Problem statement:

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'

Hints:

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

Test Train Split - Split the data into Train and Test dataset in a ratio of 67:33 and use random_state =42. Model Building is to be done on Train Dataset and Model Validation is to be done on Test Dataset.

About Data

Table 1: Head of the dataset showing the first 5 records (Image not complete due to size constraint)

	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]	Debtors Velocity (Days)	Creditors Velocity (Days)	Inventory Velocity (Days)
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	0	0	45.0
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	29	101	2.0
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	97	558	0.0
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	93	63	2.0
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	3887	346	0.0

5 rows × 67 columns

The dataset was loaded and the head of the dataset was checked. Table 1 shows the first 5 records of the dataset. From this table, we can see the different variables or columns of the dataset.

```
Index(['Co_Code', 'Co_Name', 'Networth Next Year', 'Equity Paid Up',
       'Networth', 'Capital Employed', 'Total Debt', 'Gross Block ',
       'Net Working Capital ', 'Current Assets ',
       'Current Liabilities and Provisions ', 'Total Assets/Liabilities ',
       'Gross Sales', 'Net Sales', 'Other Income', 'Value Of Output',
       'Cost of Production', 'Selling Cost', 'PBIDT', 'PBDT', 'PBIT', 'PBT',
       'PAT', 'Adjusted PAT', 'CP', 'Revenue earnings in forex',
       'Revenue expenses in forex', 'Capital expenses in forex',
       'Book Value (Unit Curr)', 'Book Value (Adj.) (Unit Curr)',
       'Market Capitalisation', 'CEPS (annualised) (Unit Curr)',
       'Cash Flow From Operating Activities',
       'Cash Flow From Investing Activities',
       'Cash Flow From Financing Activities', 'ROG-Net Worth (%)',
       'ROG-Capital Employed (%)', 'ROG-Gross Block (%)',
       'ROG-Gross Sales (%)', 'ROG-Net Sales (%)',
       'ROG-Cost of Production (%)', 'ROG-Total Assets (%)', 'ROG-PBIDT (%)',
       'ROG-PBDT (%)', 'ROG-PBIT (%)', 'ROG-PBT (%)', 'ROG-PAT (%)',
       'ROG-CP (%)', 'ROG-Revenue earnings in forex (%)',
       'ROG-Revenue expenses in forex (%)', 'ROG-Market Capitalisation (%)',
       'Current Ratio[Latest]', 'Fixed Assets Ratio[Latest]',
       'Inventory Ratio[Latest]', 'Debtors Ratio[Latest]',
       'Total Asset Turnover Ratio[Latest]', 'Interest Cover Ratio[Latest]',
       '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 of Output/Gross Block'],
      dtype='object')
```

Figure 1: Names of the columns in the dataset

Figure 1 shows the names of the columns in the dataset. It is seen that there are many special characters such as '%', '/', '(', ')', '[', ']', '.', '-' and spaces. These special characters and spaces are fixed for the ease of use.

```
Index(['Co_Code', 'Co_Name', 'Networth_Next_Year', 'Equity_Paid_Up',
       'Networth', 'Capital_Employed', 'Total_Debt', 'Gross_Block',
       'Net_Working_Capital', 'Current_Assets',
       'Current_Liabilities_and_Provisions', 'Total_Assets_by_Liabilities',
       'Gross_Sales', 'Net_Sales', 'Other_Income', 'Value_Of_Output',
       'Cost_of_Production', 'Selling_Cost', 'PBIDT', 'PBDT', 'PBIT', 'PBT',
       'PAT', 'Adjusted_PAT', 'CP', 'Revenue_earnings_in_forex',
       'Revenue_expenses_in_forex', 'Capital_expenses_in_forex',
       'Book_Value_Unit_Curr', 'Book_Value_Adj_Unit_Curr',
       'Market_Capitalisation', 'CEPS_annualised_Unit_Curr',
       'Cash_Flow_From_Operating_Activities',
       'Cash_Flow_From_Investing_Activities',
       'Cash_Flow_From_Financing_Activities', 'ROG_Net_Worth_perc',
       'ROG_Capital_Employed_perc', 'ROG_Gross_Block_perc',
       'ROG_Gross_Sales_perc', 'ROG_Net_Sales_perc',
       'ROG_Cost_of_Production_perc', 'ROG_Total_Assets_perc',
       'ROG_PBIDT_perc', 'ROG_PBDT_perc', 'ROG_PBIT_perc', 'ROG_PBT_perc',
       'ROG_PAT_perc', 'ROG_CP_perc', 'ROG_Revenue_earnings_in_forex_perc',
       'ROG_Revenue_expenses_in_forex_perc', 'ROG_Market_Capitalisation_perc',
       'Current_Ratio_Latest', 'Fixed_Assets_Ratio_Latest',
       'Inventory_Ratio_Latest', 'Debtors_Ratio_Latest',
       'Total_Asset_Turnover_Ratio_Latest', 'Interest_Cover_Ratio_Latest',
       'PBIDTM_perc_Latest', 'PBITM_perc_Latest', 'PBDTM_perc_Latest',
       'CPM_perc_Latest', 'APATM_perc_Latest', 'Debtors_Velocity_Days',
       'Creditors_Velocity_Days', 'Inventory_Velocity_Days',
       'Value_of_Output_by_Total_Assets', 'Value_of_Output_by_Gross_Block'],
      dtype='object')
```

Figure 2: Names of the columns after fixing the special characters and spaces

Figure 2 shows the names of the columns after fixing the special characters and spaces.

Table 2: Head of the dataset after fixing the column names (Image not complete due to size constraint)

	Co_Code	Co_Name	Networth_Next_Year	Equity_Paid_Up	Networth	Capital_Employed	Total_Debt	Gross_Block	Net_Working_Capital	Current_Assets	...	PBIDTM_perc_Latest
0	16974	Hind.Cables	-8021.60	419.36	-7027.48	-1007.24	5936.03	474.30	-1076.34	40.50	...	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
2	14852	ABG Shipyard	-3192.58	53.84	506.86	7714.68	6944.54	1281.54	4496.25	9097.64	...	-5279.14
3	2439	GTL	-3054.51	157.30	-623.49	2353.88	2326.05	1033.69	-2612.42	1034.12	...	-3.33
4	23505	Bharati Defence	-2967.36	50.30	-1070.83	4675.33	5740.90	1084.20	1836.23	4685.81	...	-295.55

5 rows × 67 columns

Table 2 shows the head of the dataset after fixing the special characters and spaces in the column names. The data is now analysed further.

Table 3: Information of the dataset - 1

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3586 entries, 0 to 3585
Data columns (total 67 columns):
 #   Column                                Non-Null Count  Dtype
---  ------                                --------------  -----
 0   Co_Code                               3586 non-null   int64
 1   Co_Name                               3586 non-null   object
 2   Networth_Next_Year                    3586 non-null   float64
 3   Equity_Paid_Up                        3586 non-null   float64
 4   Networth                              3586 non-null   float64
 5   Capital_Employed                      3586 non-null   float64
 6   Total_Debt                            3586 non-null   float64
 7   Gross_Block                           3586 non-null   float64
 8   Net_Working_Capital                   3586 non-null   float64
 9   Current_Assets                        3586 non-null   float64
 10  Current_Liabilities_and_Provisions    3586 non-null   float64
 11  Total_Assets_by_Liabilities           3586 non-null   float64
 12  Gross_Sales                           3586 non-null   float64
 13  Net_Sales                             3586 non-null   float64
 14  Other_Income                          3586 non-null   float64
 15  Value_Of_Output                       3586 non-null   float64
 16  Cost_of_Production                    3586 non-null   float64
 17  Selling_Cost                          3586 non-null   float64
 18  PBIDT                                 3586 non-null   float64
 19  PBDT                                  3586 non-null   float64
 20  PBIT                                  3586 non-null   float64
 21  PBT                                   3586 non-null   float64
 22  PAT                                   3586 non-null   float64
 23  Adjusted_PAT                          3586 non-null   float64
 24  CP                                    3586 non-null   float64
 25  Revenue_earnings_in_forex             3586 non-null   float64
```

Table 4: Information of the dataset - 2

26	Revenue_expenses_in_forex	3586 non-null	float64
27	Capital_expenses_in_forex	3586 non-null	float64
28	Book_Value_Unit_Curr	3586 non-null	float64
29	Book_Value_Adj_Unit_Curr	3582 non-null	float64
30	Market_Capitalisation	3586 non-null	float64
31	CEPS_annualised_Unit_Curr	3586 non-null	float64
32	Cash_Flow_From_Operating_Activities	3586 non-null	float64
33	Cash_Flow_From_Investing_Activities	3586 non-null	float64
34	Cash_Flow_From_Financing_Activities	3586 non-null	float64
35	ROG_Net_Worth_perc	3586 non-null	float64
36	ROG_Capital_Employed_perc	3586 non-null	float64
37	ROG_Gross_Block_perc	3586 non-null	float64
38	ROG_Gross_Sales_perc	3586 non-null	float64
39	ROG_Net_Sales_perc	3586 non-null	float64
40	ROG_Cost_of_Production_perc	3586 non-null	float64
41	ROG_Total_Assets_perc	3586 non-null	float64
42	ROG_PBIDT_perc	3586 non-null	float64
43	ROG_PBDT_perc	3586 non-null	float64
44	ROG_PBIT_perc	3586 non-null	float64
45	ROG_PBT_perc	3586 non-null	float64
46	ROG_PAT_perc	3586 non-null	float64
47	ROG_CP_perc	3586 non-null	float64
48	ROG_Revenue_earnings_in_forex_perc	3586 non-null	float64
49	ROG_Revenue_expenses_in_forex_perc	3586 non-null	float64
50	ROG_Market_Capitalisation_perc	3586 non-null	float64
51	Current_Ratio_Latest	3585 non-null	float64
52	Fixed_Assets_Ratio_Latest	3585 non-null	float64
53	Inventory_Ratio_Latest	3585 non-null	float64
54	Debtors_Ratio_Latest	3585 non-null	float64
55	Total_Asset_Turnover_Ratio_Latest	3585 non-null	float64
56	Interest_Cover_Ratio_Latest	3585 non-null	float64

Table 5: Information of the dataset - 3

57	PBIDTM_perc_Latest	3585 non-null	float64
58	PBITM_perc_Latest	3585 non-null	float64
59	PBDTM_perc_Latest	3585 non-null	float64
60	CPM_perc_Latest	3585 non-null	float64
61	APATM_perc_Latest	3585 non-null	float64
62	Debtors_Velocity_Days	3586 non-null	int64
63	Creditors_Velocity_Days	3586 non-null	int64
64	Inventory_Velocity_Days	3483 non-null	float64
65	Value_of_Output_by_Total_Assets	3586 non-null	float64
66	Value_of_Output_by_Gross_Block	3586 non-null	float64

dtypes: float64(63), int64(3), object(1)
memory usage: 1.8+ MB

The dataset has 67 columns and 3586 rows seen in Table 3, Table 4 and Table 5. There are 3586 records in most of the columns except a few. This means there are missing records based on this initial analysis that was done.

Table 6: Data type of the columns in the dataset

```
Co_Code                                int64
Co_Name                               object
Networth_Next_Year                   float64
Equity_Paid_Up                       float64
Networth                             float64
                                      ...
Debtors_Velocity_Days                  int64
Creditors_Velocity_Days                int64
Inventory_Velocity_Days              float64
Value_of_Output_by_Total_Assets      float64
Value_of_Output_by_Gross_Block       float64
Length: 67, dtype: object
```

The column 'Co_Name' is of object data type while all the other columns are of integer or float data type. This is seen in Table 3, Table 4, Table 5 and Table 6. There are 67 independent variables. The target variable 'Default' is derived from the variable 'Networth_Next_Year'.

```
The no. of rows:  3586
The no. of columns:  67
```

Figure 3: Shape of the dataset

The shape of the data is (3586, 67) meaning the dataset has 3586 rows and 67 columns shown in Figure 3.

Table 7: Missing value of the columns in the dataset

```
Co_Code                                0
Co_Name                                0
Networth_Next_Year                     0
Equity_Paid_Up                         0
Networth                               0
                                     ...
Debtors_Velocity_Days                  0
Creditors_Velocity_Days                0
Inventory_Velocity_Days              103
Value_of_Output_by_Total_Assets        0
Value_of_Output_by_Gross_Block         0
Length: 67, dtype: int64
```

Figure 4: Columns with missing values

```
(array([29, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 64]),)
```

The dataset was further checked for missing values and it is seen from Table 7 that there are missing values in the dataset. This was also seen earlier in Table 3, Table 4 and Table 5. Figure 4 shows the index value of the column with missing values.

```
Total number of missing values in the dataset is:  118
```

Figure 5: Number of missing values in the dataset

Figure 5 shows the number of missing values in the dataset. The missing values will be treated later.

The dataset is now checked for duplicates.

```
The no. of duplicated rows is:  0
```

Figure 6: Number of duplicates in the dataset

The dataset is checked for duplicates and it was found that there were no duplicate rows as seen in Figure 6.

Table 8: Descriptive statistics of the numerical columns in the dataset

	count	mean	std	min	25%	50%	75%	max
Co_Code	3586.0	16065.388734	19776.817379	4.00	3029.2500	6077.500	24269.5000	72493.00
Networth_Next_Year	3586.0	725.045251	4769.681004	-8021.60	3.9850	19.015	123.8025	111729.10
Equity_Paid_Up	3586.0	62.966584	778.761744	0.00	3.7500	8.290	19.5175	42263.46
Networth	3586.0	649.746299	4091.988792	-7027.48	3.8925	18.580	117.2975	81657.35
Capital_Employed	3586.0	2799.611054	26975.135385	-1824.75	7.6025	39.090	226.6050	714001.25
...	...	...	...	...	...	...	...	...
Debtors_Velocity_Days	3586.0	603.894032	10636.759580	0.00	8.0000	49.000	106.0000	514721.00
Creditors_Velocity_Days	3586.0	2057.854992	54169.479197	0.00	8.0000	39.000	89.0000	2034145.00
Inventory_Velocity_Days	3483.0	79.644559	137.847792	-199.00	0.0000	35.000	96.0000	996.00
Value_of_Output_by_Total_Assets	3586.0	0.819757	1.201400	-0.33	0.0700	0.480	1.1600	17.63
Value_of_Output_by_Gross_Block	3586.0	61.884548	976.824352	-61.00	0.2700	1.530	4.9100	43404.00

66 rows × 8 columns

Table 8 shows the description or the summary of the numerical columns in the dataset. The values of mean, standard deviation, minimum and maximum, 25th, 50th and 75th percentile is mentioned in

the above table. It can be seen that there are missing values in few columns. By looking at Table 8, we are able to deduce that the mean and standard deviation value of each variable has vast differences. This is probably because each column or variable are different measurements. For example: Debtors Velocity is measured in days while Networth or Equity Paid Up is a measure of currency.

The criteria in the project question requires the learner to do outlier treatment and missing value treatment. While solving this project question, **missing values is imputed first and then the outliers are treated**. It is done in this order because while capping the data, a small bias is brought to the actual dataset. The outliers will finally affect the model. If missing values is imputed first and then the outliers are treated, **the bias comes up only once**. If it is done the other way round, the bias is brought into the system while capping the data and imputation (mean or median) of the data. This will further increase the bias as the imputation of the unknown values will be based on the capped values as well.

The variable 'Co_Name' is dropped as it adds no value to the data. Now all the variables in the dataset are of numerical data type.

1.2 Missing Value Treatment

Table 9: Columns with the number of missing values

Book_Value_Adj_Unit_Curr	4
Current_Ratio_Latest	1
Fixed_Assets_Ratio_Latest	1
Inventory_Ratio_Latest	1
Debtors_Ratio_Latest	1
Total_Asset_Turnover_Ratio_Latest	1
Interest_Cover_Ratio_Latest	1
PBIDTM_perc_Latest	1
PBITM_perc_Latest	1
PBDTM_perc_Latest	1
CPM_perc_Latest	1
APATM_perc_Latest	1
Inventory_Velocity_Days	103
dtype: int64	

It is seen from Table 7 and Figure 4 that there are missing values in the dataset. However, considering the size of the dataset (3586 rows), there are not many missing values comparatively. There are a total of 118 missing records across 13 columns.

The columns with missing values were **imputed with median**. The replacement of missing values with median eliminates impact of outliers in the treatment. Therefore, imputation is done using median values.

There were no missing values after the treatment as seen in Table 10 and Figure 7.

Table 10: After treating the missing values

```
Co_Code                                0
Networth_Next_Year                     0
Equity_Paid_Up                         0
Networth                               0
Capital_Employed                       0
                                      ..
Debtors_Velocity_Days                  0
Creditors_Velocity_Days                0
Inventory_Velocity_Days                0
Value_of_Output_by_Total_Assets        0
Value_of_Output_by_Gross_Block         0
Length: 66, dtype: int64
```

```
The number of missing values after treatment is:  0
```

Figure 7: Number of missing values after treatment

The dataset that was split into independent and dependent variable for missing value treatment is concatenated after the treatment.

1.1 Outlier Treatment

A value is considered an outlier if it is more than 1.5 times the interquartile range below the first quartile (Q1) or above the third quartile (Q3).

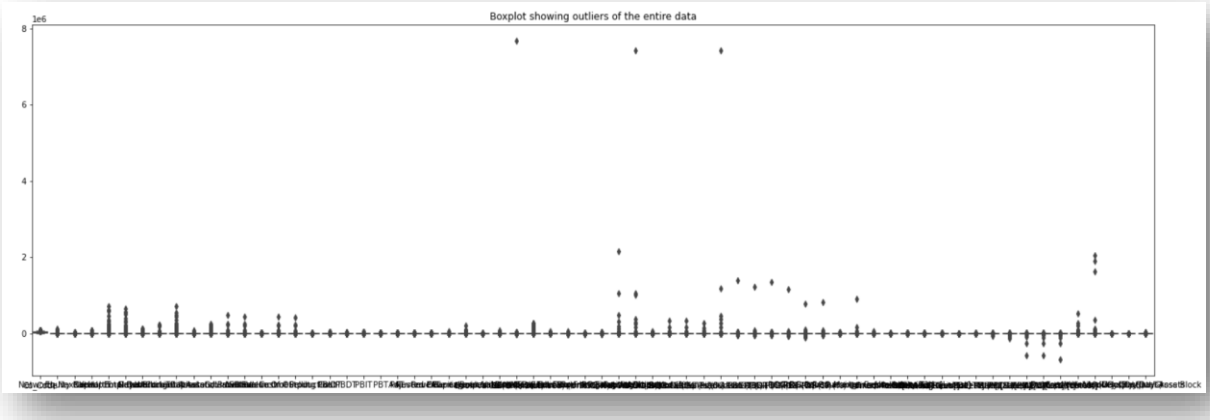

Figure 8: Boxplot showing the outliers in the data

Boxplot is used to plot the data to see if there are outliers in the data. The black points in Figure 8 represent the outliers.

Most of the numerical variables have outliers present.

Before treating the outliers, the data is split into independent and dependent or target variable. There is no target variable defined. Since the objective is to build a model for the investor to decode which company to invest in – the variable 'Networth_Next_Year' could be used to transform into

target variable. The outlier treatment is done only for the independent variables as the treatment should not affect the dependent variable and hence the result. Therefore, outlier treatment is not done for the variable 'Networth_Next_Year'.

Table 11 shows the number of outliers in each column and Figure 9 shows the total number of outliers in the entire dataset.

Table 11: Column names and number of outliers in each column

Co_Code	291
Equity_Paid_Up	448
Networth	650
Capital_Employed	596
Total_Debt	583
	...
Debtors_Velocity_Days	398
Creditors_Velocity_Days	391
Inventory_Velocity_Days	279
Value_of_Output_by_Total_Assets	150
Value_of_Output_by_Gross_Block	481

Length: 65, dtype: int64

```
The number of outliers present in the dataset:  41666
```

Figure 9: Total Number of outliers in the entire dataset

The outliers are treated by replacing values. The values above the third quartile (Q3) are replaced with the third quartile value. The values above the first quartile (Q1) are replaced with the first quartile value. Figure 10 shows the boxplot of the data after treating the outliers. It is seen that there are no outliers in the graph.

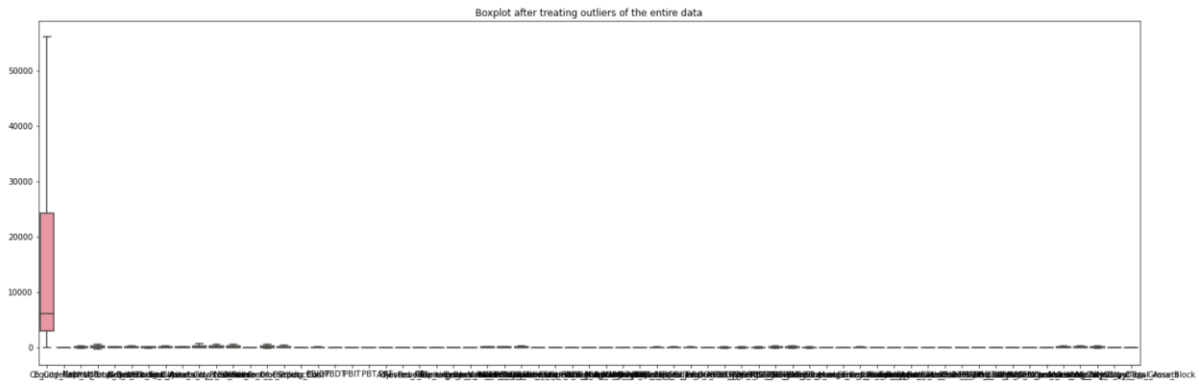

Figure 10: Boxplot of the data after treating the outliers

Table 12: Column names and number of outliers after treatment

```
Co_Code                                0
Equity_Paid_Up                         0
Networth                               0
Capital_Employed                       0
Total_Debt                             0
                                      ..
Debtors_Velocity_Days                  0
Creditors_Velocity_Days                0
Inventory_Velocity_Days                0
Value_of_Output_by_Total_Assets        0
Value_of_Output_by_Gross_Block         0
Length: 65, dtype: int64
```

```
The number of outliers present in the dataset after outlier treatment:  0
```

Figure 11: Number of outliers after treatment

Table 12 and Figure 11 shows that there are no outliers after treatment.

The dataset that was split into independent and dependent variable for outlier treatment is concatenated after the treatment.

1.3 Transform Target variable into 0 and 1

There is no target variable defined. Since the objective is to build a model for the investor to decode which company to invest in – the variable 'Networth_Next_Year' could be used to transform into target variable. A new dependent variable named 'Default' was created based on the variable 'Networth_Next_Year'.

If the company's 'Networth_Next_Year' value is equal to or less than 0 (negative), then the company would likely not return a good investment to investor. These values are therefore transformed as 1.

If the company's 'Networth_Next_Year' value is greater than 0 (positive), then the company would return a good investment to investor. These values are therefore transformed as 0.

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

- 1 – if the 'Networth_Next_Year' is negative for the company (default)
- 0 – if the 'Networth_Next_Year' is positive for the company (non-default)

Table 13: Head of the dataset after adding the dependent variable 'Default' (Image not complete due to size constraint)

Creditors_Velocity_Days	Inventory_Velocity_Days	Value_of_Output_by_Total_Assets	Value_of_Output_by_Gross_Block	Networth_Next_Year	Default
0.0	45.0	0.00	0.00	-8021.60	1
101.0	2.0	0.31	0.24	-3986.19	1
210.5	0.0	-0.03	-0.26	-3192.58	1
63.0	2.0	0.24	1.90	-3054.51	1
210.5	0.0	0.01	0.05	-2967.36	1

Table 14 and Table 15 shows the head and tail of 'Default' and 'Networth_Next_Year' variables alone.

Table 14: A snipped of 'Default' variable created based on 'Networth_Next_Year' (showing head)

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

Table 15: A snipped of 'Default' variable created based on 'Networth_Next_Year' (showing tail)

	Default	Networth_Next_Year
3581	0	72677.77
3582	0	79162.19
3583	0	88134.31
3584	0	91293.70
3585	0	111729.10

After generating the dependent column, the split of data is checked based on the dependent variable. Figure 12 shows the count plot of the 'Default' variable and Figure 13 shows the value counts of the 'Default' variable.

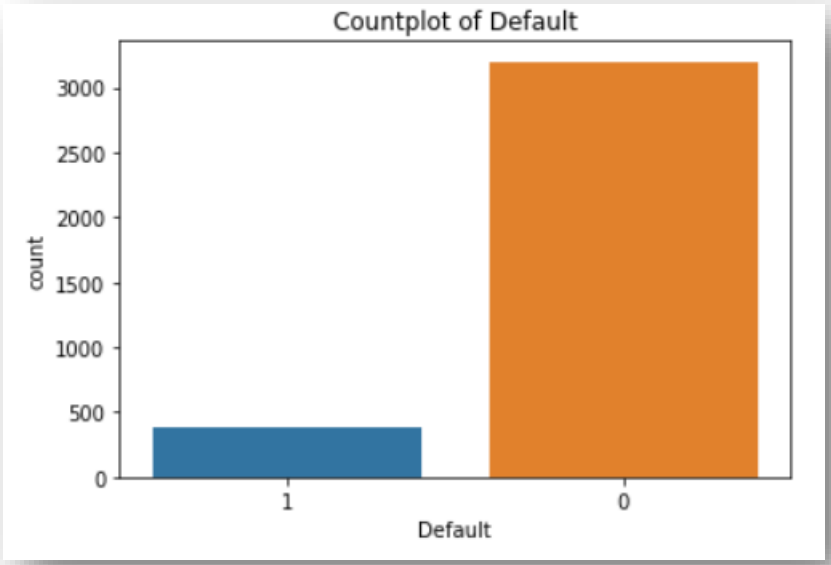

Figure 12: Countplot of 'Default' variable

```
0    3198
1     388
Name: Default, dtype: int64
```

Figure 13: Value counts of 'Default' variable

From Figure 13, it is seen that there are 388 defaulters and 3198 non-defaulters in the dataset. The proportion of 'Default' is checked and shown in Figure 14. It is seen that there are 10.8% default and 89.1% non-default in the data.

```
0    0.891801
1    0.108199
Name: Default, dtype: float64
```

Figure 14: Proportion of 'Default'

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

The **outliers and missing values in the data is already treated** before performing the exploratory data analysis (EDA).

The **EDA was only done for the variables that were significant in model building**. The variables that are important for the case study were found using Variance Inflation Factor (VIF) and the model was built. Then EDA was done for those variables. Figure 15 shows the boxplot of the significant variables. There are no outliers as they were already treated prior to performing EDA.

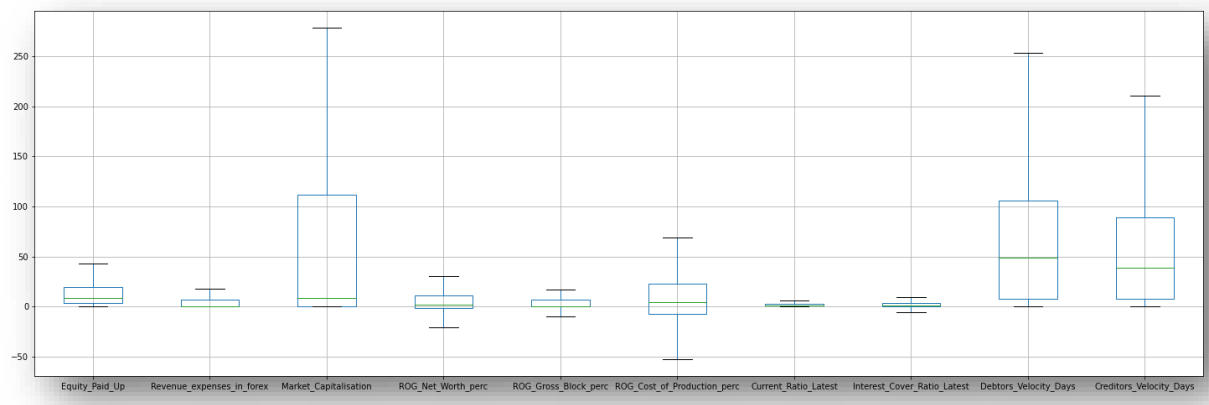

Figure 15: Boxlpot of significant variables

Univariate Analysis:

Numerical Variables:

All the significant variables are numerical. These numerical variables were used to calculate the skewness, plot the univariate distribution and the boxplot.

Table 16: Skewness of the significant variables of the dataset

Market_Capitalisation	1.203955
Revenue_expenses_in_forex	1.193516
Creditors_Velocity_Days	1.143302
Equity_Paid_Up	1.141900
Debtors_Velocity_Days	1.136086
Current_Ratio_Latest	1.049903
Interest_Cover_Ratio_Latest	0.497618
ROG_Gross_Block_perc	0.429378
ROG_Net_Worth_perc	0.203921
ROG_Cost_of_Production_perc	0.170903
dtype: float64	

1. Equity_Paid_Up

Table 17: Description of 'Equity_Paid_Up'

```
Description of Equity_Paid_Up
-----------------------------------
count    3586.000000
mean       13.994651
std        14.001442
min         0.000000
25%         3.750000
50%         8.290000
75%        19.517500
max        43.168750
Name: Equity_Paid_Up, dtype: float64
```

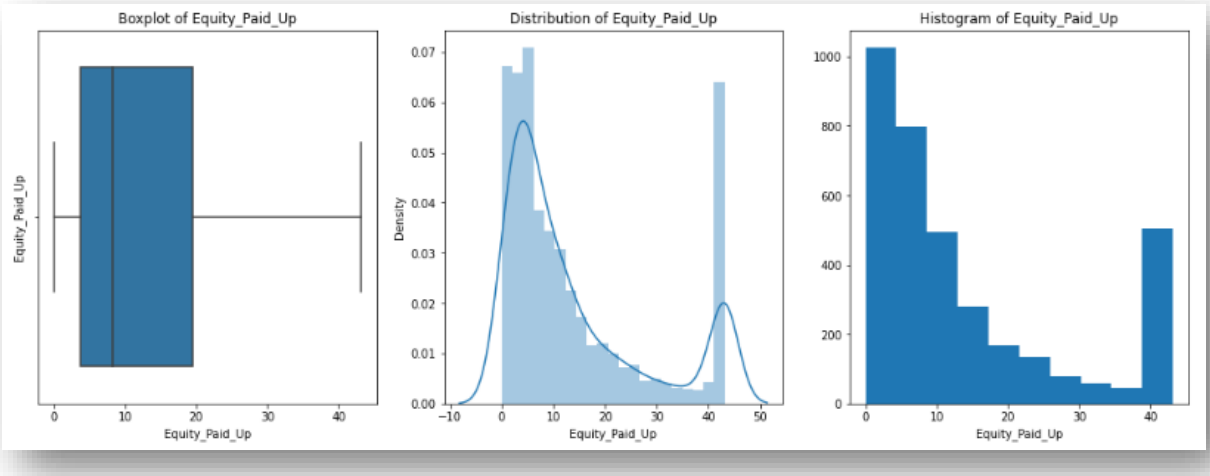

Figure 16: Boxplot, univariate distribution and histogram of 'Equity_Paid_Up'

Univariate analysis of 'Equity_Paid_Up' is done to understand the patterns and distribution of the data. From Figure 16, we can see that the Box plot of 'Equity_Paid_Up' variable has no outliers. The distribution of the data is moderately right skewed which is seen in Figure 16. This is also seen in Table 16 where the skewness values are given. The skewness value of 'Equity_Paid_Up' variable is 1.141900. From Table 17, it is seen that the mean of the data is 13.994651 meaning amount that has been received by the company through the issue of shares to the shareholders is 13.99 on average.

2. Revenue_expenses_in_forex

Table 18: Description of 'Revenue_expenses_in_forex'

```
Description of Revenue_expenses_in_forex
-----------------------------------------------
count    3586.000000
mean        4.370090
std         7.024458
min         0.000000
25%         0.000000
50%         0.000000
75%         6.987500
max        17.468750
Name: Revenue_expenses_in_forex, dtype: float64
```

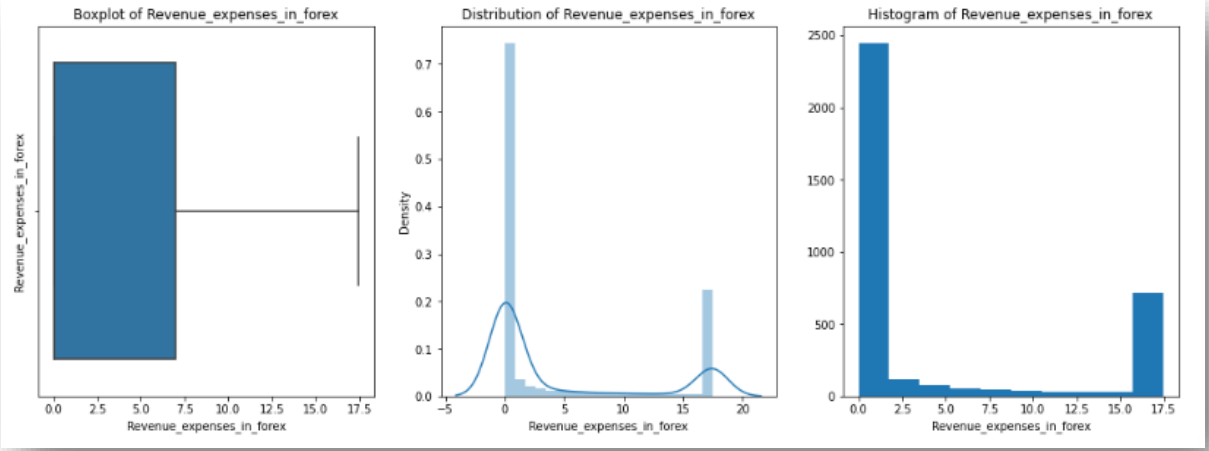

Figure 17: Boxplot, univariate distribution and histogram of 'Revenue_expenses_in_forex'

Univariate analysis of 'Revenue_expenses_in_forex' is done to understand the patterns and distribution of the data. From Figure 17, we can see that the Box plot of 'Revenue_expenses_in_forex' variable has no outliers. The distribution of the data is moderately right skewed which is seen in Figure 17. This is also seen in Table 16 where the skewness values are given. The skewness value of 'Revenue_expenses_in_forex' variable is 1.193516. From Table 18, it is seen that the mean of the data is 4.370090 meaning the expenses due to foreign currency transactions is 4.37 on average.

3. Market_Capitalisation

Table 19: Description of 'Market_Capitalisation'

```
Description of Market_Capitalisation
-----------------------------------------
count    3586.000000
mean       72.370378
std       107.359374
min         0.000000
25%         0.000000
50%         8.370000
75%       111.457500
max       278.643750
Name: Market_Capitalisation, dtype: float64
```

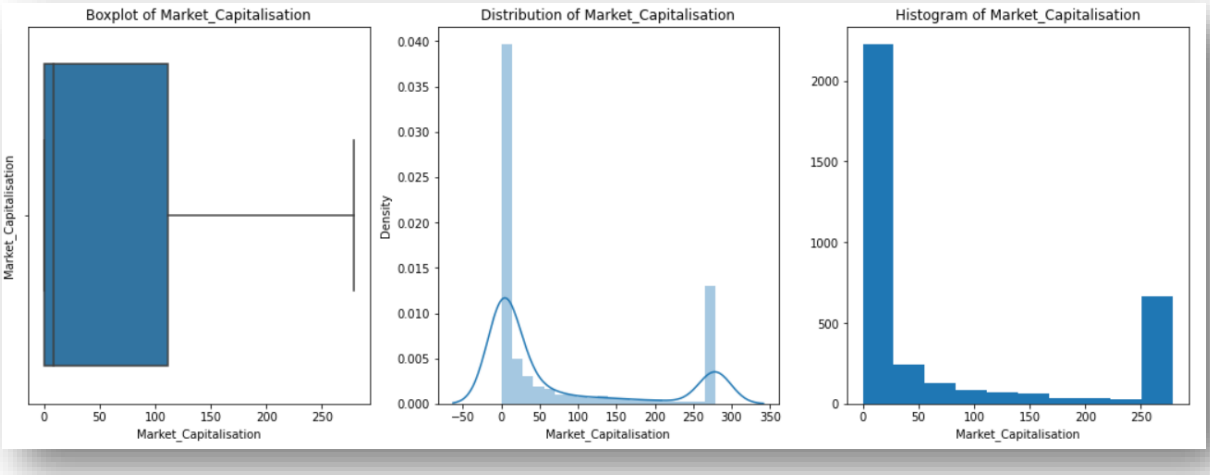

Figure 18: Boxplot, univariate distribution and histogram of 'Market_Capitalisation'

Univariate analysis of 'Market_Capitalisation' is done to understand the patterns and distribution of the data. From Figure 18, we can see that the Box plot of 'Market_Capitalisation' variable has no outliers. The distribution of the data is moderately right skewed which is seen in Figure 18. This is also seen in Table 16 where the skewness values are given. The skewness value of 'Market_Capitalisation' variable is 1.203955. From Table 19, it is seen that the mean of the data is 72.370378 meaning the product of the total number of a company's outstanding shares and the current market price of one share is 72.37 on average.

4. ROG_Net_Worth_perc

Table 20: Description of 'ROG_Net_Worth_perc'

```
Description of ROG_Net_Worth_perc
----------------------------------------
count    3586.000000
mean        4.123604
std        14.300085
min       -20.762500
25%        -1.487500
50%         1.840000
75%        11.362500
max        30.637500
Name: ROG_Net_Worth_perc, dtype: float64
```

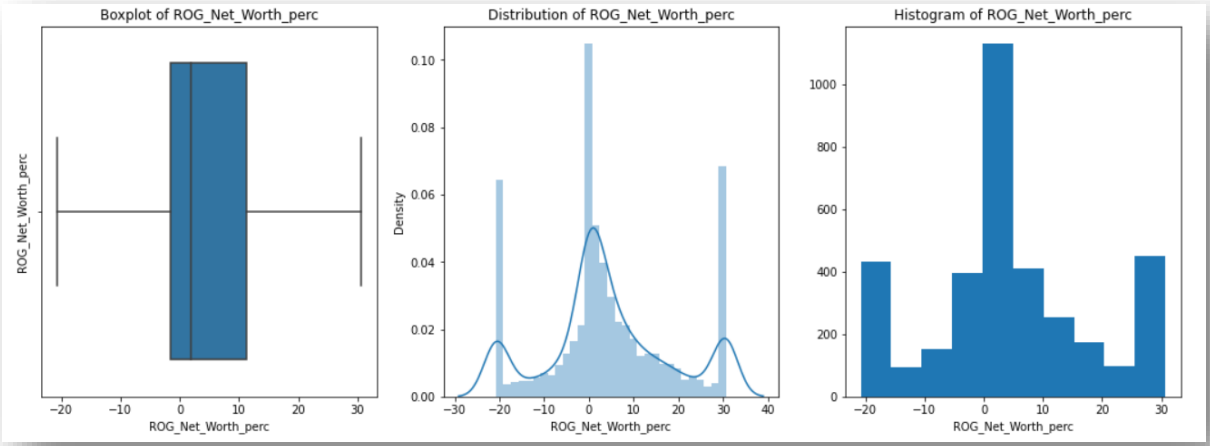

Figure 19: Boxplot, univariate distribution and histogram of 'ROG_Net_Worth_perc'

Univariate analysis of 'ROG_Net_Worth_perc' is done to understand the patterns and distribution of the data. From Figure 19, we can see that the Box plot of 'ROG_Net_Worth_perc' variable has no outliers. The distribution of the data is moderately right skewed which is seen in Figure 19. This is also seen in Table 16 where the skewness values are given. The skewness value of 'ROG_Net_Worth_perc' variable is 0.203921. From Table 20, it is seen that the mean of the data is 4.123604 meaning the rate of growth-networth is 4.12 on average.

5. ROG_Gross_Block_perc

Table 21: Description of 'ROG_Gross_Block_perc'

```
Description of ROG_Gross_Block_perc
-------------------------------------------
count    3586.000000
mean        2.946049
std         7.652767
min       -10.080000
25%         0.000000
50%         0.250000
75%         6.720000
max        16.800000
Name: ROG_Gross_Block_perc, dtype: float64
```

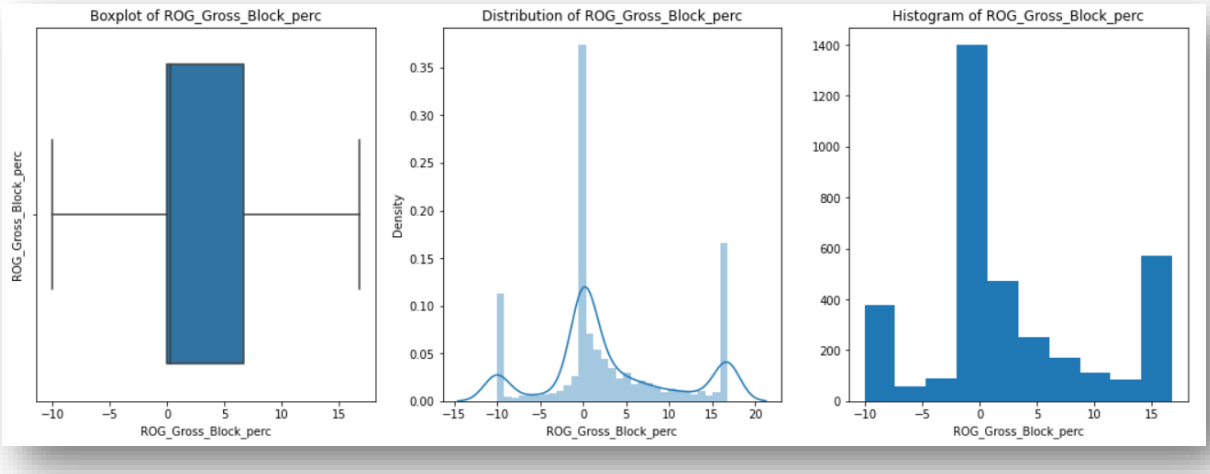

Figure 20: Boxplot, univariate distribution and histogram of 'ROG_Gross_Block_perc'

Univariate analysis of 'ROG_Gross_Block_perc' is done to understand the patterns and distribution of the data. From Figure 20, we can see that the Box plot of 'ROG_Gross_Block_perc' variable has no outliers. The distribution of the data is moderately right skewed which is seen in Figure 20. This is also seen in Table 16 where the skewness values are given. The skewness value of 'ROG_Gross_Block_perc' variable is 0.429378. From Table 21, it is seen that the mean of the data is 2.946049 meaning the rate of growth-gross block is 2.94 on average.

6. ROG_Cost_of_Production_perc

Table 22: Description of 'ROG_Cost_of_Production_perc'

```
Description of ROG_Cost_of_Production_perc
--------------------------------------------------
count    3586.000000
mean        7.889492
std        33.097493
min       -52.790000
25%        -7.242500
50%         4.415000
75%        23.122500
max        68.670000
Name: ROG_Cost_of_Production_perc, dtype: float64
```

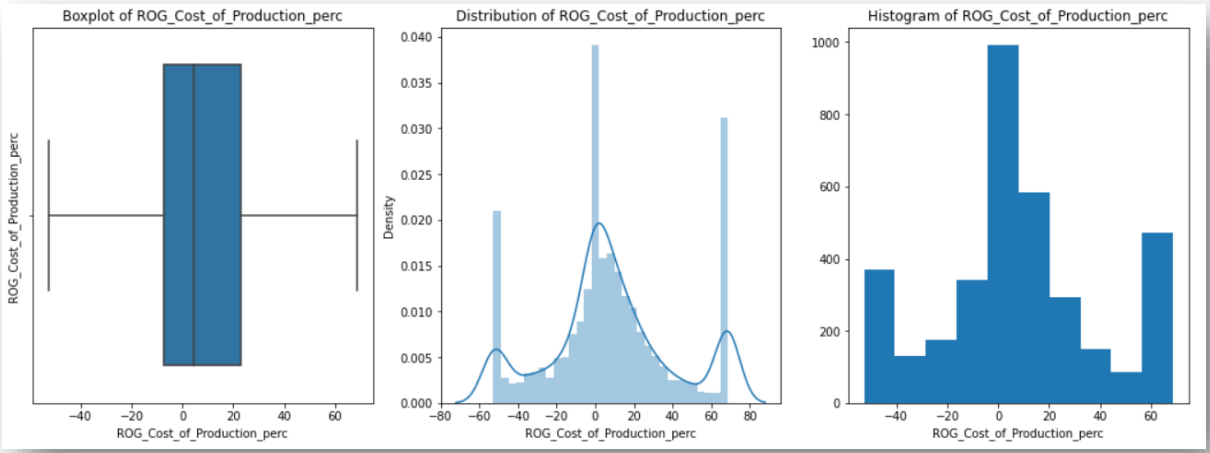

Figure 21: Boxplot, univariate distribution and histogram of 'ROG_Cost_of_Production_perc'

Univariate analysis of 'ROG_Cost_of_Production_perc' is done to understand the patterns and distribution of the data. From Figure 21, we can see that the Box plot of 'ROG_Cost_of_Production_perc' variable has no outliers. The distribution of the data is moderately right skewed which is seen in Figure 21. This is also seen in Table 16 where the skewness values are given. The skewness value of 'ROG_Cost_of_Production_perc' variable is 0.170903. From Table 22, it is seen that the mean of the data is 7.889492 meaning the rate of growth-cost of production is 7.89 on average.

7. Current_Ratio_Latest

Table 23: Description of 'Current_Ratio_Latest'

```
Description of Current_Ratio_Latest
-----------------------------------------
count    3586.000000
mean        2.084084
std         1.806351
min         0.000000
25%         0.880000
50%         1.360000
75%         2.770000
max         5.605000
Name: Current_Ratio_Latest, dtype: float64
```

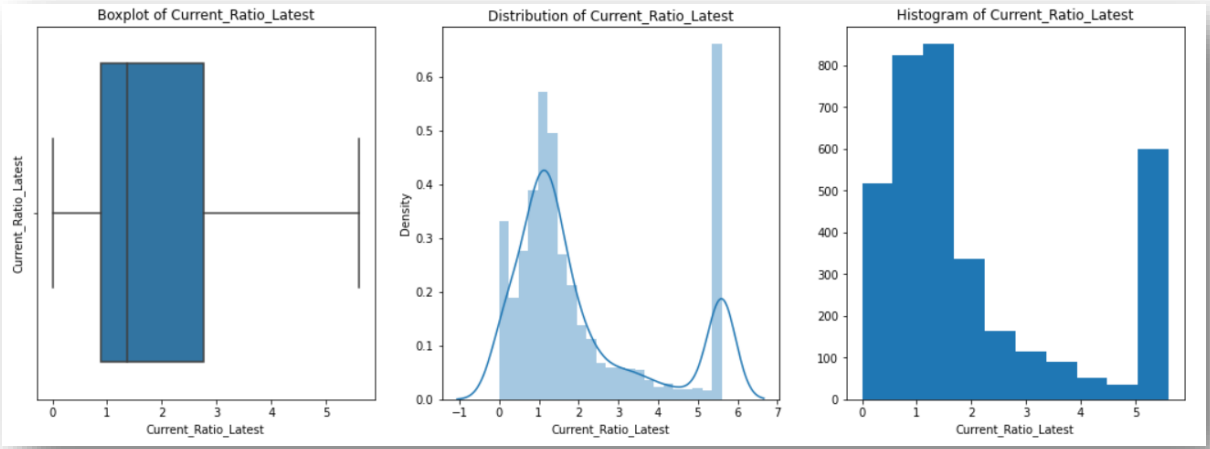

Figure 22: Boxplot, univariate distribution and histogram of 'Current_Ratio_Latest'

Univariate analysis of 'Current_Ratio_Latest' is done to understand the patterns and distribution of the data. From Figure 22Figure 21, we can see that the Box plot of 'Current_Ratio_Latest' variable has no outliers. The distribution of the data is moderately right skewed which is seen in Figure 22. This is also seen in Table 16 where the skewness values are given. The skewness value of 'Current_Ratio_Latest' variable is 1.049903. From Table 23, it is seen that the mean of the data is 2.084084 meaning the liquidity ratio, company's ability to pay short-term obligations or those due within one year is 2.0 on average.

8. Interest_Cover_Ratio_Latest

Table 24: Description of 'Interest_Cover_Ratio_Latest'

```
Description of Interest_Cover_Ratio_Latest
--------------------------------------------------
count    3586.000000
mean        2.078465
std         3.912223
min        -5.565000
25%         0.000000
50%         1.080000
75%         3.710000
max         9.275000
Name: Interest_Cover_Ratio_Latest, dtype: float64
```

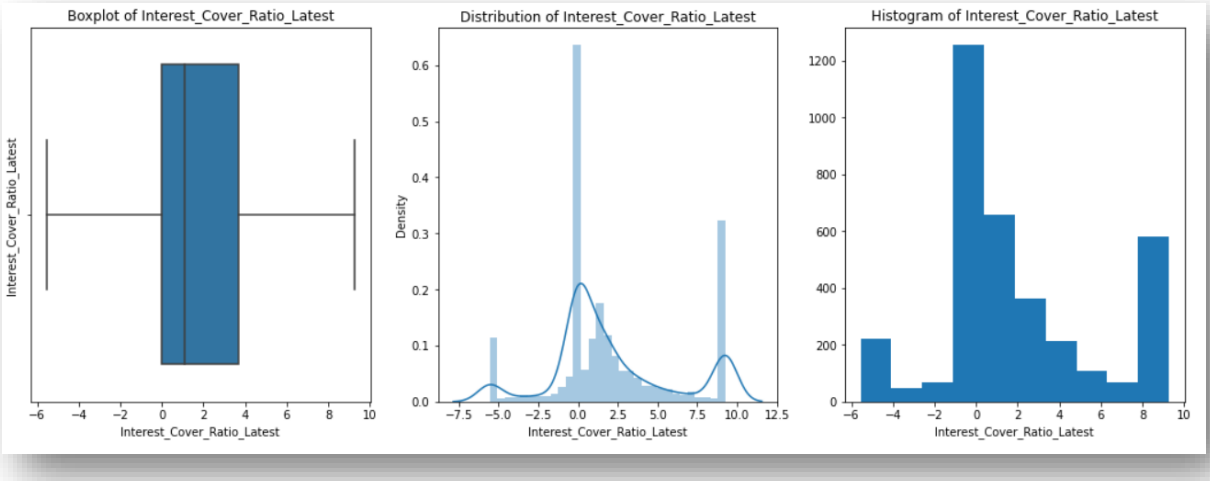

Figure 23: Boxplot, univariate distribution and histogram of 'Interest_Cover_Ratio_Latest'

Univariate analysis of 'Interest_Cover_Ratio_Latest' is done to understand the patterns and distribution of the data. From Figure 23Figure 21, we can see that the Box plot of 'Interest_Cover_Ratio_Latest' variable has no outliers. The distribution of the data is moderately right skewed which is seen in Figure 23. This is also seen in Table 16 where the skewness values are given. The skewness value of 'Interest_Cover_Ratio_Latest' variable is 0.497618. From Table 24, it is seen that the mean of the data is 2.078465. The 'Interest_Cover_Ratio_Latest' determines how easily a company can pay interest on its outstanding debt.

9. Debtors_Velocity_Days

Table 25: Description of 'Debtors_Velocity_Days'

```
Description of Debtors_Velocity_Days
-----------------------------------------
count    3586.000000
mean       75.286670
std        81.861954
min         0.000000
25%         8.000000
50%        49.000000
75%       106.000000
max       253.000000
Name: Debtors_Velocity_Days, dtype: float64
```

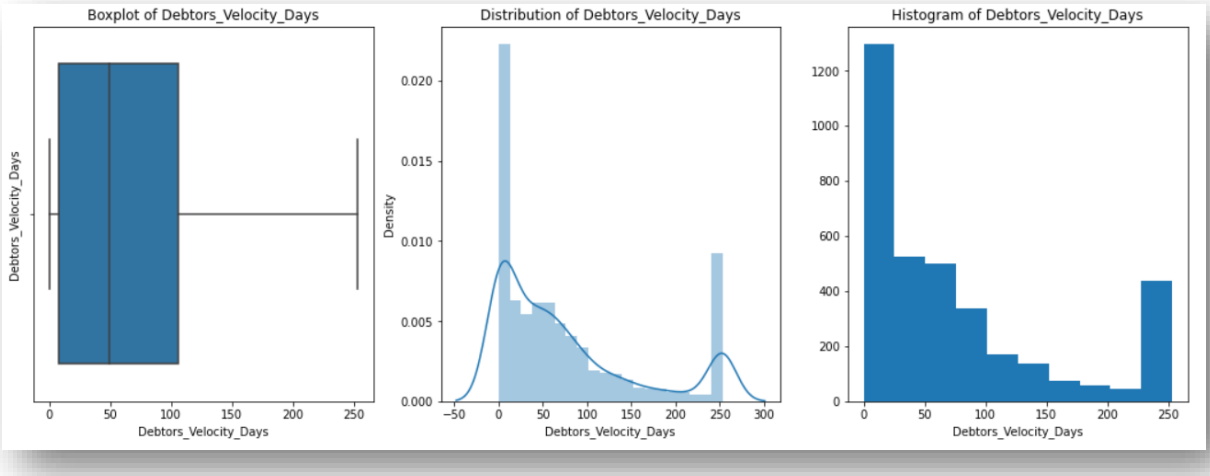

Figure 24: Boxplot, univariate distribution and histogram of 'Debtors_Velocity_Days'

Univariate analysis of 'Debtors_Velocity_Days' is done to understand the patterns and distribution of the data. From Figure 24Figure 21, we can see that the Box plot of 'Debtors_Velocity_Days' variable has no outliers. The distribution of the data is moderately right skewed which is seen in Figure 24. This is also seen in Table 16 where the skewness values are given. The skewness value of 'Debtors_Velocity_Days' variable is 1.136086. From Table 25, it is seen that the mean of the data is 75.286670 meaning the average days required for receiving the payments is 75 days on average.

10. Current_Ratio_Latest

Table 26: Description of 'Creditors_Velocity_Days'

```
Description of Creditors_Velocity_Days
-----------------------------------------------
count    3586.000000
mean       62.440742
std        68.144543
min         0.000000
25%         8.000000
50%        39.000000
75%        89.000000
max       210.500000
Name: Creditors_Velocity_Days, dtype: float64
```

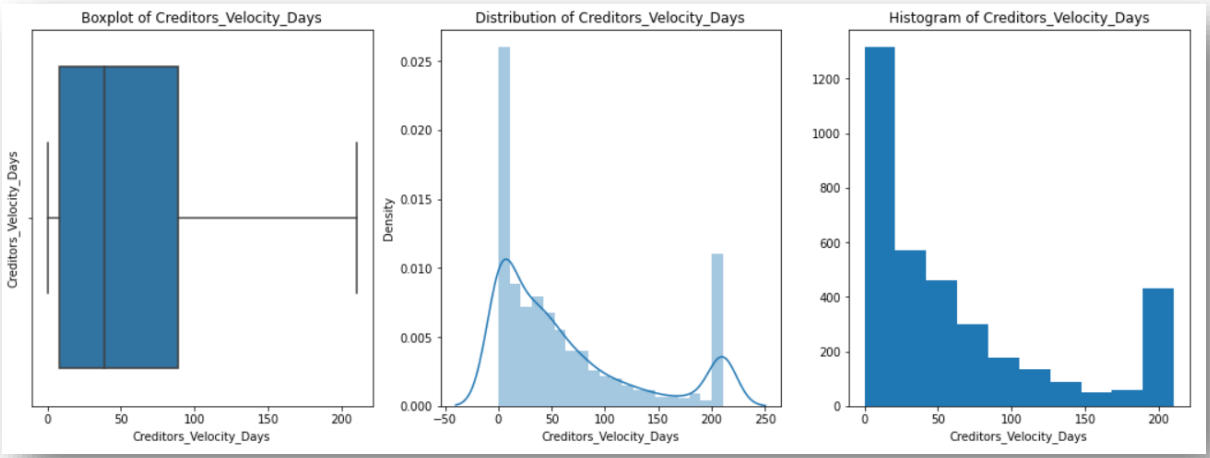

Figure 25: Boxplot, univariate distribution and histogram of 'Creditors_Velocity_Days'

Univariate analysis of 'Creditors_Velocity_Days' is done to understand the patterns and distribution of the data. From Figure 25Figure 21, we can see that the Box plot of 'Creditors_Velocity_Days' variable has no outliers. The distribution of the data is moderately right skewed which is seen in Figure 25. This is also seen in Table 16 where the skewness values are given. The skewness value of 'Creditors_Velocity_Days' variable is 1.143302. From Table 26, it is seen that the mean of the data is 62.440742 meaning the average number of days company takes to pay suppliers is 62 days on average.

Categorical Variable:

1. **Default**

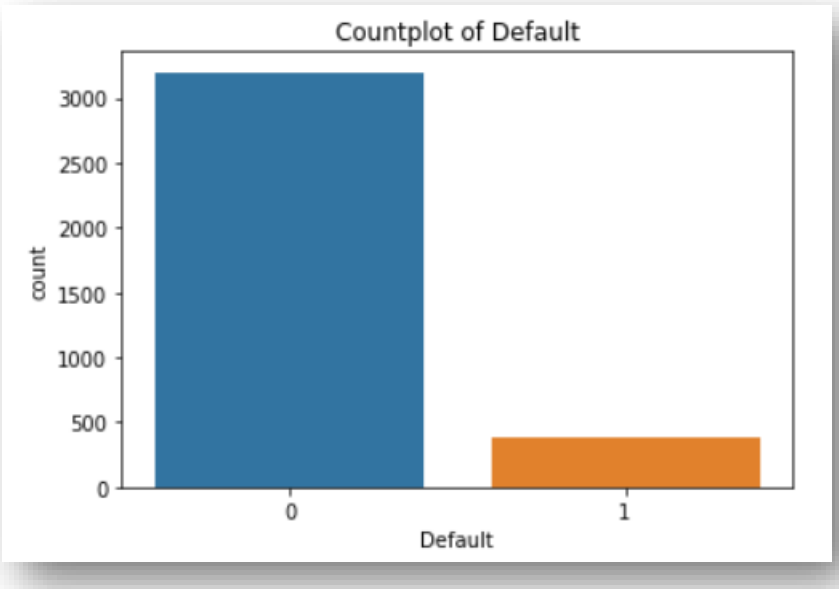

Figure 26: Countplot of target variable – Default

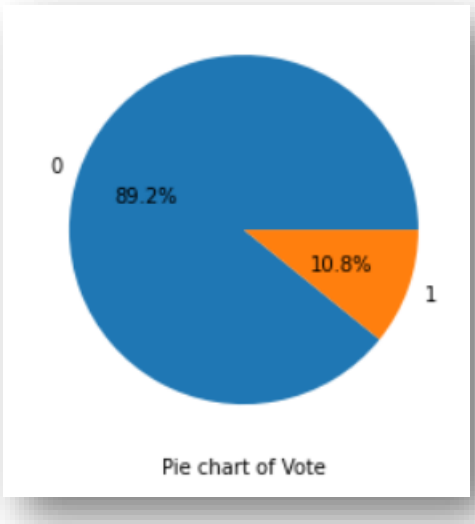

Figure 27: Pie chart of target variable – Default

Univariate analysis of 'Default' is done to understand the patterns and distribution of the data. From Figure 26 and Figure 27, we can understand that defaulters in the data are 10.8%. 89.2% of the data are non-defaulters.

Bivariate Analysis:

Figure 28: Heat map showing the bivariate analysis of the significant variables

Bivariate analysis is done using the help of a heat map. A heat map is used to understand the correlation between two numerical values in a dataset. Multicollinearity is an important issue which can destruct the model. Heatmap can help in identifying this issue. Figure 28 shows the heatmap of the dataset. However, the issue of **multicollinearity is addressed already while selecting the significant variables** using VIF.

Observations:

- The highest correlation is between the different features like 'Equity_Paid_Up' and 'Market_Capitalisation' (56%) and 'Revenue_expenses_in_forex' and 'Market_Capitalisation' (56%) although it is not a strong correlation
- There is less correlation in table with the other features
- 'Revenue_expenses_in_forex' is most negatively correlated with 'Current_Ratio_Latest' (-23%)
- Therefore, multicollinearity will not be an issue in these significant variables.

Multivariate Analysis:

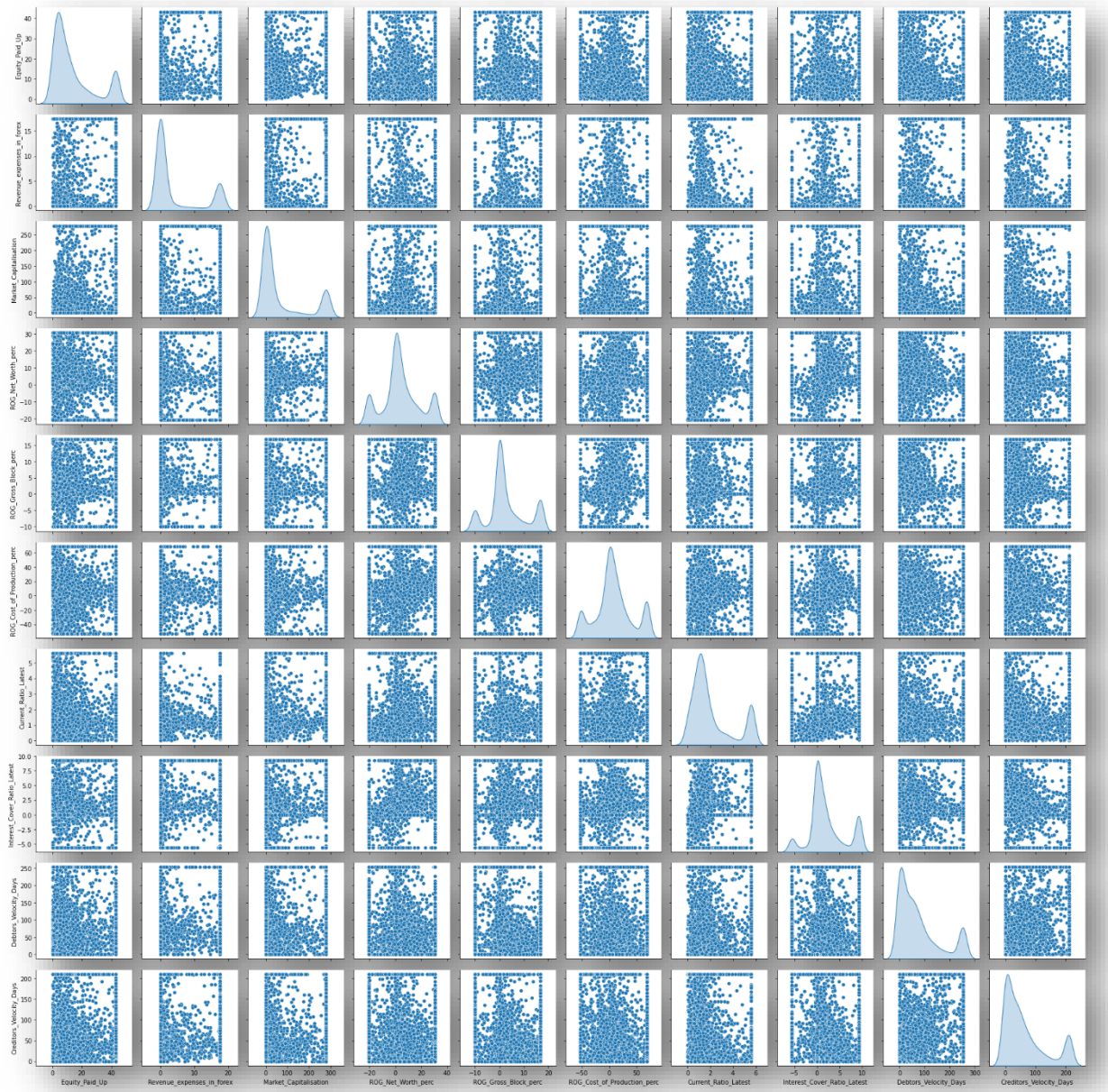

Figure 29: Pair plot showing the multivariate analysis of the significant variables in the dataset

Multivariate analysis is done using the help of a pair plot to understand the relationship between all the numerical values in the dataset. Pair plot can be used to compare all the variables with each other to understand the patterns or trends in the dataset. Figure 29 shows the pair plot of the significant variables.

1.5 Train Test Split

Data split: Splitting the data into train and test

The data is first split into train and test data before building the models. There is no fixed rule for separation training and testing data sets. The data was split into train and test in **67:33 ratio** and the random state was set to be 42 based on the criteria given in the project question. Stratify is done on 'Default' variable to make sure both train and test data have similar proportion of defaulters and non-defaulters. This is done as the dataset is imbalanced having more of non-defaulters (89.1%) as shown in Figure 12 and Figure 14.

```
Shape of the data after splitting into train and test set:
The training set for the independent variables:  (2402, 66)
The training set for the dependent variables:  (2402,)
The test set for the independent variables:  (1184, 66)
The test set for the independent variables:  (1184,)
```

Figure 30: Shape of the train and test set

The training set for the independent variables (X_train) denotes 67% training dataset with 66 columns (without the dependent variable – 'Default'). The training set for the dependent variable (y_train) denotes 67% test dataset with only the target column or the dependent variable 'Default'. The test set for the independent variables (X_test) denotes 33% training dataset with 66 columns (without the dependent variable – 'Default'). The test set for the dependent variable (y_test) denotes 33% test dataset with only the target column or the dependent variable 'Default'.

Figure 30 shows the shape of the train and test set.

Statsmodel requires the labelled data. Therefore, concatenating X_train y_train to form a train set and X_test y_test to form test set. Figure 31 shows the shape of the train and test set. This data can now be used for model building.

```
Shape of the data after concatenating into train and test set:
The shape of training set:  (2402, 67)
The shape of test set:  (1184, 67)
```

Figure 31: Shape of the train and test set after concatenating

```
The proportion of defaulters in training set:  0.10824313072439634
The proportion of defaulters in test set:  0.10810810810810811
```

Figure 32: Proportion of 'Default' in train and test set

```
The proportion of non-defaulters in training set:  0.8917568692756037
The proportion of non-defaulters in test set:  0.8918918918918919
```

Figure 33: Proportion of non-default in train and test set

The proportion of 'Default' in train and test set is checked and shown in Figure 32. It is seen that there are 10.8% default.

The proportion of non-default n train and test set is checked and shown in Figure 33. It is seen that there are 89.1% non-default in the data.

From Figure 32 and Figure 33 it is seen that the proportion of default and non-default is the same in train and test set. The proportion is also the same when compared to the default and non-default before the train test split as seen in Figure 14.

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

Logistic Regression

Logistic Regression model is supervised learning algorithm which can be used for classification type of problems. It establishes relation between dependent class variable and independent variables using regression.

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

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

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

Figure 34: Equation of Logistic Regression

Feature Selection

Before starting model building, the problem of multicollinearity has to be addressed. Multicollinearity occurs when two or more independent variables are highly correlated with one another in a regression model.

The optimal machine learning problem approach is to perform extensive EDA on dataset and understand properties of the predictors before even getting into training models on these variables. However, this is not always possible. Sometimes the dataset has lot many variables, sometimes even hundreds or even thousands of variables, which can quickly outrun human comprehension.

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

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

Feature selection is done using Variance Inflation Factor (VIF). Variance inflation factor (VIF) is a measure of the amount of multicollinearity in a set of multiple regression variables. Mathematically, the VIF for a regression model variable is equal to the ratio of the overall model variance to the variance of a model that includes only that single independent variable.

Feature selection method is done first and then it is validated back using manual feature selection using backward elimination approach.

Variance Inflation Factor (VIF)

Variance Inflation Factor (VIF) is done to measure the multicollinearity and to find the most significant columns in the dataset.

Figure 35 shows the columns in the train set.

```
Index(['Co_Code', 'Equity_Paid_Up', 'Networth', 'Capital_Employed',
       'Total_Debt', 'Gross_Block', 'Net_Working_Capital', 'Current_Assets',
       'Current_Liabilities_and_Provisions', 'Total_Assets_by_Liabilities',
       'Gross_Sales', 'Net_Sales', 'Other_Income', 'Value_Of_Output',
       'Cost_of_Production', 'Selling_Cost', 'PBIDT', 'PBDT', 'PBIT', 'PBT',
       'PAT', 'Adjusted_PAT', 'CP', 'Revenue_earnings_in_forex',
       'Revenue_expenses_in_forex', 'Capital_expenses_in_forex',
       'Book_Value_Unit_Curr', 'Book_Value_Adj_Unit_Curr',
       'Market_Capitalisation', 'CEPS_annualised_Unit_Curr',
       'Cash_Flow_From_Operating_Activities',
       'Cash_Flow_From_Investing_Activities',
       'Cash_Flow_From_Financing_Activities', 'ROG_Net_Worth_perc',
       'ROG_Capital_Employed_perc', 'ROG_Gross_Block_perc',
       'ROG_Gross_Sales_perc', 'ROG_Net_Sales_perc',
       'ROG_Cost_of_Production_perc', 'ROG_Total_Assets_perc',
       'ROG_PBIDT_perc', 'ROG_PBDT_perc', 'ROG_PBIT_perc', 'ROG_PBT_perc',
       'ROG_PAT_perc', 'ROG_CP_perc', 'ROG_Revenue_earnings_in_forex_perc',
       'ROG_Revenue_expenses_in_forex_perc', 'ROG_Market_Capitalisation_perc',
       'Current_Ratio_Latest', 'Fixed_Assets_Ratio_Latest',
       'Inventory_Ratio_Latest', 'Debtors_Ratio_Latest',
       'Total_Asset_Turnover_Ratio_Latest', 'Interest_Cover_Ratio_Latest',
       'PBIDTM_perc_Latest', 'PBITM_perc_Latest', 'PBDTM_perc_Latest',
       'CPM_perc_Latest', 'APATM_perc_Latest', 'Debtors_Velocity_Days',
       'Creditors_Velocity_Days', 'Inventory_Velocity_Days',
       'Value_of_Output_by_Total_Assets', 'Value_of_Output_by_Gross_Block',
       'Networth_Next_Year', 'Default'],
      dtype='object')
```

Figure 35: Columns in the train set

Table 27: Variables in the train set with their VIF values

	variables	VIF
65	Networth_Next_Year	1.267104
35	ROG_Gross_Block_perc	1.545711
48	ROG_Market_Capitalisation_perc	1.707102
62	Inventory_Velocity_Days	1.927059
38	ROG_Cost_of_Production_perc	2.030572
...	...	...
10	Gross_Sales	765.401445
11	Net_Sales	1420.352199
25	Capital_expenses_in_forex	NaN
46	ROG_Revenue_earnings_in_forex_perc	NaN
47	ROG_Revenue_expenses_in_forex_perc	NaN

66 rows × 2 columns

From Table 27, we can observe that the value of VIF is high for many variables. Therefore, the variables with VIF more than 5 (very high correlation) are dropped and the model is built. Table 28 and Table 29 show the variables with VIF values less than 5.

Table 28: Variables with VIF values less than 5 - 1

	variables	VIF
65	Networth_Next_Year	1.267104
35	ROG_Gross_Block_perc	1.545711
48	ROG_Market_Capitalisation_perc	1.707102
62	Inventory_Velocity_Days	1.927059
38	ROG_Cost_of_Production_perc	2.030572
0	Co_Code	2.117919
61	Creditors_Velocity_Days	2.318367
51	Inventory_Ratio_Latest	2.343611
49	Current_Ratio_Latest	2.468457
60	Debtors_Velocity_Days	2.507265
52	Debtors_Ratio_Latest	2.544727
54	Interest_Cover_Ratio_Latest	2.550858
31	Cash_Flow_From_Investing_Activities	2.606963
32	Cash_Flow_From_Financing_Activities	2.997172
33	ROG_Net_Worth_perc	3.012510
23	Revenue_earnings_in_forex	3.102019

Table 29: Variables with VIF values less than 5 - 2

39	ROG_Total_Assets_perc	3.271459
24	Revenue_expenses_in_forex	3.727331
34	ROG_Capital_Employed_perc	3.846218
30	Cash_Flow_From_Operating_Activities	4.064302
12	Other_Income	4.591733
28	Market_Capitalisation	4.600072
1	Equity_Paid_Up	4.678059
15	Selling_Cost	4.915284

Table 30: Variables used to build the model

```
65                      Networth_Next_Year
35                    ROG_Gross_Block_perc
48          ROG_Market_Capitalisation_perc
62                  Inventory_Velocity_Days
38             ROG_Cost_of_Production_perc
0                                  Co_Code
61                Creditors_Velocity_Days
51                 Inventory_Ratio_Latest
49                   Current_Ratio_Latest
60                  Debtors_Velocity_Days
52                   Debtors_Ratio_Latest
54             Interest_Cover_Ratio_Latest
31    Cash_Flow_From_Investing_Activities
32    Cash_Flow_From_Financing_Activities
33                     ROG_Net_Worth_perc
23               Revenue_earnings_in_forex
39                   ROG_Total_Assets_perc
24               Revenue_expenses_in_forex
34               ROG_Capital_Employed_perc
30    Cash_Flow_From_Operating_Activities
12                             Other_Income
28                  Market_Capitalisation
1                             Equity_Paid_Up
15                              Selling_Cost
Name: variables, dtype: object
```

Table 30 shows the variables with VIF values less than 5 that were used to build the model. From these variables, 'Networth_Next_Year' and 'Co_Code' is not used to build the model as

'Networth_Next_Year' is the variable from which the dependent variable 'Default' is derived and 'Co_Code' is a unique code given to each company which doesn't add value to the model. A total of 22 variables were retained after doing VIF.

Model Building

Model 1

Logit Regression Results

Dep. Variable:	Default	**No. Observations:**	2402
Model:	Logit	**Df Residuals:**	2378
Method:	MLE	**Df Model:**	23
Date:	Mon, 20 Jun 2022	**Pseudo R-squ.:**	0.4137
Time:	19:06:59	**Log-Likelihood:**	-482.82
converged:	True	**LL-Null:**	-823.47
Covariance Type:	nonrobust	**LLR p-value:**	3.853e-129

Figure 36: Model 1 results - 1

	coef	std err	z	P>|z|	[0.025	0.975]
Intercept	-0.3360	0.196	-1.717	0.086	-0.720	0.048
Co_Code	-3.175e-05	7.07e-06	-4.491	0.000	-4.56e-05	-1.79e-05
Equity_Paid_Up	0.0210	0.008	2.768	0.006	0.006	0.036
Other_Income	0.0300	0.037	0.807	0.420	-0.043	0.103
Selling_Cost	-0.0303	0.044	-0.685	0.493	-0.117	0.056
Revenue_earnings_in_forex	-0.0301	0.019	-1.562	0.118	-0.068	0.008
Revenue_expenses_in_forex	0.0327	0.020	1.626	0.104	-0.007	0.072
Market_Capitalisation	-0.0096	0.002	-5.196	0.000	-0.013	-0.006
Cash_Flow_From_Operating_Activities	-0.0071	0.013	-0.540	0.589	-0.033	0.019
Cash_Flow_From_Investing_Activities	0.0150	0.025	0.595	0.552	-0.034	0.064
Cash_Flow_From_Financing_Activities	0.0086	0.023	0.366	0.714	-0.037	0.054
ROG_Net_Worth_perc	-0.0489	0.008	-5.768	0.000	-0.065	-0.032
ROG_Capital_Employed_perc	-0.0016	0.008	-0.196	0.844	-0.017	0.014
ROG_Gross_Block_perc	-0.0276	0.015	-1.893	0.058	-0.056	0.001
ROG_Cost_of_Production_perc	-0.0089	0.003	-3.354	0.001	-0.014	-0.004
ROG_Total_Assets_perc	-0.0097	0.008	-1.212	0.225	-0.025	0.006
ROG_Market_Capitalisation_perc	-0.0006	0.002	-0.301	0.763	-0.005	0.003
Current_Ratio_Latest	-0.7468	0.085	-8.826	0.000	-0.913	-0.581
Inventory_Ratio_Latest	-0.0120	0.014	-0.865	0.387	-0.039	0.015
Debtors_Ratio_Latest	-0.0180	0.015	-1.179	0.238	-0.048	0.012
Interest_Cover_Ratio_Latest	-0.1543	0.032	-4.858	0.000	-0.217	-0.092
Debtors_Velocity_Days	-0.0036	0.001	-3.397	0.001	-0.006	-0.002
Creditors_Velocity_Days	0.0040	0.001	3.438	0.001	0.002	0.006
Inventory_Velocity_Days	-0.0015	0.001	-1.266	0.205	-0.004	0.001

Figure 37: Model 1 results – 2

```
The adjusted pseudo R-square value is 0.3857435369722072
```

Figure 38: Adjusted pseudo R-square value - model 1

Figure 36 and Figure 37 show the results of the model built using the selected 22 variables. From these figures, we can see that few variables are insignificant as they have a p-value greater than 0.05. There variables may not be useful to discriminate cases of default.

Figure 38 shows that the adjusted pseudo R-square seems to be lower than Pseudo R-square value (0.4137) which also proves that there are insignificant variables present in the model.

The variables whose p-value greater than 0.05 is removed and the model is rebuilt.

Model 2

The variables 'Equity_Paid_Up', 'Market_Capitalisation', 'ROG_Net_Worth_perc', 'ROG_Gross_Block_perc', 'ROG_Cost_of_Production_perc', 'Current_Ratio_Latest', 'Interest_Cover_Ratio_Latest', 'Debtors_Velocity_Days' and 'Creditors_Velocity_Days' were selected as they have p-value less than 0.05.

The variables 'Other_Income', 'Selling_Cost', 'Revenue_expenses_in_forex' 'Revenue_earnings_in_forex', 'Cash_Flow_From_Operating_Activities', 'Cash_Flow_From_Investing_Activities', 'Cash_Flow_From_Financing_Activities', 'ROG_Capital_Employed_perc', 'ROG_Total_Assets_perc', 'Interest_Cover_Ratio_Latest ', 'ROG_Market_Capitalisation_perc', 'Debtors_Ratio_Latest' and 'Inventory_Velocity_Days' were dropped as they have a p-value greater than 0.05.

Logit Regression Results

Dep. Variable:	Default	**No. Observations:**	2402
Model:	Logit	**Df Residuals:**	2392
Method:	MLE	**Df Model:**	9
Date:	Mon, 20 Jun 2022	**Pseudo R-squ.:**	0.3868
Time:	19:07:06	**Log-Likelihood:**	-504.98
converged:	True	**LL-Null:**	-823.47
Covariance Type:	nonrobust	**LLR p-value:**	2.413e-131

	coef	std err	z	P>|z|	[0.025	0.975]
Intercept	-0.8732	0.166	-5.252	0.000	-1.199	-0.547
Equity_Paid_Up	0.0172	0.007	2.495	0.013	0.004	0.031
Market_Capitalisation	-0.0101	0.002	-6.205	0.000	-0.013	-0.007
ROG_Net_Worth_perc	-0.0559	0.007	-7.990	0.000	-0.070	-0.042
ROG_Gross_Block_perc	-0.0441	0.013	-3.290	0.001	-0.070	-0.018
ROG_Cost_of_Production_perc	-0.0097	0.003	-3.797	0.000	-0.015	-0.005
Current_Ratio_Latest	-0.7665	0.088	-8.700	0.000	-0.939	-0.594
Interest_Cover_Ratio_Latest	-0.1703	0.030	-5.604	0.000	-0.230	-0.111
Debtors_Velocity_Days	-0.0040	0.001	-3.960	0.000	-0.006	-0.002
Creditors_Velocity_Days	0.0042	0.001	3.759	0.000	0.002	0.006

Figure 39: Model 2 results

```
The adjusted pseudo R-square value is 0.3758339515632504
```

Figure 40: Adjusted pseudo R-square value - model 2

From Figure 39, we can see that all variables are significant and may be useful to discriminate cases of default.

Figure 40 shows that the adjusted pseudo R-square is now close to Pseudo R-square value (0.3868) thus suggesting lesser insignificant variables in the model.

The multicollinearity of the model is checked using Variance Inflation Factor (VIF) for the predictor variables.

Table 31: VIF values for the predictor variables

	variables	VIF
4	ROG_Cost_of_Production_perc	1.135404
3	ROG_Gross_Block_perc	1.292666
2	ROG_Net_Worth_perc	1.370568
6	Interest_Cover_Ratio_Latest	1.553236
5	Current_Ratio_Latest	1.612578
8	Creditors_Velocity_Days	2.007755
7	Debtors_Velocity_Days	2.082292
1	Market_Capitalisation	2.440010
0	Equity_Paid_Up	2.683910

Table 31 shows that multicollinearity still exists. However, these variables are not dropped as the VIF values are not very high. From Figure 39, we can notice that current model (model 2) has no insignificant variables and therefore can be used for prediction purposes.

Therefore, model 2 is used for prediction. The prediction of the model is tested on train and test dataset.

The performance of Predictions on Train and Test sets is checked using model performance metrics such as Accuracy, Confusion Matrix and classification report. These measures are used to validate the predictions.

Model Performance Metrics:

Confusion matrix:

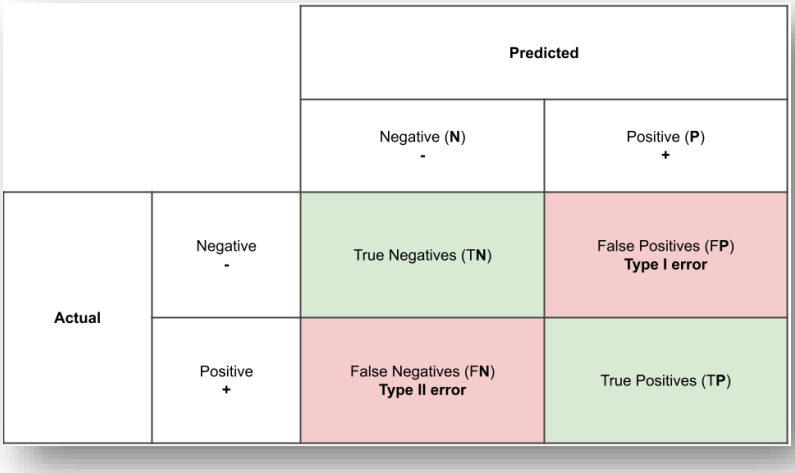

Figure 41: Confusion Matrix

A confusion matrix is a 2x2 tabular structure reflecting the performance of the model in four blocks. Figure 41 shows how a confusion matrix will look like. True Positive (TP) and True Negative (TN) are correct predictions. False Positive (FP) and False Negative (FN) are incorrect predictions.

Table 32: Confusion Matrix formulas

Metric Name	Formula from Confusion Matrix
Accuracy	$\frac{TP + TN}{TP + TN + FP + FN}$
Precision	$\frac{TP}{TP+FP}$
Recall, Sensitivity, TPR	$\frac{TP}{TP+FN}$
Specificity, 1-FPR	$\frac{TN}{TN+FP}$
F1	$\frac{2 * precision * recall}{precision + recall}$

- **Accuracy** – it is a measure of how accurately or cleanly the model classifies the data points. Lesser the false predictions, more the accuracy. In a classification problem, the best score is **100%** accuracy.
- **Precision** – it is a measure of how many among the points identified as positive by the model, are really positive. A precision score of 1.0 means that all the points are identified as positive by the model. **1.0** is a perfect precision score. However, it does not indicate about the number of observations that were not labelled correctly.

- **Recall or sensitivity** – is the ratio of correctly predicted positive observations to the all observations in actual class. A perfect recall score is **1.0** but a recall score above 0.5 is considered as a good recall score. However, the recall score does not indicate about how many observations are incorrectly predicted.
- **Specificity** – is a measure of how many of the actual negative data points are identified as negative by the model.
- **F1** – it is the measure of the model's accuracy on a dataset. The F-score is a way of combining the precision and recall of the model and it is defined as the harmonic mean of the model's precision and recall. An F1 score is considered perfect when it's **1**, while the model is a total failure when the score is 0.

All these can be calculated from the confusion matrix by using the formulas given in Table 32.

Classification report:

A Classification report is used to measure the quality of predictions from a classification algorithm. The classification report shows the main classification metrics and their scores. The metrics are precision, recall, f1-score and accuracy for the actual and predicted data. The metrics are calculated by using true and false positives, true and false negatives from the confusion matrix.

ROC Curve:

Receiver Operating Characteristics (ROC) Curve is a technique for visualizing classifier performance. It is a graph between true positive (TP) rate and false positive (FP) rate.

$$\text{TP rate} = \frac{TP}{total\ positive}$$

$$\text{FP rate} = \frac{FP}{total\ negative}$$

ROC graph is a trade-off between benefits (TP) and costs (FP). The steeper the ROC Curve, the stronger the model will be and vice versa.

ROC_AUC score:

Area under the ROC Curve (AUC) is the measure of the area under the ROC Curve. The ROC_AUC score gives us the value of the area under the ROC Curve. The larger the area under the curve, the better the model.

The distribution plot of the logit function values is first checked and shown in Figure 42.

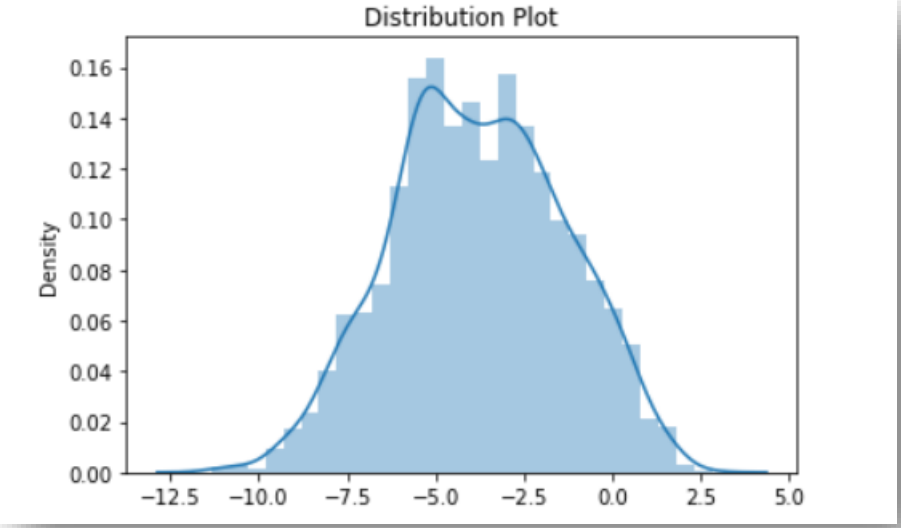

Figure 42: Distribution plot of the logit function

Predictions on the train set

The predicted probability values for the train set are shown in Table 33.

Table 33: Predicted probability values – train set

842	0.005488
1057	0.000770
1595	0.000303
100	0.566242
1191	0.032449
...	...
1815	0.005662
2852	0.177780
1505	0.001374
375	0.437327
3428	0.001094

Length: 2402, dtype: float64

Choosing optimum cut-off

The boxplot of the target variable and the train data is shown in Figure 43.

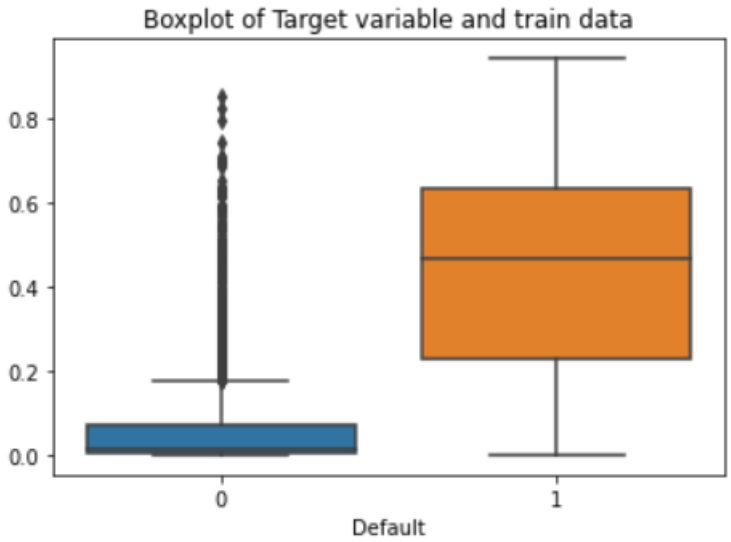

Figure 43: Boxplot of Target variable and Train data

From Figure 43, a value of cut-off needs to be decided. The cut off must be one such value which will give us the most reasonable descriptive power of the model. Let us take a cut-off of 0.07 and check.

Cut-off value 0.07

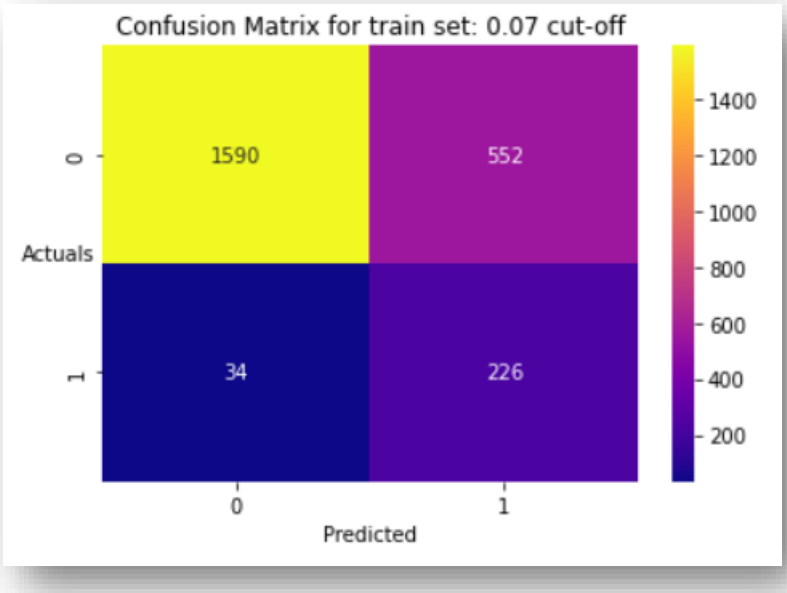

Figure 44: Confusion Matrix for Training Data – 0.07 Cut-off

```
True Negative: 1590
False Positives: 552
False Negatives: 34
True Positives: 226
```

Figure 45: Confusion Matrix results for Training Data – 0.07 Cut-off

The confusion matrix was calculated and shown in Figure 44. The results are also shown in Figure 45. The accuracy score for the training data was found to be 0.756 which is shown in Figure 46.

	precision	recall	f1-score	support
0	0.979	0.742	0.844	2142
1	0.290	0.869	0.435	260
accuracy			0.756	2402
macro avg	0.635	0.806	0.640	2402
weighted avg	0.905	0.756	0.800	2402

Figure 46: Classification Report for Training Data – 0.07 Cut-off

The classification report for the logistic regression model with the scores for the different model performance measures is calculated and shown in Figure 46. As observed, we can see that the accuracy of the model i.e., %overall correct predictions is 75.6%. Sensitivity of the model is 86.9% i.e., 86.9% of those defaulted were correctly identified as defaulters by the model.

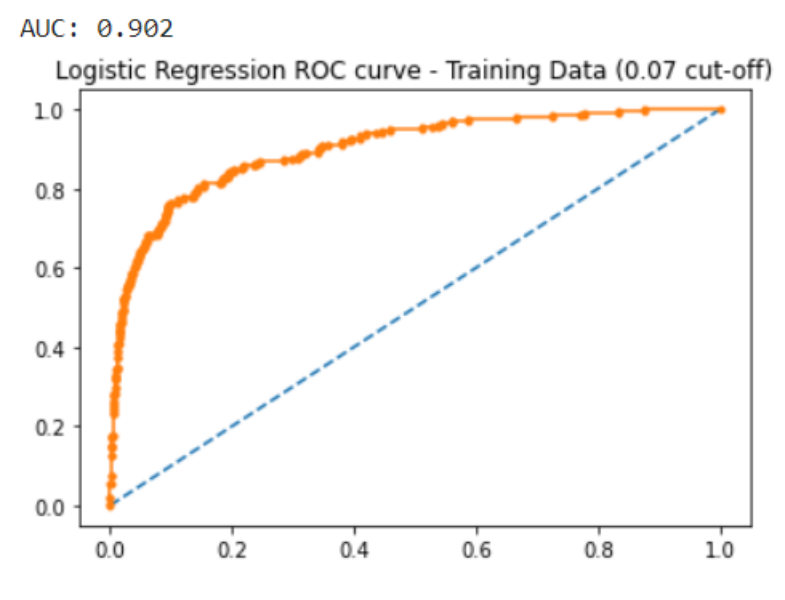

Figure 47: Logistic Regression ROC Curve for training data (0.07 cut-off)

The ROC_AUC score for the training data was calculated to be 0.902. The ROC curve was plotted and is shown in Figure 47.

Cut-off value 0.08

A cut-off of 0.08 is now taken and the results are checked to see if the predictions have improved.

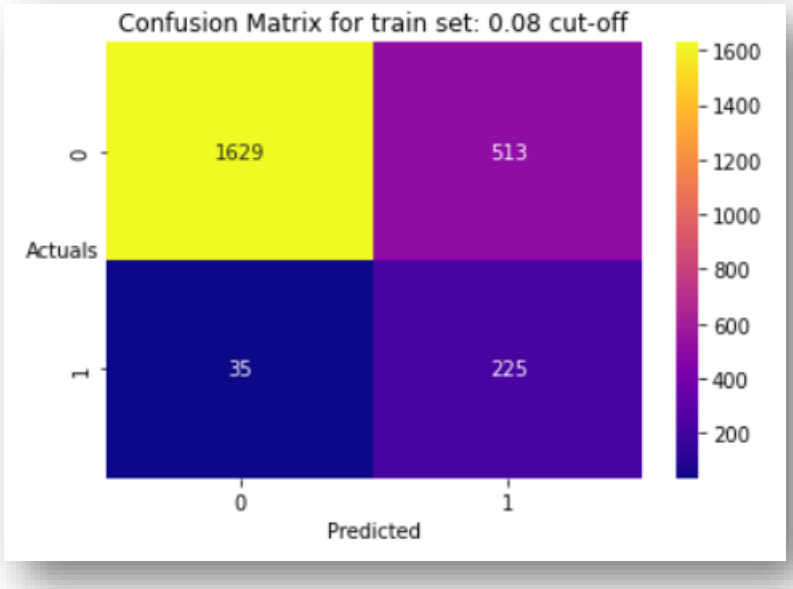

Figure 48: Confusion Matrix for Training Data – 0.08 Cut-off

```
True Negative: 1629
False Positives: 513
False Negatives: 35
True Positives: 225
```

Figure 49: Confusion Matrix results for Training Data – 0.08 Cut-off

The confusion matrix was calculated and shown in Figure 48. The results are also shown in Figure 49. The accuracy score for the training data was found to be 0.772 which is shown in Figure 50.

	precision	recall	f1-score	support
0	0.979	0.761	0.856	2142
1	0.305	0.865	0.451	260
accuracy			0.772	2402
macro avg	0.642	0.813	0.653	2402
weighted avg	0.906	0.772	0.812	2402

Figure 50: Classification Report for Training Data – 0.08 Cut-off

The classification report for the logistic regression model with the scores for the different model performance measures is calculated and shown in Figure 50. As observed, we can see that the accuracy of the model i.e., %overall correct predictions has increased from 75.6% to 77.2%. However, the sensitivity of the model has dropped slightly from 86.9% to 86.5%.

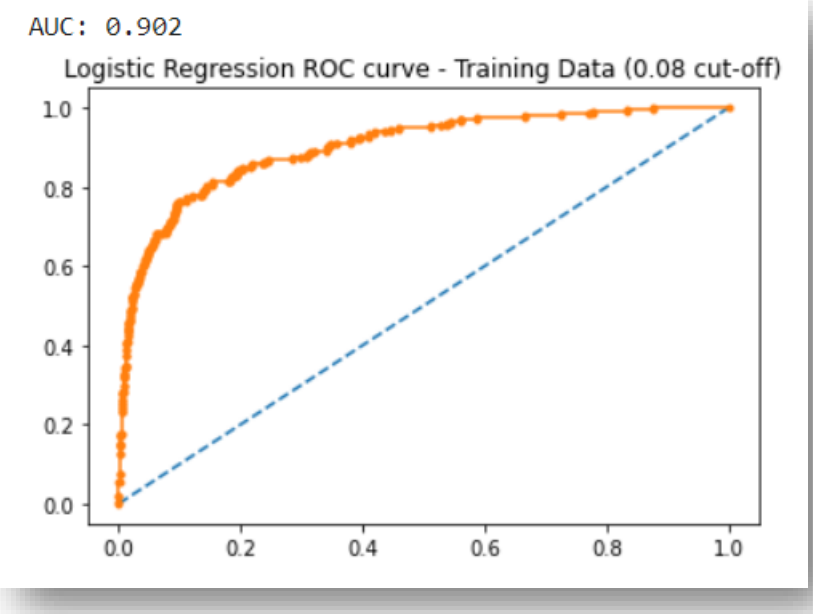

Figure 51: Logistic Regression ROC Curve for training data (0.08 cut-off)

The ROC_AUC score for the training data was calculated to be 0.902. The ROC curve was plotted and is shown in Figure 51.

The optimum cut-off can be chosen as 0.08 as it gave higher model sensitivity and overall accuracy of the model. However, the predictions on the test dataset should also be checked.

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

Predictions on the test set

The predicted probability values for the test set are shown in Table 34.

Table 34: Predicted probability values - test set

251	0.307286
3493	0.009355
3063	0.013006
2384	0.050853
1679	0.000512
	...
1321	0.042306
2666	0.003960
773	0.002560
3488	0.001319
2956	0.085441

Length: 1184, dtype: float64

Cut-off value 0.07

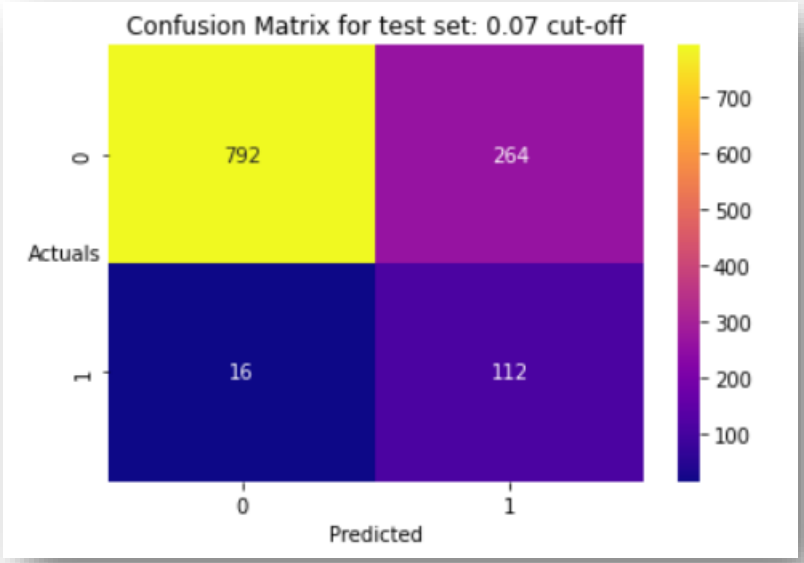

Figure 52: Confusion Matrix for Test Data – 0.07 Cut-off

```
True Negative: 792
False Positives: 264
False Negatives: 16
True Positives: 112
```

Figure 53: Confusion Matrix results for Test Data – 0.07 Cut-off

The confusion matrix was calculated and shown in Figure 52. The results are also shown in Figure 53. The accuracy score for the training data was found to be 0.764 which is shown in Figure 54.

	precision	recall	f1-score	support
0	0.980	0.750	0.850	1056
1	0.298	0.875	0.444	128
accuracy			0.764	1184
macro avg	0.639	0.812	0.647	1184
weighted avg	0.906	0.764	0.806	1184

Figure 54: Classification Report for Test Data – 0.07 Cut-off

The classification report for the logistic regression model with the scores for the different model performance measures is calculated and shown in Figure 54. As observed, we can see that the

accuracy of the model i.e., %overall correct predictions is 76.4%. Sensitivity of the model is 87.5% i.e., 87.5% of those defaulted were correctly identified as defaulters by the model.

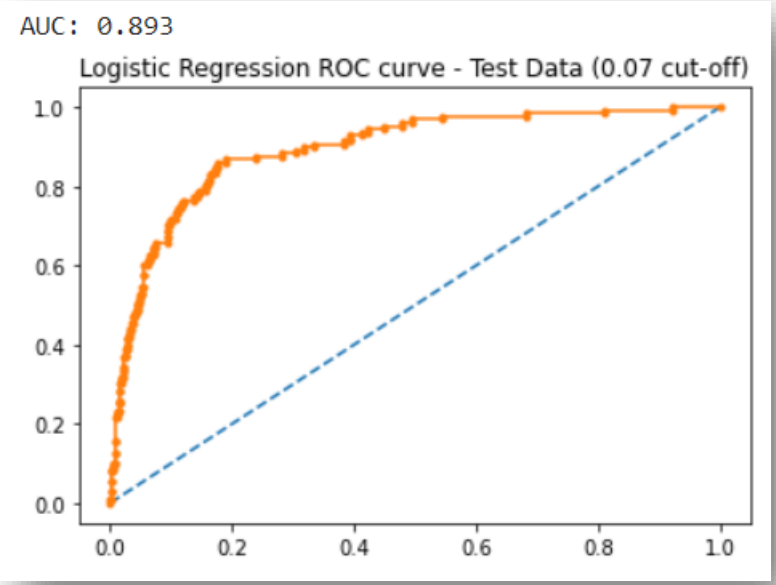

Figure 55: Logistic Regression ROC Curve for test data (0.07 cut-off)

The ROC_AUC score for the test data was calculated to be 0.893. The ROC curve was plotted and is shown in Figure 55.

Cut-off value 0.08

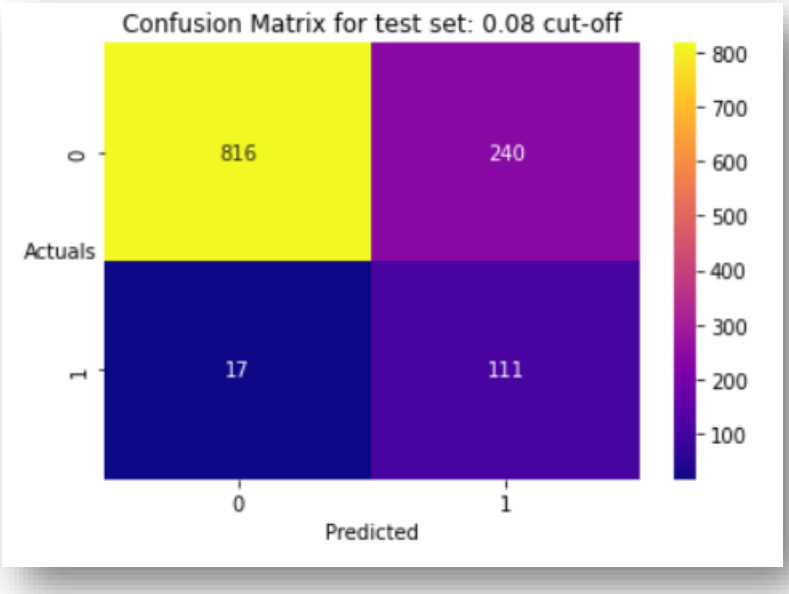

Figure 56: Confusion Matrix for Test Data – 0.08 Cut-off

```
True Negative: 816
False Positives: 240
False Negatives: 17
True Positives: 111
```

Figure 57: Confusion Matrix results for Test Data – 0.08 Cut-off

The confusion matrix was calculated and shown in Figure 56. The results are also shown in Figure 57. The accuracy score for the training data was found to be 0.783 which is shown in Figure 58.

	precision	recall	f1-score	support
0	0.980	0.773	0.864	1056
1	0.316	0.867	0.463	128
accuracy			0.783	1184
macro avg	0.648	0.820	0.664	1184
weighted avg	0.908	0.783	0.821	1184

Figure 58: Classification Report for Test Data – 0.08 Cut-off

The classification report for the logistic regression model with the scores for the different model performance measures is calculated and shown in Figure 58. As observed, we can see that the accuracy of the model i.e., %overall correct predictions has increased from 76.4% to 78.3%. However, the sensitivity of the model has dropped slightly from 87.5% to 86.7%.

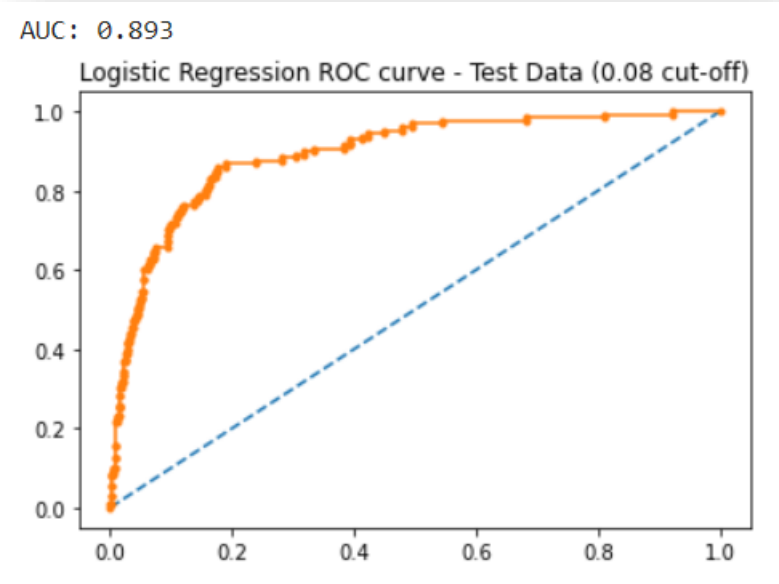

Figure 59: Logistic Regression ROC Curve for test data (0.08 cut-off)

The ROC_AUC score for the test data was calculated to be 0.893. The ROC curve was plotted and is shown in Figure 59.

Optimum cut-off as 0.08 as it gave higher model sensitivity and overall accuracy of the model in test dataset.

Interpretation from the model

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.

There are 9 most important variables that determine whether a business or the company will default or not. The 9 variables are 'Equity_Paid_Up', 'Market_Capitalisation', 'ROG_Net_Worth_perc', 'ROG_Gross_Block_perc', 'ROG_Cost_of_Production_perc', 'Current_Ratio_Latest', 'Interest_Cover_Ratio_Latest', 'Debtors_Velocity_Days' and 'Creditors_Velocity_Days'.

From a business point of view:

The companies must keep an eye on these 9 important factors to make sure their business is on the right track. If the companies concentrate on these factors and find ways to increase their business, the company will be able to keep up their debt and hence will not fall prey to default.

From an investor's point of view:

When an investor is planning to invest in the stocks of a company, the investor must study the net worth of the company, if it is capable of handling its financial obligations, if it can grow quickly, and is able to manage the growth scale. The investor can use these 9 important factors to determine if the company is likely to default or not. This is an indicator for the investor to decide whether to invest in a business or not.

From a bank's point of view:

The bank can decide whether to grant loan to a company based on the performance of the company, its net worth, probability of default etc., the bank can use these 9 variables as a way of predicting whether the company will default and hence to grant the loan or not.

Logistic model interpretation

The logistic model has been built using the statsmodel and logit function. The cut-off was chosen to be 0.08. The

The classification report for the logistic regression model on the test data is shown in Figure 58. As observed, we can see that the accuracy of the model i.e., %overall correct predictions is 78.3% and the sensitivity of the model is 86.7%.

The AUC score is 0.893 on the test data meaning it classifies defaulters and non-defaulters correctly 89.3% of the time.

The scores look good and this model can be implemented to determine whether a company will default or not.

Model Performance Metrics:

Confusion matrix:

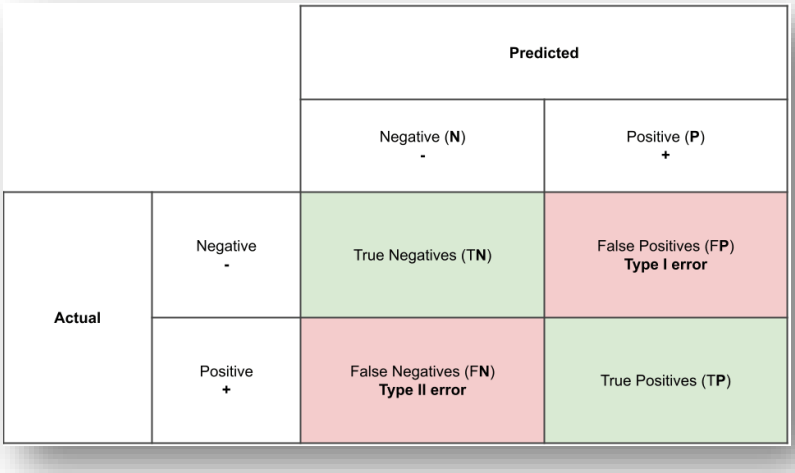

Figure 60: Confusion Matrix

A confusion matrix is a 2x2 tabular structure reflecting the performance of the model in four blocks. Figure 41 shows how a confusion matrix will look like. True Positive (TP) and True Negative (TN) are correct predictions. False Positive (FP) and False Negative (FN) are incorrect predictions.

Table 35: Confusion Matrix formulas

Metric Name	Formula from Confusion Matrix
Accuracy	$\frac{TP + TN}{TP + TN + FP + FN}$
Precision	$\frac{TP}{TP+FP}$
Recall, Sensitivity, TPR	$\frac{TP}{TP+FN}$
Specificity, 1-FPR	$\frac{TN}{TN+FP}$
F1	$\frac{2 * precision * recall}{precision + recall}$

- **Accuracy** – it is a measure of how accurately or cleanly the model classifies the data points. Lesser the false predictions, more the accuracy. In a classification problem, the best score is **100%** accuracy.
- **Precision** – it is a measure of how many among the points identified as positive by the model, are really positive. A precision score of 1.0 means that all the points are identified as positive by the model. **1.0** is a perfect precision score. However, it does not indicate about the number of observations that were not labelled correctly.

- **Recall or sensitivity** – is the ratio of correctly predicted positive observations to the all observations in actual class. A perfect recall score is **1.0** but a recall score above 0.5 is considered as a good recall score. However, the recall score does not indicate about how many observations are incorrectly predicted.
- **Specificity** – is a measure of how many of the actual negative data points are identified as negative by the model.
- **F1** – it is the measure of the model's accuracy on a dataset. The F-score is a way of combining the precision and recall of the model and it is defined as the harmonic mean of the model's precision and recall. An F1 score is considered perfect when it's **1**, while the model is a total failure when the score is 0.

All these can be calculated from the confusion matrix by using the formulas given in Table 32.

Classification report:

A Classification report is used to measure the quality of predictions from a classification algorithm. The classification report shows the main classification metrics and their scores. The metrics are precision, recall, f1-score and accuracy for the actual and predicted data. The metrics are calculated by using true and false positives, true and false negatives from the confusion matrix.

ROC Curve:

Receiver Operating Characteristics (ROC) Curve is a technique for visualizing classifier performance. It is a graph between true positive (TP) rate and false positive (FP) rate.

$$\text{TP rate} = \frac{TP}{total\ positive}$$

$$\text{FP rate} = \frac{FP}{total\ negative}$$

ROC graph is a trade-off between benefits (TP) and costs (FP). The steeper the ROC Curve, the stronger the model will be and vice versa.

ROC_AUC score:

Area under the ROC Curve (AUC) is the measure of the area under the ROC Curve. The ROC_AUC score gives us the value of the area under the ROC Curve. The larger the area under the curve, the better the model.

1.8 Build a Random Forest Model on Train Dataset. Also showcase your model building approach

Random Forest Classifier – Model building

Random Forest technique is an ensemble technique wherein we construct multiple models and take the average output of all the models to take a final decision or make a prediction. Many combinations of parameters were tried using Grid Search Cross Validation or the Grid Search CV function. Multiple values were passed for the different parameters. Figure 61 shows the best parameters for random forest classifier model.

```
{'max_depth': 10, 'max_features': 7, 'min_samples_leaf': 3, 'min_samples_split': 15, 'n_estimators': 280}
RandomForestClassifier(max_depth=10, max_features=7, min_samples_leaf=3,
                       min_samples_split=15, n_estimators=280, random_state=42)
```

Figure 61: Best parameters for Random Forest Classifier Model

- The **'max_depth'** parameter is used to prune the trees. It is the number of decision trees or the level of the trees. The optimum 'max_depth' parameter is found to be 10 meaning the depth level of the decision tree is **10**.
- The **'max_features'** parameter determines how many numbers of the independent variables or features a random forest classifier uses for evaluating and splitting the decision nodes. The 'max_features' was found to be **7** in this dataset.
- The **'min_samples_leaf'** is a parameter that determines how many observations need to be present in each of the terminal nodes or the leaf nodes in all the decision trees in the random forest. In this data, the 'min_samples_leaf' was determined to be **3**.
- The parameter **'min_samples_split'** determines how many observations a node should have for it to be split into left and right child nodes. **15** was found to be the optimum 'min_samples_split' value.
- The **'n_estimators'** is the number of trees you want to build in a random forest and the optimum value was found to be **280**.

Table 36: Feature Importances of Random Forest Classifier

	Imp
Networth_Next_Year	0.297995
Networth	0.164431
Book_Value_Unit_Curr	0.144347
Book_Value_Adj_Unit_Curr	0.138015
Capital_Employed	0.024893
...	...
Inventory_Ratio_Latest	0.000573
Revenue_earnings_in_forex	0.000234
ROG_Revenue_expenses_in_forex_perc	0.000000
ROG_Revenue_earnings_in_forex_perc	0.000000
Capital_expenses_in_forex	0.000000

[66 rows x 1 columns]

Feature importance is a measure of the features used in the split and the contribution of the different features in the split. A value of 0 means that that particular independent variable was never used in splitting the nodes. These values are a relative measure of feature importance and not absolute. From Table 36, we can see that 'Networth_Next_Year' was the most important feature in splitting the nodes whereas 'ROG_Revenue_expenses_in_forex_perc' and 'Capital_expenses_in_forex' were the least important feature used in split.

Random Forest Classifier – Performance metrics train data

Training Data

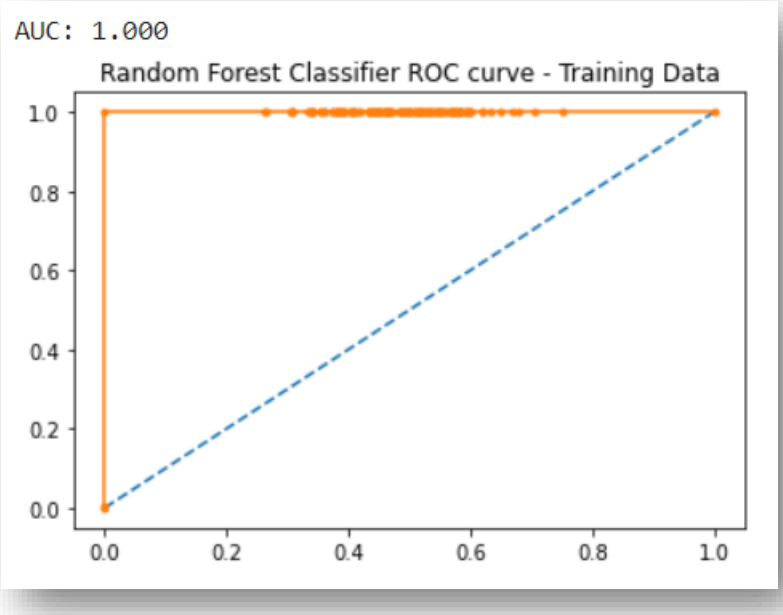

Figure 62: Random Forest Classifier ROC Curve for Training Data

The ROC_AUC score for the training data was calculated to be 1.000. The ROC curve was plotted and is shown in Figure 62.

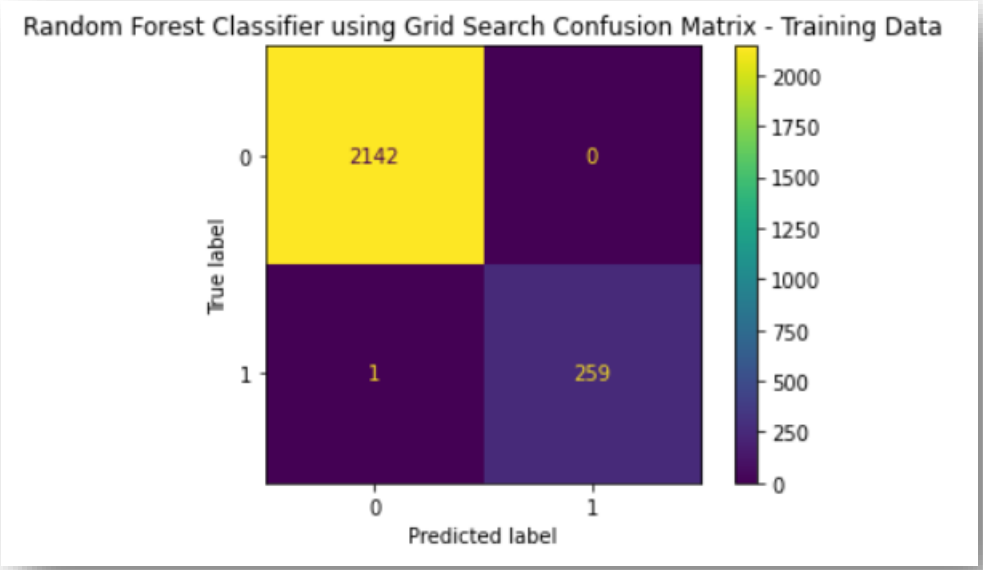

Figure 63: Confusion Matrix for Random Forest Classifier Model for Training Data

The confusion matrix was calculated and shown in Figure 63. The accuracy score for the training data was found to be 1.00 which is shown in Figure 64.

	precision	recall	f1-score	support
0	1.00	1.00	1.00	2142
1	1.00	1.00	1.00	260
accuracy			1.00	2402
macro avg	1.00	1.00	1.00	2402
weighted avg	1.00	1.00	1.00	2402

Figure 64: Random Forest Classifier Classification Report for Training Data

The classification report for the random forest classifier model with the scores for the different model performance measures is calculated and shown in Figure 64. From this figure we can see that the precision of the model is 1.00 means 100% of the data points identified as positive by the model, are really positive. The f1-score is 1.00 means the model is 100% accurate on this data set. The model has 100% accuracy. The recall score is 1.00 which means 100% of the positive observations are correctly predicted.

1.9 Validate the Random Forest Model on test Dataset and state the performance matrices. Also state interpretation from the model

Random Forest Classifier – Performance metrics test data

Test Data

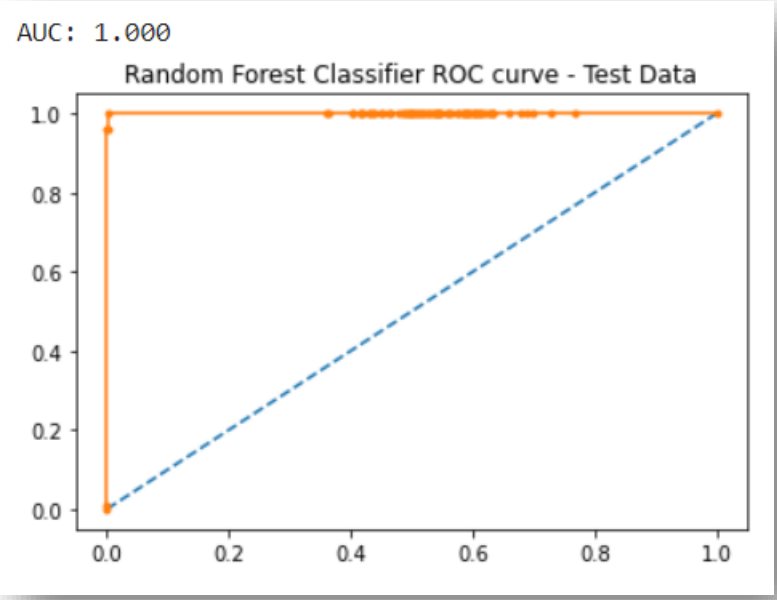

Figure 65: Random Forest Classifier ROC Curve for Test Data

The ROC_AUC score for the test data was calculated to be 1.000. The ROC curve was plotted and is shown in Figure 65.

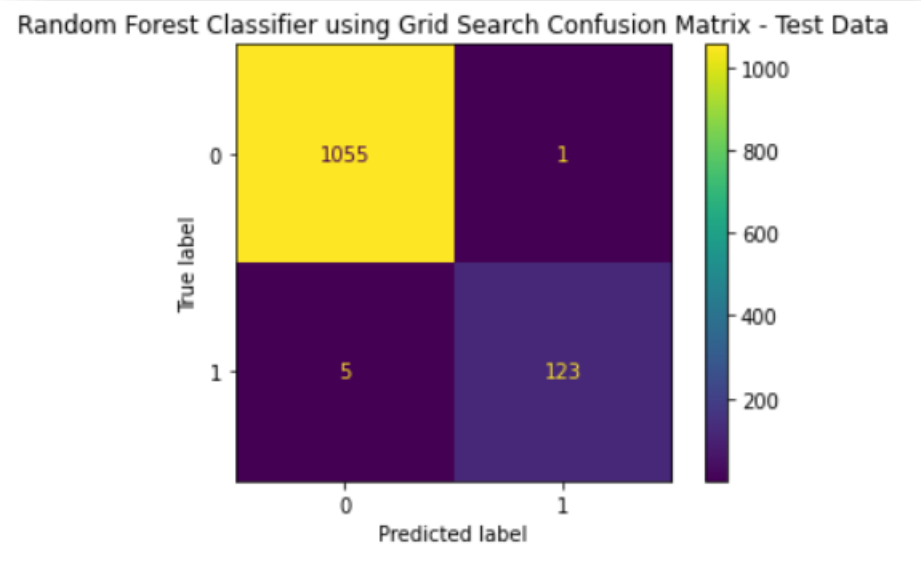

Figure 66: Confusion Matrix for Random Forest Classifier Model for Test Data

The confusion matrix was calculated and shown in Figure 66. The accuracy score for the test data was found to be 0.99 which is shown in Figure 67.

	precision	recall	f1-score	support
0	1.00	1.00	1.00	1056
1	0.99	0.96	0.98	128
accuracy			0.99	1184
macro avg	0.99	0.98	0.99	1184
weighted avg	0.99	0.99	0.99	1184

Figure 67: Random Forest Classifier Classification Report for Test Data

The classification report for the random forest classifier model with the scores for the different model performance measures is calculated and shown in Figure 67. From this figure we can see that the precision of the model is 0.99 means 99% of the data points identified as positive by the model, are really positive. The f1-score is 0.98 means the model is 98% accurate on this data set. The model has 99% accuracy. The recall score is 0.96 which means 96% of the positive observations are correctly predicted.

1.10 Build a LDA Model on Train Dataset. Also showcase your model building approach

Linear Discriminant Analysis – Model building

Linear Discriminant Analysis (LDA) uses linear combinations of independent variables to predict the class in the response variable of a given observation. LDA assumes that the independent variables (p) are normally distributed and there is equal variance / covariance for the classes. LDA is popular, because it can be used for both classification and dimensionality reduction. Values were passed for the different parameters and the best combination of parameters is found using GridSearchCV. The best parameters are shown in Figure 68.

```
{'shrinkage': 'auto', 'solver': 'lsqr', 'tol': 0.01}
LinearDiscriminantAnalysis(shrinkage='auto', solver='lsqr', tol=0.01)
```

Figure 68: Best parameters for Linear Discriminant Analysis Model

- The **'solver'** is the algorithm used in the process of backpropagation to calculate the weights of the coefficients in the logistic regression model. The 'solver' used is **'lsqr'** solver.
- The **'tol'** parameter is the threshold level. The lower the threshold, higher the accuracy and lesser the number of times the model will execute and vice versa. The threshold value used was **'0.01'**.
- The **'shrinkage'** is a form of regularization used to improve the estimation of covariance matrices in situations where the number of training samples is small compared to the number of features. The shrinkage parameter is **'auto'**. This automatically determines the optimal shrinkage parameter in an analytic way.

```
The coefficient for Co_Code is -2.248674546283934e-05
The coefficient for Equity_Paid_Up is 0.0008160010153744485
The coefficient for Networth is -0.02588103975869721
The coefficient for Capital_Employed is 0.002305134876254939
The coefficient for Total_Debt is 0.003200492351840587
The coefficient for Gross_Block is 0.0033361589104549446
The coefficient for Net_Working_Capital is 0.0017207753528725358
The coefficient for Current_Assets is -0.004623836833827238
The coefficient for Current_Liabilities_and_Provisions is 0.013473389646968216
The coefficient for Total_Assets_by_Liabilities is 0.0052343941881512505
The coefficient for Gross_Sales is 0.0002577553988792952
The coefficient for Net_Sales is 0.0011736216485269504
The coefficient for Other_Income is 0.13046061421393956
The coefficient for Value_Of_Output is -7.002387095325635e-05
The coefficient for Cost_of_Production is -0.004243188432630088
The coefficient for Selling_Cost is -0.07021958990297025
The coefficient for PBIDT is -0.0204505386037137
The coefficient for PBDT is -0.005335086412026898
The coefficient for PBIT is -0.0006695917842594705
The coefficient for PBT is 0.043243870850806415
The coefficient for PAT is -0.0040797531879215795
The coefficient for Adjusted_PAT is 0.038385170942151346
The coefficient for CP is -0.020481397524675288
The coefficient for Revenue_earnings_in_forex is -0.023837815837454854
The coefficient for Revenue_expenses_in_forex is 0.023787337155944957
The coefficient for Capital_expenses_in_forex is -4.384207840518739e-12
The coefficient for Book_Value_Unit_Curr is -0.0047787642091243615
The coefficient for Book_Value_Adj_Unit_Curr is -0.023349743313352697
The coefficient for Market_Capitalisation is 0.00024298732520740675
The coefficient for CEPS_annualised_Unit_Curr is 0.06218901185752841
```

Figure 69: LDA - Coefficient values for each column - 1

```
The coefficient for Cash_Flow_From_Operating_Activities is 0.008105566531058154
The coefficient for Cash_Flow_From_Investing_Activities is -0.0041299735811558165
The coefficient for Cash_Flow_From_Financing_Activities is 0.020608188747003625
The coefficient for ROG_Net_Worth_perc is -0.05060535910391244
The coefficient for ROG_Capital_Employed_perc is 0.007072198488272022
The coefficient for ROG_Gross_Block_perc is 0.0007198646525224464
The coefficient for ROG_Gross_Sales_perc is 0.004351792731706799
The coefficient for ROG_Net_Sales_perc is -0.005868468093474659
The coefficient for ROG_Cost_of_Production_perc is -0.008153738996244803
The coefficient for ROG_Total_Assets_perc is -0.020042370079860195
The coefficient for ROG_PBIDT_perc is -0.0020057726132876163
The coefficient for ROG_PBDT_perc is 0.0017174610837633443
The coefficient for ROG_PBIT_perc is 0.00011591376545600808
The coefficient for ROG_PBT_perc is 0.000875359859141542
The coefficient for ROG_PAT_perc is -0.0007741410279769358
The coefficient for ROG_CP_perc is -0.0006745864447440311
The coefficient for ROG_Revenue_earnings_in_forex_perc is -1.6903145549918008e-13
The coefficient for ROG_Revenue_expenses_in_forex_perc is -1.084687895058778e-13
The coefficient for ROG_Market_Capitalisation_perc is -0.0007163500264803068
The coefficient for Current_Ratio_Latest is -0.5849584825847625
The coefficient for Fixed_Assets_Ratio_Latest is 0.00023884195898855876
The coefficient for Inventory_Ratio_Latest is -0.022653907060266612
The coefficient for Debtors_Ratio_Latest is -0.02466597356542327
The coefficient for Total_Asset_Turnover_Ratio_Latest is -0.5619830391517825
The coefficient for Interest_Cover_Ratio_Latest is -0.03855909003008862
The coefficient for PBIDTM_perc_Latest is -0.002868211872907832
The coefficient for PBITM_perc_Latest is -0.05402740052761412
The coefficient for PBDTM_perc_Latest is 0.009056735047548142
The coefficient for CPM_perc_Latest is -0.0481993419813551
The coefficient for APATM_perc_Latest is 0.1175626868952973
```

Figure 70: LDA - Coefficient values for each column - 2

```
The coefficient for Debtors_Velocity_Days is -0.005447132752769146
The coefficient for Creditors_Velocity_Days is 0.0027563440791009905
The coefficient for Inventory_Velocity_Days is -0.0022968122785103056
The coefficient for Value_of_Output_by_Total_Assets is 0.30013469822874317
The coefficient for Value_of_Output_by_Gross_Block is -0.016107065438381606
The coefficient for Networth_Next_Year is -6.430444620175724e-06
```

Figure 71: LDA - Coefficient values for each column - 3

There are 66 attributes in this dataset and hence there are 66 coefficients. For every one-unit change in x (the different independent variables), y (the dependent variable) changes m (the coefficient value) times. The coefficient values for each column are shown in Figure 69, Figure 70 and Figure 71.

Linear Discriminant Analysis – Performance metrics train data

Training Data

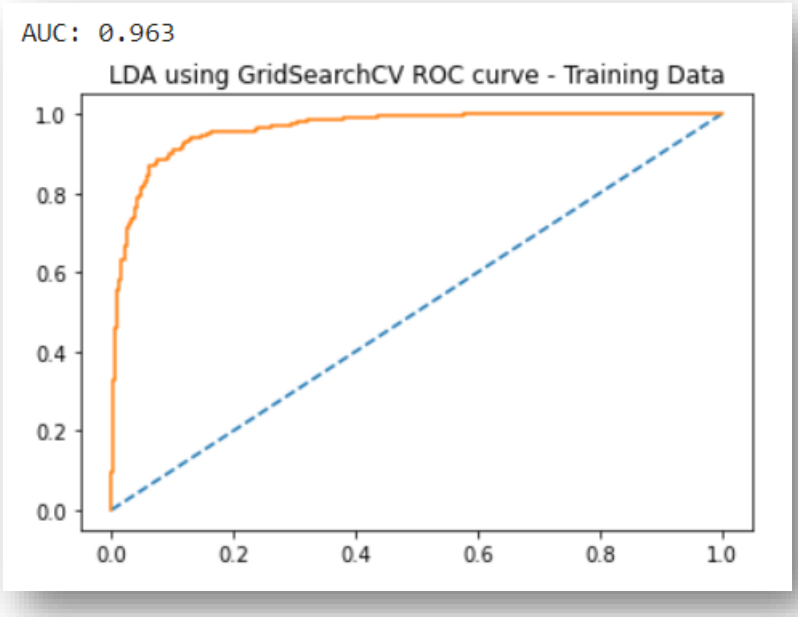

Figure 72: Linear Discriminant Analysis ROC Curve for Training Data

The ROC_AUC score for the training data was calculated to be 0.963. The ROC curve was plotted and is shown in Figure 72.

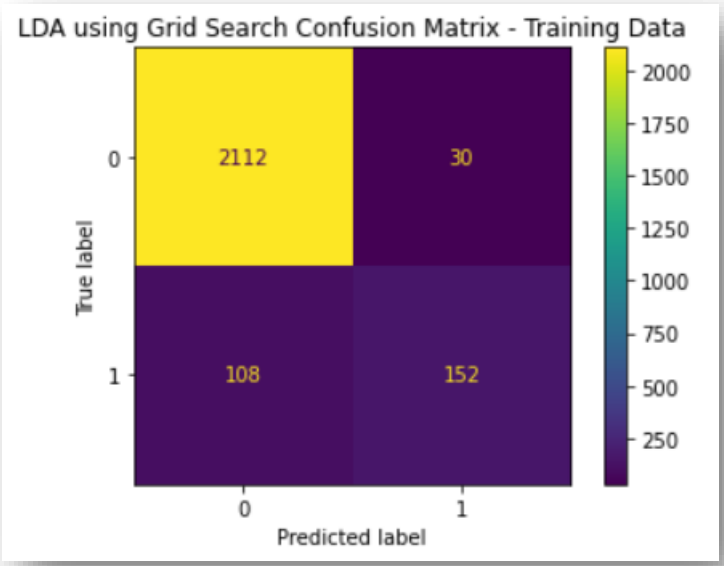

Figure 73: Confusion Matrix for Linear Discriminant Analysis Model for Training Data

The confusion matrix was calculated and shown in Figure 73. The accuracy score for the training data was found to be 0.94 which is shown in Figure 74.

	precision	recall	f1-score	support
0	0.95	0.99	0.97	2142
1	0.84	0.58	0.69	260
accuracy			0.94	2402
macro avg	0.89	0.79	0.83	2402
weighted avg	0.94	0.94	0.94	2402

Figure 74: Linear Discriminant Analysis Classification Report for Training Data

The classification report for the linear discriminant model with the scores for the different model performance measures is calculated and shown in Figure 74. From this figure we can see that the precision of the model is 0.84 means 84% of the data points identified as positive by the model, are really positive. The f1-score is 0.69 means the model is 69% accurate on this data set. The model has 94% accuracy. The recall score is 0.58 which means 58% of the positive observations are correctly predicted.

1.11 Validate the LDA Model on test Dataset and state the performance matrices. Also state interpretation from the model

Linear Discriminant Analysis – Performance metrics test data

Test Data

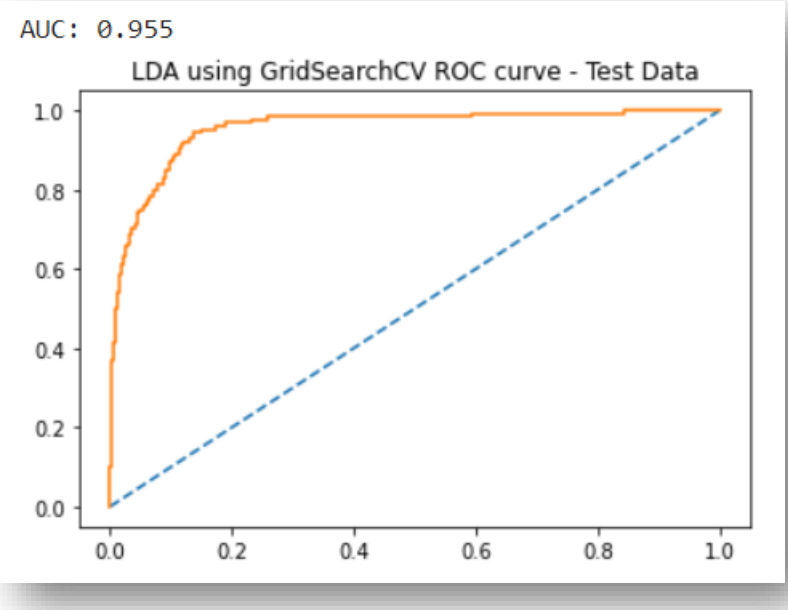

Figure 75: Linear Discriminant Analysis ROC Curve for Test Data

The ROC_AUC score for the test data was calculated to be 0.955. The ROC curve was plotted and is shown in Figure 75.

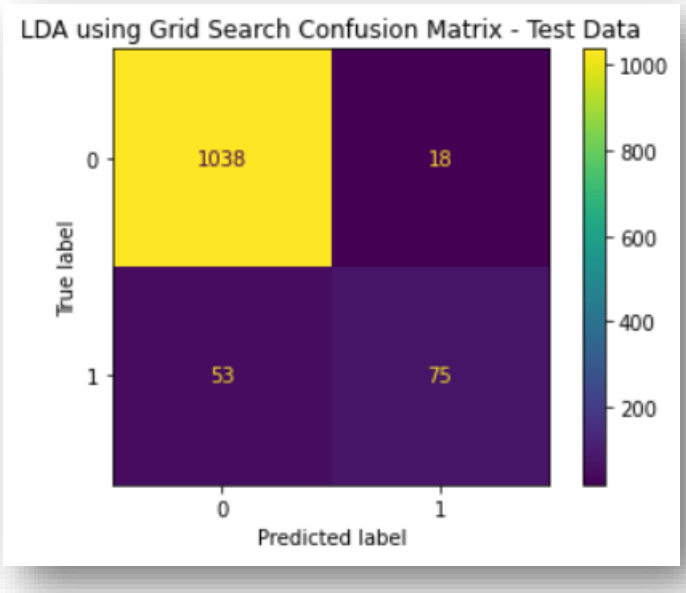

Figure 76: Confusion Matrix for Linear Discriminant Analysis Model for Test Data

The confusion matrix was calculated and shown in Figure 76. The accuracy score for the test data was found to be 0.94 which is shown in Figure 77.

	precision	recall	f1-score	support
0	0.95	0.98	0.97	1056
1	0.81	0.59	0.68	128
accuracy			0.94	1184
macro avg	0.88	0.78	0.82	1184
weighted avg	0.94	0.94	0.94	1184

Figure 77: Linear Discriminant Analysis Classification Report for Test Data

The classification report for the linear discriminant model with the scores for the different model performance measures is calculated and shown in Figure 77. From this figure we can see that the precision of the model is 0.81 means 81% of the data points identified as positive by the model, are really positive. The f1-score is 0.68 means the model is 68% accurate on this data set. The model has 94% accuracy. The recall score is 0.59 which means 59% of the positive observations are correctly predicted.

1.12 Compare the performances of Logistics, Random Forest and LDA models (include ROC Curve)

A comparison of all the three models was done to understand which model is the best suited for our case study. The model performance measures of all the three models were tabulated and it is shown in Table 37.

Table 37: Comparison of all the three models

	Logistic Train	Logistic Test	Random Forest Train	Random Forest Test	LDA Grid Train	LDA Grid Test
Accuracy	0.770	0.780	1.0	0.995	0.943	0.940
AUC	0.902	0.893	1.0	1.000	0.963	0.955
Recall	0.870	0.870	1.0	0.960	0.580	0.590
Precision	0.300	0.320	1.0	0.990	0.840	0.810
F1 Score	0.450	0.460	1.0	0.980	0.690	0.680

From Table 37, we are able to understand that the ROC_AUC score for the random forest model is the highest compared to the other two models. The larger the area under the curve, the better the model. This is also true in case of accuracy where the score is highest in the random forest model compared to the other two models. The precision and f1 score of the random forest test data is the highest. The recall score for the random forest train data is the highest and also for the test data. Since all the other model performance measures are good for the random forest model, it is chosen as the optimized model for our problem.

This is further analysed with the help of the ROC Curve.

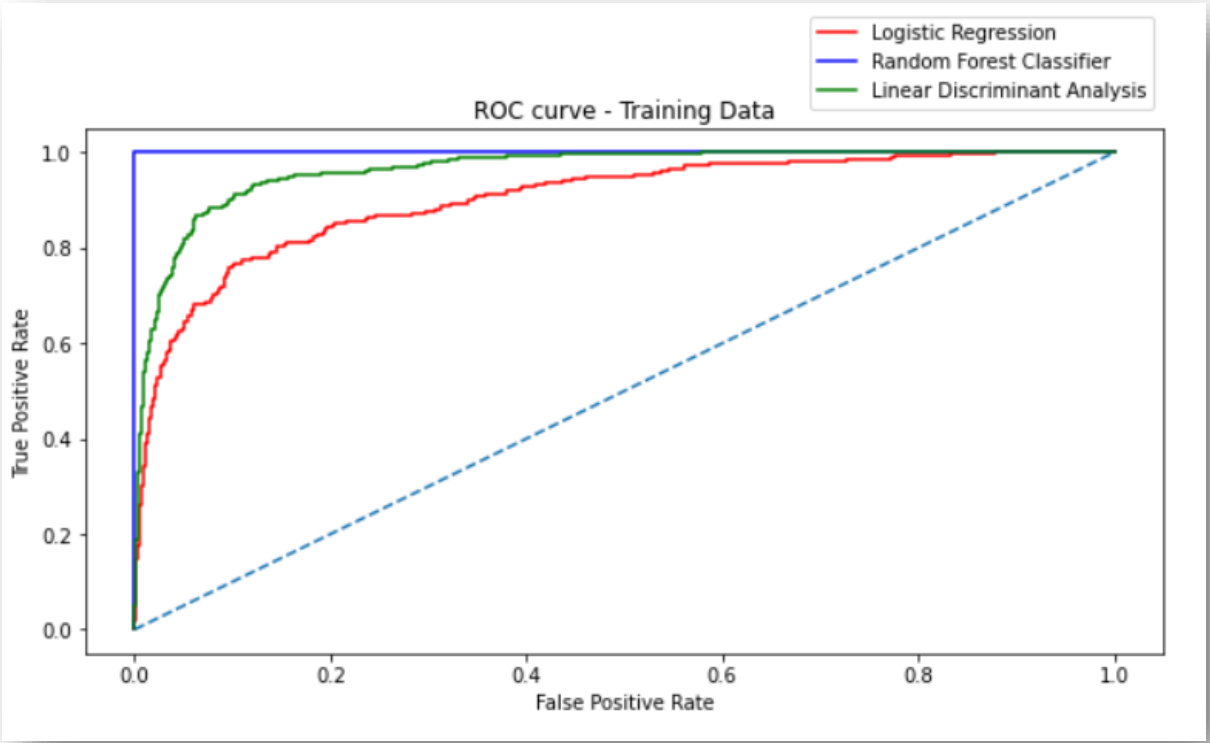

Figure 78: ROC Curve for all 3 models - Training Data

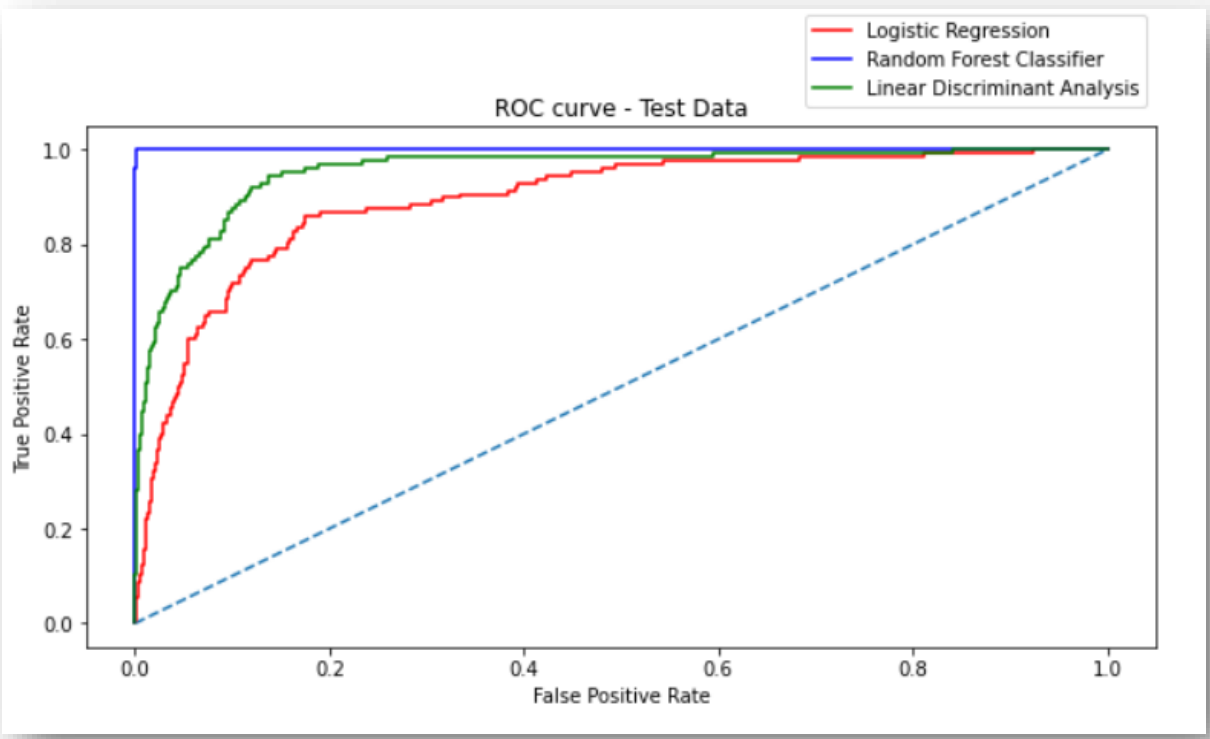

Figure 79: ROC Curve for all 3 models - Test Data

From Figure 78 and Figure 79, we are able to see that the Random Forest Classifier model is the most optimum for our case study as the curve for the Random Forest Classifier model is the steepest. The steeper the ROC Curve, the stronger the model.

Therefore, **Random Forest Classifier** is selected as the best model for our problem as it has better accuracy, precision, recall and f1 score compared to decision tree and artificial neural network model.

1.13 State Recommendations from the above models

Recommendations from the models

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.

Using the information gained from the above exercise, we can say that Random Forest is the best model.

We also looked at the coefficients derived from the best logistic (logit) model built using Stats model to derive some more insights. The results of the model are shown in Figure 80.

There are 9 most important variables that determine whether a business or the company will default or not. The 9 variables are ‘Equity_Paid_Up’, ‘Market_Capitalisation’, ‘ROG_Net_Worth_perc’, ‘ROG_Gross_Block_perc’, ‘ROG_Cost_of_Production_perc’, ‘Current_Ratio_Latest’, ‘Interest_Cover_Ratio_Latest’, ‘Debtors_Velocity_Days’ and ‘Creditors_Velocity_Days’.

Logit Regression Results

Dep. Variable:	Default	**No. Observations:**	2402
Model:	Logit	**Df Residuals:**	2392
Method:	MLE	**Df Model:**	9
Date:	Wed, 22 Jun 2022	**Pseudo R-squ.:**	0.3868
Time:	12:43:55	**Log-Likelihood:**	-504.98
converged:	True	**LL-Null:**	-823.47
Covariance Type:	nonrobust	**LLR p-value:**	2.413e-131

	coef	std err	z	P>|z|	[0.025	0.975]
Intercept	-0.8732	0.166	-5.252	0.000	-1.199	-0.547
Equity_Paid_Up	0.0172	0.007	2.495	0.013	0.004	0.031
Market_Capitalisation	-0.0101	0.002	-6.205	0.000	-0.013	-0.007
ROG_Net_Worth_perc	-0.0559	0.007	-7.990	0.000	-0.070	-0.042
ROG_Gross_Block_perc	-0.0441	0.013	-3.290	0.001	-0.070	-0.018
ROG_Cost_of_Production_perc	-0.0097	0.003	-3.797	0.000	-0.015	-0.005
Current_Ratio_Latest	-0.7665	0.088	-8.700	0.000	-0.939	-0.594
Interest_Cover_Ratio_Latest	-0.1703	0.030	-5.604	0.000	-0.230	-0.111
Debtors_Velocity_Days	-0.0040	0.001	-3.960	0.000	-0.006	-0.002
Creditors_Velocity_Days	0.0042	0.001	3.759	0.000	0.002	0.006

Figure 80: Important variables derived from logistic regression model

From the above analysis, we can infer below business insights. Following things should be kept in mind while investing in these companies.

1. 'Equity_Paid_Up' is the amount that has been received by the company through the issue of shares to the shareholders. Higher the 'Equity_paid_Up', higher is the chance of a default.
2. 'Market_Capitalisation' is the product of the total number of a company's outstanding shares and the current market price of one share. Higher the 'Market_Capitalisation', lower chance of default.
3. Lower the 'ROG_Net_Worth_perc' i.e., the rate of growth of networth, higher the chance of default.
4. Higher the 'ROG_Gross_Block_perc' i.e., the rate of growth of gross block, lower the chance of default.
5. Lower the 'ROG_Cost_of_Production_perc' i.e., the rate of growth of cost of production, higher the chance of default.
6. 'Current_Ratio_Latest' is the liquidity ratio. It is the company's ability to pay short-term obligations or those due within one year. Higher the 'Current_Ratio_Latest', lower the chance of default.
7. 'Interest_Cover_Ratio_Latest' is the activity ratio. It specifies the number of times the stock or inventory has been replaced and sold by the company. Lower the 'Interest_Cover_Ratio_Latest', higher the chance of default.
8. 'Debtors_Velocity_Days' is the average days required for receiving the payments. Higher the 'Debtors_Velocity_Days', lower the chance of default.

9. ‘Creditors_Velocity_Days’ is the average number of days company takes to pay suppliers. Higher the ‘Creditors_Velocity_Days’, higher the chance of default.

‘Current_Ratio_Lates’ is the most important criteria amongst the above parameters while ‘Equity_Paid_Up’ is the least important when considering these 9 parameters. However, all these 9 parameters remain important compared to the other variables in the dataset.

From a business point of view:

The companies must keep an eye on these 9 important factors to make sure their business is on the right track. If the companies concentrate on these factors and find ways to increase their business, the company will be able to keep up their debt and hence will not fall prey to default.

From an investor’s point of view:

When an investor is planning to invest in the stocks of a company, the investor must study the net worth of the company, if it is capable of handling its financial obligations, if it can grow quickly, and is able to manage the growth scale. The investor can use these 9 important factors to determine if the company is likely to default or not. This is an indicator for the investor to decide whether to invest in a business or not.

From a bank’s point of view:

The bank can decide whether to grant loan to a company based on the performance of the company, its net worth, probability of default etc., the bank can use these 9 variables as a way of predicting whether the company will default and hence to grant the loan or not.