greatlearning
Power Ahead

Financial Risk Analysis Project

Milestone 1

BUSINESS REPORT

greatlearning
Power Ahead

Business Problem

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.

Dataset for Problem 1: Company_Data2015.xlsx
Data Dictionary: Data_Dictionary.xlsx

Loading the Dataset and data pre-processing

All messy column names are fixed. All spaces, commas, brackets are replaced with either '_'

The first 5 rows of the dataset are as follows:

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

5 rows × 67 columns

The dataset has been loaded successfully.
It has 67 columns

Preliminary exploration of the Dataset

- Check for Duplicate Records

  No duplicate records exist

- Data types of the variables

Data type	No of variables
Object	1
float64	63
Int64	3

Co_Code is treated as an integer because it has numeric values. It is a Categorical variable and is therefore, converted to type 'object'. However, this is not used in any model building or EDA, and is therefore dropped.

Co_Name is the object variable and is dropped. We are then left with 65 variables.

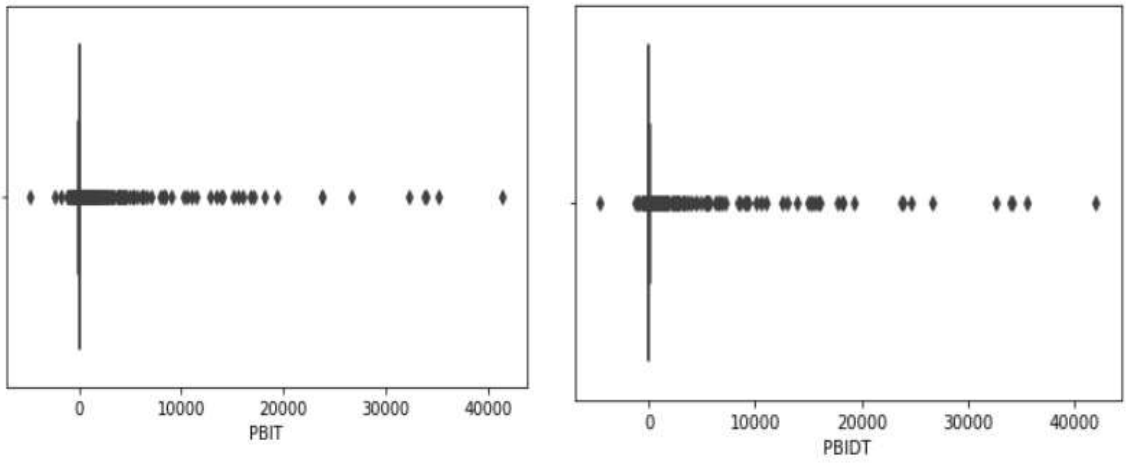
- A description of the first few columns of the dataset is given below:

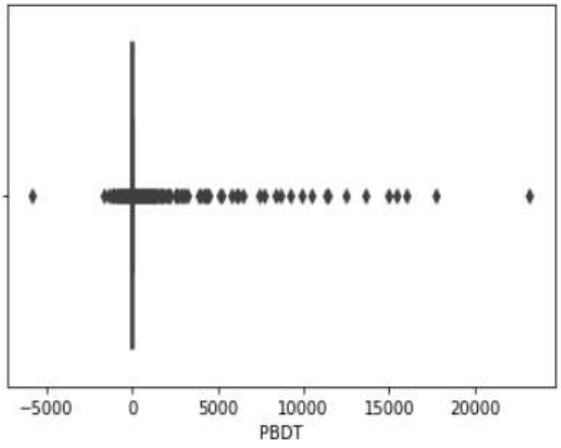
	Networth_Next_Year	Equity_Paid_Up	Networth	Capital_Employed	Total_Debt	Gross_Block	Net_Working_Capital	Curr_Assets
count	3586.000000	3586.000000	3586.000000	3586.000000	3586.000000	3586.000000	3586.000000	3586.000000
mean	725.045251	62.966584	649.746299	2799.611054	1994.823779	594.178829	410.809665	1960.349172
std	4769.681004	778.761744	4091.988792	26975.135385	23652.842746	4871.547802	6301.218546	22577.570829
min	-8021.600000	0.000000	-7027.480000	-1824.750000	-0.720000	-41.190000	-13162.420000	-0.910000
25%	3.985000	3.750000	3.892500	7.602500	0.030000	0.570000	0.942500	4.000000
50%	19.015000	8.290000	18.580000	39.090000	7.490000	15.870000	10.145000	24.540000
75%	123.802500	19.517500	117.297500	226.605000	72.350000	131.895000	61.175000	135.277500
max	111729.100000	42263.460000	81657.350000	714001.250000	652823.810000	128477.590000	223257.560000	721186.000000

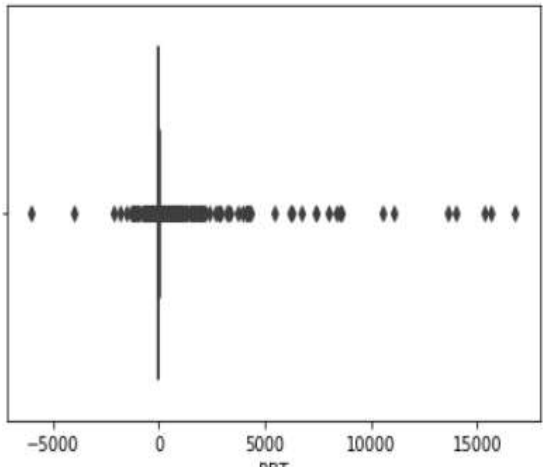
From the descriptive stats, we can see that most of the variables have outliers.

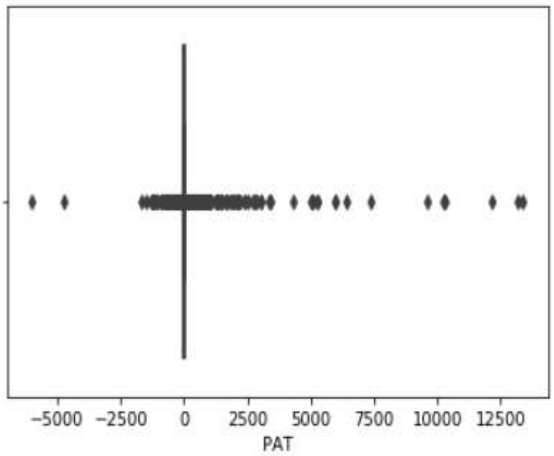
1.1 Outlier Treatment

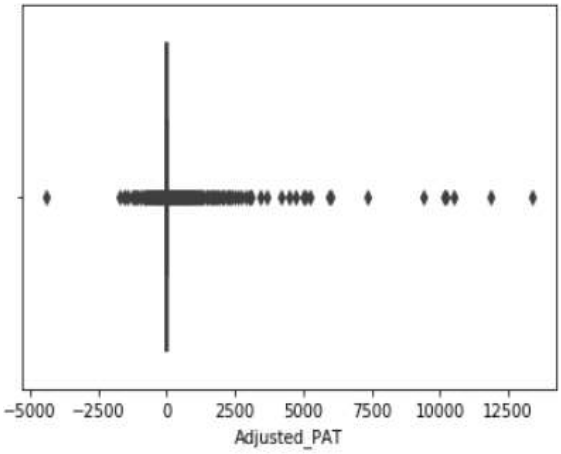
Detecting Outliers



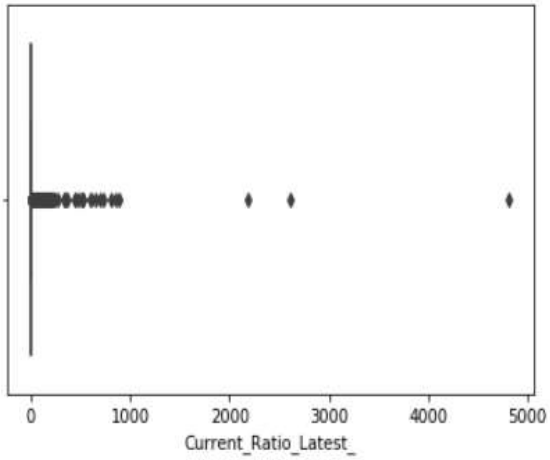


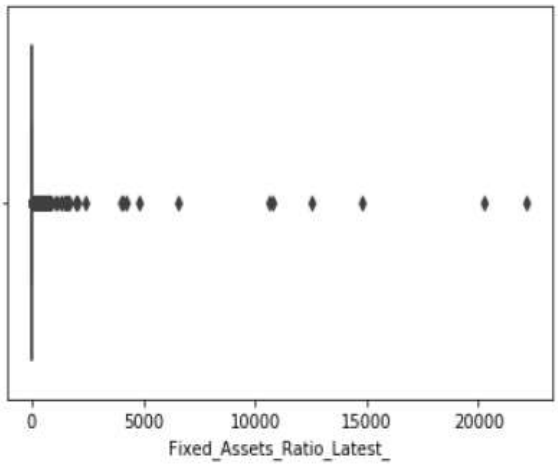




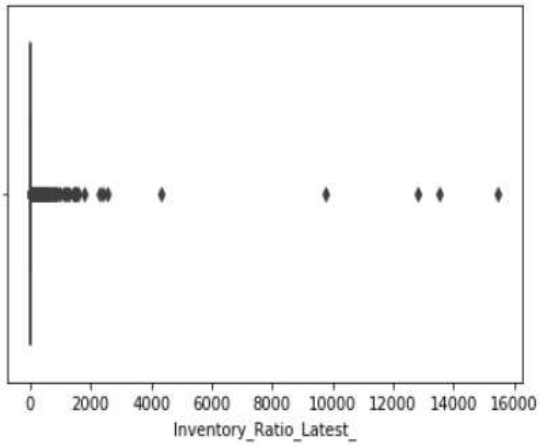


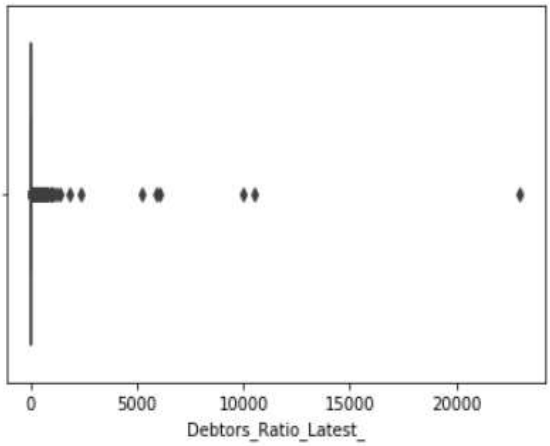
3

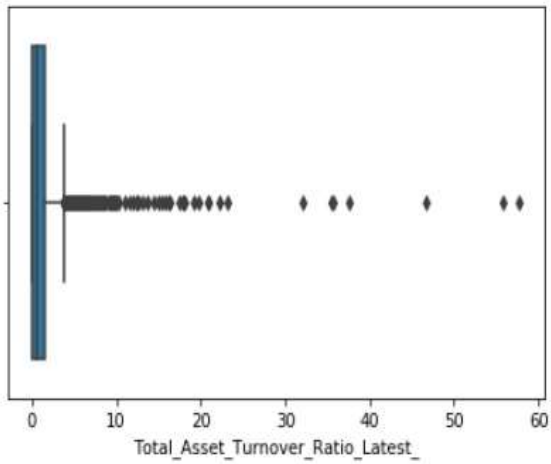


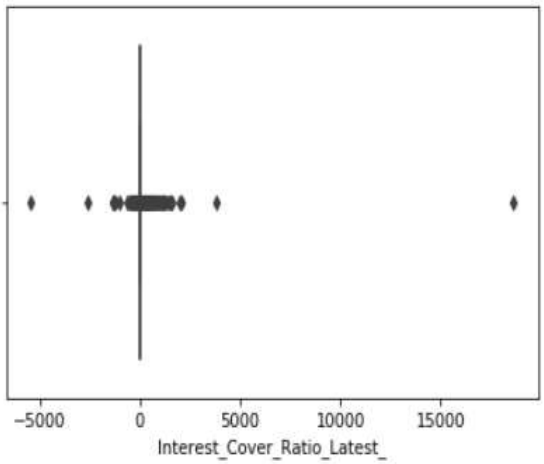


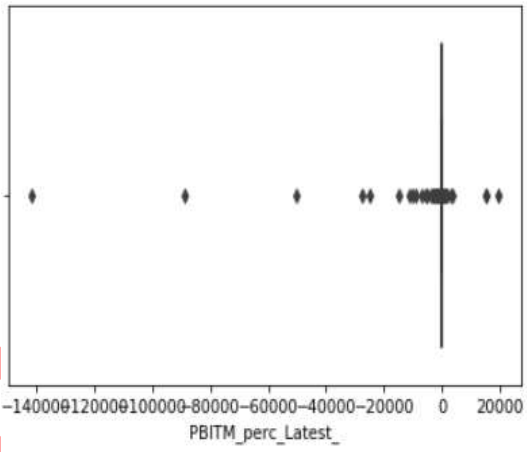
4

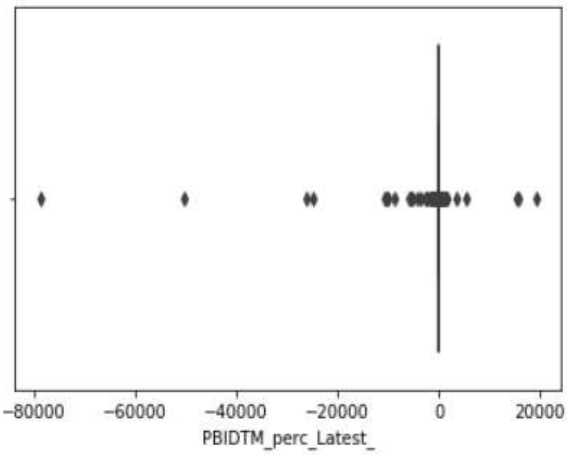


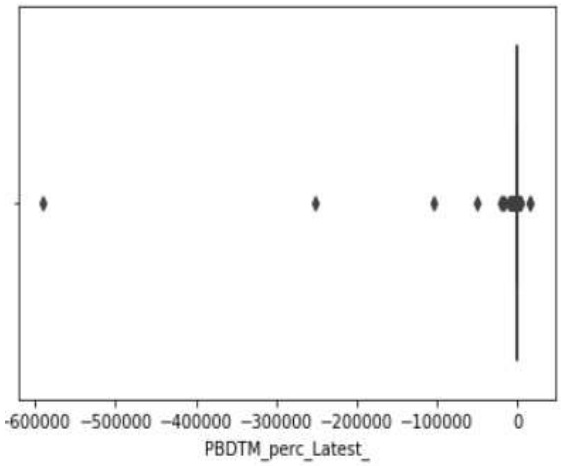












From the above boxplots, we can see there are outliers present in several variables

Another method of detecting Outliers:

Conventionally, outliers are identified based on the inter-quartile distance as follows:

Q1 – 25th Percentile

Q3 – 75th Percentile

IQR = Q3 – Q1

Lower outlier = Value < 1.5 * IQR

Upper Outlier = Value > 1.5 * IQR

Based on this definition, the number of outliers in the dataset is as follows:

```
Outlier = 1.5 * (75th - 25th)

No of Lower Outliers =  10604
No of Upper Outliers =  31427
```

A view for each of the individual variables is as follows:

	lower	upper	total
Networth_Next_Year	73	603	676
Equity_Paid_Up	0	448	448
Networth	66	584	650
Capital_Employed	5	591	596
Total_Debt	0	583	583
...	...	...	...
APATM_perc_Latest	0	398	398
Debtors_Vel_Days	0	391	391
Creditors_Vel_Days	1	261	262
Inventory_Vel_Days	0	150	150
Value_of_Output_to_Total_Assets	3	478	481

65 rows × 3 columns

Observations

- Many variables have outliers as defined by the above criteria.
- If all the records with outliers are dropped, only 112 records remain. Modeling cannot be done with only these records, so, dropping records is not an option.

The next option is to cap the outliers at both the ends.

By increasing the boundaries to 10th and 90th Percentile, the number of outliers is as follows:

```
Outlier = 1.5 * (90th - 10th)

No of Lower Outliers =  3025
No of Upper Outliers =  10042
```

By further increasing the boundaries to 5th and 95th Percentile, the number of outliers is as follows:

```
Outlier = 1.5 * (95th - 5th)

No of Lower Outliers =  1349
No of Upper Outliers =  4726
```

This seems to be a reasonable option, so, outliers have been treated as follows:

- Q1 – 25th Percentile
- Q3 – 75th Percentile
- IQR = Q3 – Q1
- Q05 – 5th Percentile
- Q95 – 95th Percentile

Rules:

- If Value < 1.5 * IQR, then replace it with Q05 (5th percentile cap on lower outliers)
- If Value > 1.5 * IQR, then replace it with Q95 (95th percentile cap on upper outliers)

With this, the outliers are now capped. A snapshot of first 5 records is seen below:

	Networth_Next_Year	Equity_Paid_Up	Networth	Capital_Employed	Total_Debt	Gross_Block	Net_Working_Capital	Curr_Assets
0	-17.445	131.24	-11.6975	0.4225	1572.61	1409.325	-11.945	40.50
1	-17.445	131.24	-11.6975	3634.9150	1572.61	1409.325	-11.945	2014.74
2	-17.445	131.24	1829.0825	3634.9150	1572.61	1409.325	827.735	2014.74
3	-17.445	131.24	-11.6975	3634.9150	1572.61	1409.325	-11.945	2014.74
4	-17.445	131.24	-11.6975	3634.9150	1572.61	1409.325	827.735	2014.74

For Networth_Next_Year, the value of -17.445 represents the 5th percentile of the original dataset (capped at the lower end). Similarly, for Networth, the value of 1829.0825 is the cap at the upper end.

As is to be expected, the median values of the original data and after outlier treatment do not change.

```
1 # Check that median does not change
2
3 Q50_outliers = df1.quantile(0.50)
4 Q50_originals = df.quantile(0.50)
5 (Q50_outliers != Q50_originals).sum().sum()
```

```
0
```

The description of the dataset after outlier treatment is seen below (for some variables)

	Networth_Next_Year	Equity_Paid_Up	Networth	Capital_Employed	Total_Debt	Gross_Block	Net_Working_Capital	Curr_Assets
count	3586.000000	3586.000000	3586.000000	3586.000000	3586.000000	3586.000000	3586.000000	3586.000000
mean	362.329521	24.997418	327.605738	660.513019	273.710134	247.798957	139.629154	362.093625
std	729.176980	41.065972	664.726629	1325.621474	573.436859	493.358861	289.823768	726.404599
min	-175.360000	0.000000	-162.010000	-286.870000	-0.720000	-41.190000	-89.250000	-0.910000
25%	3.985000	3.750000	3.892500	7.602500	0.030000	0.570000	0.942500	4.000000
50%	19.015000	8.290000	18.580000	39.090000	7.490000	15.870000	10.145000	24.540000
75%	123.802500	19.517500	117.297500	226.605000	72.350000	131.895000	61.175000	135.277500
max	1978.822500	131.240000	1829.082500	3634.915000	1572.610000	1409.325000	827.735000	2014.740000

4.2 Missing Value Treatment

The dataset with the treated outliers is used for further analysis. There is a total of 118 missing values as seen below:

	No of Missing Values
Creditors_Vel_Days	103
Book_Value_Unit_Curr	4
ROG_Market_Capitalisation_perc	1
Curr_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

Approach

Since the variables have outliers, median is the best measure of central tendency to fill in missing values.

Accordingly, the missing values have been treated using the SimpleImputer with strategy 'median'.

After imputing the missing values, it is confirmed that there are no more missing values:

```
1 df2.isna().sum().sum()
```

```
0
```

4.3 Transform Target Variable into 0 and 1

The definition of 'default' is as follows:

- If 'Networth_Next_Year' < 0, default = 1
- If 'Networth_Next_Year' > 0, default = 0

```
The distribution of the default values is
0    3199
1     387
```

- In terms of percentage distribution, it is as follows:

```
0    0.89208
1    0.10792
Name: default, dtype: float64
```

This is not a well-balanced dataset, and we could use techniques like SMOTE to balance it.

Median values of a few variables for default and non-default

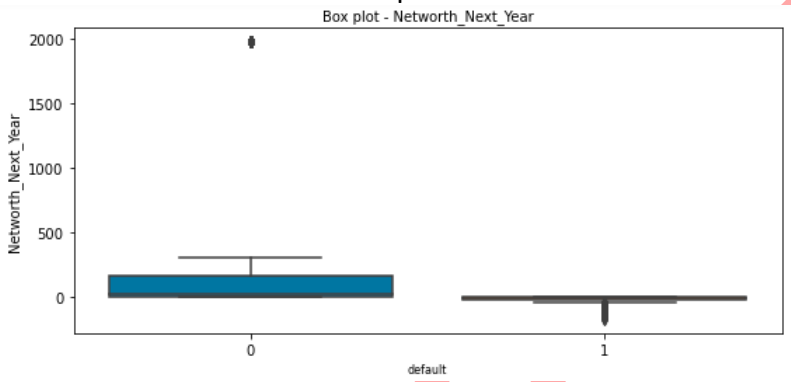
default	0	1
Networth_Next_Year	26.47	-13.00
Equity_Paid_Up	8.17	9.56
Networth	25.51	-8.56
Capital_Employed	44.23	4.23
Total_Debt	6.09	20.18
Gross_Block	15.04	20.92
Net_Working_Capital	12.27	-0.01
Curr_Assets	26.10	7.28
Curr_Liab_and_Prov	9.10	10.28
Total_Assets_to_Liab	55.68	20.37
Gross_Sales	37.17	5.79
Net_Sales	36.52	5.77
Other_Income	0.50	0.25
Value_Of_Output	36.81	4.32
Cost_of_Prod	29.21	7.06

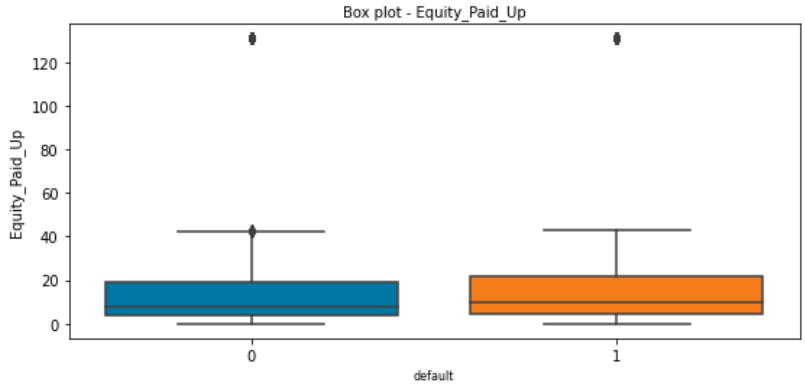
We can observe a difference in the median of the variables for the default and non-default.

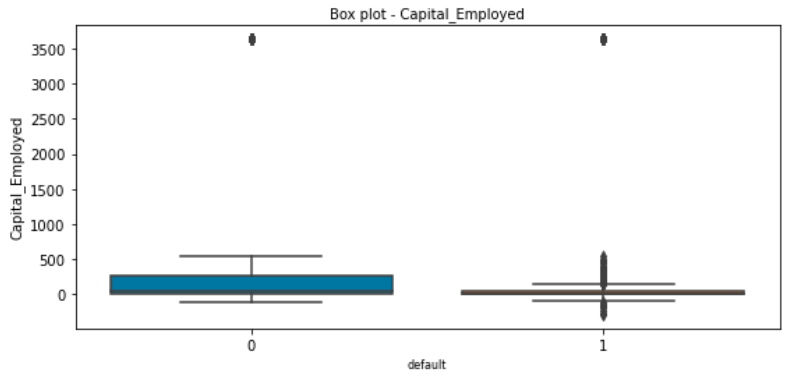
4.4 Univariate & Bivariate analysis with proper interpretation Univariate Analysis

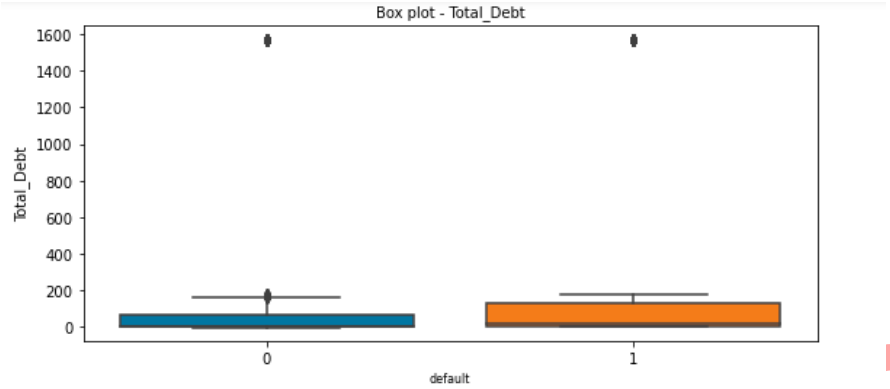
The purpose of Univariate Analysis is to find out which variables have clear separation for the target variable (default = 0 and 1).

The boxplot is a good visual technique to identify such variables as seen below for some of the independent variables:





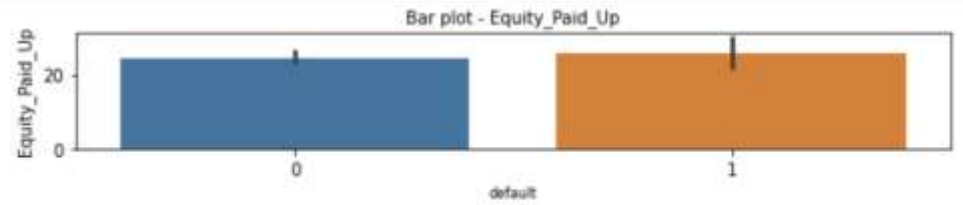


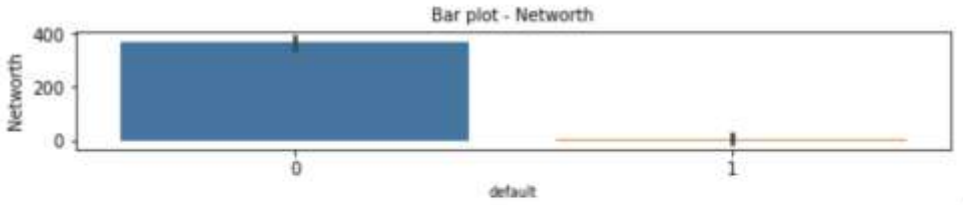


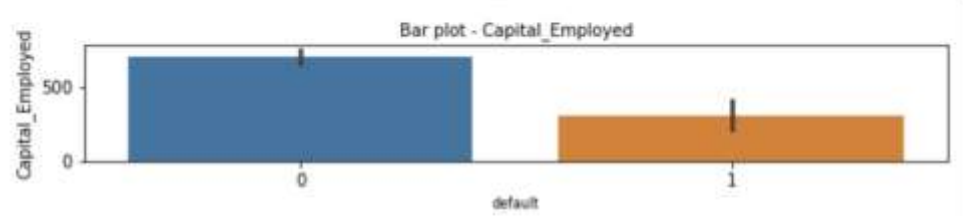
From the above boxplots, It is seen that there is no clear separation in the median values of the variables 'Equity_Paid_Up' and 'Total_Debt' between companies who defaulted (1) and those who did not. Such variables are not good predictors.

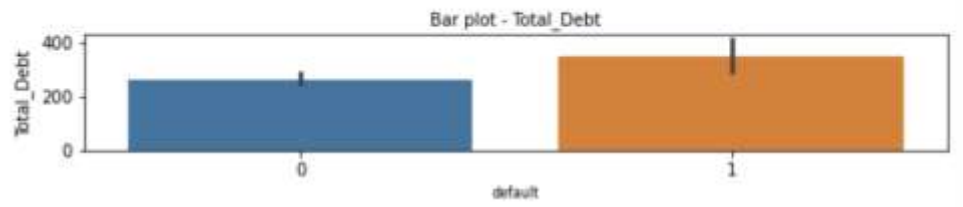
On the other hand, variables like 'Networth' and 'Capital_Employed' have clearer separation. Such variables are better predictors.

We also tried the barplots (plot the mean of the variables) for the same variables as shown in the above boxplots



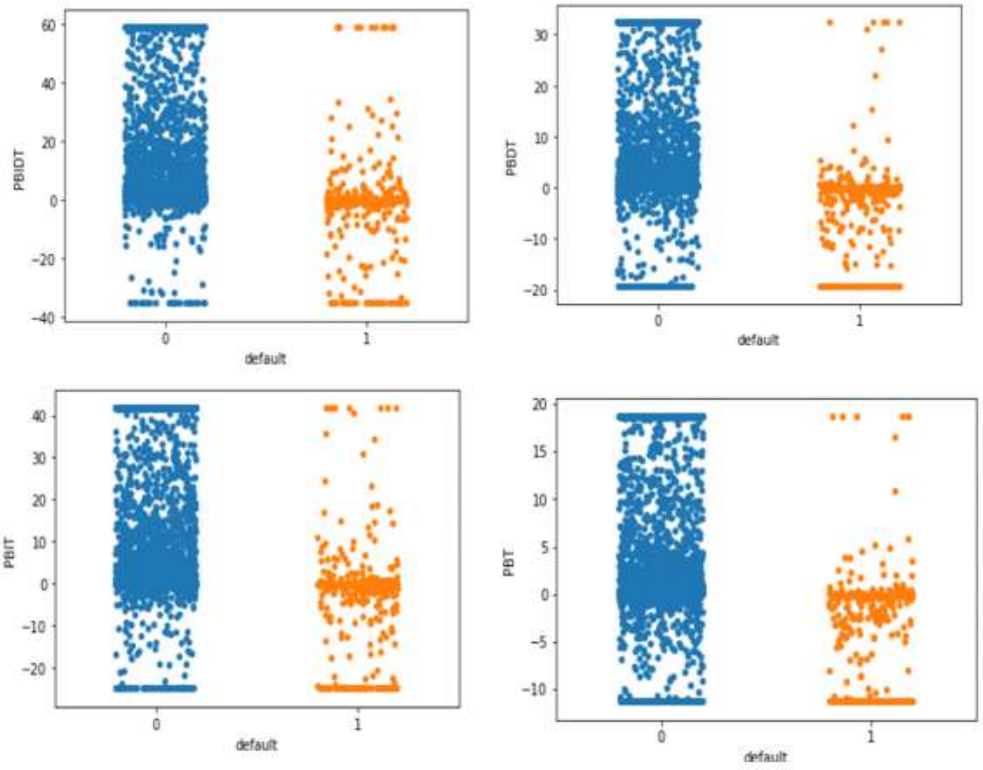


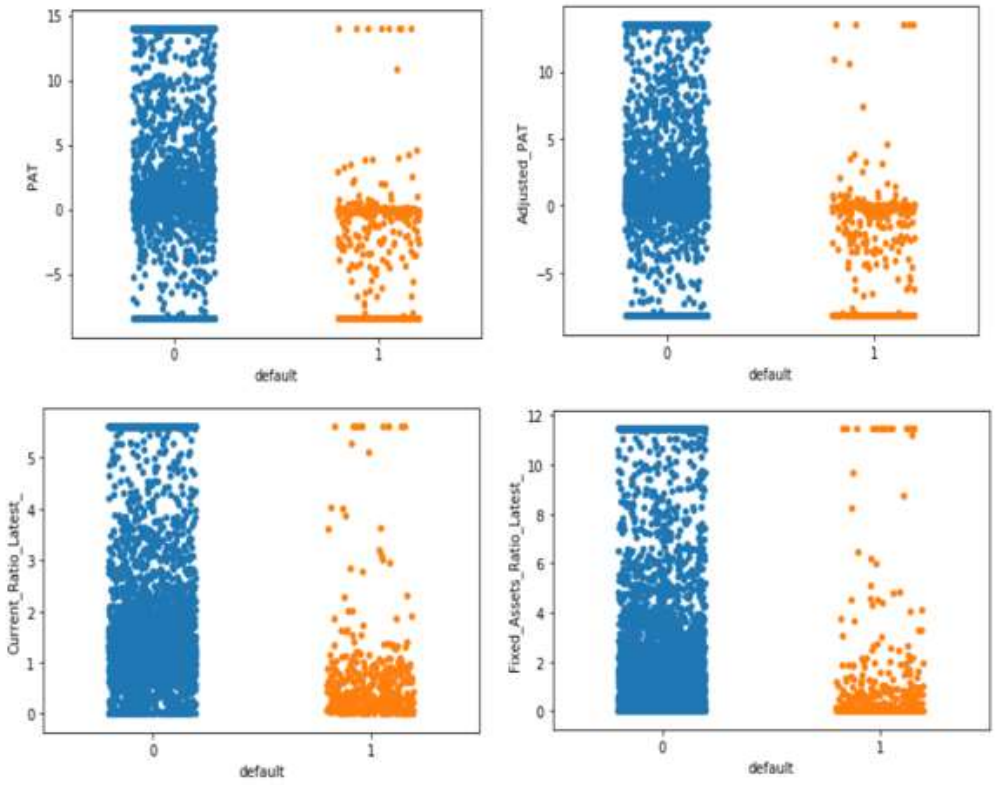




The barplots are better representations of the separation of the mean values. Thus, it is easily visible that there is a significant difference in the mean values of 'Networth' for companies that do not default (~ 400) compared to those who do default (~ 0). Similarly, for 'Capital_Employed', companies who default have a mean of ~250 as compared to those who do not default (> 500).

Strip Plot





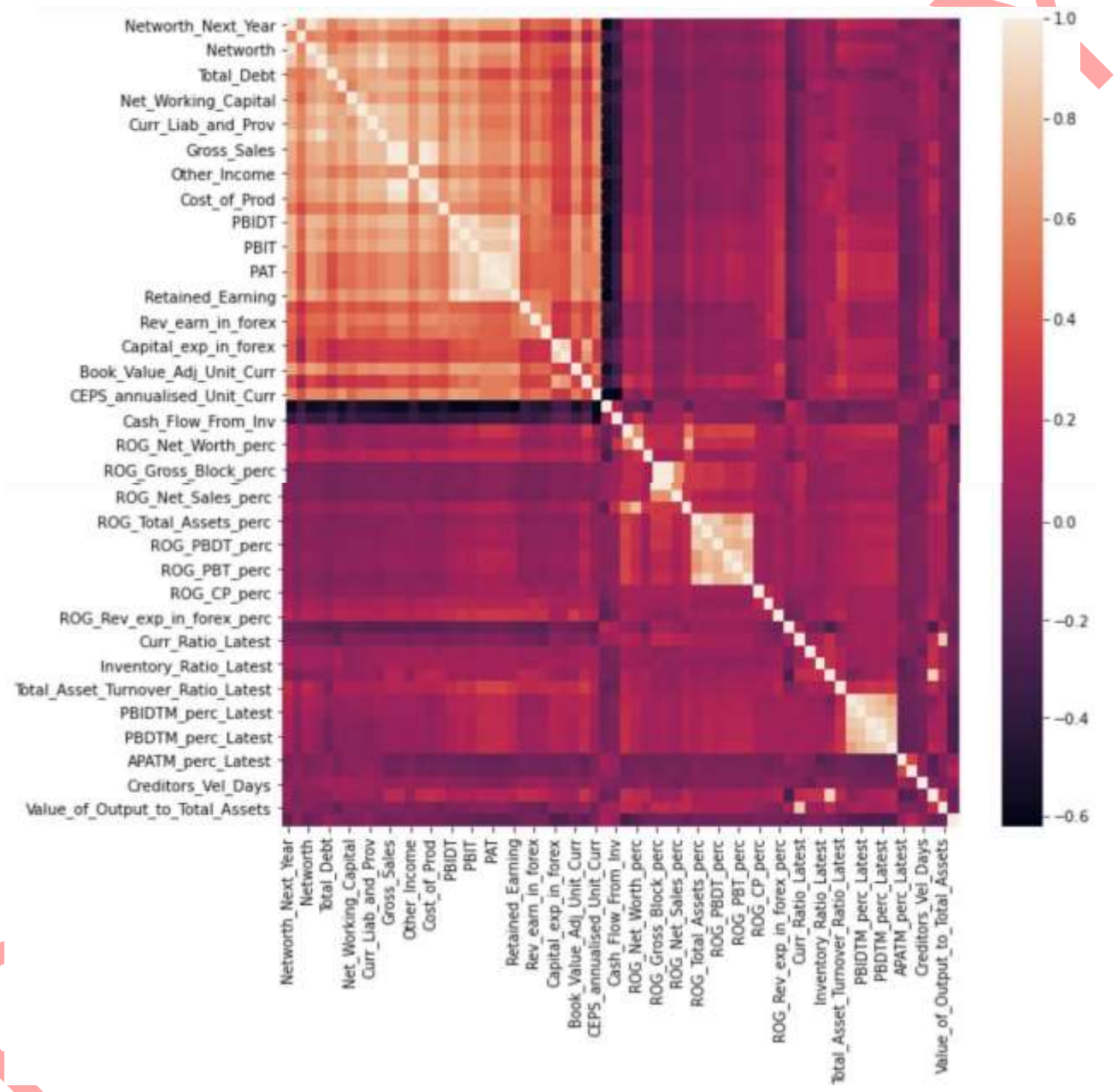
Observations:

1. The density of non-defaulters is pretty high
2. Those who default their parameters are generally close to 0 or negative

Bivariate Analysis

Since all the variables are continuous variables, the heatmap of the correlation matrix can give a very good idea of the correlations between the independent variables and also with the dependent variable.

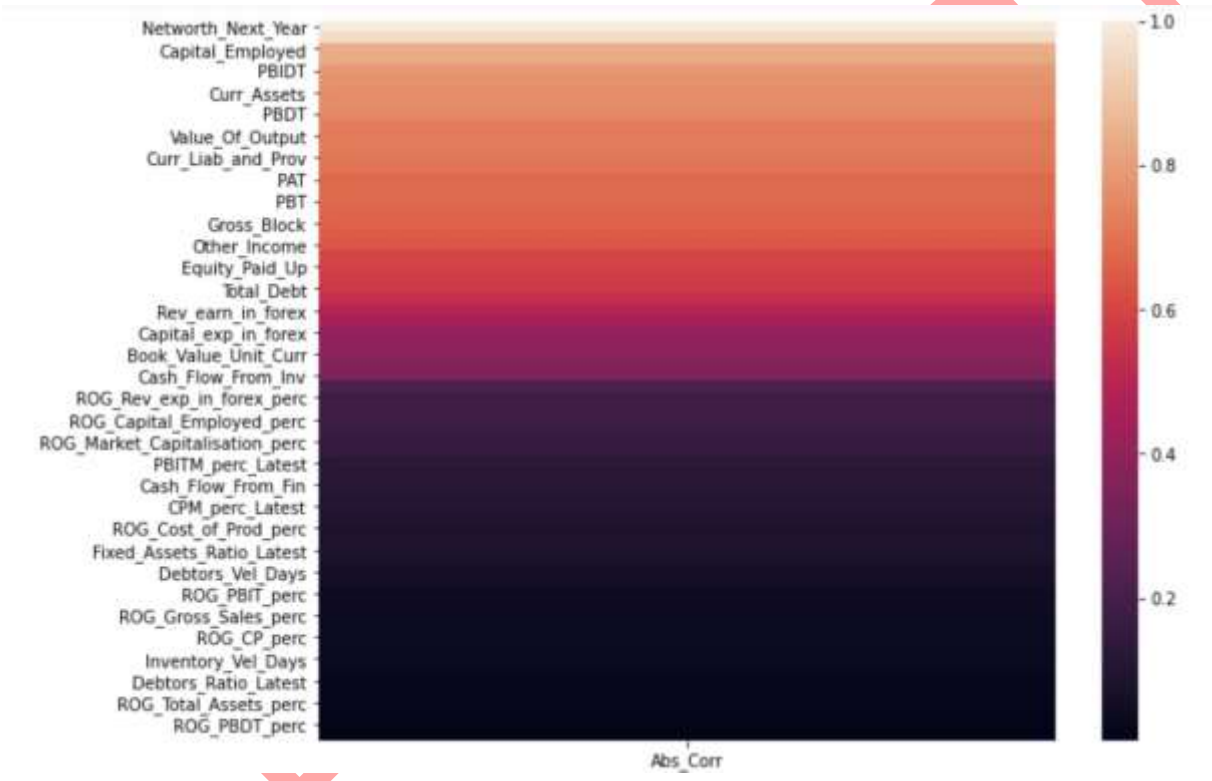
The heatmap is shown below:



It is seen that there is a high correlation between several independent variables.

Thus, for eg, Networth is highly correlated to Total_Debt and Net_Working_Capital; Gross_Sales to Other_Income, PBIDT is correlated to PBIT, PAT etc. This clearly establishes that the problem of Multi-collinearity exists.

To look at the correlation between the dependent variable (Networth_Next_Year) and the other independent variables, a heatmap of the absolute value of the correlations is shown below sorted in descending order:



Therefore, we see that variables like Networth, Capital_Employed, PBIDT, Curr_Assets, PBDT are highly correlated with Networth_Next_Year

4.5 Train-Test Split

As stated in the problem, the dataset has to be split in the ratio of 67:33 using random_state 42. Accordingly, the datasets have been split as follows:

```
There are  2402  records in the Train Dataset, and  1184  records in the Test Dataset
There are  65  variables in both Datasets
```

4.6 Build Logistic Regression Model (using statsmodel library) on most important variables on Train Dataset and choose the optimum cutoff

Approach to Model Building – Option 1 (Model A)

1. The first step is to address the problem of multi-collinearity. This occurs when the independent variables are highly correlated to each other. The VIF criterion used to exclude variables with high multi-collinearity is as follows:
   - If VIF > 5, then exclude these variables from the model

2. The next step is to build the logistic regression model with the remaining variables in an iterative manner. The Null Hypothesis is that there is no relationship between the independent variables and the dependent variable. We will reject the Null Hypothesis if p < 0.05; i.e, all variables whose p values < 0.05 have a relationship with the dependent variable, and are therefore, retained for the next iteration.

   All variables whose p values > 0.05 do not have an influence on the dependent variable, and therefore, are excluded from the next iteration of the model building exercise

3. We continue this iteration till all the variables have p values < 0.05
4. This model is designated Model A

VIF Results

```
No of Variables with VIF > 5 and therefore not used for model-building is  36
No of Variables with VIF <= 5 and therefore used for model-building is  28
```

The list of variables with VIF > 5 and therefore, excluded is as follows:

	Variables	VIF
4	Gross_Block	5.470572
49	Curr_Ratio_Latest	5.488591
63	Value_of_Output_to_Total_Assets	5.497634
58	CPM_perc_Latest	6.062378
41	ROG_PBDT_perc	6.071907
39	ROG_Total_Assets_perc	6.546595
7	Curr_Liab_and_Prov	6.802104
26	Book_Value_Unit_Curr	6.848143
1	Networth	7.146772
43	ROG_PBT_perc	8.691733
25	Capital_exp_in_forex	9.031952
52	Debtors_Ratio_Latest	9.602556
42	ROG_PBIT_perc	9.627200
44	ROG_PAT_perc	9.965258
54	Interest_Cover_Ratio_Latest	10.645208
62	Inventory_Vel_Days	10.810667
6	Curr_Assets	11.053218
55	PBIDTM_perc_Latest	11.383398
40	ROG_PBIDT_perc	11.915602
13	Cost_of_Prod	12.078660
57	PBDTM_perc_Latest	12.533986
56	PBITM_perc_Latest	13.591216
17	PBIT	13.662091
15	PBIDT	15.048930
2	Capital_Employed	18.791157
18	PBT	19.175663
20	Adjusted_PAT	20.714121
21	Retained_Earning	22.332441
8	Total_Assets_to_Liab	23.087069
16	PBDT	23.909271
19	PAT	33.245424
9	Gross_Sales	55.886002

36	ROG_Gross_Sales_perc	94.273344
35	ROG_Gross_Block_perc	94.680462
12	Value_Of_Output	149.567259
10	Net_Sales	195.744386

The list of variables with VIF <= 5 and therefore, retained is as follows:

	Variables	VIF
45	ROG_CP_perc	1.134416
46	ROG_Rev_earn_in_forex_perc	1.166949
34	ROG_Capital_Employed_perc	1.340372
48	ROG_Market_Capitalisation_perc	1.379426
47	ROG_Rev_exp_in_forex_perc	1.533276
60	Debtors_Vel_Days	1.551861
50	Fixed_Assets_Ratio_Latest	1.583765
61	Creditors_Vel_Days	1.619587
59	APATM_perc_Latest	1.653791
51	Inventory_Ratio_Latest	1.658631
53	Total_Asset_Turnover_Ratio_Latest	1.696196
37	ROG_Net_Sales_perc	1.906414
32	Cash_Flow_From_Fin	2.221831
31	Cash_Flow_From_Inv	2.365223
22	CP	2.445688
24	Rev_exp_in_forex	2.645107
0	Equity_Paid_Up	2.700335
14	Selling_Cost	2.888785
11	Other_Income	2.908752
23	Rev_earn_in_forex	2.911660
30	Cash_Flow_From_Opr	2.952774
27	Book_Value_Adj_Unit_Curr	3.233561
38	ROG_Cost_of_Prod_perc	3.573501
33	ROG_Net_Worth_perc	3.741435
28	Market_Capitalisation	4.060504
3	Total_Debt	4.091577
5	Net_Working_Capital	4.179803
29	CEPS_annualised_Unit_Curr	4.212300

Iteration 1:

The formula used is as follows:

```
default ~  + ROG_CP_perc + ROG_Rev_earn_in_forex_perc + ROG_Capital_Employed_perc + ROG_Market_Capitalisation_perc + ROG_Rev_ex
p_in_forex_perc + Debtors_Vel_Days + Fixed_Assets_Ratio_Latest + Creditors_Vel_Days + APATM_perc_Latest + Inventory_Ratio_Lates
t + Total_Asset_Turnover_Ratio_Latest + ROG_Net_Sales_perc + Cash_Flow_From_Fin + Cash_Flow_From_Inv + CP + Rev_exp_in_forex +
Equity_Paid_Up + Selling_Cost + Other_Income + Rev_earn_in_forex + Cash_Flow_From_Opr + Book_Value_Adj_Unit_Curr + ROG_Cost_of_
Prod_perc + ROG_Net_Worth_perc + Market_Capitalisation + Total_Debt + Net_Working_Capital + CEPS_annualised_Unit_Curr
```

Results of Iteration 1:

```
No of variables with p > 0.05 in Iteration 1 =  16
```

	p_value
ROG_Rev_exp_in_forex_perc	0.872975
Cash_Flow_From_Inv	0.780807
ROG_CP_perc	0.644683
ROG_Net_Worth_perc	0.467376
Equity_Paid_Up	0.427212
Selling_Cost	0.334400
CP	0.286900
Other_Income	0.284908
Rev_exp_in_forex	0.264298
CEPS_annualised_Unit_Curr	0.182748
Cash_Flow_From_Opr	0.141988
ROG_Cost_of_Prod_perc	0.075512
ROG_Rev_earn_in_forex_perc	0.062689
Rev_earn_in_forex	0.062013
Creditors_Vel_Days	0.059317
Fixed_Assets_Ratio_Latest	0.055460

These variables are excluded in the next iteration of the model building exercise.

The following 12 variables with p values < 0.05 are considered in Iteration 2:

	p_value
ROG_Capital_Employed_perc	2.276783e-02
ROG_Market_Capitalisation_perc	1.669855e-08
Debtors_Vel_Days	3.658206e-04
APATM_perc_Latest	6.656047e-06
Inventory_Ratio_Latest	4.463983e-02
Total_Asset_Turnover_Ratio_Latest	2.056906e-03
ROG_Net_Sales_perc	1.617034e-02
Cash_Flow_From_Fin	6.725161e-06
Book_Value_Adj_Unit_Curr	3.936448e-04
Market_Capitalisation	1.203234e-04
Total_Debt	4.354414e-09
Net_Working_Capital	6.086964e-05

Iteration 2:

The formula used is as follows:

```
'default ~  + ROG_Capital_Employed_perc + ROG_Market_Capitalisation_perc + Debtors_Vel_Days + APATM_perc_Latest + Inventory_Rat
io_Latest + Total_Asset_Turnover_Ratio_Latest + ROG_Net_Sales_perc + Cash_Flow_From_Fin + Book_Value_Adj_Unit_Curr + Market_Cap
italisation + Total_Debt + Net_Working_Capital'
```

Results

The model summary is given below:

Logit Regression Results

Dep. Variable:	default	**No. Observations:**	2402
Model:	Logit	**Df Residuals:**	2389
Method:	MLE	**Df Model:**	12
Date:	Sun, 17 Jan 2021	**Pseudo R-squ.:**	0.3169
Time:	23:01:48	**Log-Likelihood:**	-561.10
converged:	True	**LL-Null:**	-821.36
Covariance Type:	nonrobust	**LLR p-value:**	9.512e-104

The variables with their p values is seen below:

	coef	std err	z	P>|z|	[0.025	0.975]
Intercept	-1.3818	0.128	-10.781	0.000	-1.633	-1.131
ROG_Capital_Employed_perc	-0.0141	0.005	-2.969	0.003	-0.023	-0.005
ROG_Market_Capitalisation_perc	-0.0549	0.010	-5.395	0.000	-0.075	-0.035
Debtors_Vel_Days	0.0014	0.000	3.845	0.000	0.001	0.002
APATM_perc_Latest	-0.0017	0.000	-4.421	0.000	-0.002	-0.001
Inventory_Ratio_Latest	-0.0202	0.007	-2.750	0.006	-0.035	-0.006
Total_Asset_Turnover_Ratio_Latest	-0.0314	0.010	-3.214	0.001	-0.051	-0.012
ROG_Net_Sales_perc	-0.0028	0.001	-2.425	0.015	-0.005	-0.001
Cash_Flow_From_Fin	-0.0173	0.003	-5.311	0.000	-0.024	-0.011
Book_Value_Adj_Unit_Curr	-0.0006	0.000	-4.619	0.000	-0.001	-0.000
Market_Capitalisation	-0.1004	0.021	-4.841	0.000	-0.141	-0.060
Total_Debt	0.0011	0.000	5.506	0.000	0.001	0.001
Net_Working_Capital	-0.0022	0.000	-4.394	0.000	-0.003	-0.001

All the variables have p value < 0.05, and are therefore, significant.

It is seen below that multi-collinearity amongst the independent variables is drastically reduced (VIF < 5). So, we retain all the variables

VIF values for Variables used in Iteration 2

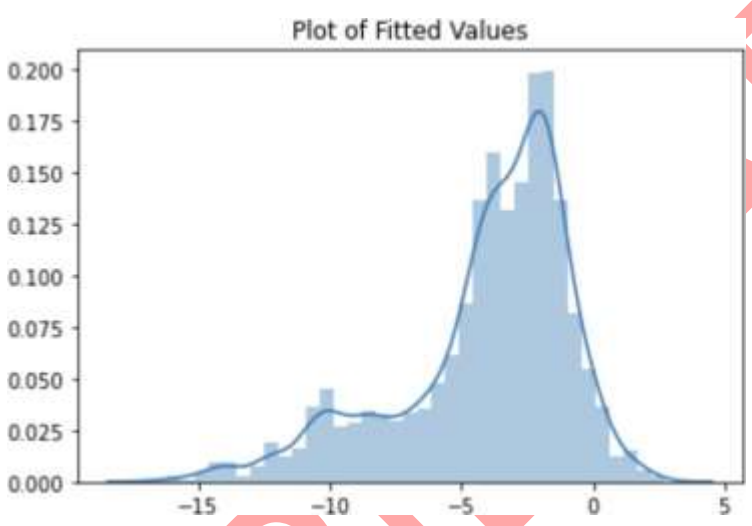
	Variables	VIF
6	ROG_Net_Sales_perc	1.146829
1	ROG_Market_Capitalisation_perc	1.166606
0	ROG_Capital_Employed_perc	1.192211
7	Cash_Flow_From_Fin	1.263466
4	Inventory_Ratio_Latest	1.286261
2	Debtors_Vel_Days	1.431154
5	Total_Asset_Turnover_Ratio_Latest	1.433099
3	APATM_perc_Latest	1.490907
9	Market_Capitalisation	1.616628
8	Book_Value_Adj_Unit_Curr	1.994955
10	Total_Debt	2.213925
11	Net_Working_Capital	2.342074

```
The adjusted pseudo R-square value of Iteration 2 is 0.3023
```

The Pseudo R-squ value is 0.3169 and the adjusted Pseudo R-sq value is 0.3023. They are fairly close, and therefore, there are lesser insignificant independent variables in the iteration 2 of the model.

We can go ahead and use this to determine the optimum cutoff value.

The distribution plot of the fitted values is seen below



Approach for cutoff

We are interested in predicting whether a company is likely to default, so that a bank or a lending company may either charge a higher premium in case of a high risk, or not give a loan at all. In such a case, Recall is the most important metric. A model which maximizes Recall (and therefore minimizes False Negatives) is preferred. This ensures that loans are not given to companies which the model had predicted would not default, but later on, actually defaulted.

On the other hand, being over-cautious and not lending to any companies who have even a slight risk of default will also not be practical and will lead to loss of business and market share. Therefore, there needs to be a balance between the two.

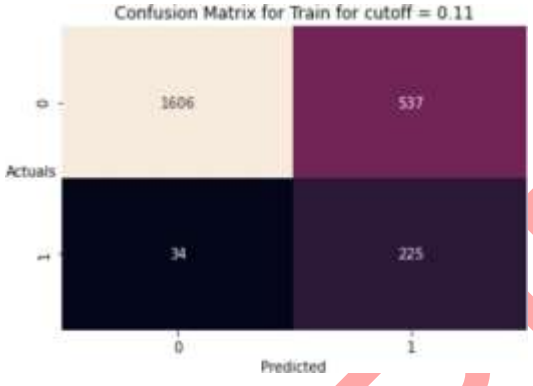
The following table gives a comparison between the different performance measures at different cutoff levels (i.e. at what value of y would be differentiate between defaulters and non-defaulters on the train dataset). A minimum cutoff value of 0.11 is considered, since the proportion of defaulters in the original dataset is ~ 0.11. The models have been evaluated with 5 different cutoff values as shown below:

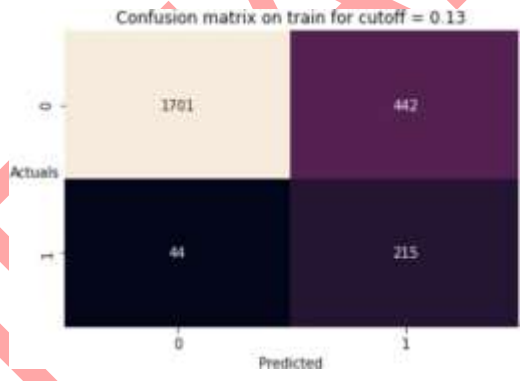
Model A Performance Measures on Training Dataset

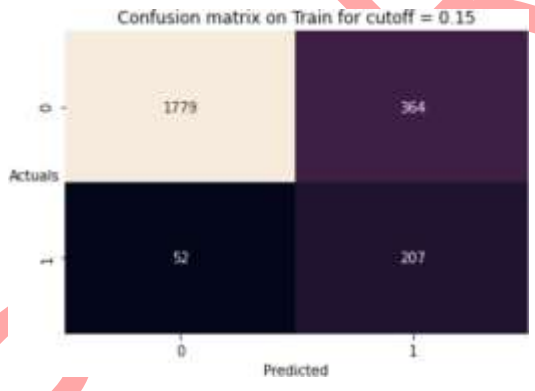
	Cutoff_0.11	Cutoff_0.13	Cutoff_0.15	Cutoff_0.17	Cutoff_0.19
Recall	0.869	0.83	0.799	0.772	0.741
F1 Score	0.441	0.469	0.499	0.531	0.552
Precision	0.295	0.327	0.363	0.404	0.439
Accuracy	0.76	0.8	0.83	0.850	0.870

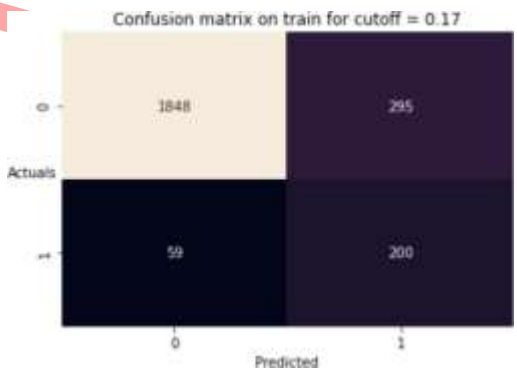
It is seen that by increasing the cutoff value, the overall Accuracy increases. However, this is offset by the reduction in Recall, which is very important.

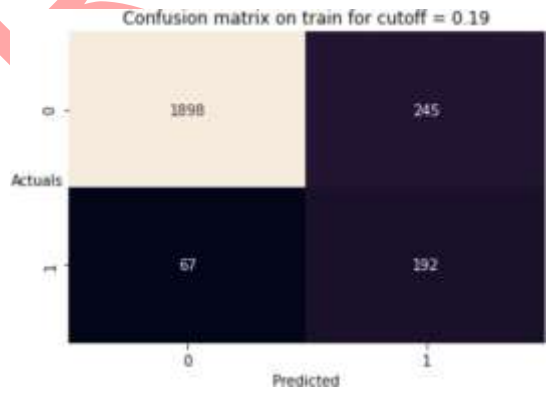
The confusion matrices for the different cutoffs on Train dataset are given below:











Optimum Cutoff

There needs to be a balance between the following 2 opposing factors:

False Positives

This means the model predicts the company is a defaulter, but in reality, is not a defaulter. This scenario represents a Lost Opportunity for the investor, since he would not have invested in the company, thinking it was a defaulter.

False negatives

This is when the model predicts the company is not a defaulter, but in reality, actually defaults. This is a big loss to the investor, and therefore, needs to be minimized.

The table below gives the incremental False Positive and False Negative scenarios at the different Cutoff levels.

Cutoff level	Number of Bad loans (FN)	Incremental Bad Loans	No of Lost opportunities (FP)	Incremental Lost opportunity	Ratio
A	B	C	D	E	E / C
Baseline (0.11)	34		537		
0.13	44	10 (44-34)	442	95 (537-442)	9.5
0.15	52	8	364	78	9.75
0.17	59	7	305	59	8.42
0.19	67	8	245	50	6.25

For eg, if the cutoff increases from 0.11 to 0.13, 10 additional loans go bad (these companies will default), but 95 additional customers are gained.

At a cutoff value of 0.15 and above, about 8 additional loans go bad, but with reducing additional business (78).

Therefore, a cutoff value of 0.15 seems to be the most optimum. This will give us the following model performance metrices **(Model A)**

Accuracy: 0.83
Recall: 0.799
Precision: 0.36

Approach to Model Building – Option 2 (Model B)

- Here, the model is built on all the variables without checking for multi-collinearity.
- Then, variables with p value > = 0.05 are dropped and the model is built again
- The iterative process is continued till all the variables used in the model have p values < 0.05

Results

- Iteration 1:

  The model is built using all the variables as follows:

```
'default ~  + Equity_Paid_Up + Networth + Capital_Employed + Total_Debt + Gross_Block + Net_Working_Capital + Curr_Assets + Cur
r_Liab_and_Prov + Total_Assets_to_Liab + Gross_Sales + Net_Sales + Other_Income + Value_Of_Output + Cost_of_Prod + Selling_Cost
+ PBIDT + PBDT + PBIT + PBT + PAT + Adjusted_PAT + Retained_Earning + CP + Rev_earn_in_forex + Rev_exp_in_forex + Capital_exp_i
n_forex + Book_Value_Unit_Curr + Book_Value_Adj_Unit_Curr + Market_Capitalisation + CEPS_annualised_Unit_Curr + Cash_Flow_From_
Opr + Cash_Flow_From_Inv + Cash_Flow_From_Fin + ROG_Net_Worth_perc + ROG_Capital_Employed_perc + ROG_Gross_Block_perc + ROG_Gro
ss_Sales_perc + ROG_Net_Sales_perc + ROG_Cost_of_Prod_perc + ROG_Total_Assets_perc + ROG_PBIDT_perc + ROG_PBDT_perc + ROG_PBIT_
perc + ROG_PBT_perc + ROG_PAT_perc + ROG_CP_perc + ROG_Rev_earn_in_forex_perc + ROG_Rev_exp_in_forex_perc + ROG_Market_Capitali
sation_perc + Curr_Ratio_Latest + Fixed_Assets_Ratio_Latest + Inventory_Ratio_Latest + Debtors_Ratio_Latest + Total_Asset_Turno
ver_Ratio_Latest + Interest_Cover_Ratio_Latest + PBIDTM_perc_Latest + PBITM_perc_Latest + PBDTM_perc_Latest + CPM_perc_Latest +
APATM_perc_Latest + Debtors_Vel_Days + Creditors_Vel_Days + Inventory_Vel_Days + Value_of_Output_to_Total_Assets'
```

  54 variables have p >= 0.05 are therefore, excluded from the next iteration

```
No of Variables with p >= 0.05 is  54
```

  The following 10 variables have p < 0.05 and are included in the next iteration (note: Intercept is not a variable) :

	p_value
Intercept	0.010936
Net_Sales	0.005846
Value_Of_Output	0.004497
PBIT	0.011526
Rev_earn_in_forex	0.002291
Book_Value_Unit_Curr	0.000836
CEPS_annualised_Unit_Curr	0.011892
Cash_Flow_From_Opr	0.009977
Cash_Flow_From_Fin	0.015818
ROG_Market_Capitalisation_perc	0.000616
APATM_perc_Latest	0.001874

- Iteration 2

  The model is built using the above variables.

  The following variables have p value >= 0.05 and are therefore, excluded from the next iteration:

	p_value
CEPS_annualised_Unit_Curr	0.671861
Cash_Flow_From_Opr	0.527582
APATM_perc_Latest	0.080604
PBIT	0.058704

- Iteration 3

  The model us built using the remaining 6 variables. The results are as follows:

Logit Regression Results

Dep. Variable:	default	**No. Observations:**	2402
Model:	Logit	**Df Residuals:**	2395
Method:	MLE	**Df Model:**	6
Date:	Wed, 20 Jan 2021	**Pseudo R-squ.:**	0.5999
Time:	18:21:56	**Log-Likelihood:**	-328.59
converged:	True	**LL-Null:**	-821.36
Covariance Type:	nonrobust	**LLR p-value:**	1.198e-209

	coef	std err	z	P>|z|	[0.025	0.975]
Intercept	-0.9411	0.123	-7.680	0.000	-1.181	-0.701
Net_Sales	0.0276	0.010	2.807	0.005	0.008	0.047
Value_Of_Output	-0.0277	0.010	-2.837	0.005	-0.047	-0.009
Rev_earn_in_forex	0.0046	0.002	3.013	0.003	0.002	0.008
Book_Value_Unit_Curr	-0.1776	0.013	-14.062	0.000	-0.202	-0.153
Cash_Flow_From_Fin	-0.0166	0.003	-5.614	0.000	-0.022	-0.011
ROG_Market_Capitalisation_perc	-0.0466	0.011	-4.067	0.000	-0.069	-0.024

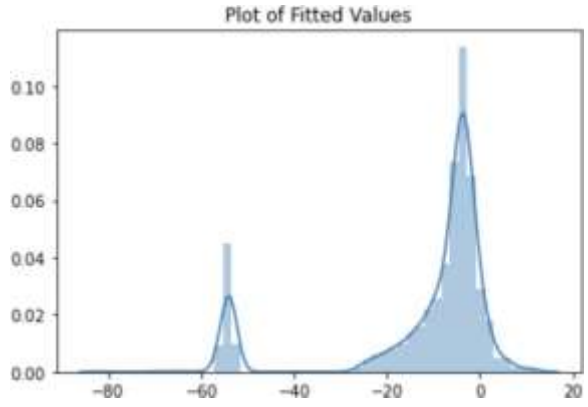
  It is seen that all the variables have p value < 0.05.

  The adjusted pseudo R-square value (0.5926) is very close to Pseudo R-square value of 0.599, indicating lesser number of insignificant variables:

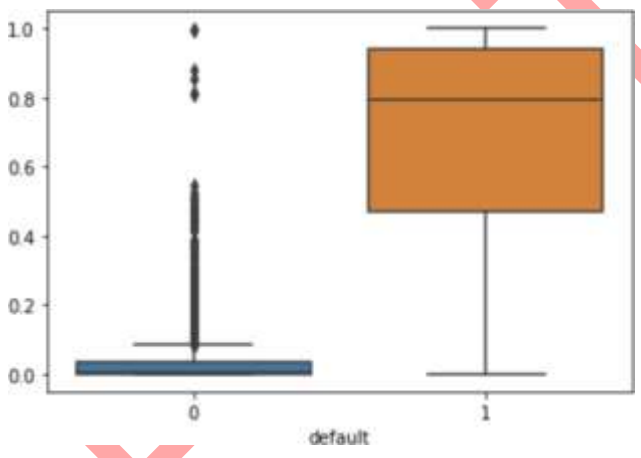
```
The adjusted pseudo R-square value of Iteration 2 is 0.5926
```

Model B Performance Measures on Training Dataset

The fitted values are plotted as follows:

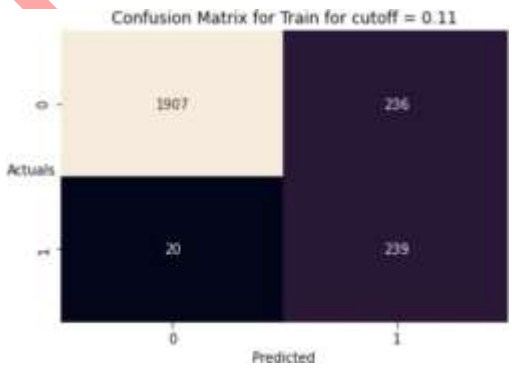


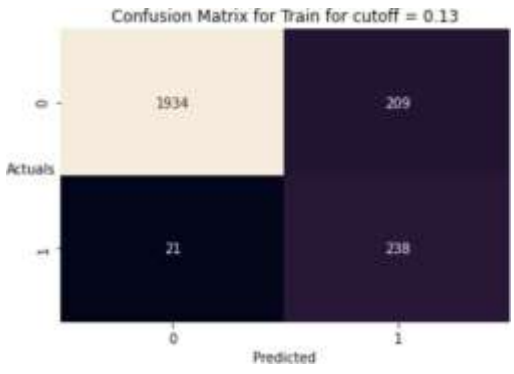
The predictions on the train set are plotted for the different default values:

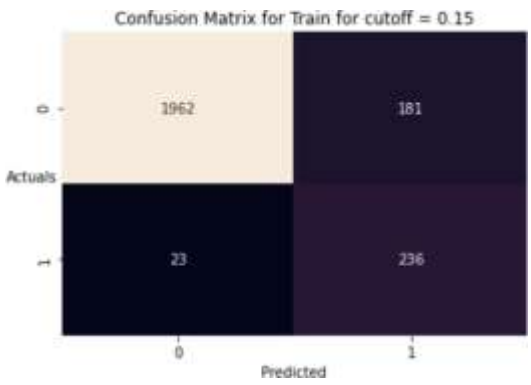


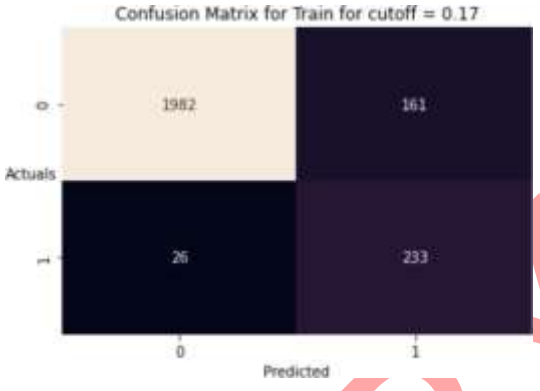
The Confusion matrix and the classification reports were run for different cutoff levels of 0.11, 0.13, 0.15, 0.17 and 0.19.

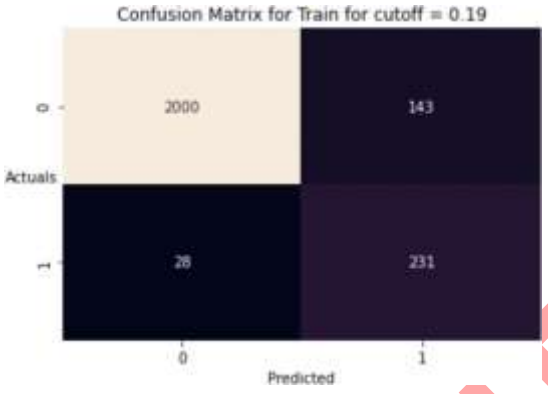
The results are shown below:











A summary of the Model B performance measures is given in the table below:

	Cutoff_0.11	Cutoff_0.13	Cutoff_0.15	Cutoff_0.17	Cutoff_0.19
Recall	0.923	0.919	0.911	0.900	0.892
F1 Score	0.651	0.674	0.698	0.714	0.730
Precision	0.503	0.532	0.566	0.591	0.618
Accuracy	0.89	0.9	0.92	0.920	0.930

Optimum Cutoff Level for Model B

As seen earlier, a balance between Recall and Precision and Accuracy is required.

A chart showing the cutoff levels with the incremental impact is shown below:

Cutoff level	Number of Bad loans (FN)	Incremental Bad Loans	No of Lost opportunities (FP)	Incremental Lost opportunity	Ratio
A	B	C	D	E	E / C
Baseline (0.11)	20		236		
0.13	21	1 (21-20)	209	27 (236-209)	27
0.15	23	2	181	28	14
0.17	26	3	161	20	6.67
0.19	28	2	143	18	9

By the logic explained earlier, the optimum cutoff level is 0.15. At this level, we will get 28 additional customers at the expense of 2 additional bad loans.

The classification report on the training dataset for the cutoff value of 0.15 is given below **(Model B):**

	precision	recall	f1-score	support
0	0.99	0.92	0.95	2143
1	0.57	0.91	0.70	259
accuracy			0.92	2402
macro avg	0.78	0.91	0.82	2402
weighted avg	0.94	0.92	0.92	2402

A comparison of the different performance measures of the two best models is given below:

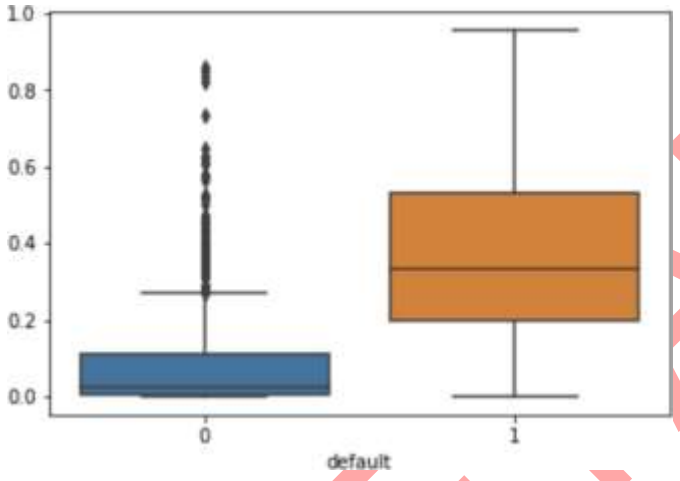
	Model A (Cutoff: 0.15)	Model B (Cutoff: 0.15)
Recall	0.799	0.911
Precision	0.363	0.566
Accuracy	0.83	0.92
F1-score	0.499	0.698

It is clearly seen that on the training dataset, Model B is superior to Model A.

4.7 Validate the Model on Test Dataset and state the performance matrices

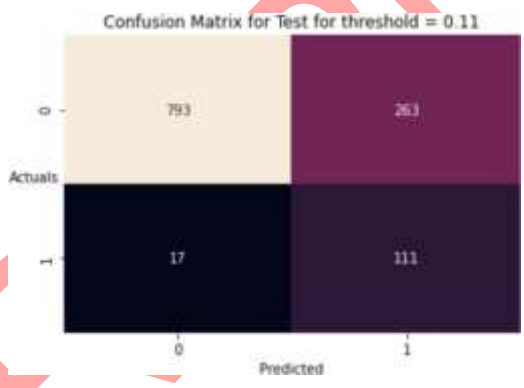
Model A:

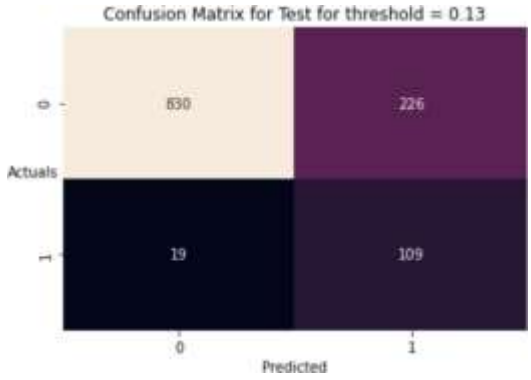
The boxplot of the predicted 'default' variable on the test dataset is shown below:

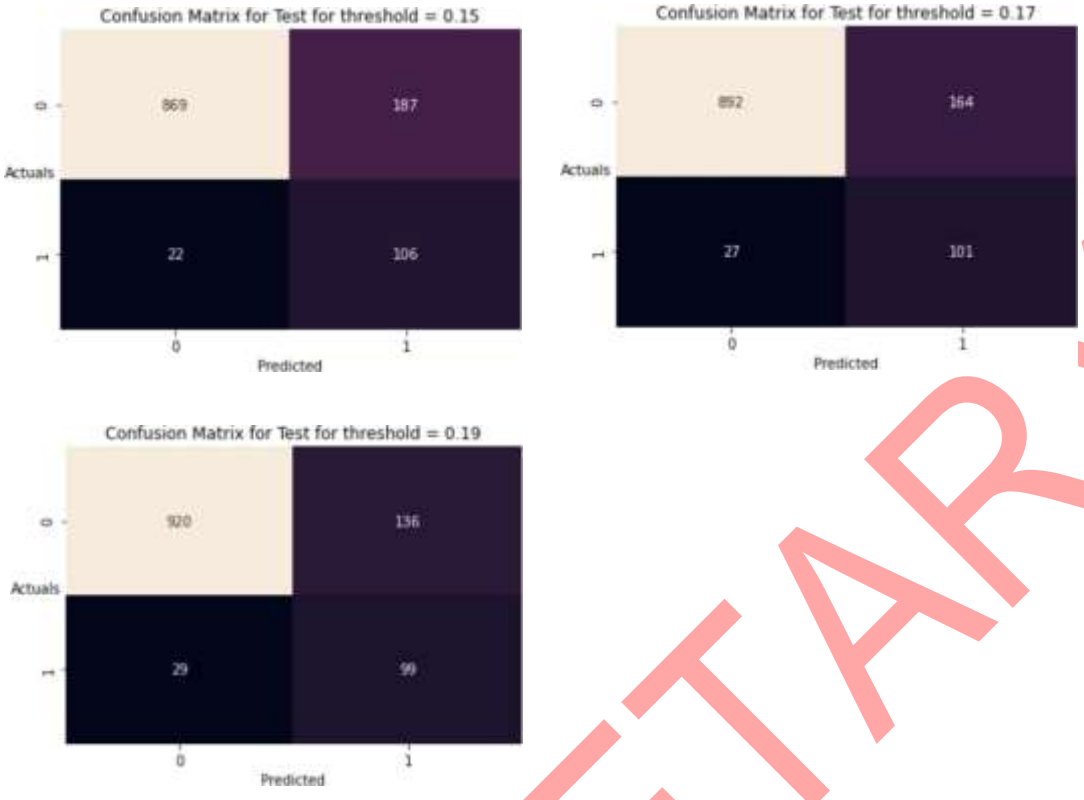


There is a clear separation between companies who default (0) and companies who do not (1).

The confusion matrices for Test Dataset with the different cutoff values are given below:







Their performance measures are summarized in the following table:

	Cutoff_0.11	Cutoff_0.13	Cutoff_0.15	Cutoff_0.17	Cutoff_0.19
Recall	0.867	0.852	0.828	0.789	0.773
F1 Score	0.442	0.471	0.504	0.514	0.545
Precision	0.297	0.325	0.362	0.381	0.421
Accuracy	0.760	0.790	0.820	0.840	0.860

The model at cutoff value of 0.15 is also fairly stable across the train and test datasets as seen below:

	Train_0.15	Test_0.15
Recall	0.799	0.828
F1 Score	0.499	0.504
Precision	0.363	0.362
Accuracy	0.83	0.820

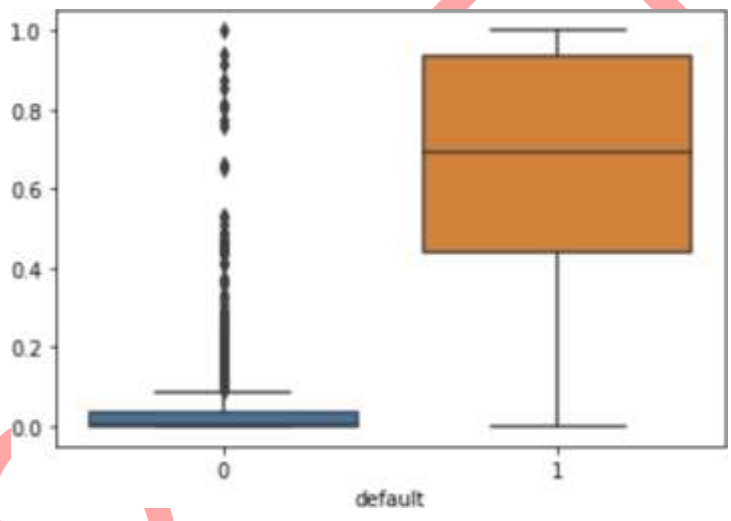
The classification report is given below:

Classification report for Test Dataset with cutoff value of 0.15

	precision	recall	f1-score	support
0	0.98	0.82	0.89	1056
1	0.36	0.83	0.50	128
accuracy			0.82	1184
macro avg	0.67	0.83	0.70	1184
weighted avg	0.91	0.82	0.85	1184

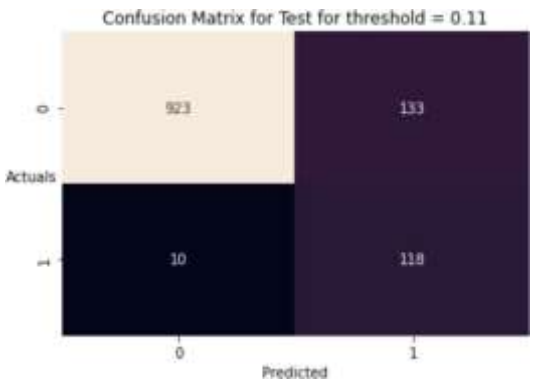
Model B:

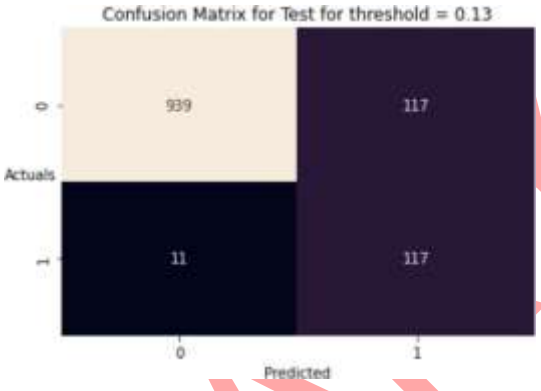
The boxplot of the predicted 'default' variable on the test dataset is shown below:

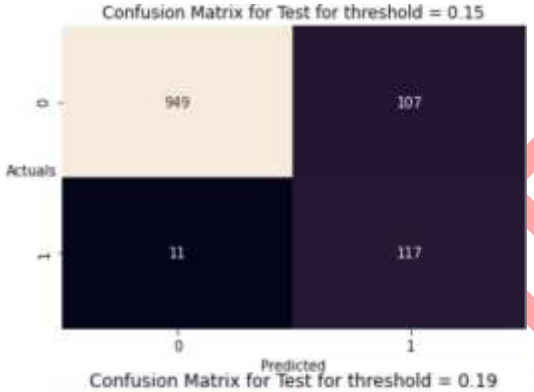


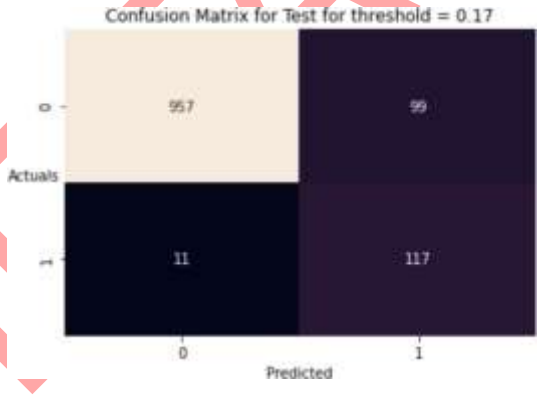
There is a clearer separation between companies who default (0) and companies who do not (1) as compared to Model A.

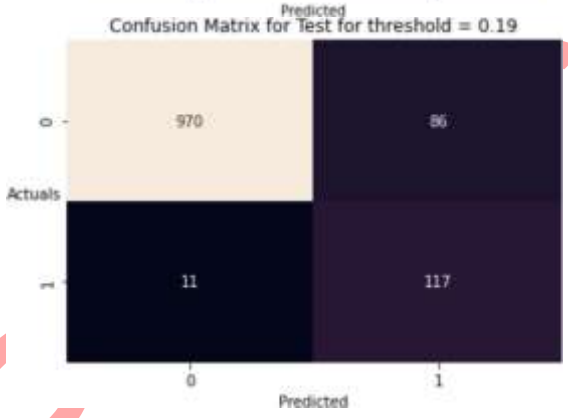
The confusion matrices for Test Dataset with the different cutoff values are given below:











The performance measures of Model B on the test dataset is given below:

	Cutoff_0.11	Cutoff_0.13	Cutoff_0.15	Cutoff_0.17	Cutoff_0.19
Recall	0.922	0.914	0.914	0.914	0.914
F1 Score	0.623	0.646	0.665	0.680	0.707
Precision	0.470	0.500	0.522	0.542	0.576
Accuracy	0.880	0.890	0.900	0.910	0.920

There is a slight improvement in Recall and slight reduction in Accuracy and Precision for Model B at cutoff value of 0.15 across the train and test datasets as seen below:

	Train_0.15	Test_0.15
Recall	0.911	0.914
F1 Score	0.698	0.665
Precision	0.566	0.522
Accuracy	0.92	0.900

However, it is not a major over-fitting problem, and the model can be considered to be fairly stable.

Another way to find Optimum Cutoff

The optimum cutoff is defined by

Opt cutoff = Max (tpr – fpr)

For Model B, which was the best model in Logistic Regression, the cutoff value is 0.282 as follows:

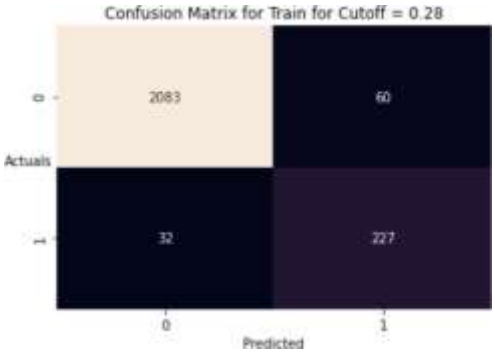
```
optimal_idx = np.argmax(tpr - fpr)
optimal_threshold = thresholds[optimal_idx]
print('The Optimal Cutoff Value is ', round(optimal_threshold,5))
```

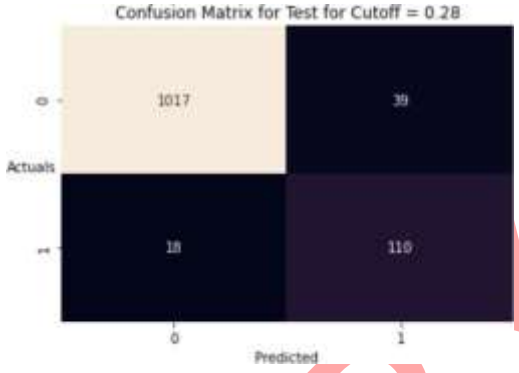
```
The Optimal Cutoff Value is  0.282
```

The performance of Model B with this cutoff is as shown below:

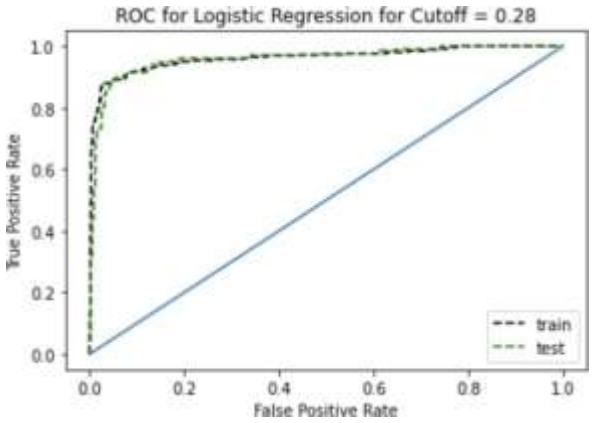
	Train_Cutoff_0.28	Test_Cutoff_0.28
Recall	0.876	0.859
F1 Score	0.832	0.794
Precision	0.791	0.738
Accuracy	0.960	0.950
auc	0.963	0.960

The confusion matrix for Train and Test dataset for cutoff = 0.28 is as follows:





The ROC curve on Train and test datasets is shown below:



The table below summarizes the comparison of Model B with the two cutoff values of 0.15 and the 'optimal' cutoff value of 0.28 as follows:

	Train_0.15	Test_0.15	Train_Cutoff_0.28	Test_Cutoff_0.28
Recall	0.911	0.914	0.876	0.859
F1 Score	0.698	0.665	0.832	0.794
Precision	0.566	0.522	0.791	0.738
Accuracy	0.920	0.900	0.960	0.950
auc	1.000	1.000	0.963	0.960

As expected, the Recall performance at Cutoff of 0.15 (0.914) is superior to Cutoff at 0.28 (0.859) on the Test dataset, and this is offset by higher F1 Score, Precision and Accuracy at Cutoff value of 0.28.

Conclusion

The two approaches used gave quite different results.

The second approach – **Model B** (where the model was built using all the independent variables and then dropping those with p values > 0.05) gave a better model than the first approach (**Model A**) (where the highly correlated independent variables using the VIF criterion were excluded and then the model is built).

Model B is also simpler than Model A with only 6 independent variables (excluding the intercept term) compared to 12 for Model A.

Moreover, there are only 2 independent variables common between the two models, with a third one very close (Book_Value_Unit_Curr in Model B and

Book_Value_Adj_Unit_Curr in Model A).