FRA | Extended Project

VIBHAV JAISWAL

01-01-2023

CONTENTS

LIST OF FIGURES

LIST OF TABLES

PROBLEM STATEMENT

You are requested to create an Indian credit risk (default) model, using the data provided in the spreadsheet.

Hints:

Dependent variable - We need to create a default variable which should take the value of 1 when net worth next year is negative & 0 when net worth next year is positive.

Validation Dataset - We need to build the model on train dataset and check the model performance measures on validation dataset.

EXECUTIVE SUMMARY

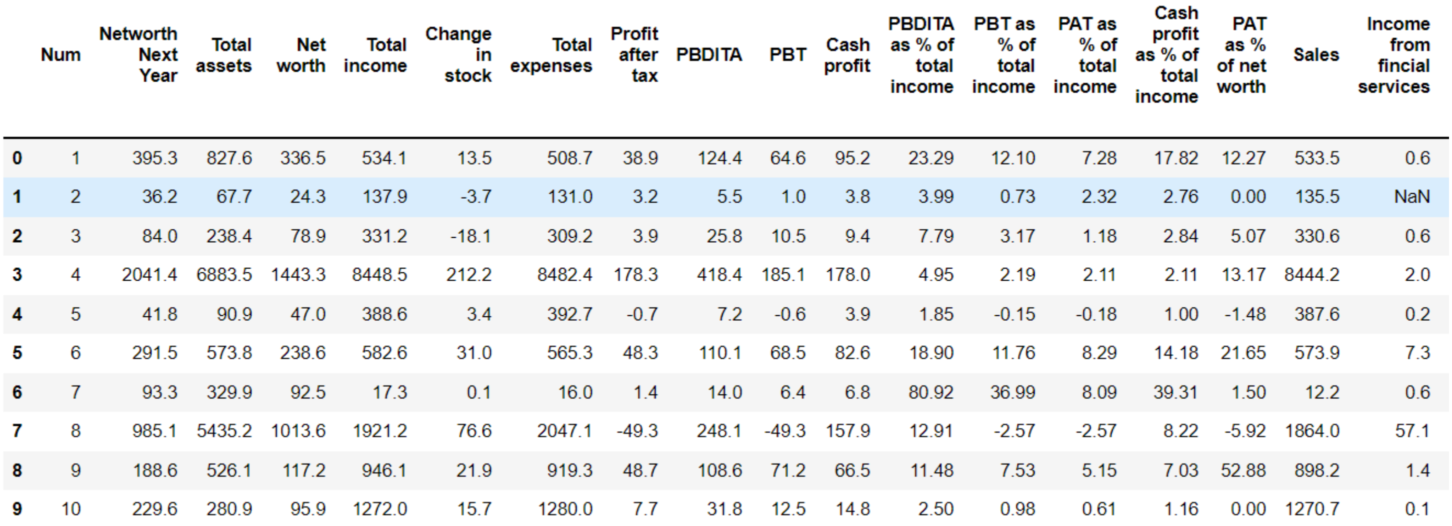
Objective is to build Indian credit risk (default) model based on different attributes of the dataset and predict which company is going to be defaulter or non-defaulter.

Purpose is to do exploratory data analysis, visualization & apply various supervised learning algorithms like **logistic regression and random forest** to predict which company will be defaulter or non-defaulter.

This assignment will help to **reduce credit risk** by predicting defaulter or non-defaulter status of companies.

CHECKING THE RECORDS OF THE DATASET:

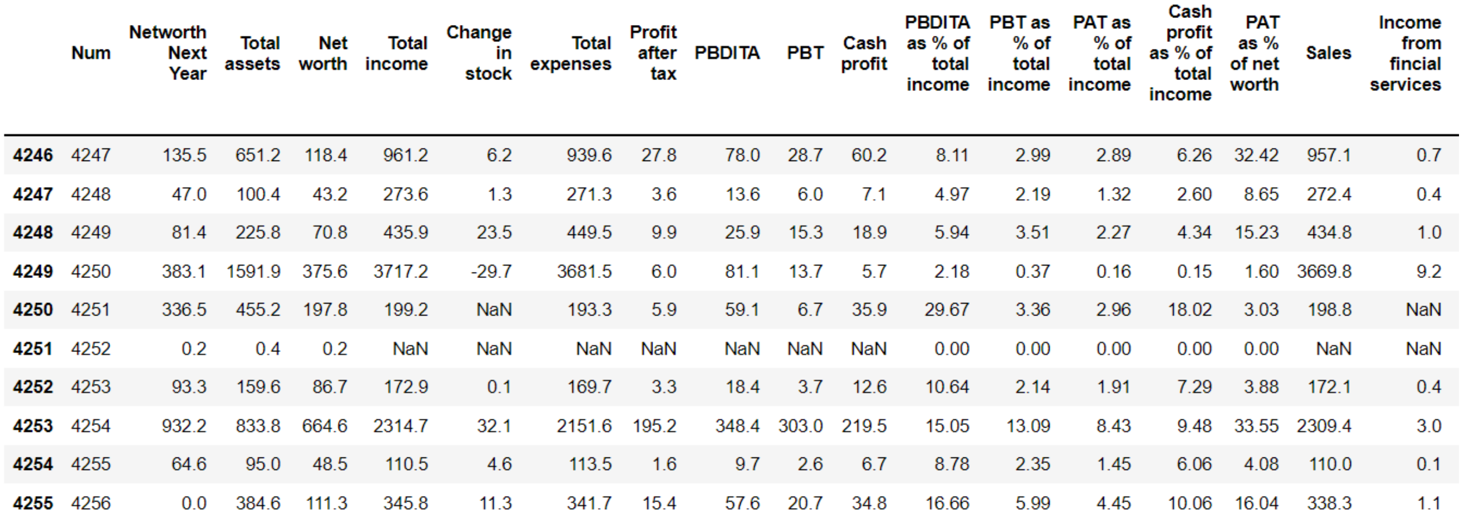
Head of Dataset



	Num	Networth Next Year	Total assets	Net worth	Total income	Change in stock	Total expenses	Profit after tax	PBDITA	PBT	Cash profit	PBDITA as % of total income	PBT as % of total income	PAT as % of total income	Cash profit as % of total income	PAT as % of net worth	Sales	Income from fincial services
0	1	395.3	827.6	336.5	534.1	13.5	508.7	38.9	124.4	64.6	95.2	23.29	12.10	7.28	17.82	12.27	533.5	0.6
1	2	36.2	67.7	24.3	137.9	-3.7	131.0	3.2	5.5	1.0	3.8	3.99	0.73	2.32	2.76	0.00	135.5	NaN
2	3	84.0	238.4	78.9	331.2	-18.1	309.2	3.9	25.8	10.5	9.4	7.79	3.17	1.18	2.84	5.07	330.6	0.6
3	4	2041.4	6883.5	1443.3	8448.5	212.2	8482.4	178.3	418.4	185.1	178.0	4.95	2.19	2.11	2.11	13.17	8444.2	2.0
4	5	41.8	90.9	47.0	388.6	3.4	392.7	-0.7	7.2	-0.6	3.9	1.85	-0.15	-0.18	1.00	-1.48	387.6	0.2
5	6	291.5	573.8	238.6	582.6	31.0	565.3	48.3	110.1	68.5	82.6	18.90	11.76	8.29	14.18	21.65	573.9	7.3
6	7	93.3	329.9	92.5	17.3	0.1	16.0	1.4	14.0	6.4	6.8	80.92	36.99	8.09	39.31	1.50	12.2	0.6
7	8	985.1	5435.2	1013.6	1921.2	76.6	2047.1	-49.3	248.1	-49.3	157.9	12.91	-2.57	-2.57	8.22	-5.92	1864.0	57.1
8	9	188.6	526.1	117.2	946.1	21.9	919.3	48.7	108.6	71.2	66.5	11.48	7.53	5.15	7.03	52.88	898.2	1.4
9	10	229.6	280.9	95.9	1272.0	15.7	1280.0	7.7	31.8	12.5	14.8	2.50	0.98	0.61	1.16	0.00	1270.7	0.1

Table 1: Records of the Dataset Head

Tail of Dataset



	Num	Networth Next Year	Total assets	Net worth	Total income	Change in stock	Total expenses	Profit after tax	PBDITA	PBT	Cash profit	PBDITA as % of total income	PBT as % of total income	PAT as % of total income	Cash profit as % of total income	PAT as % of net worth	Sales	Income from fincial services
4246	4247	135.5	651.2	118.4	961.2	6.2	939.6	27.8	78.0	28.7	60.2	8.11	2.99	2.89	6.26	32.42	957.1	0.7
4247	4248	47.0	100.4	43.2	273.6	1.3	271.3	3.6	13.6	6.0	7.1	4.97	2.19	1.32	2.60	8.65	272.4	0.4
4248	4249	81.4	225.8	70.8	435.9	23.5	449.5	9.9	25.9	15.3	18.9	5.94	3.51	2.27	4.34	15.23	434.8	1.0
4249	4250	383.1	1591.9	375.6	3717.2	-29.7	3681.5	6.0	81.1	13.7	5.7	2.18	0.37	0.16	0.15	1.60	3669.8	9.2
4250	4251	336.5	455.2	197.8	199.2	NaN	193.3	5.9	59.1	6.7	35.9	29.67	3.36	2.96	18.02	3.03	198.8	NaN
4251	4252	0.2	0.4	0.2	NaN	NaN	NaN	NaN	NaN	NaN	NaN	0.00	0.00	0.00	0.00	0.00	NaN	NaN
4252	4253	93.3	159.6	86.7	172.9	0.1	169.7	3.3	18.4	3.7	12.6	10.64	2.14	1.91	7.29	3.88	172.1	0.4
4253	4254	932.2	833.8	664.6	2314.7	32.1	2151.6	195.2	348.4	303.0	219.5	15.05	13.09	8.43	9.48	33.55	2309.4	3.0
4254	4255	64.6	95.0	48.5	110.5	4.6	113.5	1.6	9.7	2.6	6.7	8.78	2.35	1.45	6.06	4.08	110.0	0.1
4255	4256	0.0	384.6	111.3	345.8	11.3	341.7	15.4	57.6	20.7	34.8	16.66	5.99	4.45	10.06	16.04	338.3	1.1

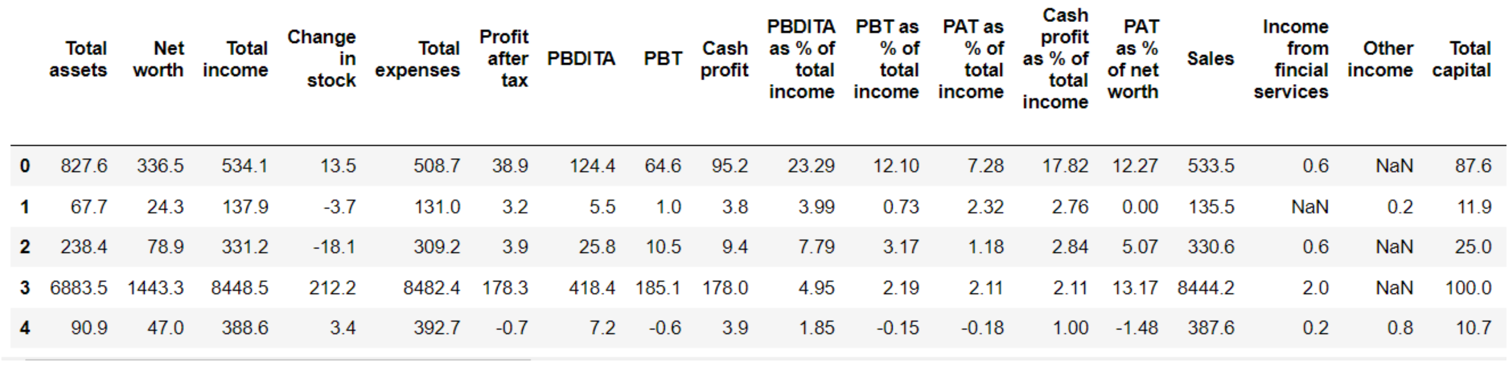
Table 2: Records of the Dataset Tail

Note:

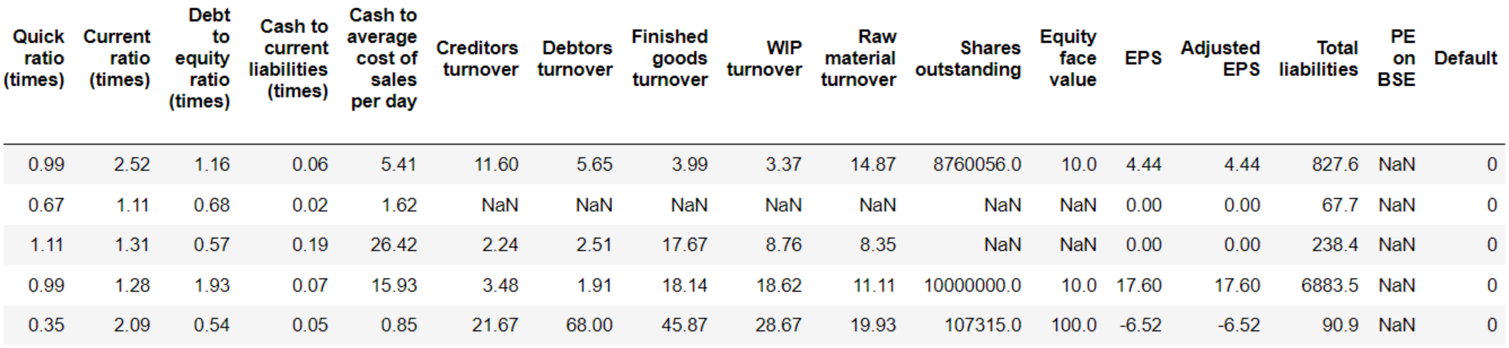
- **Num** column will be dropped as its just for row numbering doesn't contribute in analysis.
- As explained in the problem statement, default variable should take the value of 1 when net worth next year is negative & 0 when net worth next year is positive, we have to derive the target column from the existing independent variable i.e., Net worth Next Year.
- After creating the target variable, we are going to drop the Net worth Next Year attribute because dependent variable is derived from this variable and if we are not dropping this variable then it will make our model bias and other variables will not perform well in predicting the dependent variable.

RECORDS OF THE DATASET AFTER CREATING TARGET COLUMN AND DROPPING NUM AND NET WORTH NEXT YEAR COLUMN:

Head of Dataset



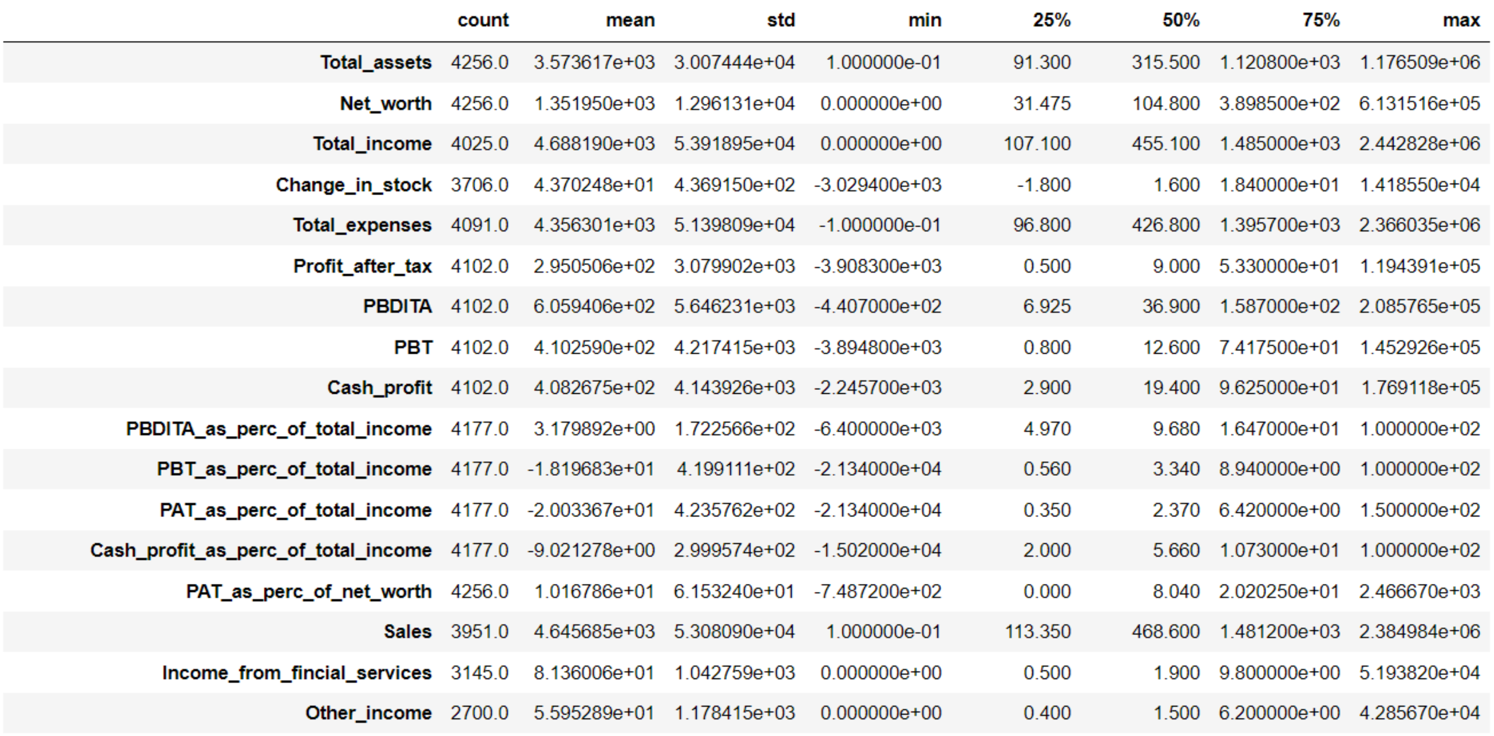
	Total assets	Net worth	Total income	Change in stock	Total expenses	Profit after tax	PBDITA	PBT	Cash profit	PBDITA as % of total income	PBT as % of total income	PAT as % of total income	Cash profit as % of total income	PAT as % of net worth	Sales	Income from fincial services	Other income	Total capital
0	827.6	336.5	534.1	13.5	508.7	38.9	124.4	64.6	95.2	23.29	12.10	7.28	17.82	12.27	533.5	0.6	NaN	87.6
1	67.7	24.3	137.9	-3.7	131.0	3.2	5.5	1.0	3.8	3.99	0.73	2.32	2.76	0.00	135.5	NaN	0.2	11.9
2	238.4	78.9	331.2	-18.1	309.2	3.9	25.8	10.5	9.4	7.79	3.17	1.18	2.84	5.07	330.6	0.6	NaN	25.0
3	6883.5	1443.3	8448.5	212.2	8482.4	178.3	418.4	185.1	178.0	4.95	2.19	2.11	2.11	13.17	8444.2	2.0	NaN	100.0
4	90.9	47.0	388.6	3.4	392.7	-0.7	7.2	-0.6	3.9	1.85	-0.15	-0.18	1.00	-1.48	387.6	0.2	0.8	10.7



Quick ratio (times)	Current ratio (times)	Debt to equity ratio (times)	Cash to current liabilities (times)	Cash to average cost of sales per day	Creditors turnover	Debtors turnover	Finished goods turnover	WIP turnover	Raw material turnover	Shares outstanding	Equity face value	EPS	Adjusted EPS	Total liabilities	PE on BSE	Default
0.99	2.52	1.16	0.06	5.41	11.60	5.65	3.99	3.37	14.87	8760056.0	10.0	4.44	4.44	827.6	NaN	0
0.67	1.11	0.68	0.02	1.62	NaN	NaN	NaN	NaN	NaN	NaN	NaN	0.00	0.00	67.7	NaN	0
1.11	1.31	0.57	0.19	26.42	2.24	2.51	17.67	8.76	8.35	NaN	NaN	0.00	0.00	238.4	NaN	0
0.99	1.28	1.93	0.07	15.93	3.48	1.91	18.14	18.62	11.11	10000000.0	10.0	17.60	17.60	6883.5	NaN	0
0.35	2.09	0.54	0.05	0.85	21.67	68.00	45.87	28.67	19.93	107315.0	100.0	-6.52	-6.52	90.9	NaN	0

Table 3: Records of the Dataset After Creating Target Column and Dropping Num and Net worth Next Year column

SUMMARY OF THE DATASET.



	count	mean	std	min	25%	50%	75%	max
Total_assets	4256.0	3.573617e+03	3.007444e+04	1.000000e-01	91.300	315.500	1.120800e+03	1.176509e+06
Net_worth	4256.0	1.351950e+03	1.296131e+04	0.000000e+00	31.475	104.800	3.898500e+02	6.131516e+05
Total_income	4025.0	4.688190e+03	5.391895e+04	0.000000e+00	107.100	455.100	1.485000e+03	2.442828e+06
Change_in_stock	3706.0	4.370248e+01	4.369150e+02	-3.029400e+03	-1.800	1.600	1.840000e+01	1.418550e+04
Total_expenses	4091.0	4.356301e+03	5.139809e+04	-1.000000e-01	96.800	426.800	1.395700e+03	2.366035e+06
Profit_after_tax	4102.0	2.950506e+02	3.079902e+03	-3.908300e+03	0.500	9.000	5.330000e+01	1.194391e+05
PBDITA	4102.0	6.059406e+02	5.646231e+03	-4.407000e+02	6.925	36.900	1.587000e+02	2.085765e+05
PBT	4102.0	4.102590e+02	4.217415e+03	-3.894800e+03	0.800	12.600	7.417500e+01	1.452926e+05
Cash_profit	4102.0	4.082675e+02	4.143926e+03	-2.245700e+03	2.900	19.400	9.625000e+01	1.769118e+05
PBDITA_as_perc_of_total_income	4177.0	3.179892e+00	1.722566e+02	-6.400000e+03	4.970	9.680	1.647000e+01	1.000000e+02
PBT_as_perc_of_total_income	4177.0	-1.819683e+01	4.199111e+02	-2.134000e+04	0.560	3.340	8.940000e+00	1.000000e+02
PAT_as_perc_of_total_income	4177.0	-2.003367e+01	4.235762e+02	-2.134000e+04	0.350	2.370	6.420000e+00	1.500000e+02
Cash_profit_as_perc_of_total_income	4177.0	-9.021278e+00	2.999574e+02	-1.502000e+04	2.000	5.660	1.073000e+01	1.000000e+02
PAT_as_perc_of_net_worth	4256.0	1.016786e+01	6.153240e+01	-7.487200e+02	0.000	8.040	2.020250e+01	2.466670e+03
Sales	3951.0	4.645685e+03	5.308090e+04	1.000000e-01	113.350	468.600	1.481200e+03	2.384984e+06
Income_from_fincial_services	3145.0	8.136006e+01	1.042759e+03	0.000000e+00	0.500	1.900	9.800000e+00	5.193820e+04
Other_income	2700.0	5.595289e+01	1.178415e+03	0.000000e+00	0.400	1.500	6.200000e+00	4.285670e+04

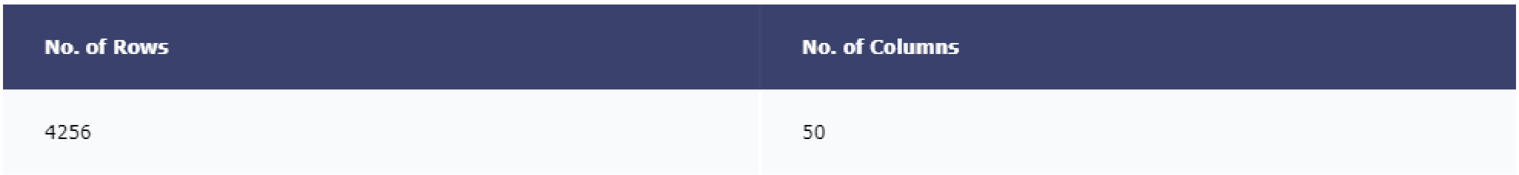
Table 4: Summary of the Dataset

Insights:

- From above table, we can get count, mean, std, 25%, 50% ,75% and min & max values of the all variables present in the dataset.
- There are no anomalies present in variables.

SHAPE OF THE DATAFRAME:

From shape attribute, we can infer total no. of rows and columns. Dataset has 4256 rows and 50 columns.

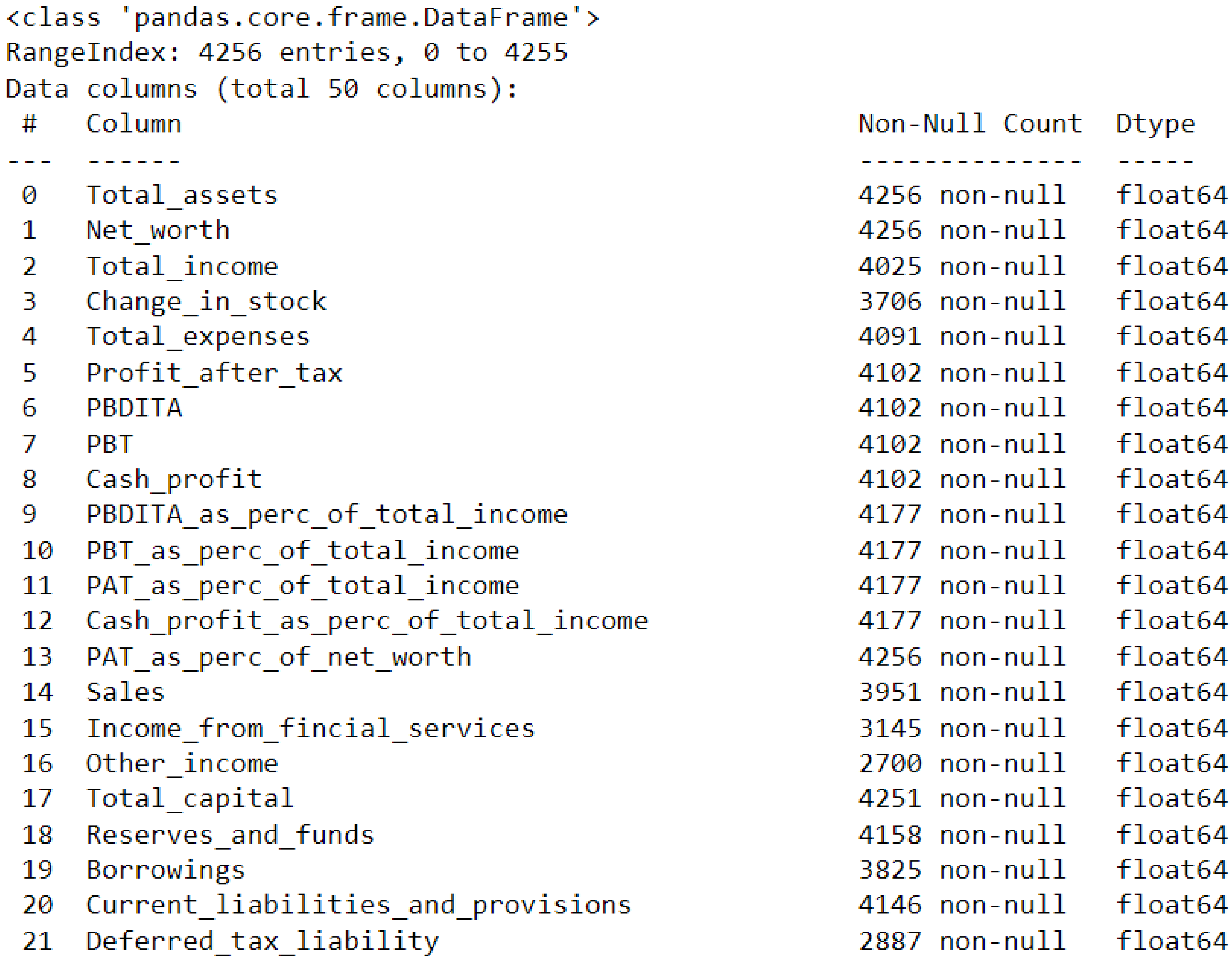


No. of Rows	No. of Columns
4256	50

Table 5: Shape of the Dataset

APPROPRIATENESS OF DATATYPES & INFORMATION OF THE DATAFRAME:

From info(), we can get column details including its datatype along with total number of rows. In addition, we get total number of non-null rows and memory usage.



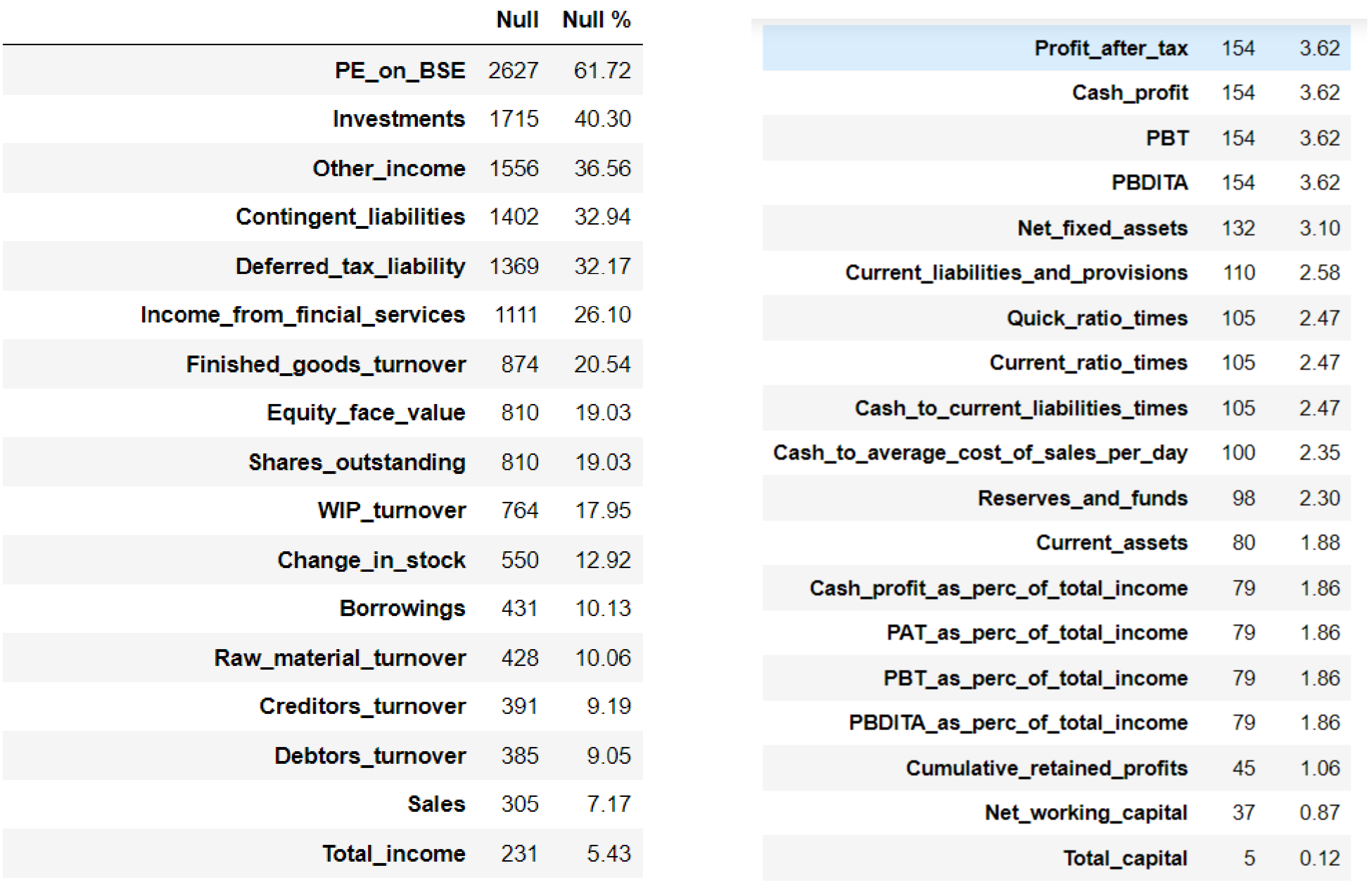
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4256 entries, 0 to 4255
Data columns (total 50 columns):
 #   Column                                  Non-Null Count  Dtype  
---  ------                                  --------------  -----  
 0   Total_assets                            4256 non-null   float64
 1   Net_worth                               4256 non-null   float64
 2   Total_income                            4025 non-null   float64
 3   Change_in_stock                         3706 non-null   float64
 4   Total_expenses                          4091 non-null   float64
 5   Profit_after_tax                        4102 non-null   float64
 6   PBDITA                                  4102 non-null   float64
 7   PBT                                     4102 non-null   float64
 8   Cash_profit                             4102 non-null   float64
 9   PBDITA_as_perc_of_total_income          4177 non-null   float64
 10  PBT_as_perc_of_total_income             4177 non-null   float64
 11  PAT_as_perc_of_total_income             4177 non-null   float64
 12  Cash_profit_as_perc_of_total_income     4177 non-null   float64
 13  PAT_as_perc_of_net_worth                4256 non-null   float64
 14  Sales                                   3951 non-null   float64
 15  Income_from_fincial_services            3145 non-null   float64
 16  Other_income                            2700 non-null   float64
 17  Total_capital                           4251 non-null   float64
 18  Reserves_and_funds                      4158 non-null   float64
 19  Borrowings                              3825 non-null   float64
 20  Current_liabilities_and_provisions      4146 non-null   float64
 21  Deferred_tax_liability                  2887 non-null   float64
```

Table 6: Appropriateness of Datatypes & Information of the Dataframe

Insights:

- From the above table, we can infer that there are null values present in the dataset.
- There is total 4256 rows and 50 columns in the dataset. Only one variable 'Default' is int type. All other variables are float type.

NULL VALUES AND % OF NULL VALUES PRESENT IN EACH VARIABLE



	Null	Null %
PE_on_BSE	2627	61.72
Investments	1715	40.30
Other_income	1556	36.56
Contingent_liabilities	1402	32.94
Deferred_tax_liability	1369	32.17
Income_from_fincial_services	1111	26.10
Finished_goods_turnover	874	20.54
Equity_face_value	810	19.03
Shares_outstanding	810	19.03
WIP_turnover	764	17.95
Change_in_stock	550	12.92
Borrowings	431	10.13
Raw_material_turnover	428	10.06
Creditors_turnover	391	9.19
Debtors_turnover	385	9.05
Sales	305	7.17
Total_income	231	5.43
Profit_after_tax	154	3.62
Cash_profit	154	3.62
PBT	154	3.62
PBDITA	154	3.62
Net_fixed_assets	132	3.10
Current_liabilities_and_provisions	110	2.58
Quick_ratio_times	105	2.47
Current_ratio_times	105	2.47
Cash_to_current_liabilities_times	105	2.47
Cash_to_average_cost_of_sales_per_day	100	2.35
Reserves_and_funds	98	2.30
Current_assets	80	1.88
Cash_profit_as_perc_of_total_income	79	1.86
PAT_as_perc_of_total_income	79	1.86
PBT_as_perc_of_total_income	79	1.86
PBDITA_as_perc_of_total_income	79	1.86
Cumulative_retained_profits	45	1.06
Net_working_capital	37	0.87
Total_capital	5	0.12

Table 7: Checking for Null Values and % of Null Values Present in Each Variable.

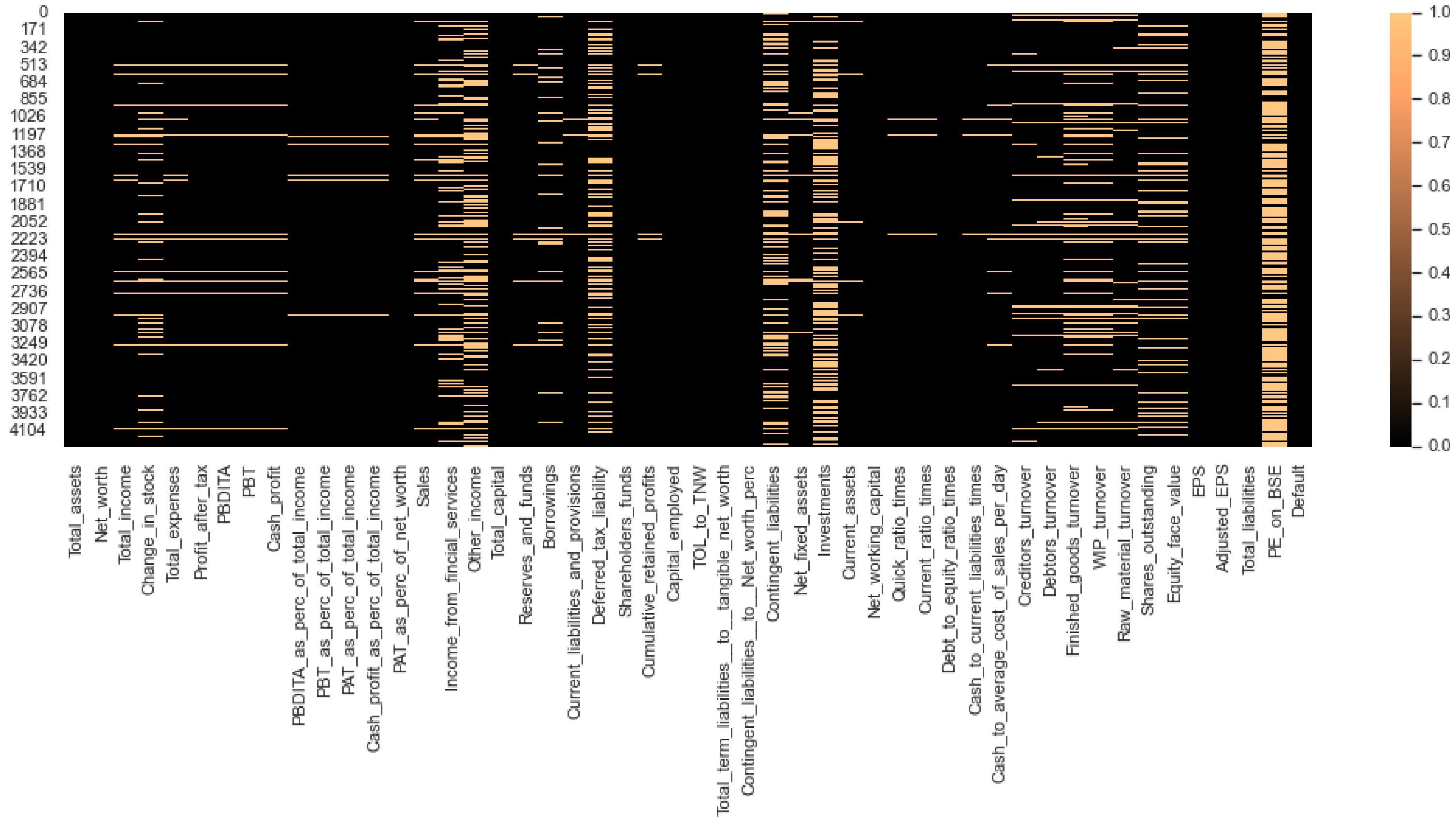
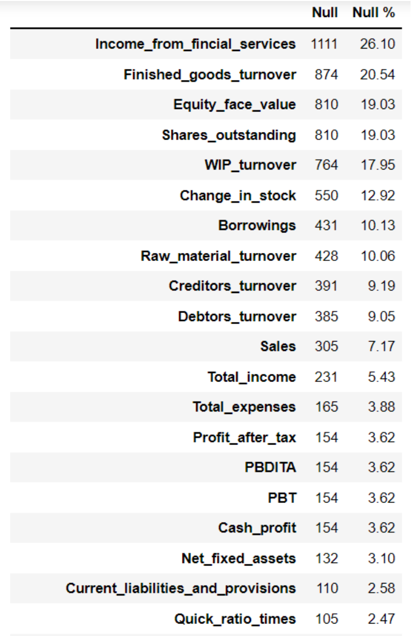


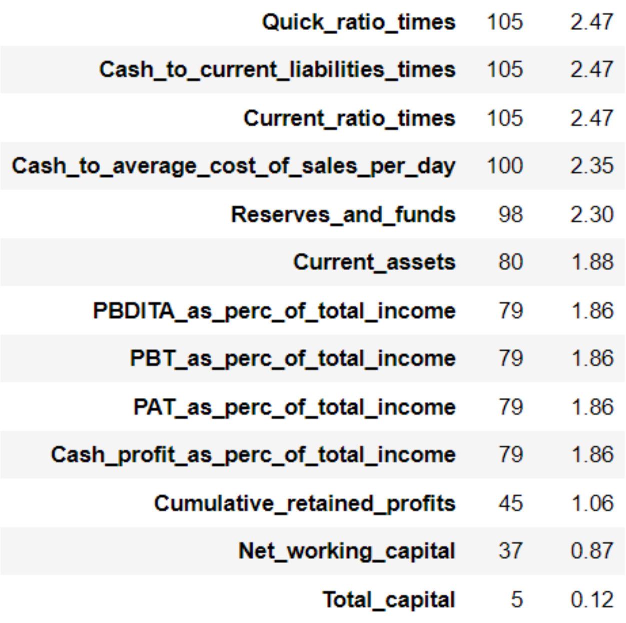
Figure 1: Null Values Present in Each Variable

We will drop columns having more than 30% null values as their contribution to analysis is minimal.

NULL VALUES AND % OF NULL VALUES PRESENT IN EACH VARIABLE AFTER DROPPING VARIABLES WHICH HAVE NULL % MORE THAN 30 %.



	Null	Null %
Income_from_fincial_services	1111	26.10
Finished_goods_turnover	874	20.54
Equity_face_value	810	19.03
Shares_outstanding	810	19.03
WIP_turnover	764	17.95
Change_in_stock	550	12.92
Borrowings	431	10.13
Raw_material_turnover	428	10.06
Creditors_turnover	391	9.19
Debtors_turnover	385	9.05
Sales	305	7.17
Total_income	231	5.43
Total_expenses	165	3.88
Profit_after_tax	154	3.62
PBDITA	154	3.62
PBT	154	3.62
Cash_profit	154	3.62
Net_fixed_assets	132	3.10
Current_liabilities_and_provisions	110	2.58
Quick_ratio_times	105	2.47



Quick_ratio_times	105	2.47
Cash_to_current_liabilities_times	105	2.47
Current_ratio_times	105	2.47
Cash_to_average_cost_of_sales_per_day	100	2.35
Reserves_and_funds	98	2.30
Current_assets	80	1.88
PBDITA_as_perc_of_total_income	79	1.86
PBT_as_perc_of_total_income	79	1.86
PAT_as_perc_of_total_income	79	1.86
Cash_profit_as_perc_of_total_income	79	1.86
Cumulative_retained_profits	45	1.06
Net_working_capital	37	0.87
Total_capital	5	0.12

Table 8: Checking for Null Values and % of Null Values Present in Each Variable After Dropping Variables which have Null % more than 30 %

Since now we have columns with null values less than 30%, we can impute them with median.

ANOMALIES IN THE DATASET

```
TOTAL_ASSETS :  2961
[ 827.6    67.7   238.4 ... 1591.9   159.6   833.8]

**************************************************

NET_WORTH :  2376
[336.5   24.3   78.9 ... 123.8   35.3 664.6]

**************************************************

TOTAL_INCOME :  2870
[ 534.1   137.9   331.2 ... 3717.2   172.9 2314.7]

**************************************************

CHANGE_IN_STOCK :  1164
[ 13.5   -3.7 -18.1 ...   41.4 306.5 321.6]
```

Table 9: Checking for Anomalies for variables in the Dataset

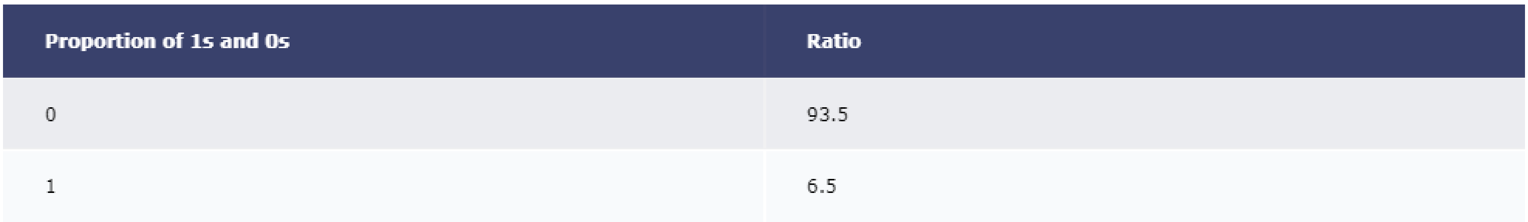
There are no anomalies present in the features. However, there are NAN values which need to be treated.

CHECKING DUPLICATE VALUES

Total number of duplicate rows in dataset is 665. Since duplicate rows do not contribute in analysis, same is dropped using .drop() function.

After dropping all duplicate rows, total number of rows is 3591.

CHECKING THE CLASS PROPORTION OF DEPENDENT VARIABLE ('DEFAULT').



Proportion of 1s and 0s	Ratio
0	93.5
1	6.5

Table 10: Checking the Class Proportion of Dependent Variable ('Default')

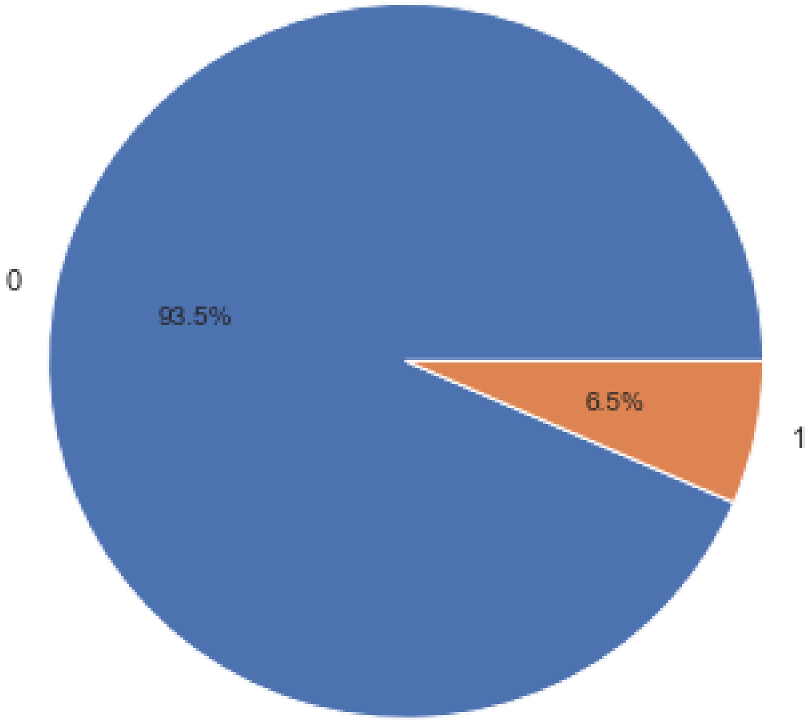


Figure 2: Pie-Plot of Class Proportion of Target Column

93.5% of datapoints belong to class ‘0’. Only 6.5% of datapoints belong to class ‘1’. As there is class imbalance, same need to be resolved using SMOTE.

UNIVARIATE ANALYSIS OF CONTINUOUS NUMERICAL VARIABLES

Histogram takes numerical variables as input and helps in identifying skewness in numerical features.

A boxplot is a standardized way of displaying the distribution of data based on a five number summary (“minimum”, first quartile (Q1), median, third quartile (Q3), and “maximum”). It can tell you about your outliers and what their values are. It can also tell you if your data is symmetrical, how tightly your data is grouped, and if and how your data is skewed.

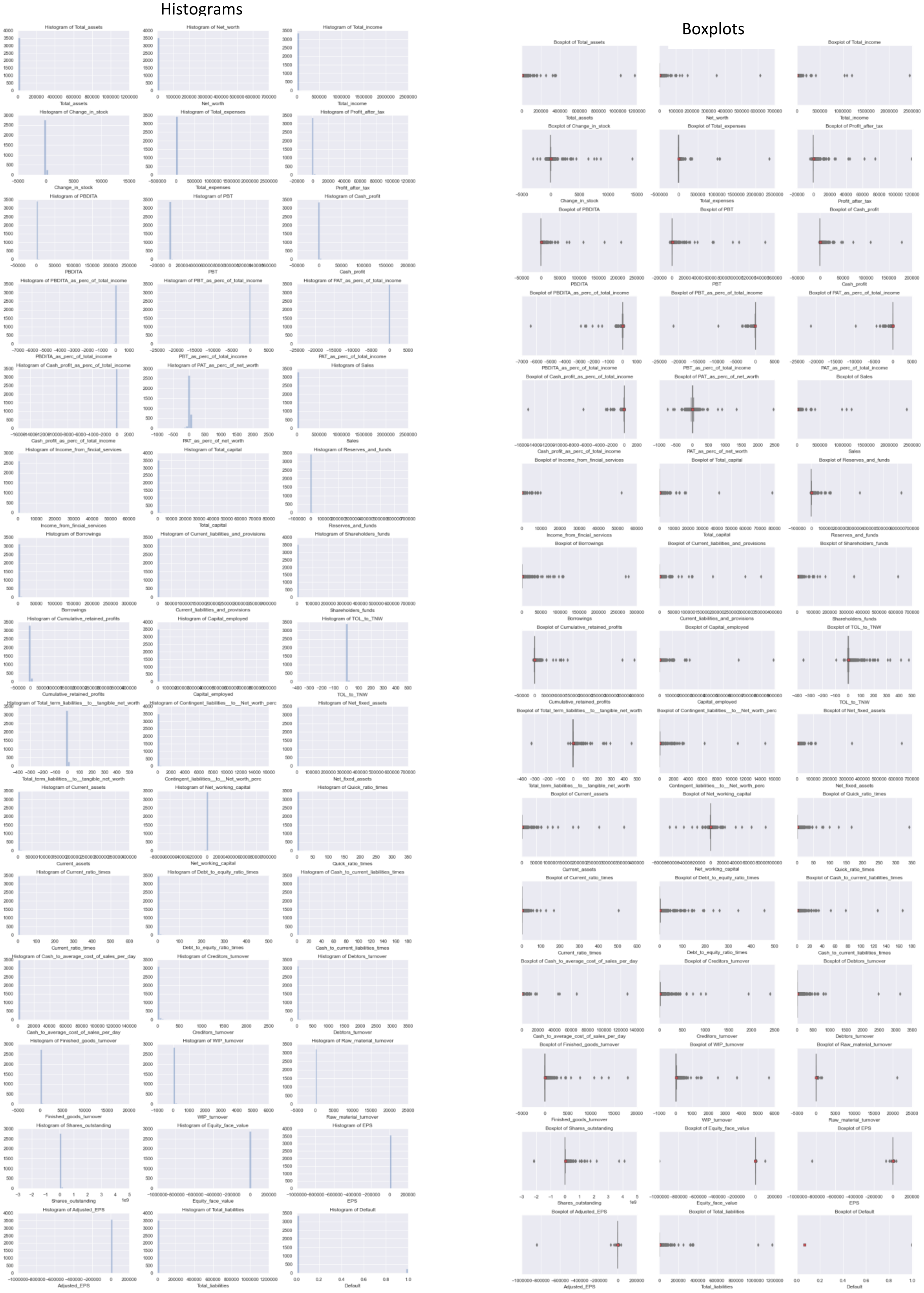


Figure 3: Histogram and Box Plots of all the Variables

Insights -

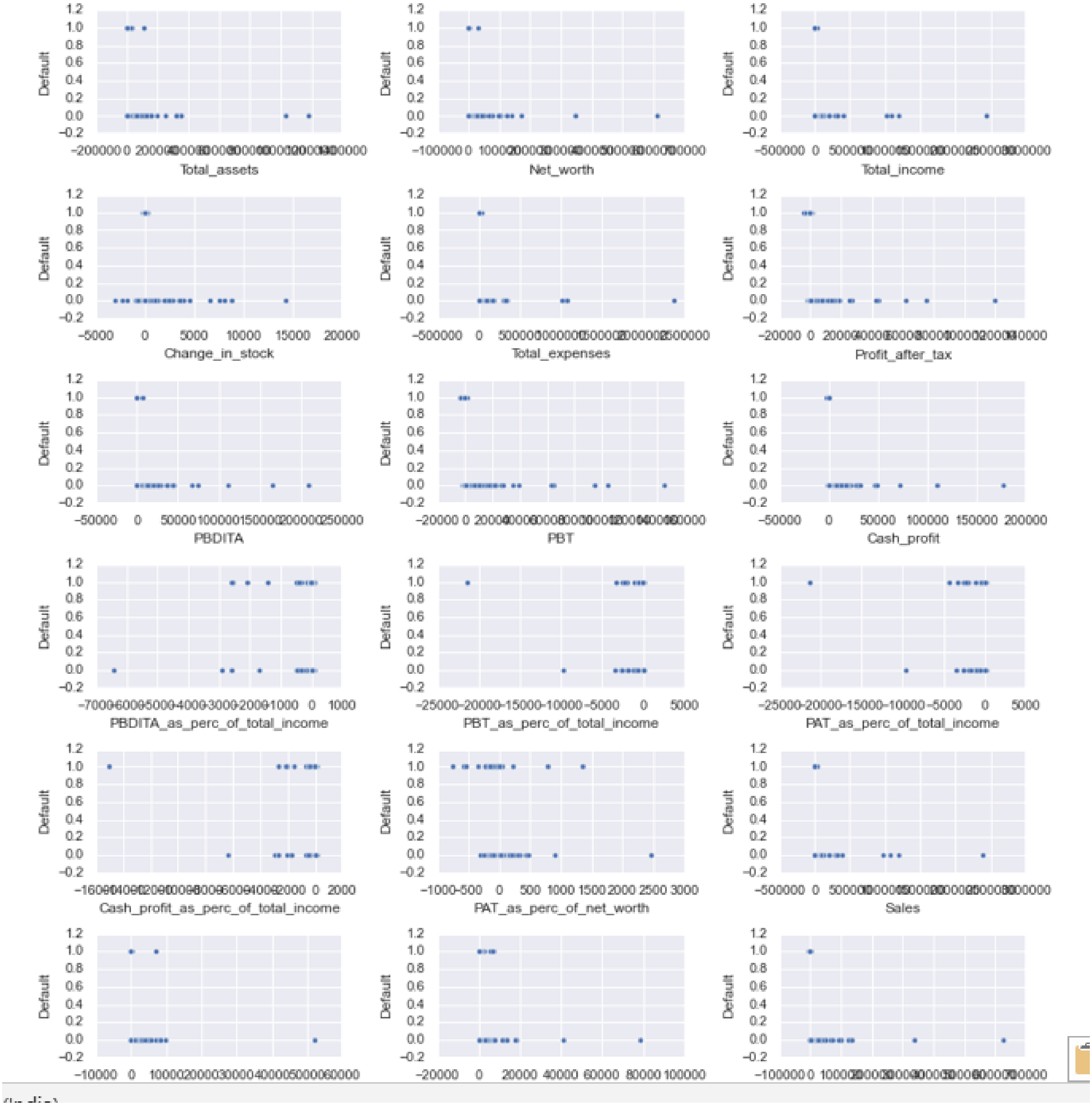
- Total_assets - Total assets of customer ranges from a minimum of 0.100 to maximum of 1176509.
- Average Total assets of customer is around 3575.297.
- Total_assets has outliers.
- Net_worth - Net worth of the customer of present year ranges from a minimum of 0.00 to maximum of 613151.60.
- Average Net worth of the customer of present year is around 1352.58.
- Net_worth has outliers.
- Total_income - Total income of the customer ranges from a minimum of 0.00 to maximum of 2442828.
- Average Total income of the customer is around 4688.190.
- Total_income has outliers.
- Change_in_stock - difference between value of current stock and the value of stock in last trading day ranges from a minimum of -3029.400 to maximum of 14185.500.
- Average difference between value of current stock and the value of stock in last trading day is around 43.702.
- Change_in_stock has outliers.
- Total_expenses - Total expense done by customer ranges from a minimum of -0.100 to maximum of 2366035.
- Average of Total expense done by customer is around 4356.301.
- Total_expenses has outliers.
- Profit_after_tax - Profit after tax deduction ranges from a minimum of -3908.300 to maximum of 119439.100.
- Average Profit after tax deduction is around 295.050.
- Profit_after_tax has outliers.
- PBDITA - Profit before depreciation, income tax and amortization ranges from a minimum of -440.700 to maximum of 208576.500.
- Average PBDITA is around 605.940.
- PBDITA has outliers.
- PBT - Profit before tax deduction ranges from a minimum of -3894.800 to maximum of 145292.600.
- Average Profit before tax deduction is around 410.259.
- PBT has outliers.
- Cash_profit - Total Cash profit ranges from a minimum of -2245.700 to maximum of 176911.800.
- Average of Total Cash profit is around 408.267.
- Cash_profit has outliers.
- PBDITA_as_perc_of_total_income - PBDITA / Total income ranges from a minimum of -6400.00 to maximum of 100.00.
- Average PBDITA_as_perc_of_total_income is around 3.181.
- PBDITA_as_perc_of_total_income has outliers.
- PBT_as_perc_of_total_income - PBT / Total income ranges from a minimum of -21340.00 to maximum of 100.00.
- Average PBT_as_perc_of_total_income is around -18.205.
- PBT_as_perc_of_total_income has outliers.
- PAT_as_perc_of_total_income - PAT / Total income ranges from a minimum of -21340.00 to maximum of 150.00.
- Average PAT_as_perc_of_total_income is around -20.043.
- PAT_as_perc_of_total_income has outliers.
- Cash_profit_as_perc_of_total_income - Cash Profit / Total income ranges from a minimum of -15020.00 to maximum of 100.00.
- Average Cash_profit_as_perc_of_total_income is around -9.025.
- Cash_profit_as_perc_of_total_income has outliers.
- PAT_as_perc_of_net_worth - PAT / Net worth ranges from a minimum of -748.720 to maximum of 2466.670.
- Average PAT_as_perc_of_net_worth is around 10.172.
- PAT_as_perc_of_net_worth has outliers.
- Sales - Sales done by customer ranges from a minimum of 0.100 to maximum of 2384984.
- Average Sales done by customer is around 4645.685.
- Sales has outliers.
- Total_capital - Total capital of the customer ranges from a minimum of 0.100 to maximum of 78273.200.
- Average Total capital of the customer is around 224.663.
- Total_capital has outliers.
- Reserves_and_funds - Total reserves and funds of the customer ranges from a minimum of -6525.900 to maximum of 625137.800.
- Average of Total reserves and funds of the customer is around 1210.561.
- Reserves_and_funds has outliers.
- Borrowings - Total amount borrowed by customer ranges from a minimum of 0.100 to maximum of 278257.300.
- Average Total amount borrowed by customer is around 1176.248.
- Borrowings has outliers.
- Current_liabilities_and_provisions - current liabilities of the customer ranges from a minimum of 0.100 to maximum of 352240.300.
- Average current liabilities of the customer is around 960.631.
- Current_liabilities_and_provisions has outliers.

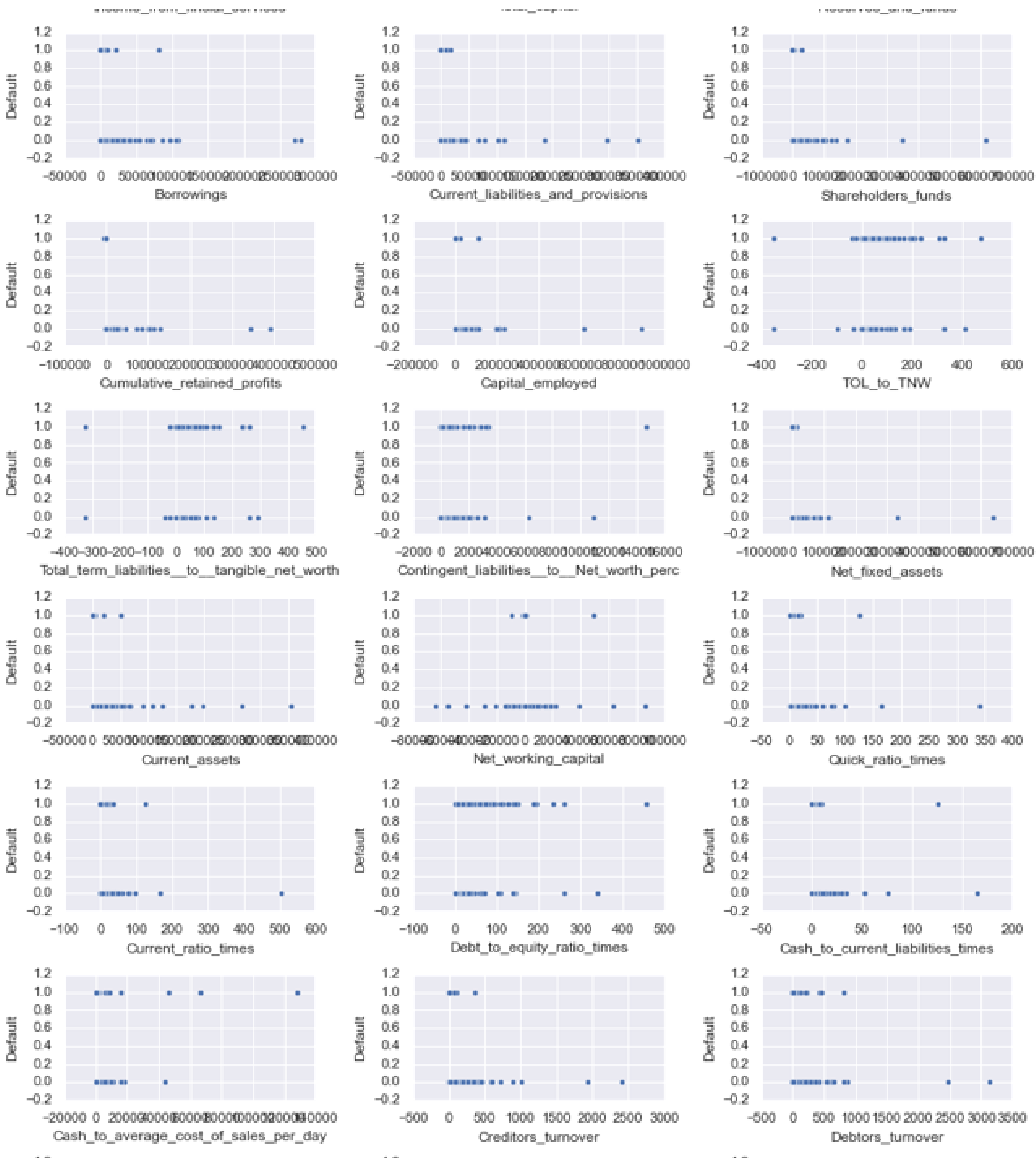
- Shareholders_funds - Amount of equity in a company, which is belong to shareholder ranges from a minimum of 0.00 to maximum of 613151.600.
- Average Amount of equity in a company, which is belong to shareholder is 1377.133.
- Shareholders_funds has outliers.
- Cumulative_retained_profits - Total cumulative profit retained by customer ranges from a minimum of -6534.300 to maximum of 390133.800.
- Average Total cumulative profit retained by customer is around 937.181.
- Cumulative_retained_profits has outliers.
- Capital_employed - Current asset minus current liabilities ranges from a minimum of 0.00 to maximum of 891408.900.
- Average Current asset minus current liabilities is around 2434.761.
- Capital_employed has outliers.
- TOL_to_TNW - Total liabilities of the customer divided by Total net worth ranges from a minimum of -350.480 to maximum of 473.00.
- Average TOL_to_TNW is around 4.027.
- TOL_to_TNW has outliers.
- Total_term_liabilities__to__tangible_net_worth - Short + long term liabilities divided by tangible net worth ranges from a minimum of -325.600 to maximum of 456.00.
- Average Total_term_liabilities__to__tangible_net_worth is around 1.855.
- Total_term_liabilities__to__tangible_net_worth has outliers.
- Contingent_liabilities__to__Net_worth_perc - Contingent liabilities / Net worth ranges from a minimum of 0.00 to maximum of 14704.270.
- Average Contingent_liabilities__to__Net_worth_perc is 55.733.
- Contingent_liabilities__to__Net_worth_perc has outliers.
- Net_fixed_assets - purchase price of all fixed assets ranges from a minimum of 0.00 to maximum of 636604.600.
- Average Net_fixed_assets is around 1209.486.
- Net_fixed_assets has outliers.
- Current_assets - Assets that are expected to be converted to cash within a year ranges from a minimum of 0.100 to maximum of 354815.200.
- Average Current_assets is around 1351.006.
- Current_assets has outliers.
- Net_working_capital - Difference of current liabilities and current assets ranges from a minimum of -63839.00 to maximum of 85782.800.
- Average Net_working_capital is around 162.951.
- Net_working_capital has outliers.
- Quick_ratio_times - Total cash divided by current liabilities ranges from a minimum of 0.00 to maximum of 341.00.
- Average Quick_ratio_times is around 1.497.
- Quick_ratio_times has outliers.
- Current_ratio_times - Current assets divided by current liabilities ranges from a minimum of 0.00 to maximum of 505.00.
- Average Current_ratio_times is around 2.257.
- Current_ratio_times has outliers.
- Debt_to_equity_ratio_times - Total liabilities divided by its shareholder equity ranges from a minimum of 0.00 to maximum of 456.00.
- Average Debt_to_equity_ratio_times is around 2.872.
- Debt_to_equity_ratio_times has outliers.
- Cash_to_current_liabilities_times - Total liquid cash divided by current liabilities ranges from a minimum of 0.00 to maximum of 165.00.
- Average Cash_to_current_liabilities_times is around 0.528.
- Cash_to_current_liabilities_times has outliers.
- Cash_to_average_cost_of_sales_per_day - Total cash divided by average cost of the sales ranges from a minimum of 0.00 to maximum of 128040.760.
- Average Cash_to_average_cost_of_sales_per_day is around 145.157.
- Cash_to_average_cost_of_sales_per_day has outliers.
- Creditors_turnover - Net credit purchase divided to average trade creditors ranges from a minimum of 0.00 to maximum of 2401.00.
- Average Creditors_turnover is around 16.812.
- Creditors_turnover has outliers.
- Debtors_turnover - Net credit sales divided by average accounts receivable ranges from a minimum of 0.00 to maximum of 3135.200.
- Average Debtors_turnover is around 17.929.
- Debtors_turnover has outliers.
- Finished_goods_turnover - Annual sales divided by average inventory ranges from a minimum of -0.090 to maximum of 17947.600.

- Average Finished_goods_turnover is around 84.369.
- Finished_goods_turnover has outliers.
- WIP_turnover - The cost of goods sold for a period divided by the average inventory for that period ranges from a minimum of -0.180 to maximum of 5651.400.
- Average WIP_turnover is around 28.684.
- WIP_turnover has outliers.
- Raw_material_turnover - Cost of goods sold is divided by the average inventory for the same period ranges from a minimum of -2.000 to maximum of 21092.000.
- Average Raw_material_turnover is around 17.733.
- Raw_material_turnover has outliers.
- Shares_outstanding - Number of issued shares minus the number of share held in the company ranges from a minimum of -2147484000 to maximum of 4130401000.
- Average Shares_outstanding is around 23764910.
- Shares_outstanding has outliers.
- Equity_face_value - cost of the equity at the time of issuing ranges from a minimum of -999998.900 to maximum of 100000.000.
- Average Equity_face_value is around -1094.828.
- Equity_face_value has outliers.
- EPS - Net income divided by total number of outstanding share ranges from a minimum of -843181.820 to maximum of 34522.530.
- Average EPS is around -196.309.
- EPS has outliers.
- Adjusted_EPS - Adjusted net earning divided by the weighted average number of common share outstanding on a diluted basis during the plan year ranges from a minimum of -843181.820 to maximum of 34522.530.
- Average Adjusted_EPS is around -197.620.
- Adjusted_EPS has outliers.
- Total_liabilities - Sum of all type of liabilities ranges from a minimum of 0.100 to maximum of 1176509.
- Average Total_liabilities is around 3575.297.
- Total_liabilities has outliers.

BIVARIATE ANALYSIS

Scatter Plot is a graph in which the values of two variables are plotted along two axes, the pattern of the resulting points revealing any correlation present.


Default
1.2
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
Total_assets
Net_worth
Total_income
Change_in_stock
Total_expenses
Profit_after_tax
PBDITA
PBT
Cash_profit
PBDITA_as_perc_of_total_income
PBT_as_perc_of_total_income
PAT_as_perc_of_total_income
Cash_profit_as_perc_of_total_income
PAT_as_perc_of_net_worth
Sales
−5000 0 5000 10000 15000 20000
−50000 0 50000 100000 150000 200000 250000
−50000 0 50000 100000 150000 200000
−25000 −20000 −15000 −10000 −5000 0 5000
−1000 −500 0 500 1000 1500 2000 2500 3000
−10000 0 10000 20000 30000 40000 50000 60000
−20000 0 20000 40000 60000 80000 100000


Default
1.2
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
Borrowings
Current_liabilities_and_provisions
Shareholders_funds
Cumulative_retained_profits
Capital_employed
TOL_to_TNW
Total_term_liabilities__to__tangible_net_worth
Contingent_liabilities__to__Net_worth_perc
Net_fixed_assets
Current_assets
Net_working_capital
Quick_ratio_times
Current_ratio_times
Debt_to_equity_ratio_times
Cash_to_current_liabilities_times
Cash_to_average_cost_of_sales_per_day
Creditors_turnover
Debtors_turnover

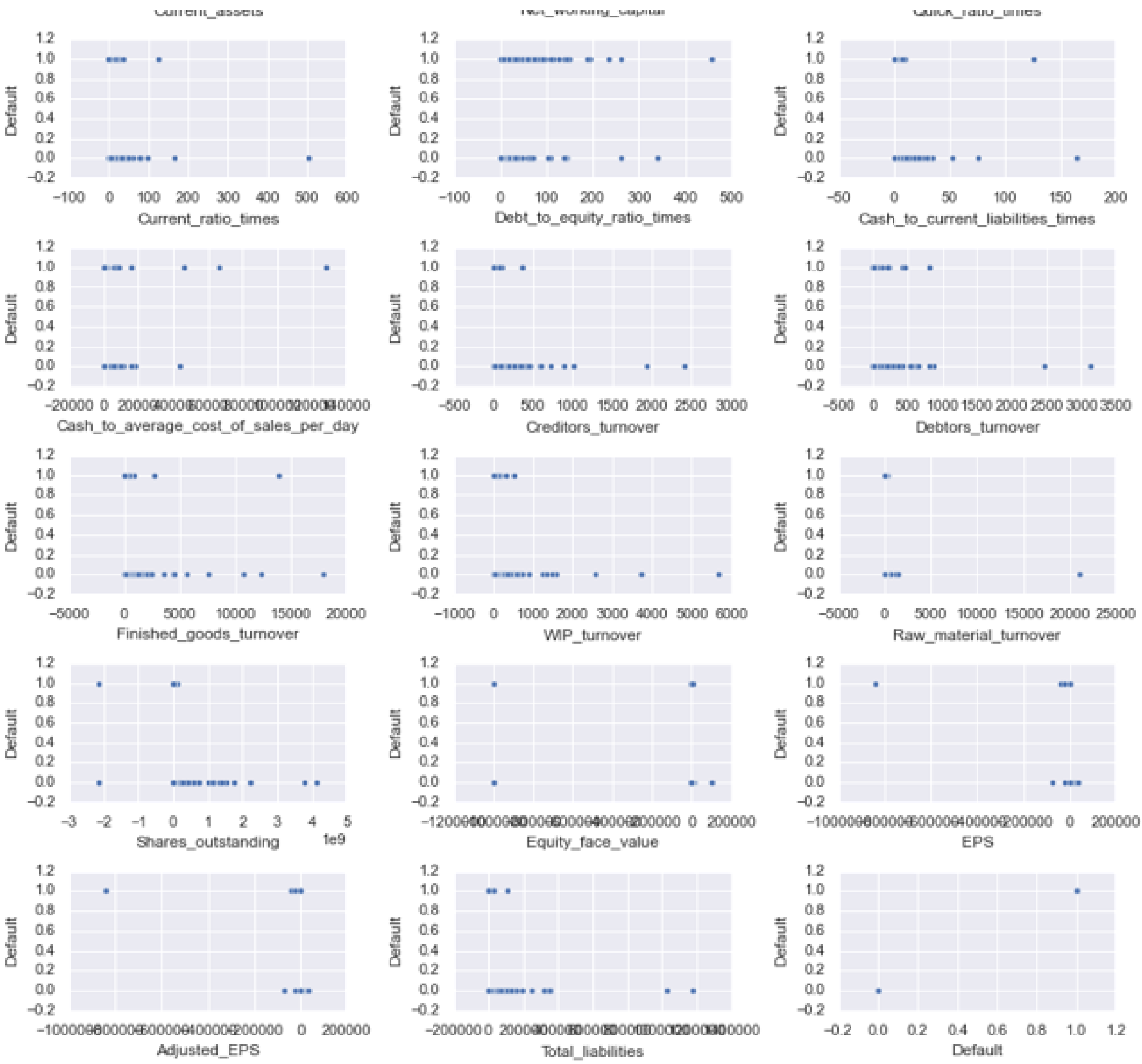


Figure 4: Scatter Plot of Default VS All the Variables

Insights:

For the upper range (-1000 – 0) of PBDITA as % of total income, chances are high that the customer will default.

For the upper range (-4000 – 0) of PBT as % of total income, chances are high that the customer will default.

For the upper range (-5000 -0) of PAT as % of total income, chances are high that the customer will default.

For the upper range (-3000 – 0) of Cash profit as % of total income, chances are high that the customer will default.

For the lower range (-500 – 0) of PAT as % of net worth , chances are high that the customer will default.

Customers in range 0-200 of TOL_to_TNW seem more likely to default.

Customers in range 0-300 of Total_term_liabilities seem more likely to default.

Customers in range 0-4000 of Contingent_liabilities seem more likely to default.

Customers in range 0-250 of Debt_to_equity_ratio_times seem more likely to default.

Customers in range 0-500 of creditors_turnover seems more likely to default.

Customers in range 0-1000 of debtors_turnover seems more likely to default.

Customers in range 0-2500 of Finished_goods_turnover seems more likely to default.

MULTIVARIATE ANALYSIS

The **heat map** can also be used to check the association between two numeric variables. All the boxes with a value higher than 0.8 are highly correlated. But in the given data set none of the variables have a value 0.8 or more. The heat map for all the numerical variables is below.

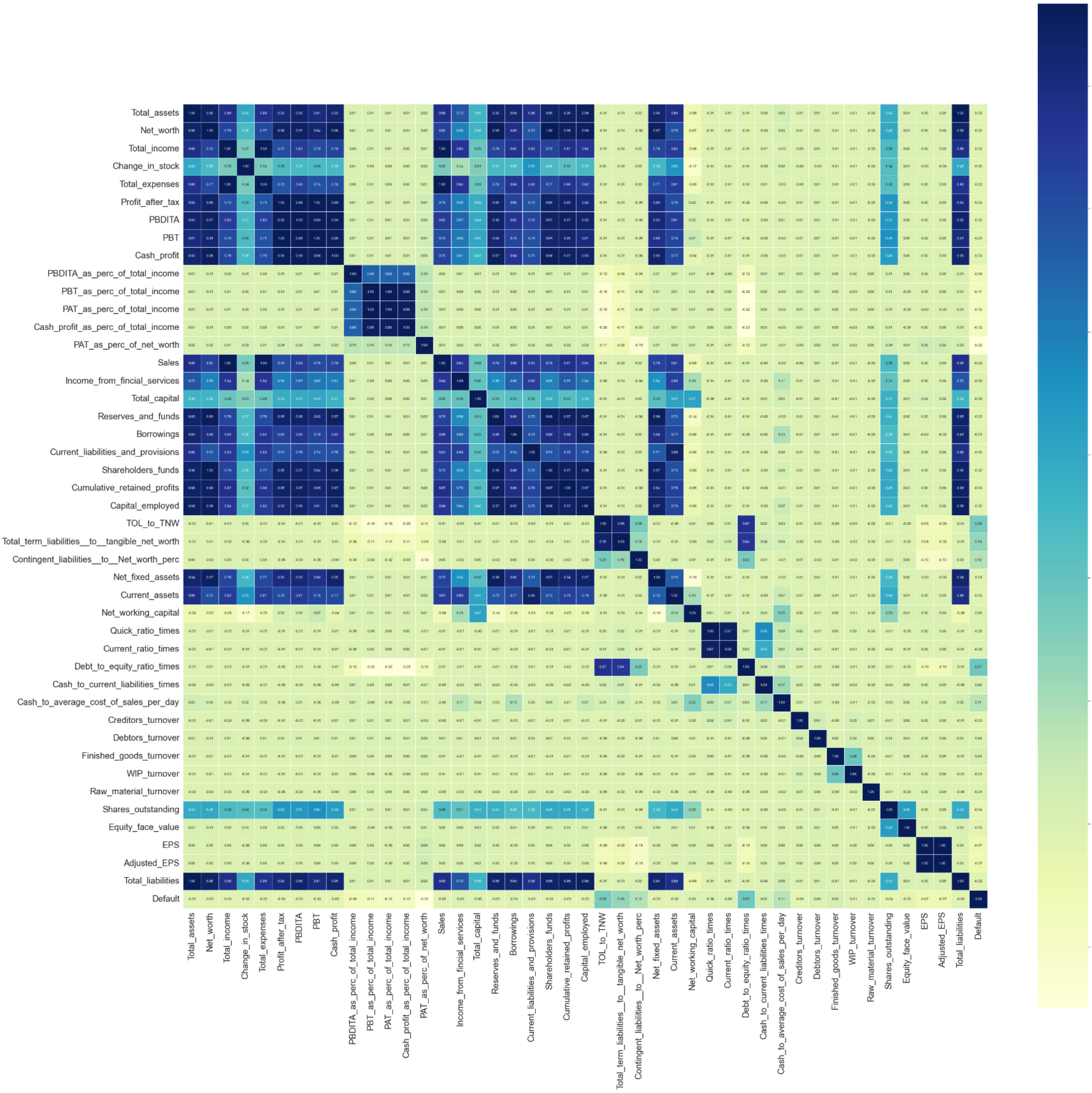


Figure 5: Heat Map of Variables to Check Multicollinearity

As seen from the heat map, variables shown in dark blue are highly corelated.

Perfect correlation is present between these variables:

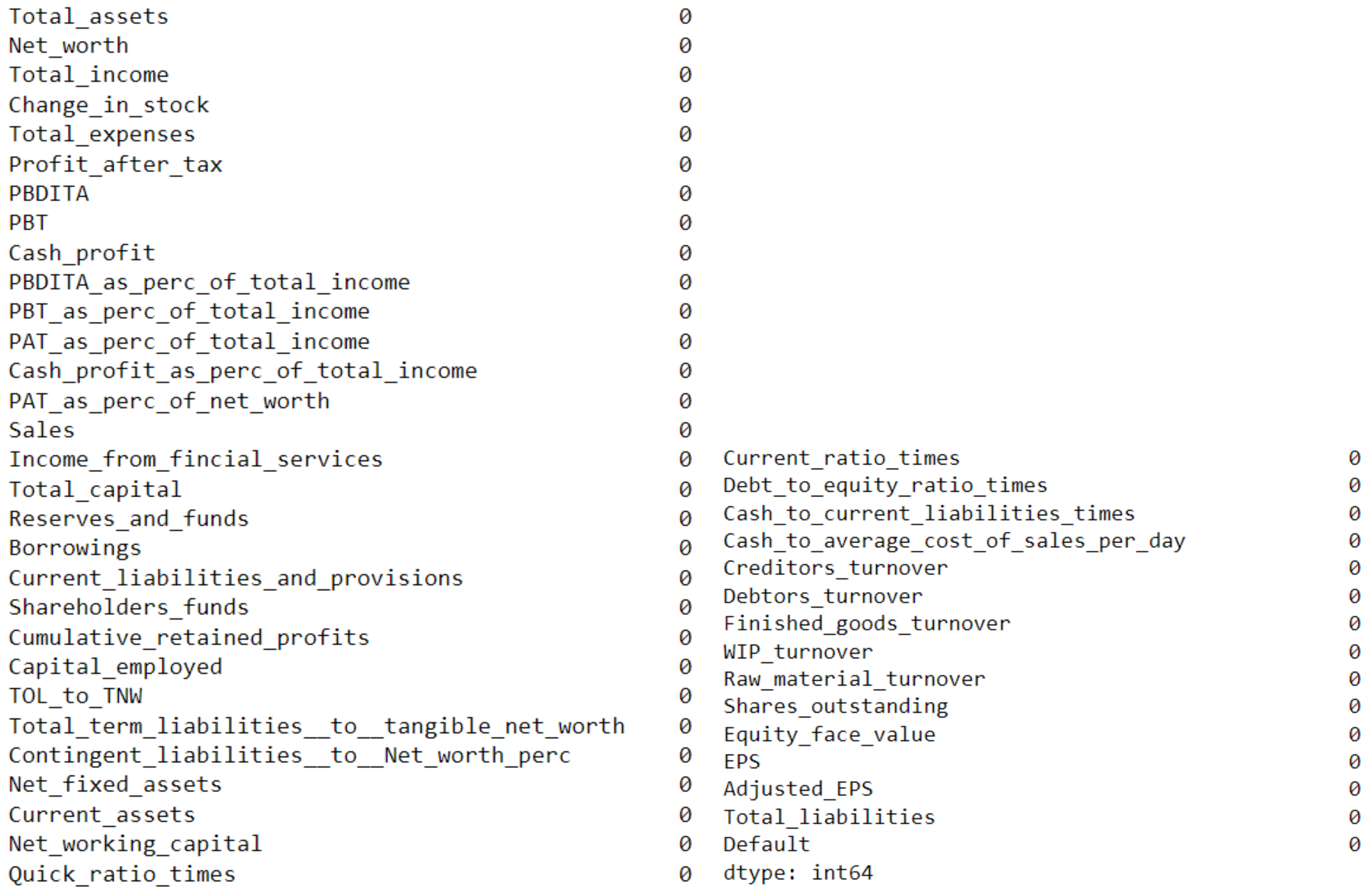
- Net_worth - Total_assets
- Total_income - Total_assets, Net_worth
- Total_expenses - Total_assets, Net_worth, Total_income
- Profit_after_tax - Total_assets, Net_worth, Total_income, Total_expenses
- Total_liabilities - Total_assets, Net_worth, Total_income, Total_expenses, Profit_after_tax, PBDITA, PBT, Cash_profit, Sales, Other_income, Borrowings, Current_liabilities_and_provisions, Deferred_tax_liability, Shareholders_funds, Cumulative_retained_profits, Capital_employed

Feature reduction technique such as VIF will be used to extract important features for model building.

NULL VALUES TREATMENT

We have already removed features with null values more than 30%. For rest of the features having null values, we will impute them with median using simple imputer.

Null Values After Imputation



```
Total_assets                                  0
Net_worth                                     0
Total_income                                  0
Change_in_stock                               0
Total_expenses                                0
Profit_after_tax                              0
PBDITA                                        0
PBT                                           0
Cash_profit                                   0
PBDITA_as_perc_of_total_income                0
PBT_as_perc_of_total_income                   0
PAT_as_perc_of_total_income                   0
Cash_profit_as_perc_of_total_income           0
PAT_as_perc_of_net_worth                      0
Sales                                         0
Income_from_fincial_services                  0
Total_capital                                 0
Reserves_and_funds                            0
Borrowings                                    0
Current_liabilities_and_provisions            0
Shareholders_funds                            0
Cumulative_retained_profits                   0
Capital_employed                              0
TOL_to_TNW                                    0
Total_term_liabilities__to__tangible_net_worth    0
Contingent_liabilities__to__Net_worth_perc    0
Net_fixed_assets                              0
Current_assets                                0
Net_working_capital                           0
Quick_ratio_times                             0
```

```
Current_ratio_times                           0
Debt_to_equity_ratio_times                    0
Cash_to_current_liabilities_times             0
Cash_to_average_cost_of_sales_per_day         0
Creditors_turnover                            0
Debtors_turnover                              0
Finished_goods_turnover                       0
WIP_turnover                                  0
Raw_material_turnover                         0
Shares_outstanding                            0
Equity_face_value                             0
EPS                                           0
Adjusted_EPS                                  0
Total_liabilities                             0
Default                                       0
dtype: int64
```

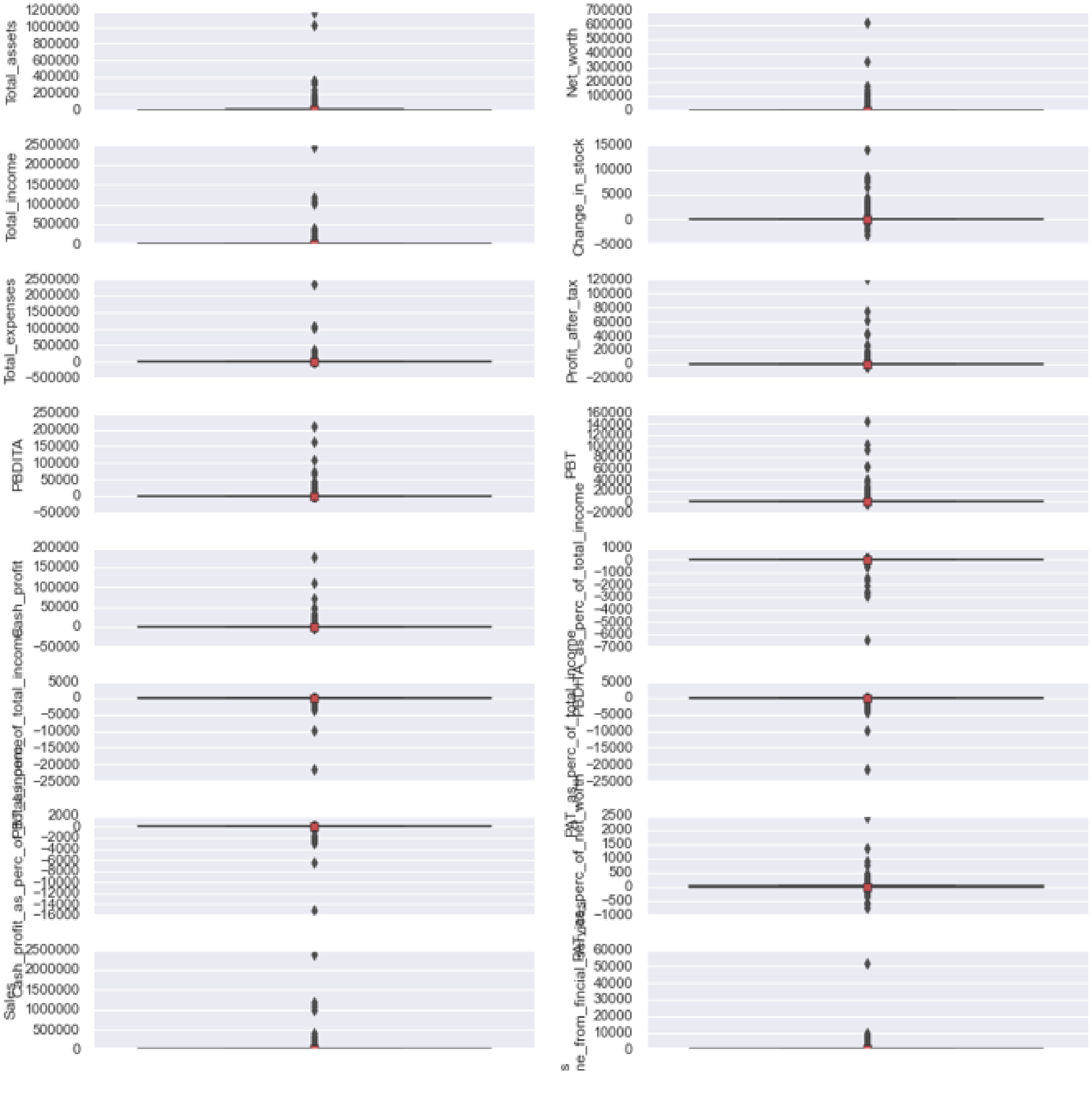
Table 11: Checking for Null Values After Imputation

From above table, we can conclude that there are no null values present in any features.

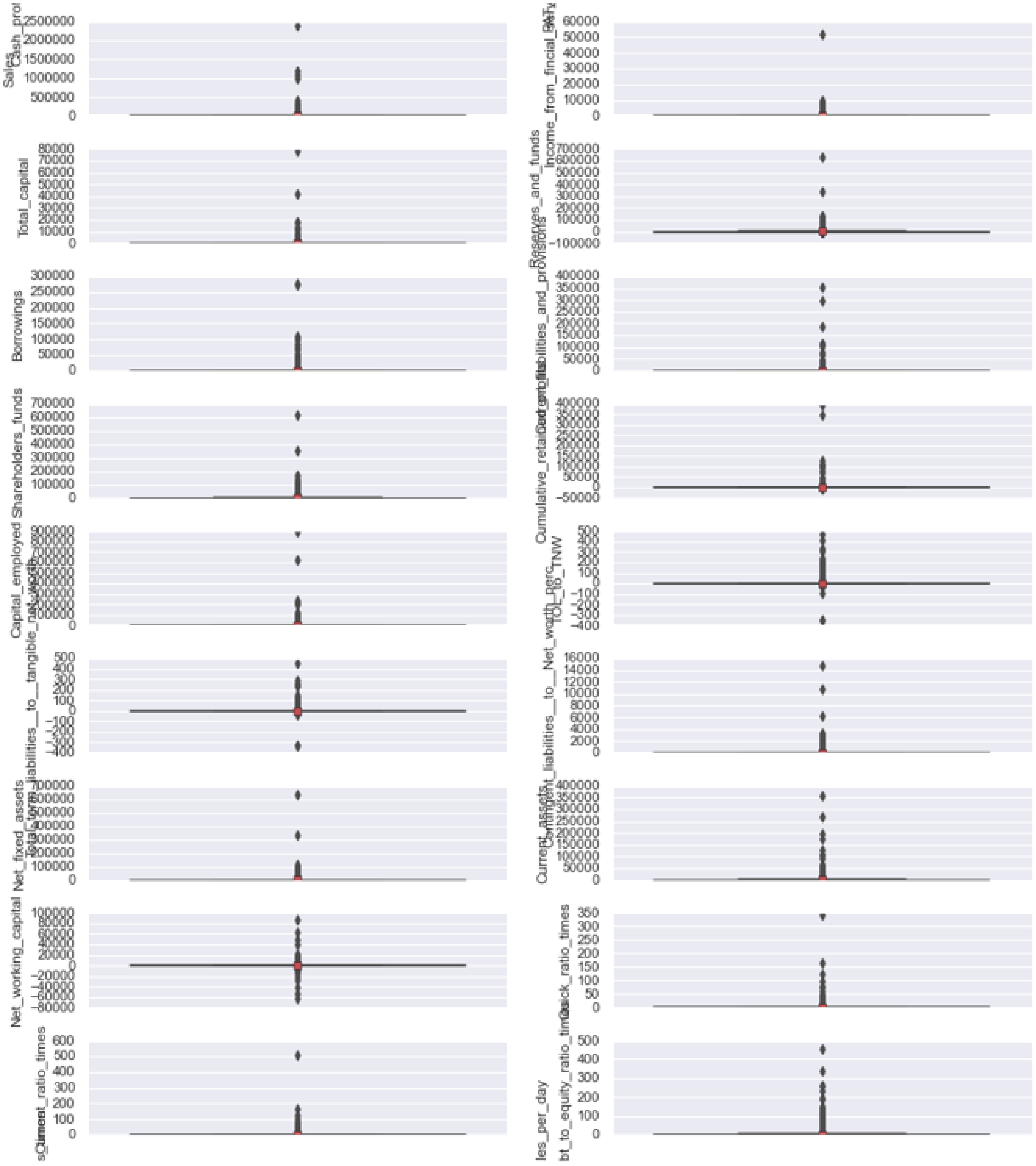
OUTLIERS

As many of the continuous variables has outliers and extreme values which shall be removed as many of the Machine learning algorithm such as Logistic Regression are sensitive to outliers.

Any values above 1.5 x IQR from Q3 shall be floored to that limit, likewise any values below 1.5 x IQR from Q1 shall be capped to that lower limit. IQR shall be calculated as difference between Q3 and Q1.



Total_assets
1200000
1000000
800000
600000
400000
200000
0
Net_worth
700000
600000
500000
400000
300000
200000
100000
0
Total_income
2500000
2000000
1500000
1000000
500000
0
Change_in_stock
15000
10000
5000
0
−5000
Total_expenses
2500000
2000000
1500000
1000000
500000
0
−500000
Profit_after_tax
120000
100000
80000
60000
40000
20000
0
−20000
PBDITA
250000
200000
150000
100000
50000
0
−50000
PBT
160000
140000
120000
100000
80000
60000
40000
20000
0
−20000
Cash_profit
200000
150000
100000
50000
0
−50000
PAT_as_perc_of_total_income
1000
0
−1000
−2000
−3000
−4000
−5000
−6000
−7000
PBT_as_perc_of_total_income
5000
0
−5000
−10000
−15000
−20000
−25000
PBDITA_as_perc_of_total_income
5000
0
−5000
−10000
−15000
−20000
−25000
Cash_profit_as_perc_of_total_income
2000
0
−2000
−4000
−6000
−8000
−10000
−12000
−14000
−16000
PAT_as_perc_of_net_worth
2500
2000
1500
1000
500
0
−500
−1000
Sales
2500000
2000000
1500000
1000000
500000
0
Income_from_fincial_services
60000
50000
40000
30000
20000
10000
0


Sales
Cash_prof
Income_from_fincial_serv
PAT
Total_capital
Reserves_and_funds
Borrowings
Current_liabilities_and_provisions
Shareholders_funds
Cumulative_retained_profits
Capital_employed
TOL_to_TNW
Net_worth_perc
Total_term_liabilities__to__tangible_net_worth
Contingent_liabilities__to__Net_worth_perc
Net_fixed_assets
Current_assets
Net_working_capital
Quick_ratio_times
Current_ratio_times
Debt_to_equity_ratio_times
Sales_per_day
2500000
2000000
1500000
1000000
500000
0
60000
50000
40000
30000
20000
10000
80000
70000
700000
600000
500000
400000
300000
200000
100000
-100000
300000
250000
150000
50000
350000
16000
14000
12000
10000
8000
6000
4000
2000
900000
800000
500
400
300
200
100
-100
-200
-300
-400
-20000
-40000
-60000
-80000
250
150
50
600

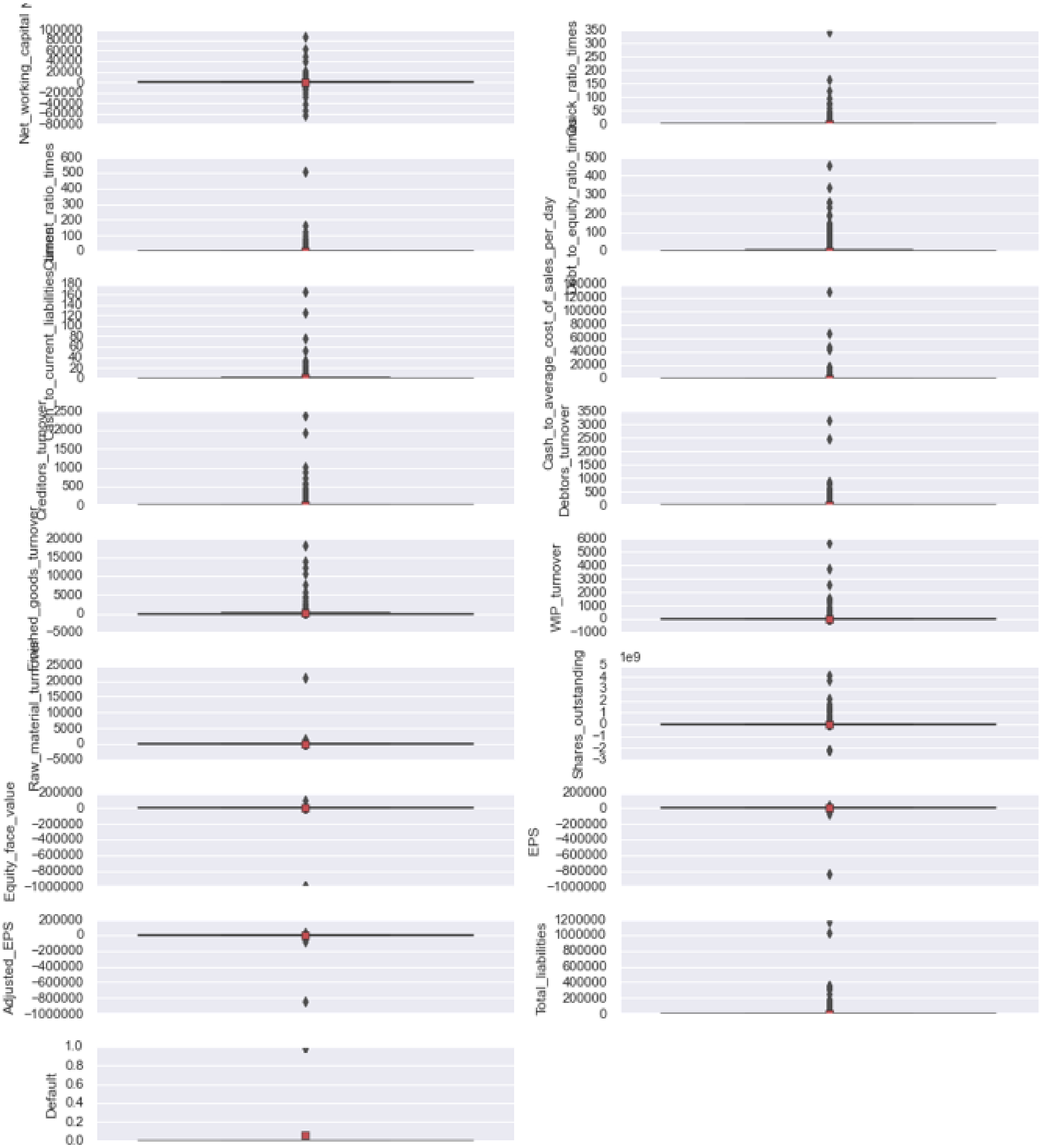


Figure 6: Checking for Outliers in the dataset

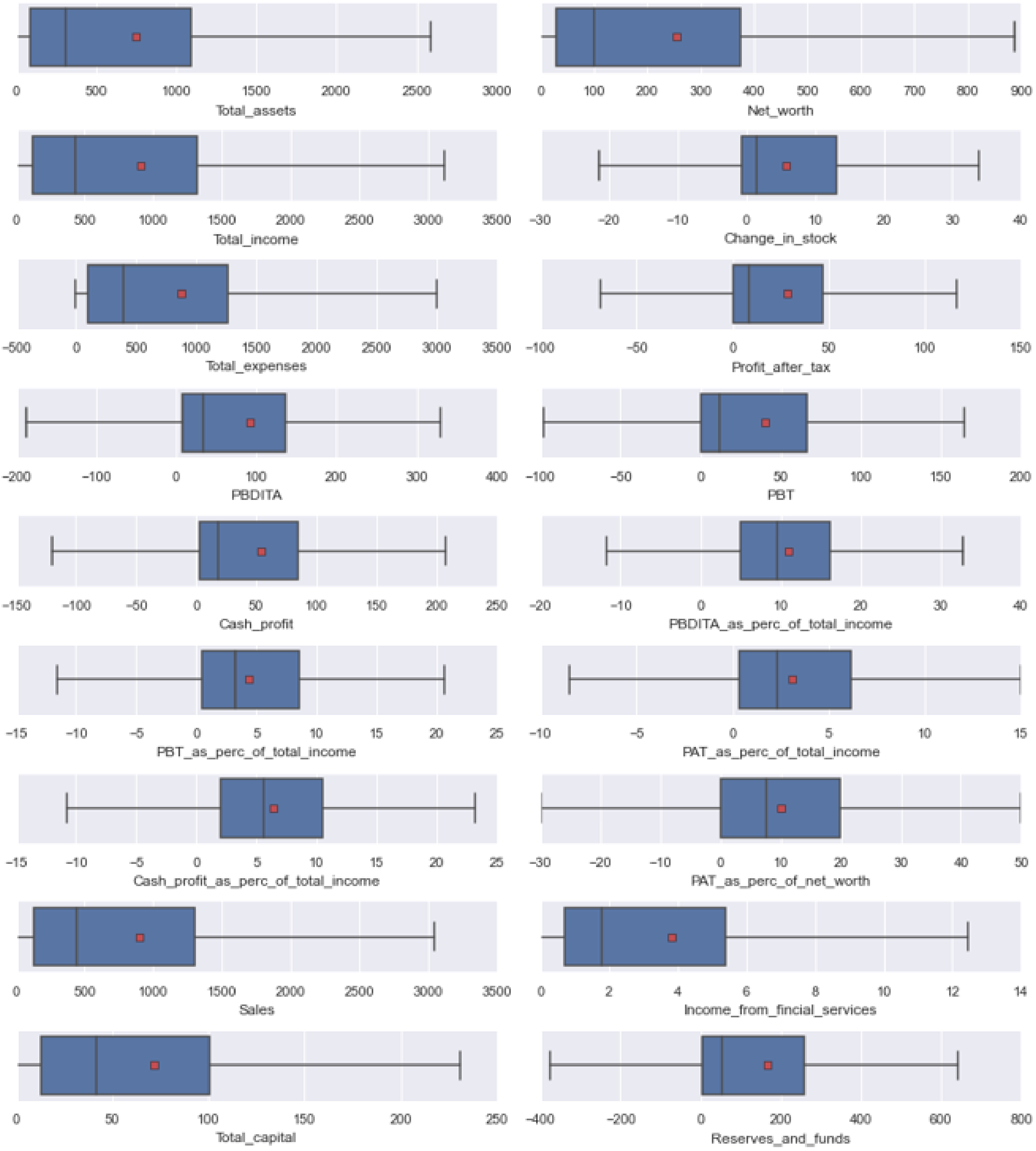
From above graph, its understood that outliers are present in the features. Outlier treatment is performed only on independent variables. Accordingly data is split into x and y before outlier treatment.

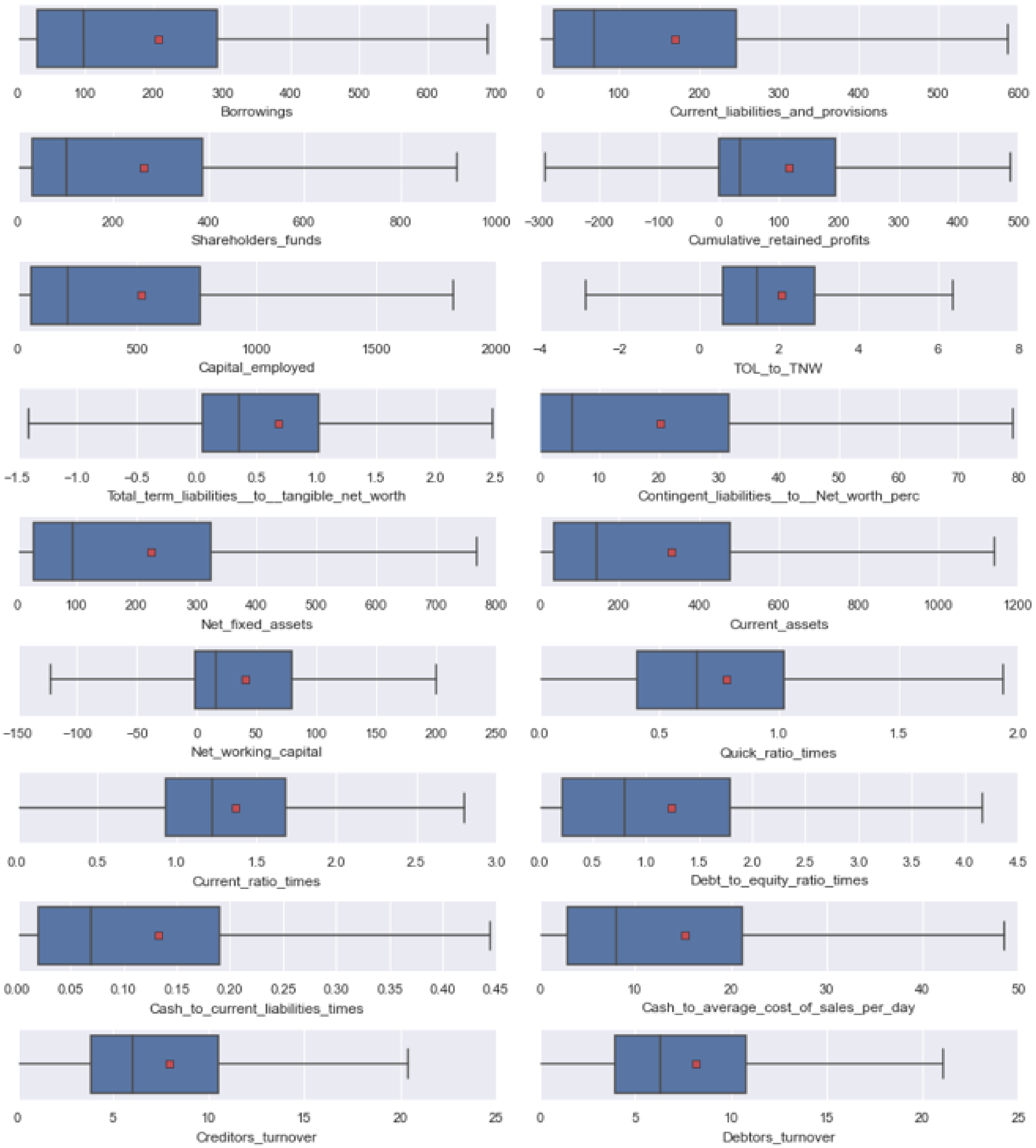
TREATMENT OF OUTLIERS

The outlier in the data set is treated using the IQR method. Inter quartile range (IQR) method – Each dataset can be divided into quartiles. The first quartile point indicates that 25% of the data points are below that value whereas the second quartile is considered as the median point of the dataset.

The inter quartile method finds the outliers on numerical datasets by following the procedure below.

Find the first quartile, Q1. Find the third quartile, Q3. Calculate the IQR. IQR= Q3-Q1. Define the normal data range with lower limit as Q1–1.5*IQR and upper limit as Q3+1.5*IQR. Any data point outside this range is considered an outlier and should be removed for further analysis. The concept of quartiles and IQR can best be visualized from the boxplot. It has the minimum and maximum point defined as Q1–1.5*IQR and Q3+1.5*IQR respectively. Any point outside this range is outlier.




0 100 200 300 400 500 600 700
Borrowings
0 100 200 300 400 500 600
Current_liabilities_and_provisions
0 200 400 600 800 1000
Shareholders_funds
−300 −200 −100 0 100 200 300 400 500
Cumulative_retained_profits
0 500 1000 1500 2000
Capital_employed
−4 −2 0 2 4 6 8
TOL_to_TNW
−1.5 −1.0 −0.5 0.0 0.5 1.0 1.5 2.0 2.5
Total_term_liabilities__to__tangible_net_worth
0 10 20 30 40 50 60 70 80
Contingent_liabilities__to__Net_worth_perc
0 100 200 300 400 500 600 700 800
Net_fixed_assets
0 200 400 600 800 1000 1200
Current_assets
−150 −100 −50 0 50 100 150 200 250
Net_working_capital
0.0 0.5 1.0 1.5 2.0
Quick_ratio_times
0.0 0.5 1.0 1.5 2.0 2.5 3.0
Current_ratio_times
0.0 0.5 1.0 1.5 2.0 2.5 3.0 3.5 4.0 4.5
Debt_to_equity_ratio_times
0.00 0.05 0.10 0.15 0.20 0.25 0.30 0.35 0.40 0.45
Cash_to_current_liabilities_times
0 10 20 30 40 50
Cash_to_average_cost_of_sales_per_day
0 5 10 15 20 25
Creditors_turnover
0 5 10 15 20 25
Debtors_turnover

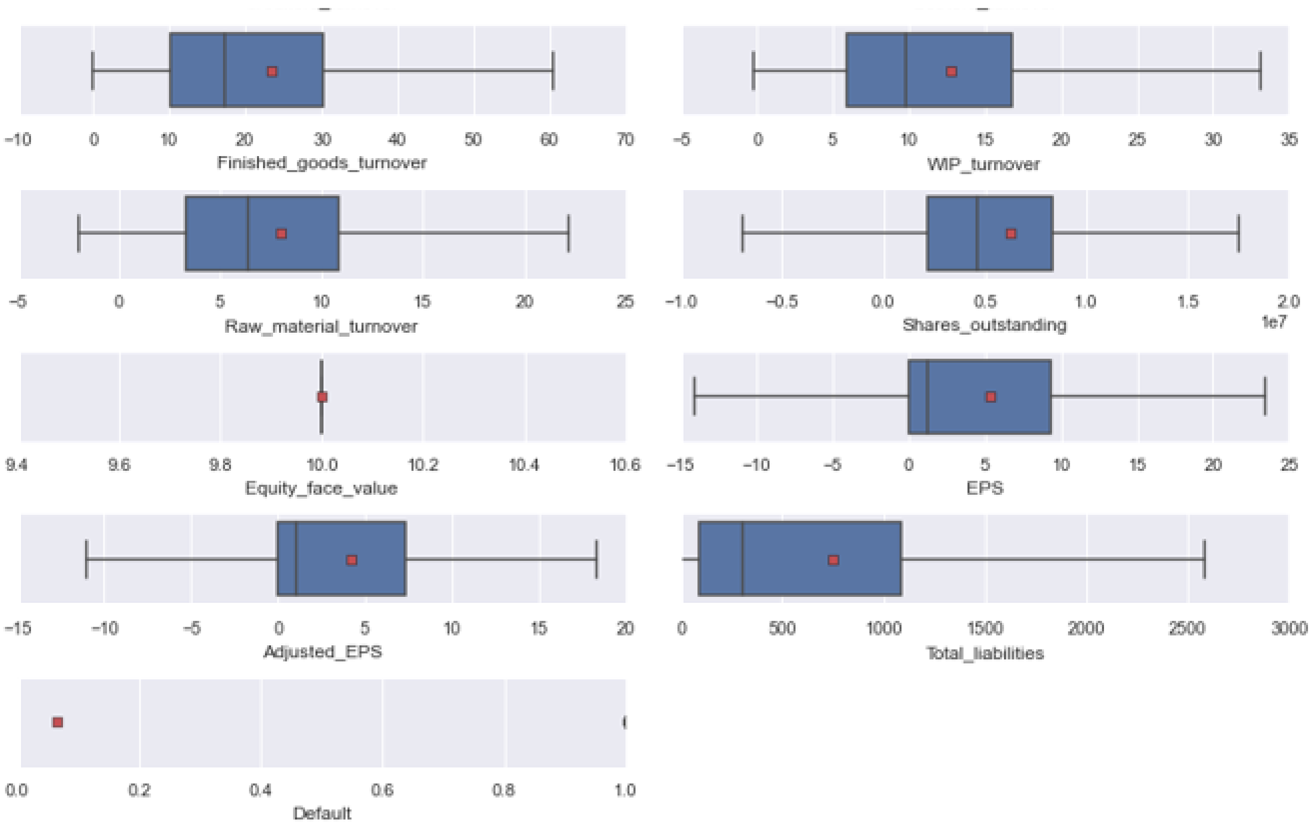


Figure 7: Checking for Outliers in the dataset after Outlier Treatment

From the above Boxplots, we can conclude that outliers for all the independent variables are successfully treated. Now we are ready for model building exercise.

MODEL BUILDING

We need to build Logistic Regression and Random Forest Model with the given data. For building these models, we need to split models into train and test set in 70:30 ratio.

From value_counts(), we get that only 6.5% are ‘1’ and rest are ‘0’, which confirms class imbalance and may cause model to either overfit or underfit.

Hence, we create models on the given dataset and check model performance on train and test set. After this we apply SMOTE on train set to remove class imbalance and validate model on train and test set.

Dimensions of the Train and Test Data:

```
Shape of x_train :  (2513, 44)
Shape of x_test :  (1078, 44)
Shape of y_train :  (2513,)
Shape of y_test :  (1078,)
```

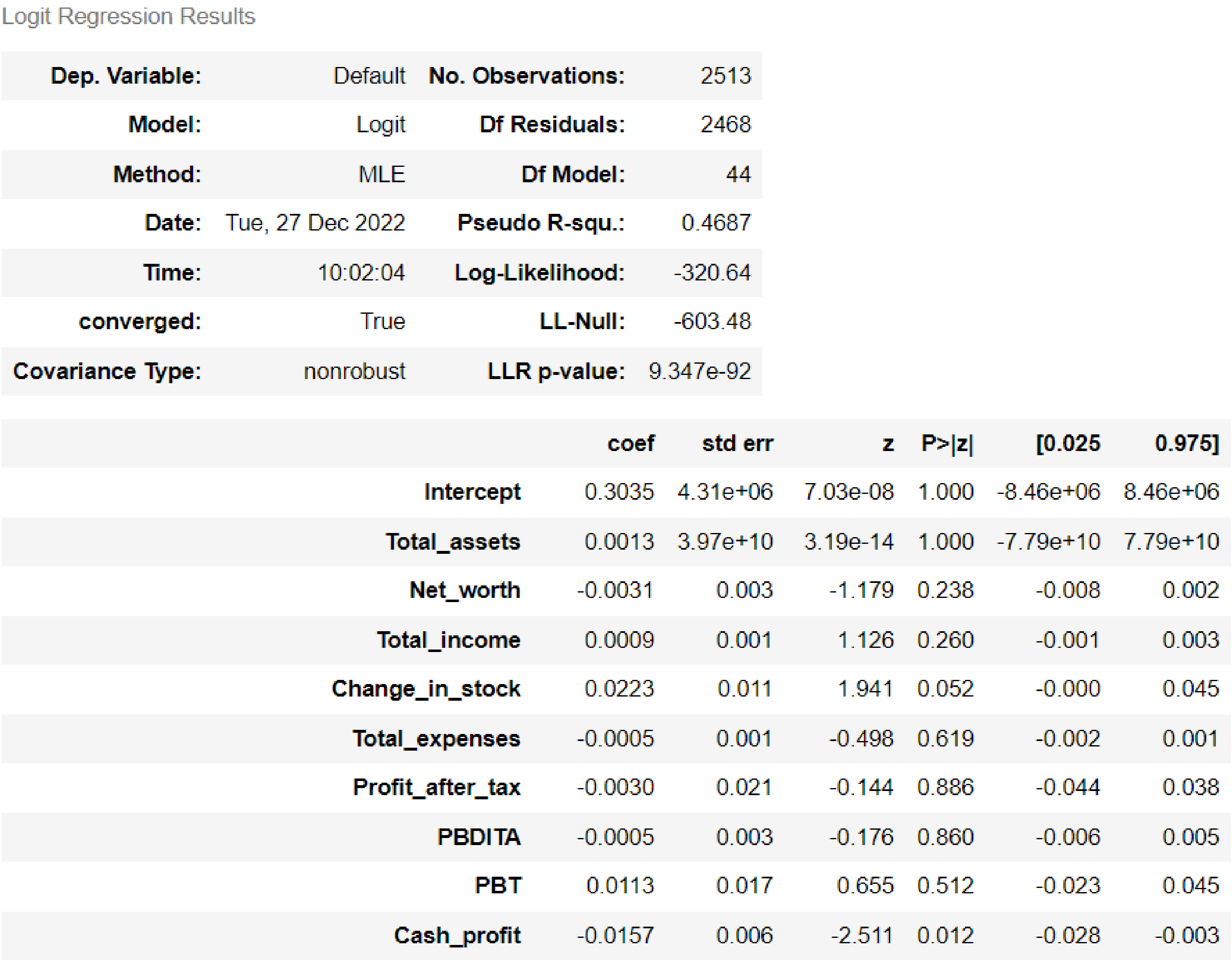
LOGISTIC REGRESSION USING STATSMODEL LIBRARY

Logistic regression is a machine learning algorithm used for predictive analysis and is probability based. The hypothesis of logistic regression tends to limit the cost function between 0 and 1.

In logistic regression, the dependent variable is a binary variable that contains data coded as 1 (yes, success, etc.) or 0 (no, failure, etc.). In other words, the logistic regression model predicts P(Y=1) as a function of X. Logistic Regression uses a more complex cost function, this cost function can be defined as the 'Sigmoid function' or also known as the 'logistic function' - In order to map predicted values to probabilities, we use the Sigmoid function. The function maps any real value into another value between 0 and 1. In machine learning, we use sigmoid to map predictions to probabilities.

Logistic Regression Model Using All Variables

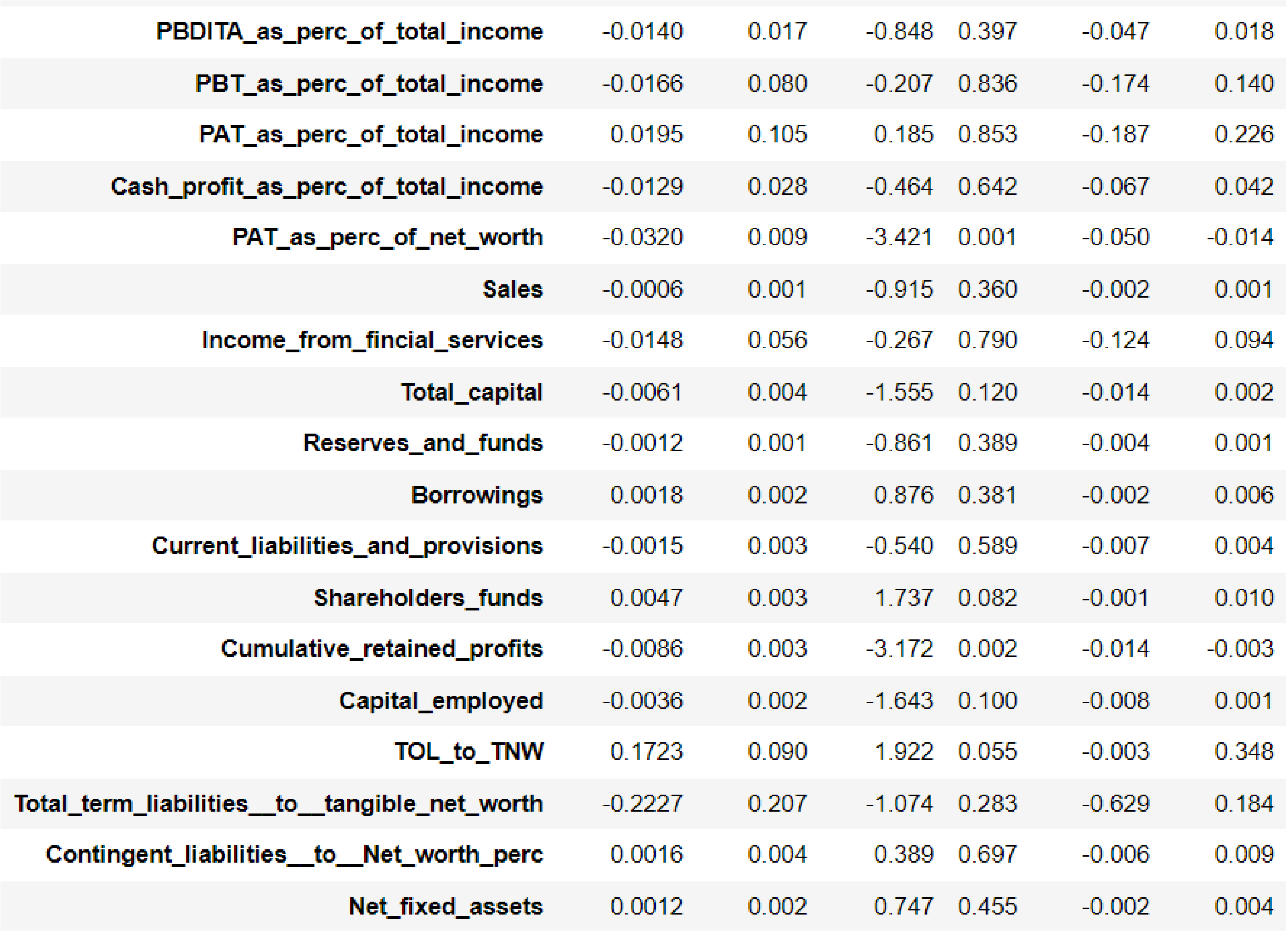
Summary of Logistic Regression

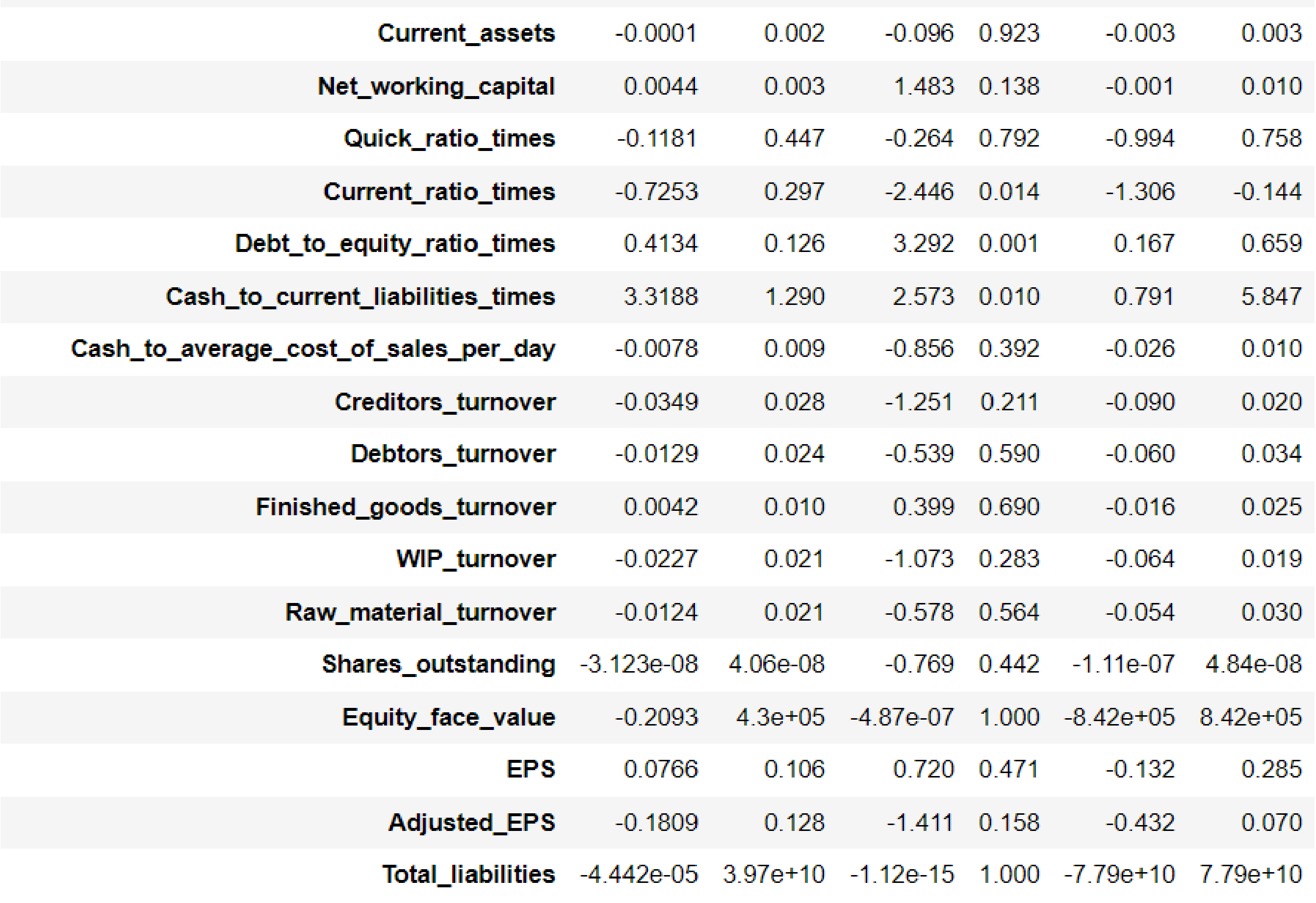


Logit Regression Results

Dep. Variable:	Default	**No. Observations:**	2513
Model:	Logit	**Df Residuals:**	2468
Method:	MLE	**Df Model:**	44
Date:	Tue, 27 Dec 2022	**Pseudo R-squ.:**	0.4687
Time:	10:02:04	**Log-Likelihood:**	-320.64
converged:	True	**LL-Null:**	-603.48
Covariance Type:	nonrobust	**LLR p-value:**	9.347e-92

	coef	**std err**	**z**	**P>|z|**	**[0.025**	**0.975]**
Intercept	0.3035	4.31e+06	7.03e-08	1.000	-8.46e+06	8.46e+06
Total_assets	0.0013	3.97e+10	3.19e-14	1.000	-7.79e+10	7.79e+10
Net_worth	-0.0031	0.003	-1.179	0.238	-0.008	0.002
Total_income	0.0009	0.001	1.126	0.260	-0.001	0.003
Change_in_stock	0.0223	0.011	1.941	0.052	-0.000	0.045
Total_expenses	-0.0005	0.001	-0.498	0.619	-0.002	0.001
Profit_after_tax	-0.0030	0.021	-0.144	0.886	-0.044	0.038
PBDITA	-0.0005	0.003	-0.176	0.860	-0.006	0.005
PBT	0.0113	0.017	0.655	0.512	-0.023	0.045
Cash_profit	-0.0157	0.006	-2.511	0.012	-0.028	-0.003





PBDITA_as_perc_of_total_income	-0.0140	0.017	-0.848	0.397	-0.047	0.018
PBT_as_perc_of_total_income	-0.0166	0.080	-0.207	0.836	-0.174	0.140
PAT_as_perc_of_total_income	0.0195	0.105	0.185	0.853	-0.187	0.226
Cash_profit_as_perc_of_total_income	-0.0129	0.028	-0.464	0.642	-0.067	0.042
PAT_as_perc_of_net_worth	-0.0320	0.009	-3.421	0.001	-0.050	-0.014
Sales	-0.0006	0.001	-0.915	0.360	-0.002	0.001
Income_from_fincial_services	-0.0148	0.056	-0.267	0.790	-0.124	0.094
Total_capital	-0.0061	0.004	-1.555	0.120	-0.014	0.002
Reserves_and_funds	-0.0012	0.001	-0.861	0.389	-0.004	0.001
Borrowings	0.0018	0.002	0.876	0.381	-0.002	0.006
Current_liabilities_and_provisions	-0.0015	0.003	-0.540	0.589	-0.007	0.004
Shareholders_funds	0.0047	0.003	1.737	0.082	-0.001	0.010
Cumulative_retained_profits	-0.0086	0.003	-3.172	0.002	-0.014	-0.003
Capital_employed	-0.0036	0.002	-1.643	0.100	-0.008	0.001
TOL_to_TNW	0.1723	0.090	1.922	0.055	-0.003	0.348
Total_term_liabilities__to__tangible_net_worth	-0.2227	0.207	-1.074	0.283	-0.629	0.184
Contingent_liabilities__to__Net_worth_perc	0.0016	0.004	0.389	0.697	-0.006	0.009
Net_fixed_assets	0.0012	0.002	0.747	0.455	-0.002	0.004
Current_assets	-0.0001	0.002	-0.096	0.923	-0.003	0.003
Net_working_capital	0.0044	0.003	1.483	0.138	-0.001	0.010
Quick_ratio_times	-0.1181	0.447	-0.264	0.792	-0.994	0.758
Current_ratio_times	-0.7253	0.297	-2.446	0.014	-1.306	-0.144
Debt_to_equity_ratio_times	0.4134	0.126	3.292	0.001	0.167	0.659
Cash_to_current_liabilities_times	3.3188	1.290	2.573	0.010	0.791	5.847
Cash_to_average_cost_of_sales_per_day	-0.0078	0.009	-0.856	0.392	-0.026	0.010
Creditors_turnover	-0.0349	0.028	-1.251	0.211	-0.090	0.020
Debtors_turnover	-0.0129	0.024	-0.539	0.590	-0.060	0.034
Finished_goods_turnover	0.0042	0.010	0.399	0.690	-0.016	0.025
WIP_turnover	-0.0227	0.021	-1.073	0.283	-0.064	0.019
Raw_material_turnover	-0.0124	0.021	-0.578	0.564	-0.054	0.030
Shares_outstanding	-3.123e-08	4.06e-08	-0.769	0.442	-1.11e-07	4.84e-08
Equity_face_value	-0.2093	4.3e+05	-4.87e-07	1.000	-8.42e+05	8.42e+05
EPS	0.0766	0.106	0.720	0.471	-0.132	0.285
Adjusted_EPS	-0.1809	0.128	-1.411	0.158	-0.432	0.070
Total_liabilities	-4.442e-05	3.97e+10	-1.12e-15	1.000	-7.79e+10	7.79e+10

Table 12: Summary of Logistic Regression

Insights:

Possibly complete quasi-separation: A fraction 0.20 of observations can be perfectly predicted. This might indicate that there is complete quasi-separation. In this case some parameters will not be identified.

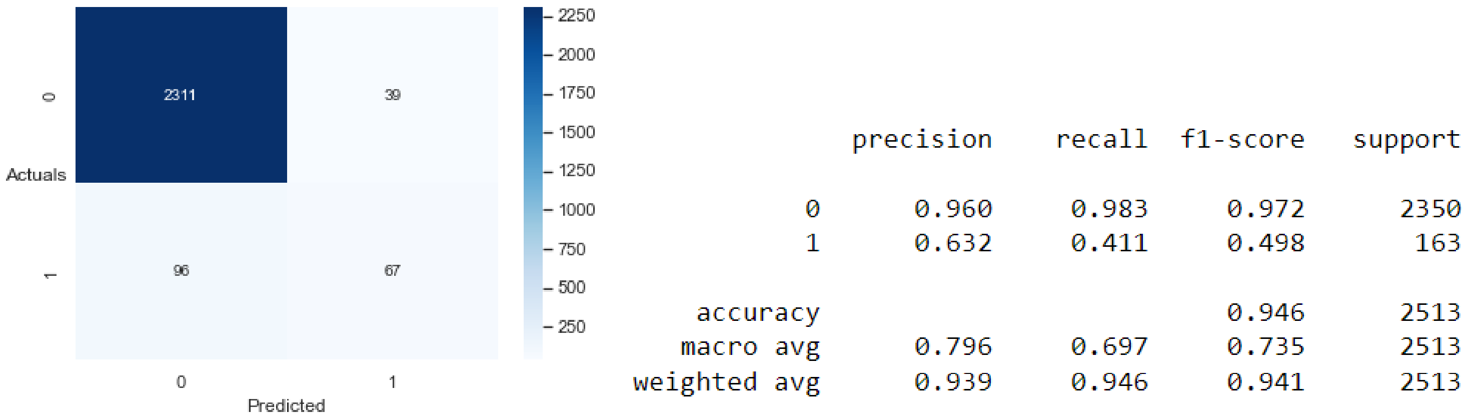
Here in the model 1 summary the value of Pseudo R-squ. is 0.4687 as we know pseudo R-squared value between of 0.2 to 0.4 indicates good fit. But here we get slightly higher value. So, we do feature selection and check again the value of Pseudo R-squ.

Log-Likelihood value is -320.64 as we know Log Likelihood value is a measure of goodness of fit for any model. Higher the value, better is the model. We should remember that Log Likelihood can lie between -Inf to +Inf. Hence, the absolute look at the value cannot give any indication. We can only compare the Log Likelihood values between multiple models. So we again build the model on selected features and check the Log-Likelihood value.

As we saw most of the variables p value is more than 0.05. As there is some kind of multicollinearity or not helpful for predicting the target variable. So, we will do feature selection and choose the features and build the model again. Then check its summary and do the validation of the model.

Evaluation on the Training Data

Confusion Matrix and Classification report for the Training Data:

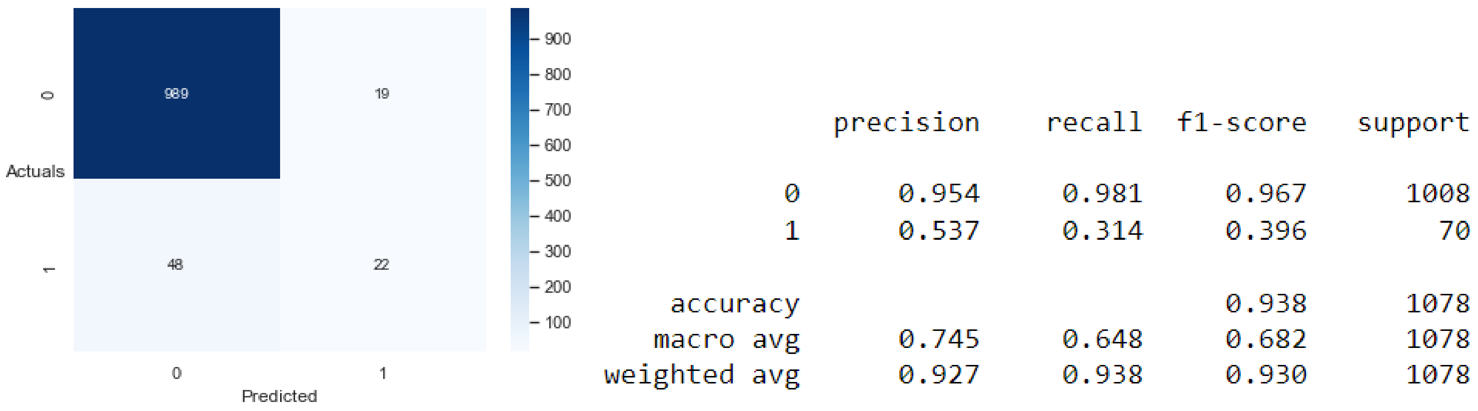


```
              precision    recall  f1-score   support

           0      0.960     0.983     0.972      2350
           1      0.632     0.411     0.498       163

    accuracy                          0.946      2513
   macro avg      0.796     0.697     0.735      2513
weighted avg      0.939     0.946     0.941      2513
```

Figure 8: Confusion Matrix Plot / Classification Report Logistic Regression Model (Train Data)

Train Data Accuracy: 0.946

Confusion Matrix and Classification report for the Test Data:



```
              precision    recall  f1-score   support

           0      0.954     0.981     0.967      1008
           1      0.537     0.314     0.396        70

    accuracy                          0.938      1078
   macro avg      0.745     0.648     0.682      1078
weighted avg      0.927     0.938     0.930      1078
```

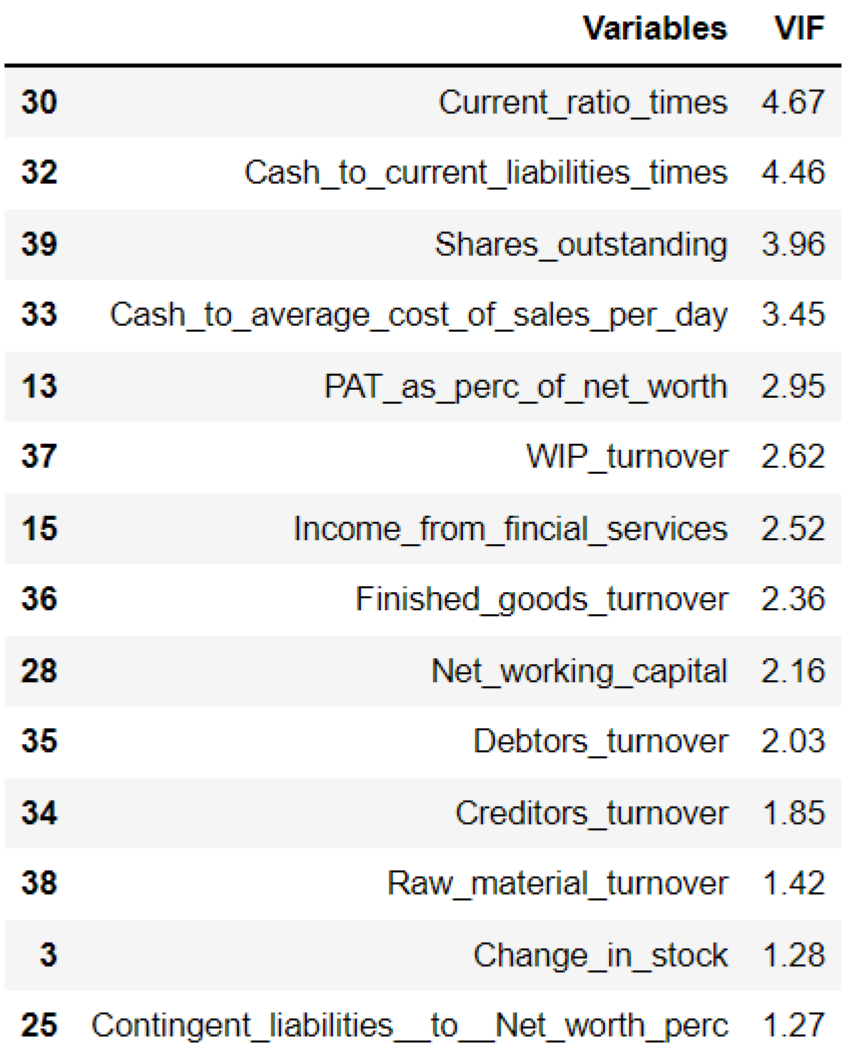
Figure 9: Confusion Matrix Plot / Classification Report Logistic Regression Model (Train Data)

Test Data Accuracy 0.938

On comparing Train & Test results, we conclude that model is not overfitting or underfitting. However, Precision, recall and F1-score is low for class ‘1’ due to data imbalance with proportion of class ‘1’ is only 6.5%.

LOGISTIC REGRESSION ON (SELECTED FEATURES) USING STATSMODEL LIBRARY

Features with VIF < 5 is used for logistic regression.

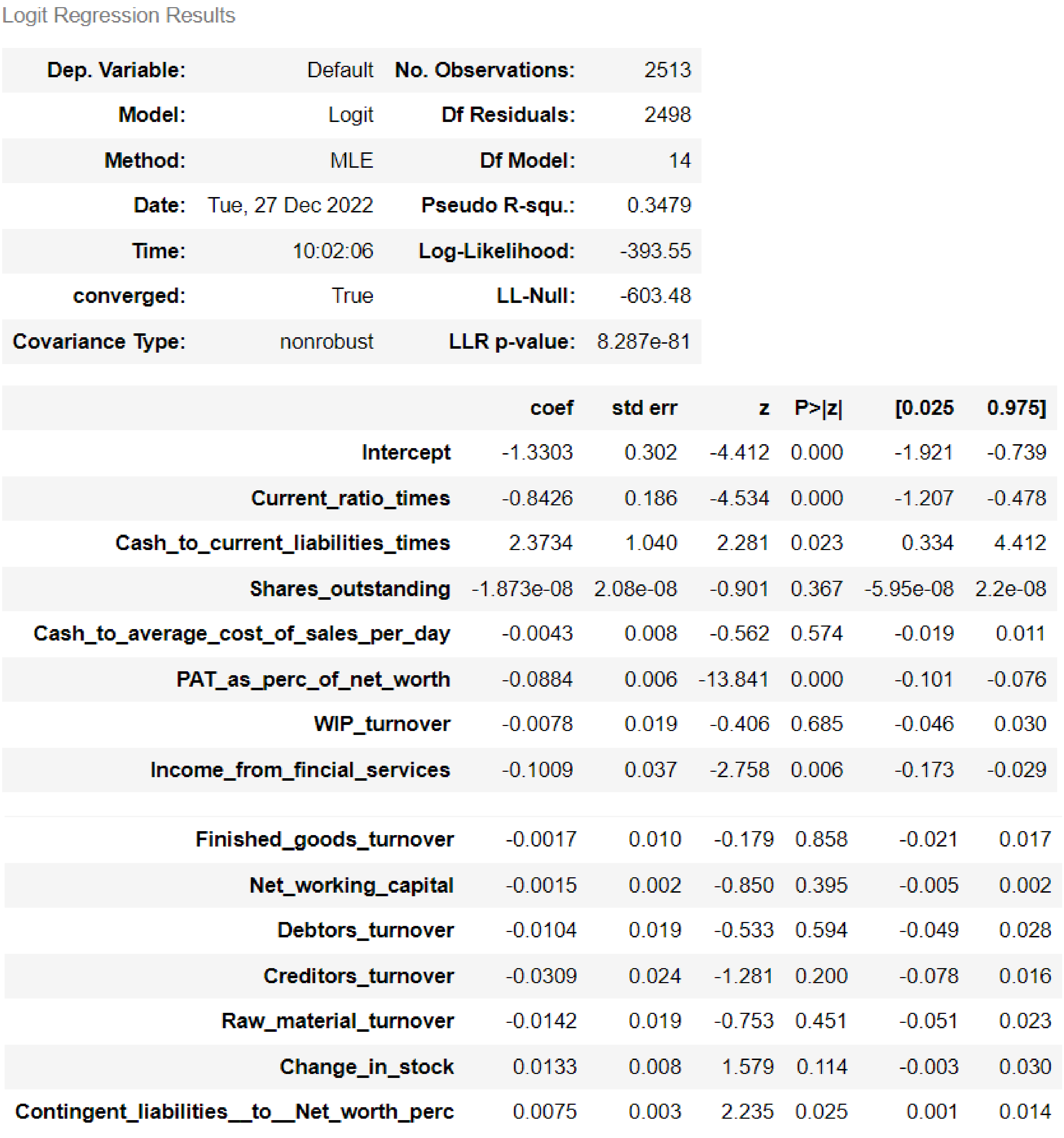


	Variables	VIF
30	Current_ratio_times	4.67
32	Cash_to_current_liabilities_times	4.46
39	Shares_outstanding	3.96
33	Cash_to_average_cost_of_sales_per_day	3.45
13	PAT_as_perc_of_net_worth	2.95
37	WIP_turnover	2.62
15	Income_from_fincial_services	2.52
36	Finished_goods_turnover	2.36
28	Net_working_capital	2.16
35	Debtors_turnover	2.03
34	Creditors_turnover	1.85
38	Raw_material_turnover	1.42
3	Change_in_stock	1.28
25	Contingent_liabilities__to__Net_worth_perc	1.27

Table 13: Features with VIF Values < 5

Above mentioned features have VIF less than 5. Hence these features will be used as independent variable for further model building.

Summary of Logistic Regression Model with Selected Features



Logit Regression Results

Dep. Variable:	Default	**No. Observations:**	2513
Model:	Logit	**Df Residuals:**	2498
Method:	MLE	**Df Model:**	14
Date:	Tue, 27 Dec 2022	**Pseudo R-squ.:**	0.3479
Time:	10:02:06	**Log-Likelihood:**	-393.55
converged:	True	**LL-Null:**	-603.48
Covariance Type:	nonrobust	**LLR p-value:**	8.287e-81

	coef	**std err**	**z**	**P>|z|**	**[0.025**	**0.975]**
Intercept	-1.3303	0.302	-4.412	0.000	-1.921	-0.739
Current_ratio_times	-0.8426	0.186	-4.534	0.000	-1.207	-0.478
Cash_to_current_liabilities_times	2.3734	1.040	2.281	0.023	0.334	4.412
Shares_outstanding	-1.873e-08	2.08e-08	-0.901	0.367	-5.95e-08	2.2e-08
Cash_to_average_cost_of_sales_per_day	-0.0043	0.008	-0.562	0.574	-0.019	0.011
PAT_as_perc_of_net_worth	-0.0884	0.006	-13.841	0.000	-0.101	-0.076
WIP_turnover	-0.0078	0.019	-0.406	0.685	-0.046	0.030
Income_from_fincial_services	-0.1009	0.037	-2.758	0.006	-0.173	-0.029
Finished_goods_turnover	-0.0017	0.010	-0.179	0.858	-0.021	0.017
Net_working_capital	-0.0015	0.002	-0.850	0.395	-0.005	0.002
Debtors_turnover	-0.0104	0.019	-0.533	0.594	-0.049	0.028
Creditors_turnover	-0.0309	0.024	-1.281	0.200	-0.078	0.016
Raw_material_turnover	-0.0142	0.019	-0.753	0.451	-0.051	0.023
Change_in_stock	0.0133	0.008	1.579	0.114	-0.003	0.030
Contingent_liabilities__to__Net_worth_perc	0.0075	0.003	2.235	0.025	0.001	0.014

Table 14: Summary of Logistic Regression with Selected Features

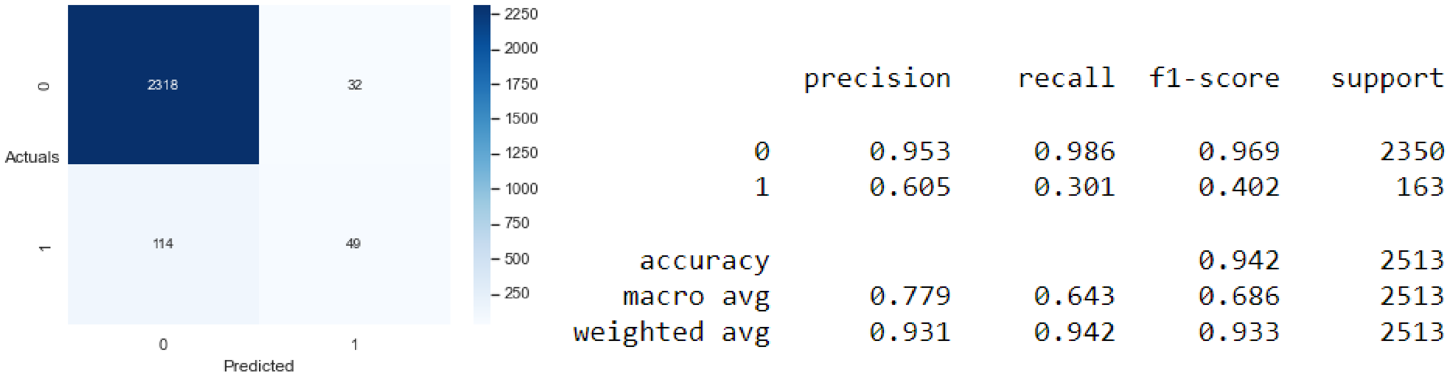
In the previous model of logistic regression with all features, summary the value of Pseudo R-squ. is 0.4687 as we know pseudo R-squared value between of 0.2 to 0.4 indicates good fit. But here we get Pseudo R-squ.: 0.3479 which better than previous model.

Previously in model of logistic regression with all features, Log-Likelihood value is -320.64 as we know Log Likelihood value is a measure of goodness of fit for any model. Higher the value, better is the model. We should remember that Log Likelihood can lie between -Inf to +Inf. Hence, the absolute look at the value cannot give any indication. We can only compare the Log Likelihood values between multiple models. But here we get Log-Likelihood value is around -393.55 which is less than model of logistic regression with all features.

As we saw here also many of the variables p value is more than 0.05.

Evaluation on the Training Data

Confusion Matrix and Classification report for the Training Data

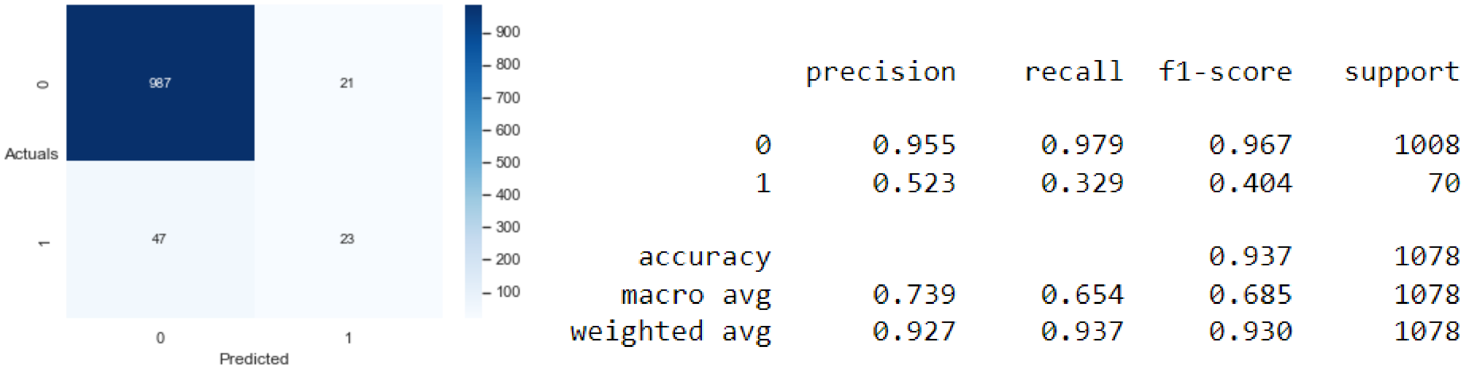


```
              precision    recall  f1-score   support

           0      0.953     0.986     0.969      2350
           1      0.605     0.301     0.402       163

    accuracy                          0.942      2513
   macro avg      0.779     0.643     0.686      2513
weighted avg      0.931     0.942     0.933      2513
```

Figure 10: Confusion Matrix Plot / Classification Report Logistic Regression Model (Train Data)

Train Data Accuracy 0.942

Confusion Matrix and Classification report for the Test Data



```
              precision    recall  f1-score   support

           0      0.955     0.979     0.967      1008
           1      0.523     0.329     0.404        70

    accuracy                          0.937      1078
   macro avg      0.739     0.654     0.685      1078
weighted avg      0.927     0.937     0.930      1078
```

Figure 11: Confusion Matrix Plot / Classification Report Logistic Regression Model (Test Data)

Test Data Accuracy 0.937

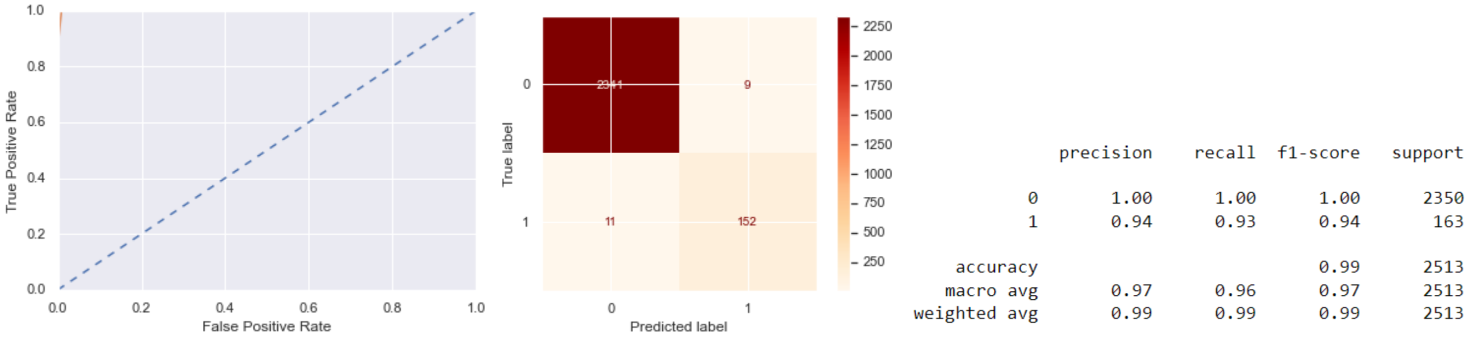
On comparing Train & Test results, we conclude that model is not overfitting or underfitting. However, Precision, recall and F1-score is low for class ‘1’ due to data imbalance with proportion of class ‘1’ is only 6.5%.

RANDOM FOREST BASE MODEL

Random forest establishes outcome based on predictions of the decision trees. It builds decision trees on different samples and takes their majority vote for classification and average in case of regression.

Model Evaluation - Random Forest Base Model

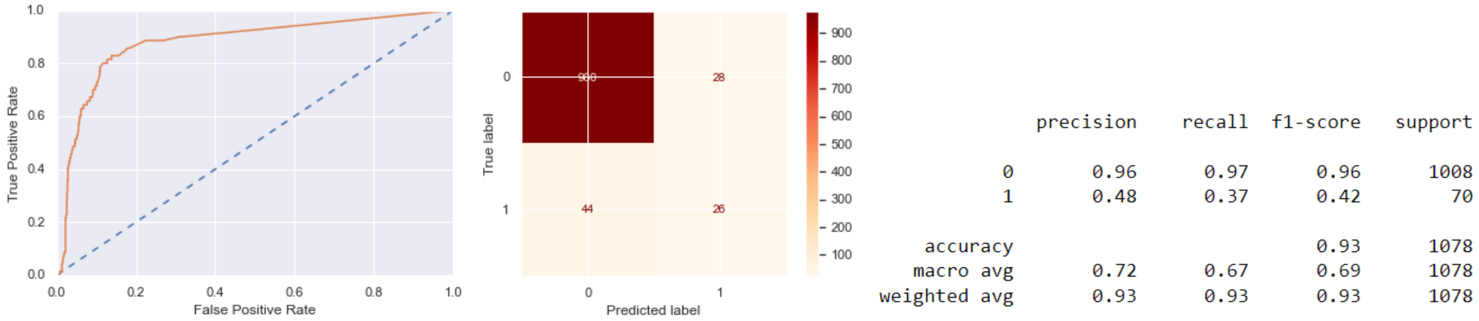
AUC ROC, Confusion Matrix and Classification report for the Training Data



```
              precision    recall  f1-score   support

           0       1.00      1.00      1.00      2350
           1       0.94      0.93      0.94       163

    accuracy                           0.99      2513
   macro avg       0.97      0.96      0.97      2513
weighted avg       0.99      0.99      0.99      2513
```

Figure 12: AUC ROC / Confusion Matrix Plot / Classification Report Random Forest Base Model (Train Data)

AUC ROC, Confusion Matrix and Classification report for the Test Data



```
              precision    recall  f1-score   support

           0       0.96      0.97      0.96      1008
           1       0.48      0.37      0.42        70

    accuracy                           0.93      1078
   macro avg       0.72      0.67      0.69      1078
weighted avg       0.93      0.93      0.93      1078
```

Figure 13: AUC ROC / Confusion Matrix Plot / Classification Report Random Forest Base Model (Test Data)

As there is huge difference in precision, recall and F1 score (more than 10%) for class 1 between train and test results, we can conclude there is problem of overfitting in the model. Hence this model preforms poorly on test set and is rejected. Problem of overfit can be resolved by applying bagging technique.

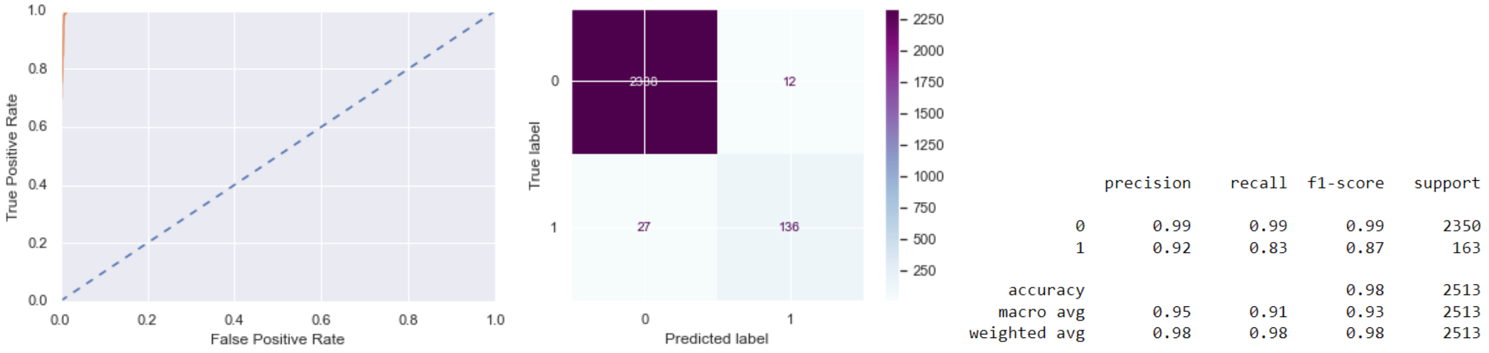
BAGGING OF RANDOM FOREST

Bagging is an ensemble algorithm that fits multiple models on different subsets of a training dataset, then combines the predictions from all models. Random forest is an extension of bagging that also randomly selects subsets of features used in each data sample.

Bootstrap aggregating, also called bagging (from bootstrap aggregating), is a machine learning ensemble meta-algorithm designed to improve the stability and accuracy of machine learning algorithms used in statistical classification and regression. It also reduces variance and helps to avoid overfitting. Let’s build Random Forest with Bagging technique.

Model Evaluation – Bagging Random Forest Model

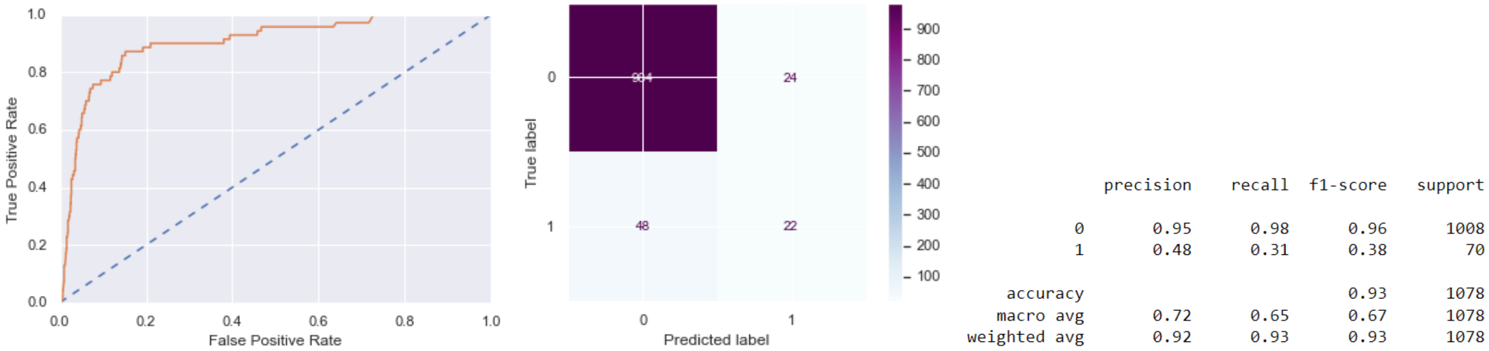
AUC ROC, Confusion Matrix and Classification report for the Training Data



```
              precision    recall  f1-score   support

           0       0.99      0.99      0.99      2350
           1       0.92      0.83      0.87       163

    accuracy                           0.98      2513
   macro avg       0.95      0.91      0.93      2513
weighted avg       0.98      0.98      0.98      2513
```

Figure 14: AUC ROC / Confusion Matrix Plot / Classification Report Bagging Random Forest Base Model (Train Data)

AUC ROC, Confusion Matrix and Classification report for the Test Data



```
              precision    recall  f1-score   support

           0       0.95      0.98      0.96      1008
           1       0.48      0.31      0.38        70

    accuracy                           0.93      1078
   macro avg       0.72      0.65      0.67      1078
weighted avg       0.92      0.93      0.93      1078
```

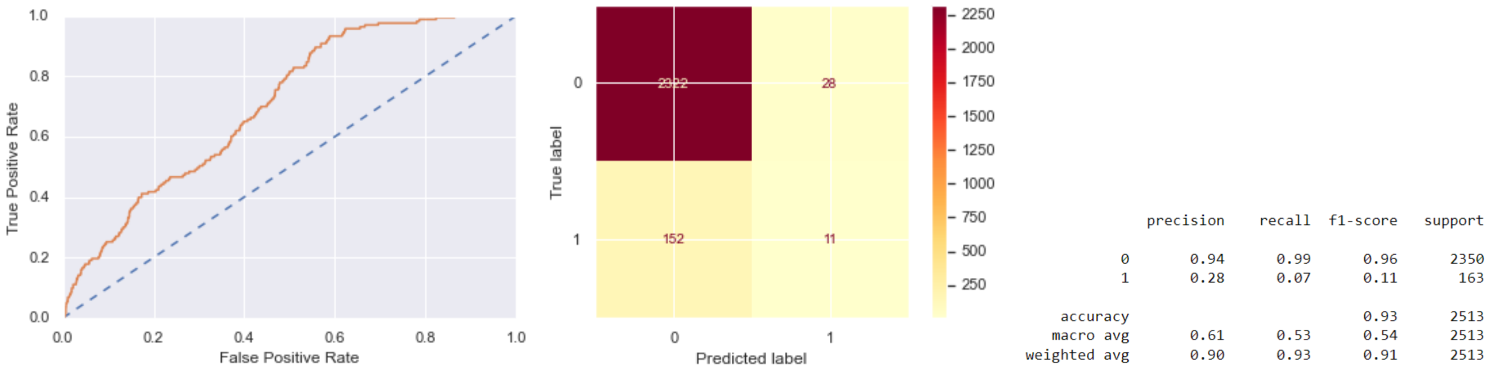
Figure 15: AUC ROC / Confusion Matrix Plot / Classification Report Bagging Random Forest Base Model (Test Data)

As there is huge difference in precision, recall and F1 score (more than 10%) for class 1 between train and test results, we can conclude there is problem of overfitting in the model. Hence this model preforms poorly on test set and is rejected.

LOGISTICS REGRESSION WITH SKLEARN

Model Evaluation-Logistic Regression

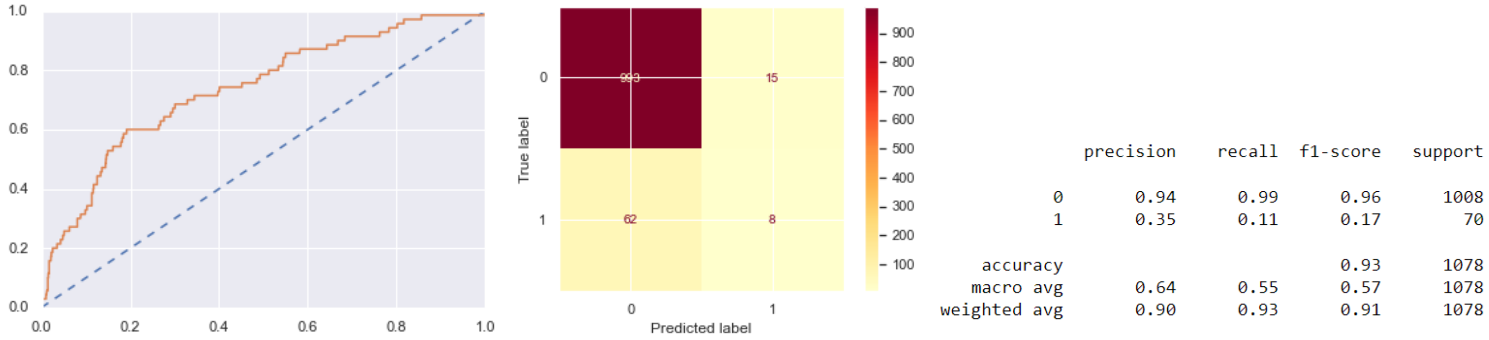
AUC ROC, Confusion Matrix and Classification report for Training Data



```
              precision    recall  f1-score   support

           0       0.94      0.99      0.96      2350
           1       0.28      0.07      0.11       163

    accuracy                           0.93      2513
   macro avg       0.61      0.53      0.54      2513
weighted avg       0.90      0.93      0.91      2513
```

Figure 16: AUC ROC / Confusion Matrix Plot / Classification Report Logistic Regression with SK learn (Train Data)

AUC ROC, Confusion Matrix and Classification report for Test Data



```
              precision    recall  f1-score   support

           0       0.94      0.99      0.96      1008
           1       0.35      0.11      0.17        70

    accuracy                           0.93      1078
   macro avg       0.64      0.55      0.57      1078
weighted avg       0.90      0.93      0.91      1078
```

Figure 17: AUC ROC / Confusion Matrix Plot / Classification Report Logistic Regression with SK learn (Test Data)

On comparing the Train & Test results of the Logistics Regression Model, we conclude there is no problem of underfitting or overfitting of the model. As precision, recall & f1 score are very poor for class 1 as compared to class 0. Hence model is not good to predict the results.

GRADIENT BOOSTING MODEL

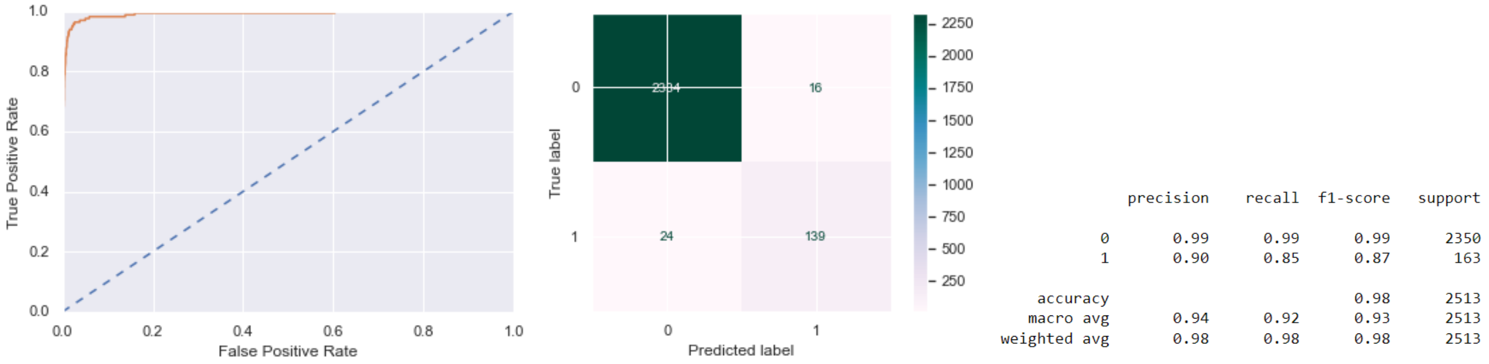
Gradient boosting is a machine learning technique for regression, classification and other tasks, which produces a prediction model in the form of an ensemble of weak prediction models, typically decision trees.

Gradient boosting is a type of machine learning boosting. It relies on the intuition that the best possible next model, when combined with previous models, minimizes the overall prediction error. The key idea is to set the target outcomes for this next model in order to minimize the error.

Gradient boosting algorithm can be used for predicting not only continuous target variable (as a Regressor) but also categorical target variable (as a Classifier). When it is used as a regressor, the cost function is Mean Square Error (MSE) and when it is used as a classifier then the cost function is Log loss.

Model Evaluation - Gradient Boosting Model

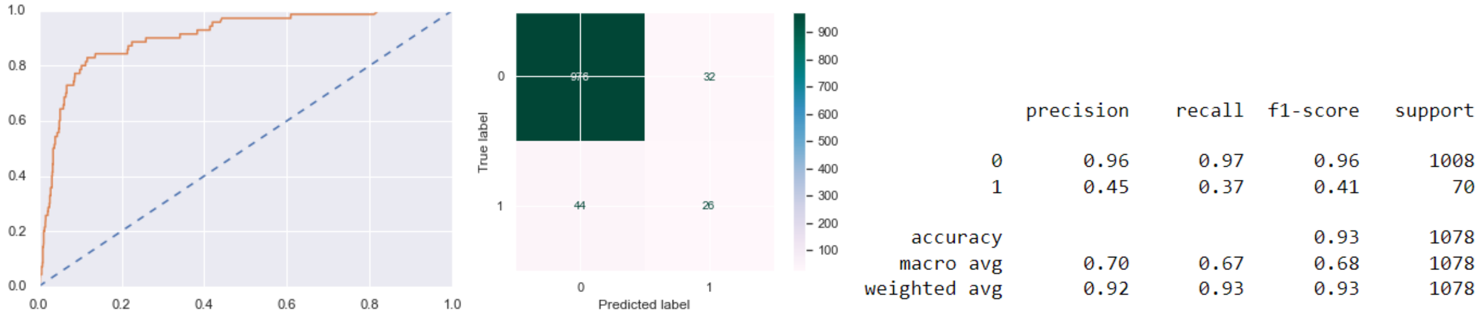
AUC ROC, Confusion Matrix and Classification report for Training Data



```
              precision    recall  f1-score   support

           0       0.99      0.99      0.99      2350
           1       0.90      0.85      0.87       163

    accuracy                           0.98      2513
   macro avg       0.94      0.92      0.93      2513
weighted avg       0.98      0.98      0.98      2513
```

Figure 18: AUC ROC / Confusion Matrix Plot / Classification Report Gradient Boosting Model (Train Data)

AUC ROC, Confusion Matrix and Classification report for Test Data



```
              precision    recall  f1-score   support

           0       0.96      0.97      0.96      1008
           1       0.45      0.37      0.41        70

    accuracy                           0.93      1078
   macro avg       0.70      0.67      0.68      1078
weighted avg       0.92      0.93      0.93      1078
```

Figure 19: AUC ROC / Confusion Matrix Plot / Classification Report Gradient Boosting Model (Test Data)

As there is huge difference in precision, recall and F1 score (more than 10%) for class 1 between train and test results, we can conclude there is problem of overfitting in the model. Hence this model preforms poorly on test set and is rejected.

SMOTE can be applied now to check class imbalance and further check model performance.

SMOTE

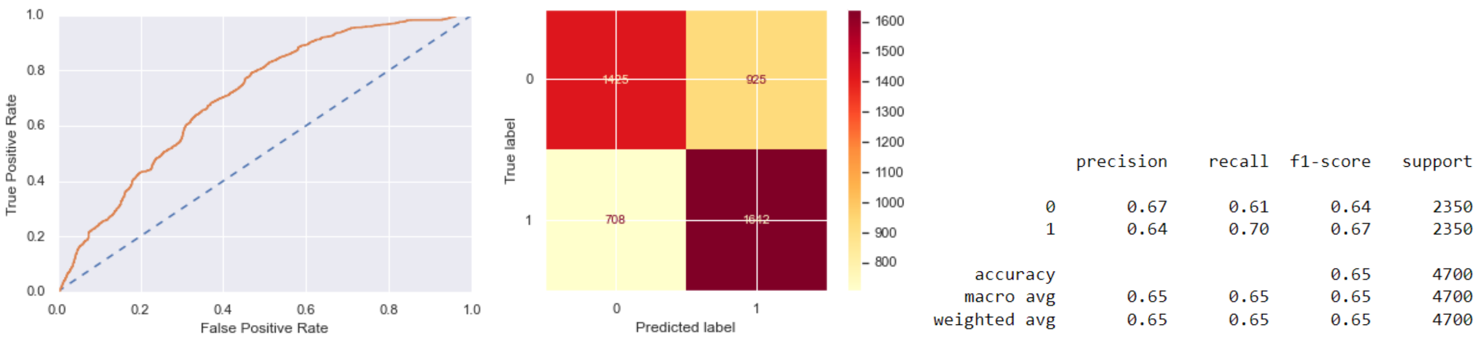
Synthetic Minority Oversampling Technique (SMOTE) is a statistical technique for increasing the number of cases in your dataset in a balanced way. The component works by generating new instances from existing minority cases that you supply as input.

Applying Smote on the Training Data. Shape after SMOTE is (4700, 44).

Logistic Regression Model with SMOTE.

Model Evaluation - Logistic Regression with SMOTE.

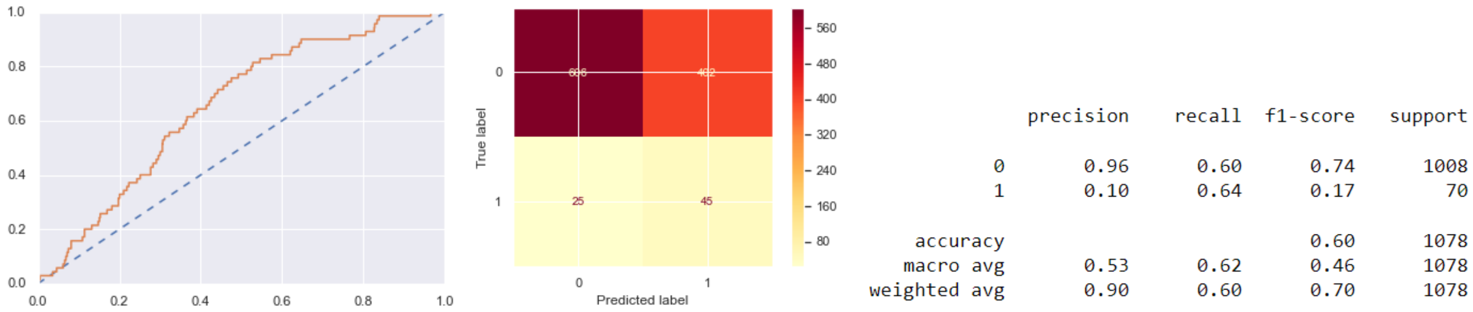
AUC ROC, Confusion Matrix and Classification report for Training Data



```
              precision    recall  f1-score   support

           0       0.67      0.61      0.64      2350
           1       0.64      0.70      0.67      2350

    accuracy                           0.65      4700
   macro avg       0.65      0.65      0.65      4700
weighted avg       0.65      0.65      0.65      4700
```

Figure 20: AUC ROC / Confusion Matrix Plot / Classification Report Logistic Regression with SMOTE (Train Data)

AUC ROC, Confusion Matrix and Classification report for Test Data



```
              precision    recall  f1-score   support

           0       0.96      0.60      0.74      1008
           1       0.10      0.64      0.17        70

    accuracy                           0.60      1078
   macro avg       0.53      0.62      0.46      1078
weighted avg       0.90      0.60      0.70      1078
```

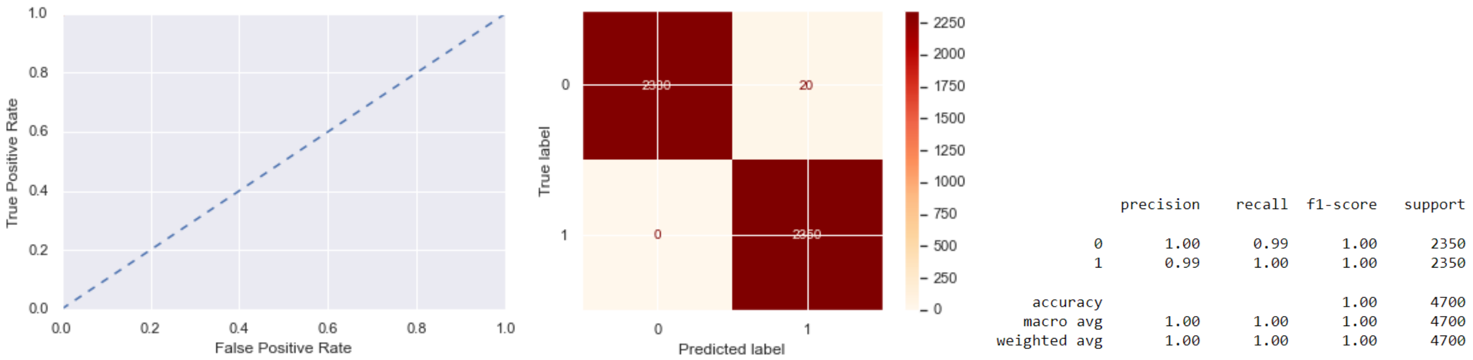
Figure 21: AUC ROC / Confusion Matrix Plot / Classification Report Logistic Regression with SMOTE (Test Data)

On comparing train and test results, we can conclude, that Logistic Regression with SMOTE performance is poor. Only recall value is similar but precision, f1 score is very poor on test data for class 1.

Random Forest with SMOTE

Model Evaluation – Random Forest with SMOTE

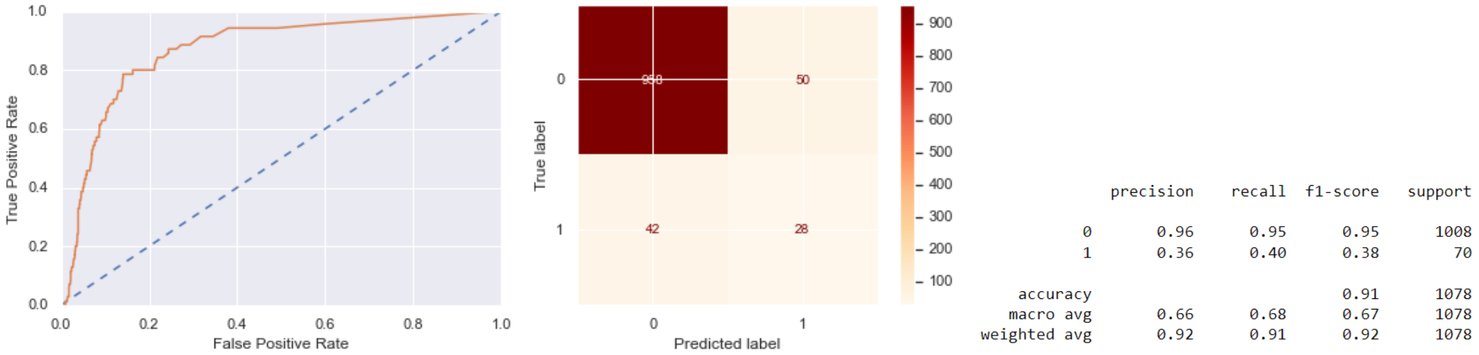
AUC ROC, Confusion Matrix and Classification report for Training Data



```
              precision    recall  f1-score   support

           0       1.00      0.99      1.00      2350
           1       0.99      1.00      1.00      2350

    accuracy                           1.00      4700
   macro avg       1.00      1.00      1.00      4700
weighted avg       1.00      1.00      1.00      4700
```

Figure 22: AUC ROC / Confusion Matrix Plot / Classification Report Random Forest with SMOTE (Train Data)

AUC ROC, Confusion Matrix and Classification report for Test Data



```
              precision    recall  f1-score   support

           0       0.96      0.95      0.95      1008
           1       0.36      0.40      0.38        70

    accuracy                           0.91      1078
   macro avg       0.66      0.68      0.67      1078
weighted avg       0.92      0.91      0.92      1078
```

Figure 23: AUC ROC / Confusion Matrix Plot / Classification Report Random Forest with SMOTE (Test Data)

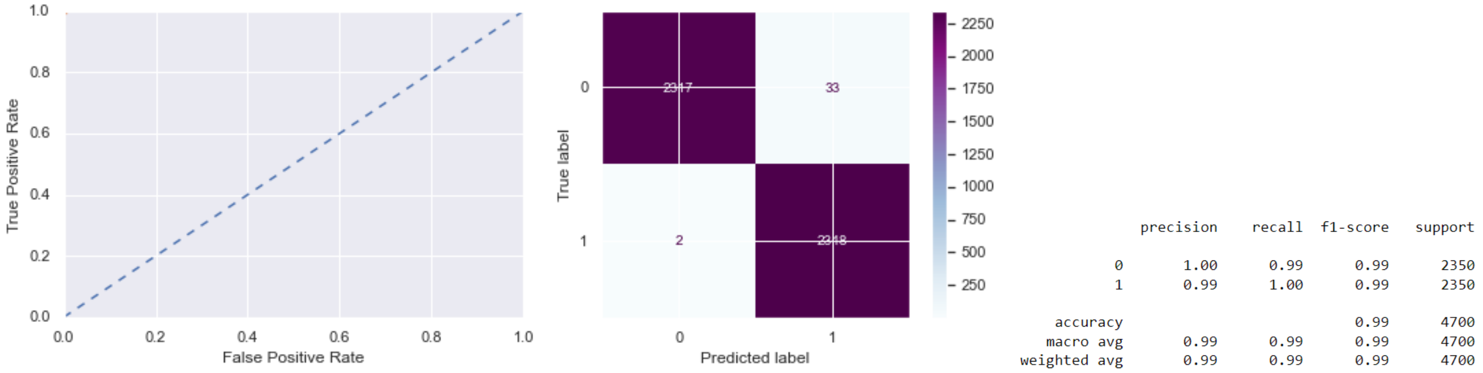
As there is huge difference in precision, recall and F1 score (more than 10%) for class 1 between train and test results, we can conclude there is problem of overfitting in the model. Hence this model preforms poorly on test set and is rejected. Problem of overfit can be resolved by applying bagging technique.

Bagging of Random Forest with SMOTE

For Bagged Random Forest model with Smote, fit SMOTE train data into the bagging classifier model with Random Forest Classifier as base estimator.

Model Evaluation - Bagging of Random Forest with SMOTE

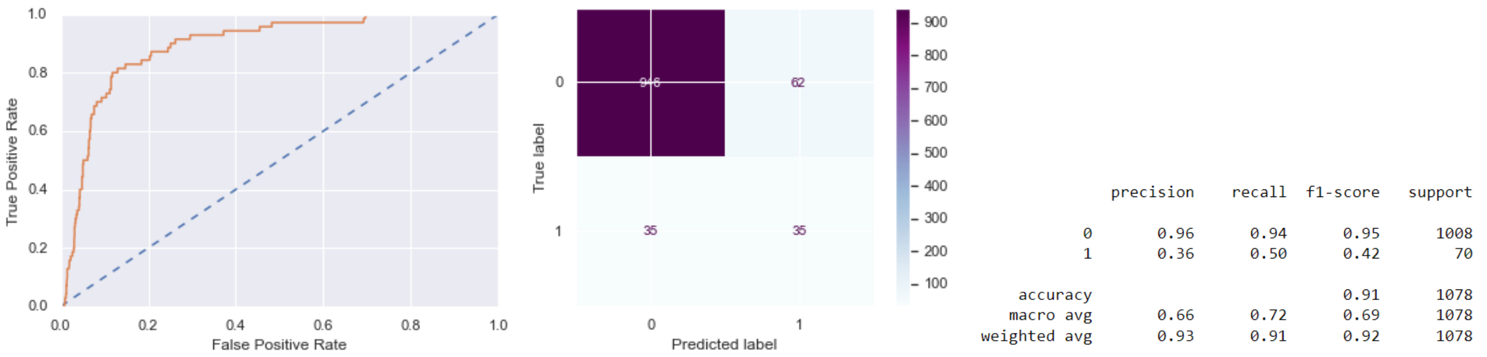
AUC ROC, Confusion Matrix and Classification report for Training Data



```
              precision    recall  f1-score   support

           0       1.00      0.99      0.99      2350
           1       0.99      1.00      0.99      2350

    accuracy                           0.99      4700
   macro avg       0.99      0.99      0.99      4700
weighted avg       0.99      0.99      0.99      4700
```

Figure 24: AUC ROC / Confusion Matrix Plot / Classification Report Bagging of Random Forest with SMOTE (Train Data)

AUC ROC, Confusion Matrix and Classification report for Test Data



```
              precision    recall  f1-score   support

           0       0.96      0.94      0.95      1008
           1       0.36      0.50      0.42        70

    accuracy                           0.91      1078
   macro avg       0.66      0.72      0.69      1078
weighted avg       0.93      0.91      0.92      1078
```

Figure 25: AUC ROC / Confusion Matrix Plot / Classification Report Bagging of Random Forest with SMOTE (Test Data)

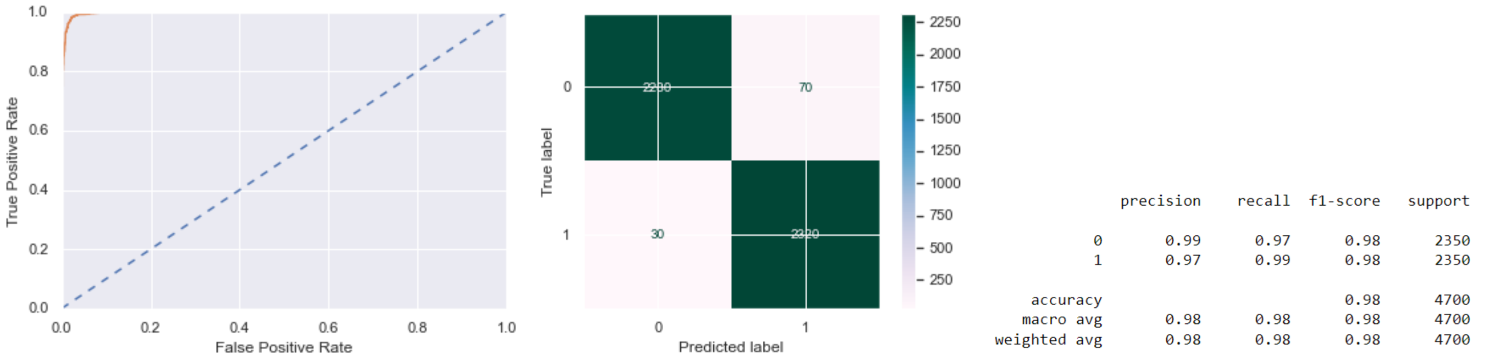
As there is huge difference in precision, recall and F1 score (more than 10%) for class 1 between train and test results, we can conclude there is problem of overfitting in the model. Hence this model preforms poorly on test set and is rejected.

Gradient Boosting Model with SMOTE

For Gradient Boosting model with Smote, fit SMOTE train data into the Gradient Boosting Classifier model.

Model Evaluation – Gradient Boosting Model with SMOTE

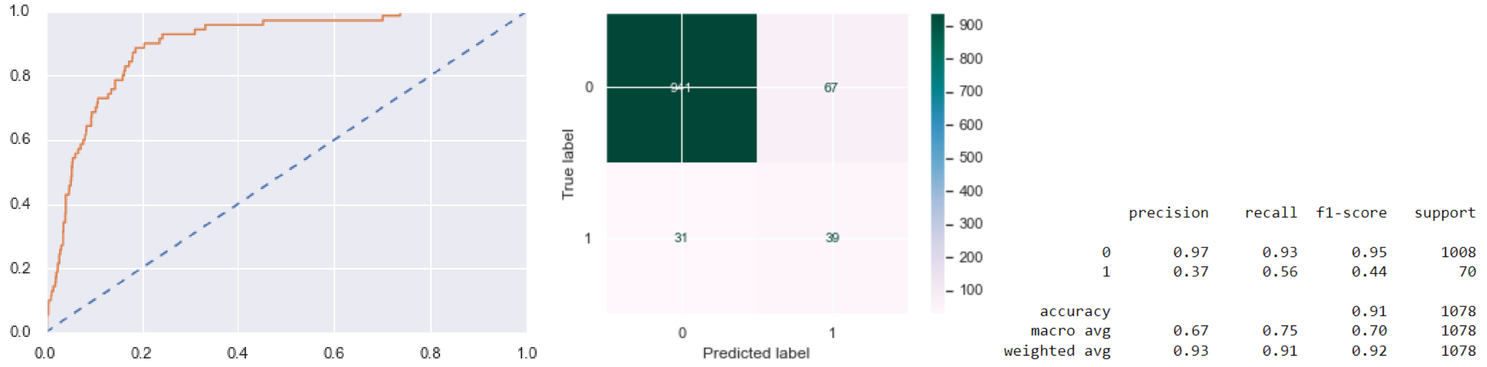
AUC ROC, Confusion Matrix and Classification report for Training Data



```
              precision    recall  f1-score   support

           0       0.99      0.97      0.98      2350
           1       0.97      0.99      0.98      2350

    accuracy                           0.98      4700
   macro avg       0.98      0.98      0.98      4700
weighted avg       0.98      0.98      0.98      4700
```

Figure 26: AUC ROC / Confusion Matrix Plot / Classification Report Gradient Boosting Model with SMOTE (Train Data)

AUC ROC, Confusion Matrix and Classification report for Test Data



```
              precision    recall  f1-score   support

           0       0.97      0.93      0.95      1008
           1       0.37      0.56      0.44        70

    accuracy                           0.91      1078
   macro avg       0.67      0.75      0.70      1078
weighted avg       0.93      0.91      0.92      1078
```

Figure 27: AUC ROC / Confusion Matrix Plot / Classification Report Gradient Boosting Model with SMOTE(Test Data)

As there is huge difference in precision, recall and F1 score (more than 10%) for class 1 between train and test results, we can conclude there is problem of overfitting in the model. Hence this model preforms poorly on test set and is rejected.

COMPARISON OF THE PERFORMANCE METRICS OF ALL THE MODELS ON TRAIN DATA AND TEST DATA

For class level predictions, Accuracy is not a reliable metric.

There is class imbalance in the data set as class '1' is only 6.5%.

As seen during various model evaluations, precision, recall is low for class '1'. Hence f1 score plays vital role while finalising model.

Further we will compare metrics for different models in train and test using tabular form to decide final model for the deployment.

Comparison of the Performance Metrics of All the Models on Train Data

Sr. No.	Model Name (Train Data)	Precision Class 0 (Train Data)	Precision Class 1 (Train Data)	Recall Class 0 (Train Data)	Recall Class 1 (Train Data)	F1 Score Class 0 (Train Data)	F1 Score Class 1 (Train Data)	Model Accuracy (Train Data)	AUC / ROC (Train Data)
1	Logistic Regression Using Statsmodel	0.960	0.632	0.983	0.411	0.972	0.498	0.946	-
2	Logistic Regression Using Statsmodel (Selected Features)	0.953	0.605	0.986	0.301	0.969	0.402	0.942	-
3	Random Forest Base Model	1.000	0.940	1.000	0.930	1.000	0.940	0.990	0.999
4	Bagging of Random Forest Base Model	0.990	0.920	0.990	0.830	0.990	0.870	0.980	0.998
5	Logistic Regression using sklearn	0.940	0.280	0.990	0.070	0.960	0.110	0.930	0.704
6	Gradient Boosting Model	0.990	0.900	0.990	0.850	0.990	0.870	0.980	0.991
7	Logistic Regression using sklearn with SMOTE	0.670	0.640	0.610	0.700	0.640	0.670	0.650	0.706
8	Random Forest Model with SMOTE	1.000	0.990	0.990	1.000	1.000	1.000	1.000	1.000
9	Bagging of Random Forest Model with SMOTE	1.000	0.990	0.990	1.000	0.990	0.990	0.990	1.000
10	Gradient Boosting Model with SMOTE	0.990	0.970	0.970	0.990	0.980	0.980	0.980	0.997

Table 15 Performance Metrics of All the Models on Train Data

Comparison of the Performance Metrics of All the Models on Test Data

Sr. No.	Model Name (Test Data)	Precision Class 0 (Test Data)	Precision Class 1 (Test Data)	Recall Class 0 (Test Data)	Recall Class 1 (Test Data)	F1 Score Class 0 (Test Data)	F1 Score Class 1 (Test Data)	Model Accuracy (Test Data)	AUC / ROC (Test Data)
1	Logistic Regression Using Statsmodel	0.954	0.537	0.981	0.314	0.967	0.396	0.938	-
2	Logistic Regression Using Statsmodel (Selected Features)	0.955	0.523	0.979	0.329	0.967	0.404	0.937	-
3	Random Forest Base Model	0.960	0.480	0.970	0.370	0.960	0.420	0.930	0.889
4	Bagging of Random Forest Base Model	0.950	0.480	0.980	0.310	0.960	0.380	0.930	0.902
5	Logistic Regression using sklearn	0.940	0.350	0.990	0.110	0.960	0.170	0.930	0.738
6	Gradient Boosting Model	0.960	0.450	0.970	0.370	0.960	0.410	0.930	0.904
7	Logistic Regression using sklearn with SMOTE	0.960	0.100	0.600	0.640	0.740	0.170	0.600	0.654
8	Random Forest Model with SMOTE	0.960	0.360	0.950	0.400	0.950	0.380	0.910	0.870
9	Bagging of Random Forest Model with SMOTE	0.960	0.360	0.940	0.400	0.950	0.420	0.910	0.894
10	Gradient Boosting Model with SMOTE	0.970	0.370	0.930	0.560	0.950	0.440	0.910	0.896

Table 16 Performance Metrics of All the Models on Test Data

Conclusion of Comparison

Based on the above-mentioned tables, we find **Logistic Regression using stats-model, Logistic Regression using stats-model (Selected Features)** as acceptable. These models are balanced and has performance improvement on test data. Also, they do not have overfitting problem on train and test data for class ‘1’.

The models that are being rejected due the problem of overfitting is:

- Random Forest Base Model
- Bagging of Random Forest Model
- Gradient Boosting Model
- Random Forest Model with SMOTE
- Bagging of Random Forest Model with SMOTE
- Gradient Boosting Model with SMOTE

Logistic Regression Model using sk-learn and Logistic Regression Model using sk-learn with SMOTE are rejected due to very poor performance in terms of precision, recall and f1 score for class ‘1’ on train and test data.

Final Model Selection

Logistic Regression using stats-model (Selected Features) as its precision, recall and f1 score are better. Also, pseudo R-squared is more for this model than Logistic Regression using stats-model.

RECOMMENDATIONS

- As seen during model building exercise, dataset is imbalanced and is not suitable for predicting class ‘1’ as only 6.5% dataset contains class ‘1’. To overcome this problem, SMOTE was used. However, results are not satisfactory. So, to get high performance model, company need to provide balanced dataset or increase dataset to increase performance.

- Features with high importance shall be used in predicting target variable. As shown during model building, models with selected features perform better on test set in predicting defaulter and non-defaulter.

- Before providing loan or credit to companies please check these ranges, as it will reduce the credit risk:

    - For the upper range (-1000 – 0) of PBDITA as % of total income, chances are high that the customer will default.
    - For the upper range (-4000 – 0) of PBT as % of total income, chances are high that the customer will default.
    - For the upper range (-5000 -0) of PAT as % of total income, chances are high that the customer will default.
    - For the upper range (-3000 – 0) of Cash profit as % of total income, chances are high that the customer will default.
    - For the lower range (-500 – 0) of PAT as % of net worth, chances are high that the customer will default.
    - Customers in range 0-200 of TOL_to_TNW seem more likely to default.
    - Customers in range 0-500 of creditors_turnover seems more likely to default.
    - Customers in range 0-1000 of debtors_turnover seems more likely to default.

- Do a proper check of customer's credit history and credit bureau report will also reduce credit risk.