-46.985387
PBDTM_perc_Latest	-47.723669
APATM_perc_Latest	-49.249980

Univariate Analysis

Data is highly skewed and most of the data is found to be right skewed.

A total of 61 variables were found having tails to the right and hence were right skewed.

There were a total of 6 variables which were found to be left skewed i.e. they had a longer tail on the left hand side of the distribution.

The top 5 variables that have the highest skew are:

```
Book_Value_Adj_Unit_Curr          59.843813
ROG_PBIT_perc                     58.925536
ROG_PBIDT_perc                    58.880737
ROG_PBDT_perc                     58.407667
ROG_PBT_perc                      57.330567
```

The top variables that have the least skew are (in decreasing order):

```
Networth                              11.738799
Total_Asset_Turnover_Ratio_Latest     10.358866
Cash_Flow_From_Opr                     6.634856
Value_of_Output_to_Total_Assets        4.704950
Inventory_Vel_Days                     3.494365
Cash_Flow_From_Fin                     1.703710
```

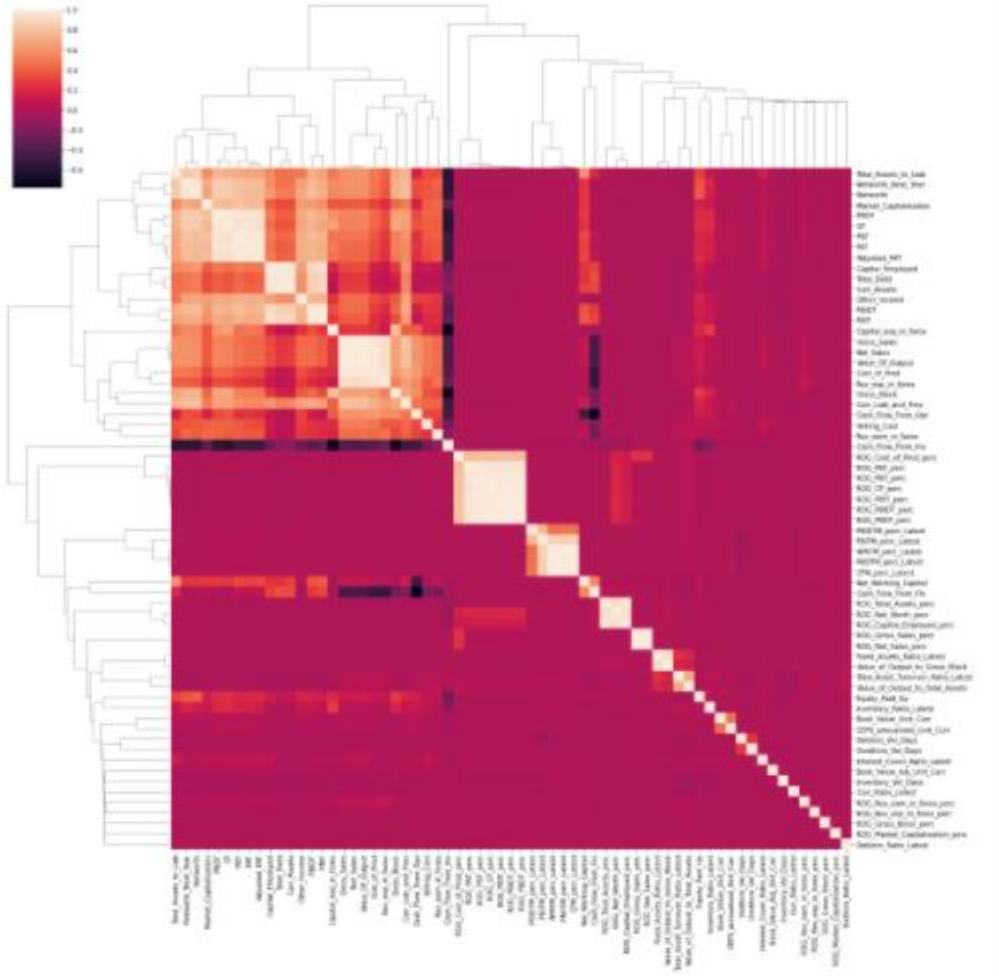
MULTIVARIATE ANALYSIS

We also performed multi Variate analysis on the data to see if there are any correlation that are observed within the data. Correlations function was used and seaborn cluster map was used to plot the correlations and to make better sense of the data.

We observed that net worth and net worth next year were highly correlated. Apart from this, we also found various Rate of Growth variables were highly correlated.

This analysis tells us that there is a problem of collinearity with this data set. Heat map has been plotted on the next page.

Out[46]: <seaborn.matrix.ClusterGrid at 0x1a49cefc280>



PROBLEM 1.5

Train Test Split

Resolution:

Since there was a great imbalance in the data set, we also created a parallel data set with SMOTE and evaluated the performance on smote as well as non smote data.

```
print("Before OverSampling the shape of X: {}".format(X.shape))
print("Before OverSampling the shape of y: {}".format(y.shape))

print("Before OverSampling, counts of label '1': {}".format(sum(y==1)))
print("Before OverSampling, counts of label '0': {} \n".format(sum(y==0)))
```

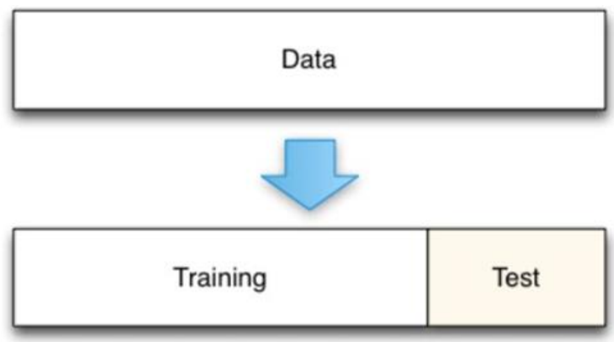
```
Before OverSampling the shape of X: (3581, 34)
Before OverSampling the shape of y: (3581,)
Before OverSampling, counts of label '1': 386
Before OverSampling, counts of label '0': 3195
```

```python
print("After OverSampling the shape of X: {}".format(X_smote.shape))
print("After OverSampling the shape of y: {}".format(y_smote.shape))

print("After OverSampling, counts of label '1': {}".format(sum(y_smote==1)))
print("After OverSampling, counts of label '0': {} \n".format(sum(y_smote==0)))
```

```
After OverSampling the shape of X: (6390, 34)
After OverSampling the shape of y: (6390,)
After OverSampling, counts of label '1': 3195
After OverSampling, counts of label '0': 3195
```

After this data was split into train and testing set, using the stratify = y argument, keeping the ratio of Default variable more or less similar in training as well as testing set.




```python
In [260]: X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.33, random_state = 42, stratify = y)

In [261]: X_train_smote, X_test_smote, y_train_smote, y_test_smote = train_test_split(
              X_smote, y_smote, test_size = 0.33, random_state = 42, stratify = y_smote)
```

Data was split in the 67:33 ratio as per project notes using sklearn's train_test_split function. Also seed value of 42 was used

PROBLEM 1.6

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

Resolution:

Prior to building the logistic regression model, we had to work on feature selection since there were too many columns to start with and we decided to eliminate a few of the columns using the Variation Inflation Factor i.e. VIF

```
calculate_vif_(X, thresh = 5)
```

```
dropping 'PBIDT' at index: 16
dropping 'PBDT' at index: 16
dropping 'APATM_perc_Latest' at index: 57
dropping 'ROG_Gross_Sales_perc' at index: 34
dropping 'Net_Sales' at index: 11
dropping 'PBDTM_perc_Latest' at index: 53
dropping 'Value_Of_Output' at index: 12
dropping 'ROG_Total_Assets_perc' at index: 34
dropping 'Capital_Employed' at index: 3
dropping 'Gross_Sales' at index: 9
dropping 'Total_Debt' at index: 3
dropping 'ROG_PBDT_perc' at index: 32
dropping 'ROG_PBIT_perc' at index: 32
dropping 'CP' at index: 15
dropping 'PAT' at index: 13
dropping 'ROG_PBIDT_perc' at index: 29
dropping 'Total_Assets_to_Liab' at index: 7
dropping 'ROG_PBT_perc' at index: 28
dropping 'PBT' at index: 11
dropping 'PBIT' at index: 10
dropping 'Cost_of_Prod' at index: 8
dropping 'Networth' at index: 2
dropping 'PBITM_perc_Latest' at index: 36
dropping 'ROG_CP_perc' at index: 25
dropping 'Cash_Flow_From_Fin' at index: 18
dropping 'ROG_Net_Worth_perc' at index: 18
dropping 'Gross_Block' at index: 2
dropping 'Fixed_Assets_Ratio_Latest' at index: 26
dropping 'Curr_Liab_and_Prov' at index: 4
dropping 'Networth_Next_Year' at index: 0
dropping 'Adjusted_PAT' at index: 5
Remaining variables:
Index(['Equity_Paid_Up', 'Net_Working_Capital', 'Curr_Assets', 'Other_Income',
       'Selling_Cost', 'Rev_earn_in_forex', 'Rev_exp_in_forex',
       'Capital_exp_in_forex', 'Book_Value_Unit_Curr',
       'Book_Value_Adj_Unit_Curr', 'Market_Capitalisation',
       'CEPS_annualised_Unit_Curr', 'Cash_Flow_From_Opr', 'Cash_Flow_From_Inv',
       'ROG_Capital_Employed_perc', 'ROG_Gross_Block_perc',
       'ROG_Net_Sales_perc', 'ROG_Cost_of_Prod_perc', 'ROG_PAT_perc',
       'ROG_Rev_earn_in_forex_perc', 'ROG_Rev_exp_in_forex_perc',
       'ROG_Market_Capitalisation_perc', 'Curr_Ratio_Latest',
       'Inventory_Ratio_Latest', 'Debtors_Ratio_Latest',
       'Total_Asset_Turnover_Ratio_Latest', 'Interest_Cover_Ratio_Latest',
       'PBIDTM_perc_Latest', 'CPM_perc_Latest', 'Debtors_Vel_Days',
       'Creditors_Vel_Days', 'Inventory_Vel_Days',
       'Value_of_Output_to_Total_Assets', 'Value_of_Output_to_Gross_Block'],
      dtype='object')
```

LOGISTIC REGRESSION

A number of variables were dropped as part of this VIF calculation. These were as below.

```
dropping 'PBIDT' at index: 16
dropping 'PBDT' at index: 16
dropping 'APATM_perc_Latest' at index: 57
dropping 'ROG_Gross_Sales_perc' at index: 34
dropping 'Net_Sales' at index: 11
dropping 'PBDTM_perc_Latest' at index: 53
dropping 'Value_Of_Output' at index: 12
dropping 'ROG_Total_Assets_perc' at index: 34
dropping 'Capital_Employed' at index: 3
dropping 'Gross_Sales' at index: 9
dropping 'Total_Debt' at index: 3
dropping 'ROG_PBDT_perc' at index: 32
dropping 'ROG_PBIT_perc' at index: 32
dropping 'CP' at index: 15
dropping 'PAT' at index: 13
dropping 'ROG_PBIDT_perc' at index: 29
dropping 'Total_Assets_to_Liab' at index: 7
dropping 'ROG_PBT_perc' at index: 28
dropping 'PBT' at index: 11
dropping 'PBIT' at index: 10
dropping 'Cost_of_Prod' at index: 8
dropping 'Networth' at index: 2
dropping 'PBITM_perc_Latest' at index: 36
dropping 'ROG_CP_perc' at index: 25
dropping 'Cash_Flow_From_Fin' at index: 18
dropping 'ROG_Net_Worth_perc' at index: 18
dropping 'Gross_Block' at index: 2
dropping 'Fixed_Assets_Ratio_Latest' at index: 26
dropping 'Curr_Liab_and_Prov' at index: 4
dropping 'Networth_Next_Year' at index: 0
dropping 'Adjusted_PAT' at index: 5
```

A total of 34 variables were retained after this exercise. These were as below.

```
Remaining variables:
Index(['Equity_Paid_Up', 'Net_Working_Capital', 'Curr_Assets', 'Other_Income',
       'Selling_Cost', 'Rev_earn_in_forex', 'Rev_exp_in_forex',
       'Capital_exp_in_forex', 'Book_Value_Unit_Curr',
       'Book_Value_Adj_Unit_Curr', 'Market_Capitalisation',
       'CEPS_annualised_Unit_Curr', 'Cash_Flow_From_Opr', 'Cash_Flow_From_Inv',
       'ROG_Capital_Employed_perc', 'ROG_Gross_Block_perc',
       'ROG_Net_Sales_perc', 'ROG_Cost_of_Prod_perc', 'ROG_PAT_perc',
       'ROG_Rev_earn_in_forex_perc', 'ROG_Rev_exp_in_forex_perc',
       'ROG_Market_Capitalisation_perc', 'Curr_Ratio_Latest',
       'Inventory_Ratio_Latest', 'Debtors_Ratio_Latest',
       'Total_Asset_Turnover_Ratio_Latest', 'Interest_Cover_Ratio_Latest',
       'PBIDTM_perc_Latest', 'CPM_perc_Latest', 'Debtors_Vel_Days',
       'Creditors_Vel_Days', 'Inventory_Vel_Days',
       'Value_of_Output_to_Total_Assets', 'Value_of_Output_to_Gross_Block'],
      dtype='object')
```

Backward elimination method was used for model tuning, we started wtih all 34 variables and built a model, evaluated the p-values at the end of it. Then we removed the variable with highest p-value and then re-ran the model. This process was re-iterated multiple times until we had all variables whose p-value was less than 0.05.

Variables with p-values less than 0.05 were dropped since their coefficients are unreliable and might very well be just a statistical coincidence.

First Model

```
Optimization terminated successfully.
         Current function value: 0.118363
         Iterations: 332
         Function evaluations: 402
         Gradient evaluations: 393
                           Logit Regression Results
==============================================================================
Dep. Variable:                Default   No. Observations:                 2399
Model:                          Logit   Df Residuals:                     2364
Method:                           MLE   Df Model:                           34
Date:                Wed, 17 Feb 2021   Pseudo R-squ.:                  0.6541
Time:                        19:20:00   Log-Likelihood:                -283.95
converged:                       True   LL-Null:                       -821.02
Covariance Type:            nonrobust   LLR p-value:                 1.349e-203
===================================================================================================
                                      coef    std err          z      P>|z|      [0.025      0.975]
---------------------------------------------------------------------------------------------------
Intercept                          -0.2130      0.176     -1.213      0.225      -0.557       0.131
Equity_Paid_Up                      0.0003      0.001      0.202      0.840      -0.002       0.003
Net_Working_Capital                 0.0003      0.000      0.822      0.411      -0.000       0.001
Curr_Assets                      8.536e-05   8.67e-05      0.984      0.325   -8.47e-05       0.000
Other_Income                       -0.0075      0.009     -0.877      0.381      -0.024       0.009
Selling_Cost                     5.552e-05      0.007      0.008      0.994      -0.014       0.014
Rev_earn_in_forex                   0.0015      0.002      0.881      0.378      -0.002       0.005
Rev_exp_in_forex                    0.0002      0.001      0.308      0.758      -0.001       0.002
Capital_exp_in_forex               -0.0073      0.047     -0.154      0.878      -0.100       0.085
Book_Value_Unit_Curr               -0.1268      0.052     -2.438      0.015      -0.229      -0.025
Book_Value_Adj_Unit_Curr           -0.0275      0.052     -0.529      0.597      -0.129       0.074
Market_Capitalisation              -0.0007      0.001     -0.937      0.349      -0.002       0.001
CEPS_annualised_Unit_Curr          -0.0922      0.017     -5.402      0.000      -0.126      -0.059
Cash_Flow_From_Opr                  0.0010      0.002      0.420      0.674      -0.004       0.006
Cash_Flow_From_Inv                 -0.0003      0.002     -0.178      0.859      -0.004       0.003
ROG_Capital_Employed_perc          -0.0005      0.001     -0.621      0.534      -0.002       0.001
ROG_Gross_Block_perc               -0.0047      0.004     -1.252      0.211      -0.012       0.003
ROG_Net_Sales_perc                 -0.0002      0.001     -0.381      0.703      -0.001       0.001
ROG_Cost_of_Prod_perc           -4.205e-05      0.000     -0.252      0.801      -0.000       0.000
ROG_PAT_perc                     6.123e-05   4.43e-05      1.382      0.167   -2.56e-05       0.000
ROG_Rev_earn_in_forex_perc         -0.0032      0.004     -0.887      0.375      -0.010       0.004
ROG_Rev_exp_in_forex_perc       -8.732e-05      0.000     -0.225      0.822      -0.001       0.001
ROG_Market_Capitalisation_perc     -0.0004      0.001     -0.545      0.586      -0.002       0.001
Curr_Ratio_Latest                  -0.4393      0.096     -4.590      0.000      -0.627      -0.252
Inventory_Ratio_Latest             -0.0026      0.002     -1.642      0.101      -0.006       0.000
Debtors_Ratio_Latest               -0.0017      0.002     -0.747      0.455      -0.006       0.003
Total_Asset_Turnover_Ratio_Latest   0.0250      0.037      0.674      0.500      -0.048       0.098
Interest_Cover_Ratio_Latest        -0.0023      0.001     -2.566      0.010      -0.004      -0.001
PBIDTM_perc_Latest               4.199e-05      0.000      0.154      0.878      -0.000       0.001
CPM_perc_Latest                    -0.0001      0.000     -0.535      0.593      -0.001       0.000
Debtors_Vel_Days                -4.735e-05   6.04e-05     -0.784      0.433      -0.000     7.1e-05
Creditors_Vel_Days               6.635e-06    7.3e-06      0.909      0.364   -7.68e-06    2.09e-05
Inventory_Vel_Days                  0.0002      0.001      0.305      0.760      -0.001       0.002
Value_of_Output_to_Total_Assets    -0.0988      0.114     -0.868      0.385      -0.322       0.124
Value_of_Output_to_Gross_Block     -0.0005      0.001     -0.383      0.702      -0.003       0.002
===================================================================================================

Possibly complete quasi-separation: A fraction 0.44 of observations can be
perfectly predicted. This might indicate that there is complete
quasi-separation. In this case some parameters will not be identified.
```

It is evident from the image that the variable Selling_Cost has a p-value of 0.993767. Since this is higher than 0.05 and the highest of all the variables, we will drop this variable in subsequent models. This process of dropping variables based on p-values and modeling continued until a model where all the p-values were relevant was achieved. The iterative process got stopped at Model30 which has 4 independent variables and each of them were relevant.

```
Optimization terminated successfully.
         Current function value: 0.124653
         Iterations: 48
         Function evaluations: 57
         Gradient evaluations: 57
                           Logit Regression Results
==============================================================================
Dep. Variable:                Default   No. Observations:                 2399
Model:                          Logit   Df Residuals:                     2394
Method:                           MLE   Df Model:                            4
Date:                Wed, 17 Feb 2021   Pseudo R-squ.:                  0.6358
Time:                        19:33:58   Log-Likelihood:                -299.04
converged:                       True   LL-Null:                       -821.02
Covariance Type:            nonrobust   LLR p-value:                 1.067e-224
===============================================================================================
                                  coef    std err          z      P>|z|      [0.025      0.975]
-----------------------------------------------------------------------------------------------
Intercept                      -0.3898      0.148     -2.641      0.008      -0.679      -0.100
Book_Value_Unit_Curr           -0.1514      0.012    -12.174      0.000      -0.176      -0.127
CEPS_annualised_Unit_Curr      -0.0972      0.013     -7.649      0.000      -0.122      -0.072
Curr_Ratio_Latest              -0.4580      0.097     -4.733      0.000      -0.648      -0.268
Interest_Cover_Ratio_Latest    -0.0024      0.001     -2.999      0.003      -0.004      -0.001
===============================================================================================

Possibly complete quasi-separation: A fraction 0.42 of observations can be
perfectly predicted. This might indicate that there is complete
quasi-separation. In this case some parameters will not be identified.

Intercept                      8.276667e-03
Interest_Cover_Ratio_Latest    2.704242e-03
Curr_Ratio_Latest              2.210014e-06
CEPS_annualised_Unit_Curr      2.020357e-14
Book_Value_Unit_Curr           4.275435e-34
dtype: float64
```

P-values of all the variables are less than 0.05 and thus all the coefficients are relevant. Book_Value_Unit_Curr has the highest coefficient and Interest_Cover_Ratio_Latest the least of all. This model will be used to validate the test dataset.

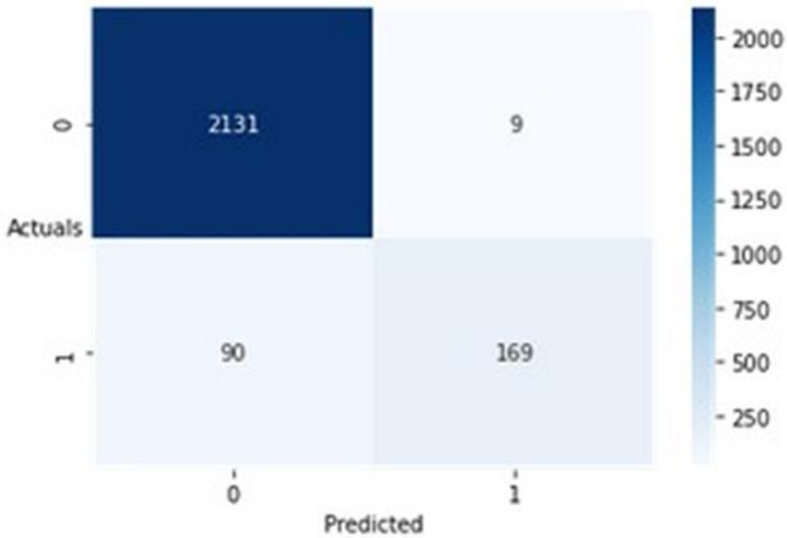
Evaluation on SMOTE set did not yield any better results. Hence we stuck to the original data set.

PROBLEM 1.7

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

Resolution:

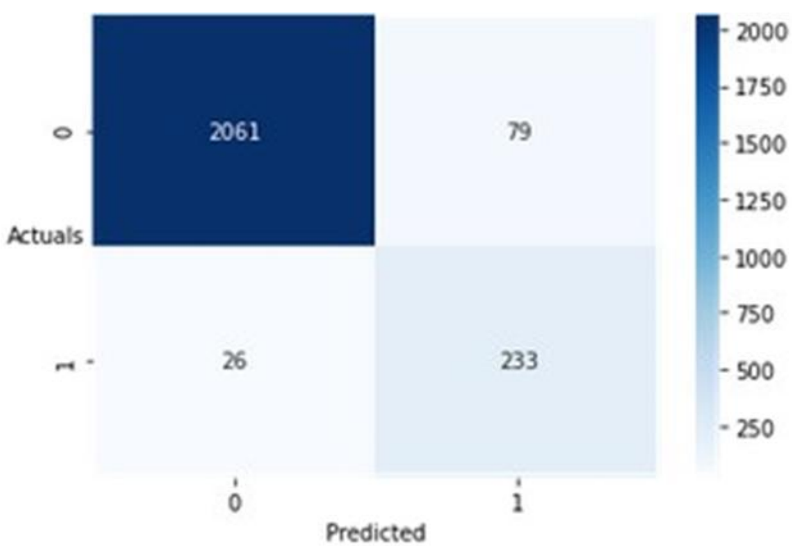
With default probability threshold of 0.5, the confusion matrix for the train set is as follows:



Correctly predicted = 2131 incorrectly predicted records = 169

This was pretty good result on its own, however to further improve the on the results. We decided to look for the optimum threshold.

After evaluating using the optimal threshold. Below was the new classification matrix.

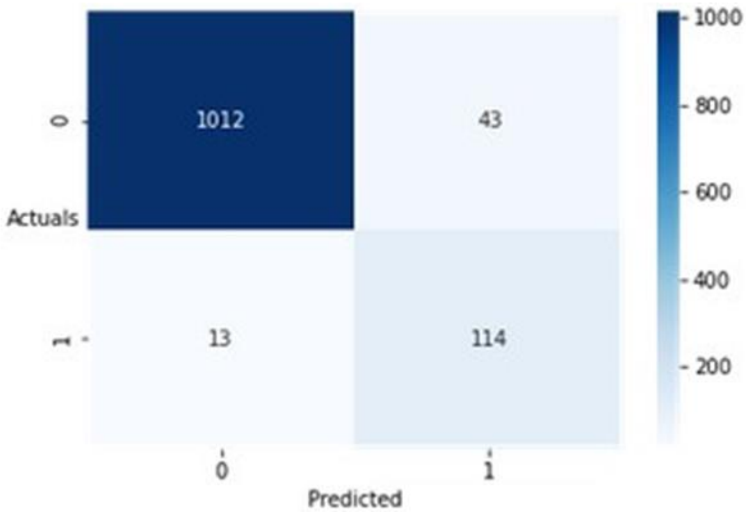


	precision	recall	f1-score	support
0	0.988	0.963	0.975	2140
1	0.747	0.900	0.816	259
accuracy			0.956	2399
macro avg	0.867	0.931	0.896	2399
weighted avg	0.962	0.956	0.958	2399

Accuracy of over 95.6% was achieved while recall, precision and f1 score were also very high at 96.3,98.8% and 97.5% respectively.

We also evaluated the test data set for the same model which was built after the above mentioned re-iterative process.

Below are statistics for the test model.



	precision	recall	f1-score	support
0	0.987	0.959	0.973	1055
1	0.726	0.898	0.803	127
accuracy			0.953	1182
macro avg	0.857	0.928	0.888	1182
weighted avg	0.959	0.953	0.955	1182

Accuracy of 95.3% and very high recall, precision and f1 score of 95.9% ,98.7% and 97.3% respectively were also observed on the test set. This clearly indicates that the model which has been built is highly efficient and has been able to capture the correct variable for prediction.

Tt has been proven to work on train as well as test data.

The End

Thakur Arun Singh

****************************^^***************************