Financial Risk Analysis Business Report

Soni Kumari
itssonipsjha@gmail.com
PGPDSBA.O.FEB23.A

Contents

PART A

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 interest 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.

EDA

	Co_Code	Co_Name	_Operating_Expense_Rate	_Research_and_development_expense_rate	_Cash_flow_rate	_Interest_bearing_debt_interest_rate	_Tax_
0	16974	Hind.Cables	8820000000.00	0.00	0.46	0.00	
1	21214	Tata Tele. Mah.	9380000000.00	4230000000.00	0.46	0.00	
2	14852	ABG Shipyard	3800000000.00	815000000.00	0.45	0.00	
3	2439	GTL	6440000000.00	0.00	0.46	0.00	
4	23505	Bharati Defence	3680000000.00	0.00	0.46	0.00	

5 rows × 58 columns

Figure 1.1 Top 5 rows of the dataset

The number of rows (observations) is **2058**
The number of columns (variables) is **58**

```
count   2058.00
mean       0.11
std        0.31
min        0.00
25%        0.00
50%        0.00
75%        0.00
max        1.00
Name: Default, dtype: float64
```

Figure 1.2 Summary Statistics of Default Variable

#	Column	Non-Null Count	Dtype
0	Co_Code	2058 non-null	int64
1	Co_Name	2058 non-null	object
2	Operating_Expense_Rate	2058 non-null	float64
3	Research_and_development_expense_rate	2058 non-null	float64
4	Cash_flow_rate	2058 non-null	float64
5	Interest_bearing_debt_interest_rate	2058 non-null	float64
6	Tax_rate_A	2058 non-null	float64
7	Cash_Flow_Per_Share	1891 non-null	float64
8	Per_Share_Net_profit_before_tax_Yuan	2058 non-null	float64
9	Realized_Sales_Gross_Profit_Growth_Rate	2058 non-null	float64
10	Operating_Profit_Growth_Rate	2058 non-null	float64
11	Continuous_Net_Profit_Growth_Rate	2058 non-null	float64
12	Total_Asset_Growth_Rate	2058 non-null	float64
13	Net_Value_Growth_Rate	2058 non-null	float64
14	Total_Asset_Return_Growth_Rate_Ratio	2058 non-null	float64
15	Cash_Reinvestment_perc	2058 non-null	float64
16	Current_Ratio	2058 non-null	float64
17	Quick_Ratio	2058 non-null	float64
18	Interest_Expense_Ratio	2058 non-null	float64
19	Total_debt_to_Total_net_worth	2037 non-null	float64
20	Long_term_fund_suitability_ratio_A	2058 non-null	float64
21	Net_profit_before_tax_to_Paid_in_capital	2058 non-null	float64
22	Total_Asset_Turnover	2058 non-null	float64
23	Accounts_Receivable_Turnover	2058 non-null	float64
24	Average_Collection_Days	2058 non-null	float64
25	Inventory_Turnover_Rate_times	2058 non-null	float64
26	Fixed_Assets_Turnover_Frequency	2058 non-null	float64
27	Net_Worth_Turnover_Rate_times	2058 non-null	float64
28	Operating_profit_per_person	2058 non-null	float64
29	Allocation_rate_per_person	2058 non-null	float64
30	Quick_Assets_to_Total_Assets	2058 non-null	float64
31	Cash_to_Total_Assets	1962 non-null	float64
32	Quick_Assets_to_Current_Liability	2058 non-null	float64
33	Cash_to_Current_Liability	2058 non-null	float64
34	Operating_Funds_to_Liability	2058 non-null	float64
35	Inventory_to_Working_Capital	2058 non-null	float64
36	Inventory_to_Current_Liability	2058 non-null	float64
37	Long_term_Liability_to_Current_Assets	2058 non-null	float64
38	Retained_Earnings_to_Total_Assets	2058 non-null	float64
39	Total_income_to_Total_expense	2058 non-null	float64
40	Total_expense_to_Assets	2058 non-null	float64
41	Current_Asset_Turnover_Rate	2058 non-null	float64
42	Quick_Asset_Turnover_Rate	2058 non-null	float64
43	Cash_Turnover_Rate	2058 non-null	float64
44	Fixed_Assets_to_Assets	2058 non-null	float64
45	Cash_Flow_to_Total_Assets	2058 non-null	float64
46	Cash_Flow_to_Liability	2058 non-null	float64
47	CFO_to_Assets	2058 non-null	float64
48	Cash_Flow_to_Equity	2058 non-null	float64
49	Current_Liability_to_Current_Assets	2044 non-null	float64
50	Liability_Assets_Flag	2058 non-null	int64
51	Total_assets_to_GNP_price	2058 non-null	float64
52	No_credit_Interval	2058 non-null	float64
53	Degree_of_Financial_Leverage_DFL	2058 non-null	float64
54	Interest_Coverage_Ratio_Interest_expense_to_EBIT	2058 non-null	float64
55	Equity_to_Liability	2058 non-null	float64
56	Default	2058 non-null	int64

dtypes: float64(53), int64(3), object(1)

Figure 1.3 - Data Variables and types

Types of Data: The dataset encompasses three primary data categories:

Integer Data: This comprises four columns, namely **"Co_Code," "_Liability_Assets_Flag," "_Net_Income_Flag," and "Default,"** all of which contain integer values.
Floating-Point Data: The majority of the dataset consists of floating-point values, present in various columns representing financial ratios and metrics. These columns involve decimal values.
Object Data: Singularly, there is one column named **"Co_Name,"** presumably holding the names of the companies.

Missing Data: Some columns exhibit missing values, denoted by the **"Non-Null Count"** in the summary. For instance, **"_Cash_Flow_Per_Share" and others like "_Total_debt_to_Total_net_worth," "_Cash_to_Total_Assets," and "_Current_Liability_to_Current_Assets"** contain missing values.

Numeric Data: The dataset predominantly comprises numeric data, with 53 columns containing floating-point values. These columns likely depict diverse financial and operational metrics of the companies.

Categorical Data: Certain integer columns, like "**_Liability_Assets_Flag" and "_Net_Income_Flag,"** may function as binary indicators or flags, constituting categorical data.

Target Variable: The "Default" column appears to serve as the target variable, as indicated in the problem statement. It is of integer type, hinting at a binary variable that signifies whether a company defaulted (1) or not (0).

Company Names: The **"Co_Name"** column presumably holds the names of the companies in the dataset. While valuable for identification, it may not be directly utilized in predictive modeling.

Prior to initiating any analysis or modeling, it is imperative to address missing values, potentially encode categorical variables, and investigate the relationships between independent variables and the target variable. Moreover, comprehending the context of financial ratios and metrics within the dataset is crucial for a meaningful analysis.

```
0    1838
1     220
Name: Default, dtype: int64
```

Figure 1.4

The "**Default**" variable within the dataset exhibits two discrete values, and their respective frequencies are outlined below:

For Default value 0: There are 1,838 occurrences indicating instances where the company did not default.

For Default value 1: There are 220 occurrences indicating instances where the company did default.

This information presents an overview of the distribution between default and non-default cases in the dataset. It's noteworthy that datasets with such imbalances (where one class significantly outweighs the other) can pose challenges during machine learning model training and evaluation. Depending on the objectives of the analysis, it may be necessary to explore techniques such as resampling, utilizing different evaluation metrics, or employing advanced modeling approaches to appropriately address the class imbalance.

Proportion of missing values is 0.25403645167340116

PART A: Outlier Treatment

	count	mean	std	min	25%	50%	75%	max
Co_Code	2058.00	17572.11	21892.89	4.00	3674.00	6240.00	24280.75	72493.00
Operating_Expense_Rate	2058.00	2052388835.76	3252623690.29	0.00	0.00	0.00	4110000000.00	9980000000.00
Research_and_development_expense_rate	2058.00	1208634256.56	2144568158.08	0.00	0.00	0.00	1550000000.00	9980000000.00
Cash_flow_rate	2058.00	0.47	0.02	0.00	0.46	0.46	0.47	1.00
Interest_bearing_debt_interest_rate	2058.00	11130223.52	90425949.04	0.00	0.00	0.00	0.00	990000000.00
Tax_rate_A	2058.00	0.11	0.15	0.00	0.00	0.04	0.22	1.00
Cash_Flow_Per_Share	1891.00	0.32	0.02	0.17	0.31	0.32	0.33	0.46
Per_Share_Net_profit_before_tax_Yuan	2058.00	0.18	0.03	0.00	0.17	0.18	0.19	0.79
Realized_Sales_Gross_Profit_Growth_Rate	2058.00	0.02	0.02	0.00	0.02	0.02	0.02	1.00
Operating_Profit_Growth_Rate	2058.00	0.85	0.00	0.74	0.85	0.85	0.85	1.00
Continuous_Net_Profit_Growth_Rate	2058.00	0.22	0.01	0.00	0.22	0.22	0.22	0.23
Total_Asset_Growth_Rate	2058.00	5287663257.05	2912614769.58	0.00	4315000000.00	6225000000.00	7220000000.00	9980000000.00
Net_Value_Growth_Rate	2058.00	5189504.37	207791797.86	0.00	0.00	0.00	0.00	9330000000.00
Total_Asset_Return_Growth_Rate_Ratio	2058.00	0.26	0.00	0.25	0.26	0.26	0.26	0.36
Cash_Reinvestment_perc	2058.00	0.38	0.03	0.03	0.37	0.38	0.39	1.00
Current_Ratio	2058.00	1336248.80	60619173.20	0.00	0.01	0.01	0.01	2750000000.00
Quick_Ratio	2058.00	27755102.05	444865390.47	0.00	0.00	0.01	0.01	9230000000.00
Interest_Expense_Ratio	2058.00	0.63	0.01	0.53	0.63	0.63	0.63	0.81
Total_debt_to_Total_net_worth	2037.00	10714285.73	269696017.59	0.00	0.00	0.01	0.01	9940000000.00
Long_term_fund_suitability_ratio_A	2058.00	0.01	0.03	0.00	0.01	0.01	0.01	1.00
Net_profit_before_tax_to_Paid_in_capital	2058.00	0.18	0.03	0.00	0.17	0.17	0.18	0.79
Total_Asset_Turnover	2058.00	0.13	0.10	0.00	0.06	0.10	0.17	0.92
Accounts_Receivable_Turnover	2058.00	41598639.46	504767266.59	0.00	0.00	0.00	0.00	9740000000.00
Average_Collection_Days	2058.00	26297862.01	410996733.83	0.00	0.00	0.01	0.01	8800000000.00
Inventory_Turnover_Rate_times	2058.00	2030227259.48	3077250265.27	0.00	0.00	19100000.00	3815000000.00	9990000000.00
Fixed_Assets_Turnover_Frequency	2058.00	1230897959.18	2649288936.44	0.00	0.00	0.00	0.01	9990000000.00
Net_Worth_Turnover_Rate_times	2058.00	0.04	0.04	0.01	0.02	0.03	0.04	1.00
Operating_profit_per_person	2058.00	0.40	0.05	0.00	0.39	0.40	0.40	1.00
Allocation_rate_per_person	2058.00	5725558.82	197949961.06	0.00	0.00	0.01	0.02	8280000000.00
Quick_Assets_to_Total_Assets	2058.00	0.34	0.21	0.00	0.17	0.31	0.48	0.99
Cash_to_Total_Assets	1982.00	0.08	0.10	0.00	0.02	0.05	0.10	0.93
Quick_Assets_to_Current_Liability	2058.00	11904761.91	312292270.93	0.00	0.00	0.01	0.01	8820000000.00
Cash_to_Current_Liability	2058.00	92825072.90	785189881.95	0.00	0.00	0.00	0.01	9170000000.00
Operating_Funds_to_Liability	2058.00	0.35	0.04	0.03	0.34	0.35	0.35	1.00
Inventory_to_Working_Capital	2058.00	0.28	0.02	0.00	0.28	0.28	0.28	1.00
Inventory_to_Current_Liability	2058.00	57863459.68	627879536.23	0.00	0.00	0.01	0.01	9600000000.00
Long_term_Liability_to_Current_Assets	2058.00	73401069.01	669352618.01	0.00	0.00	0.00	0.01	9310000000.00

Figure 1.5 - Basic measures of descriptive statistics for the continuous variables

In this summary, the provided snippet computes upper and lower outlier thresholds for each column by utilizing the **Interquartile Range (IQR)**. Subsequently, any data points falling outside these thresholds are replaced with NaN values. This approach, frequently employed for managing outliers, serves to mitigate the impact of extreme values on the analysis or modeling process. It's essential to note that this method doesn't completely eliminate outliers; instead, it conceals them by assigning NaN values to their positions.

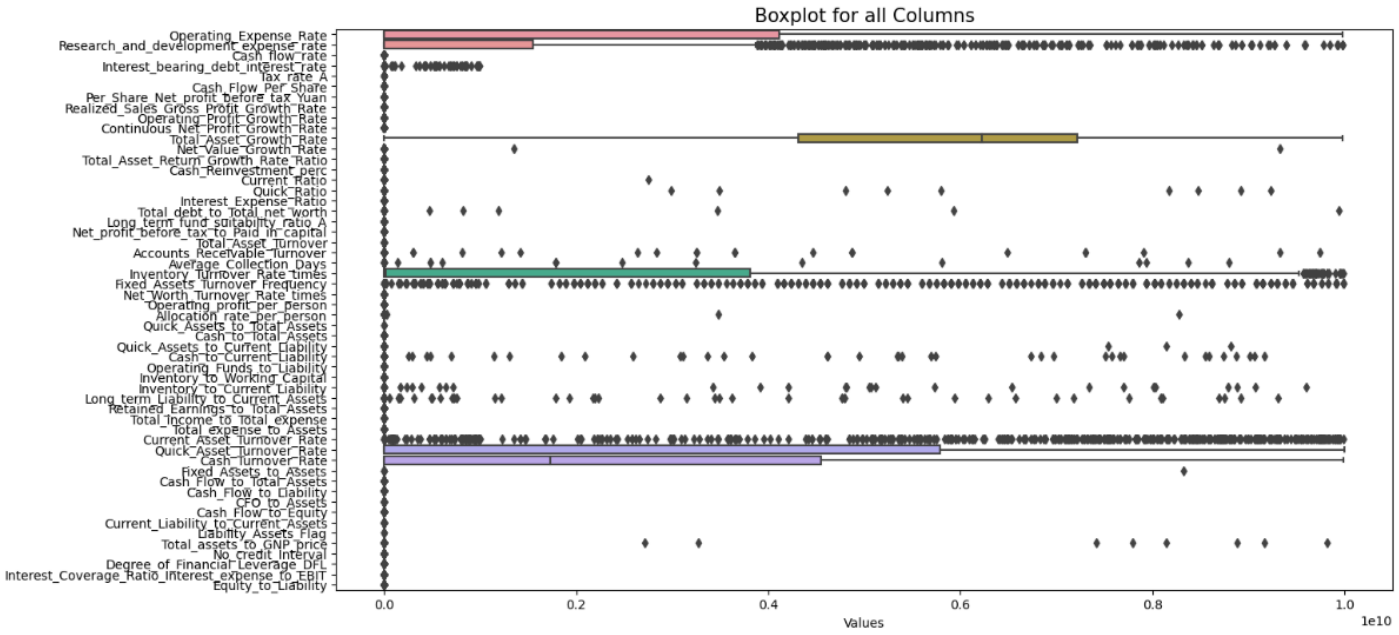

Figure 1.6 - Boxplot for all Columns

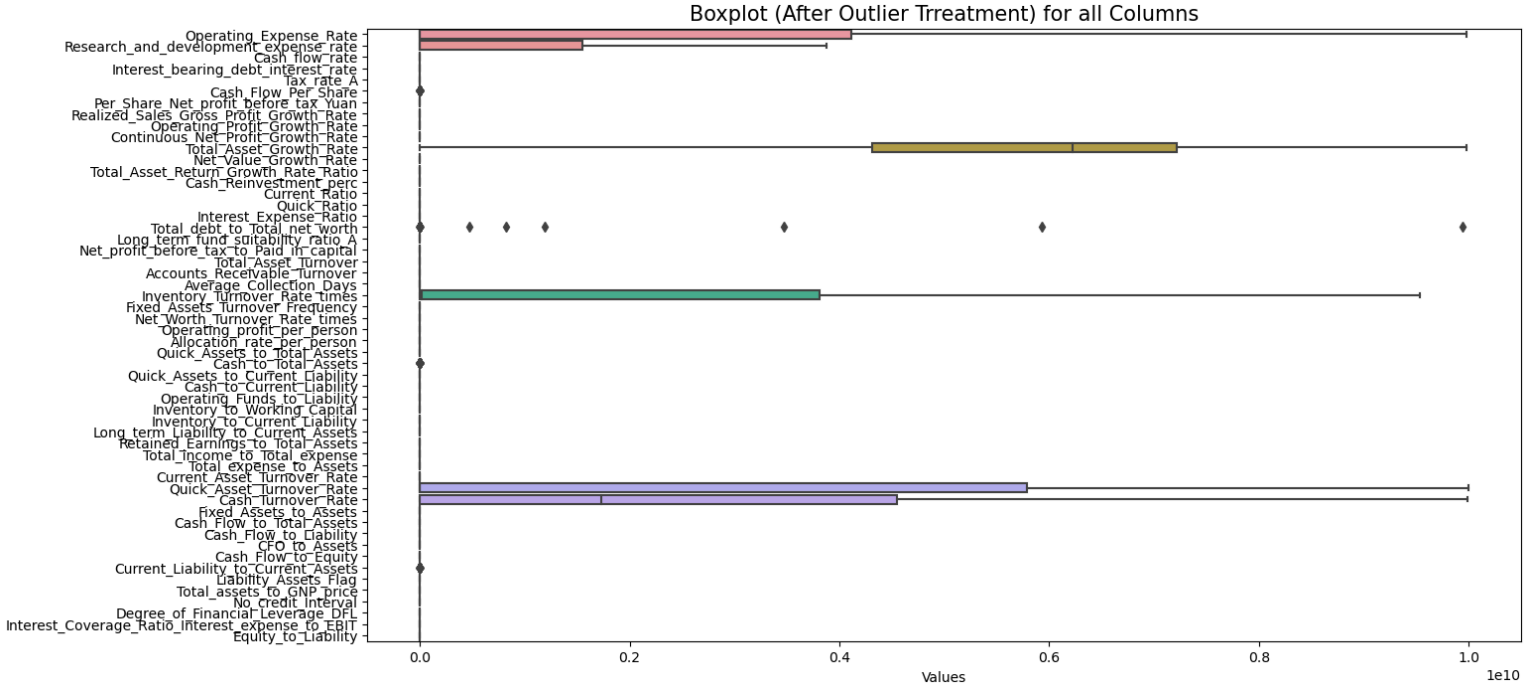

Figure 1.7 - Boxplot (After Outlier Treatment) for all Columns

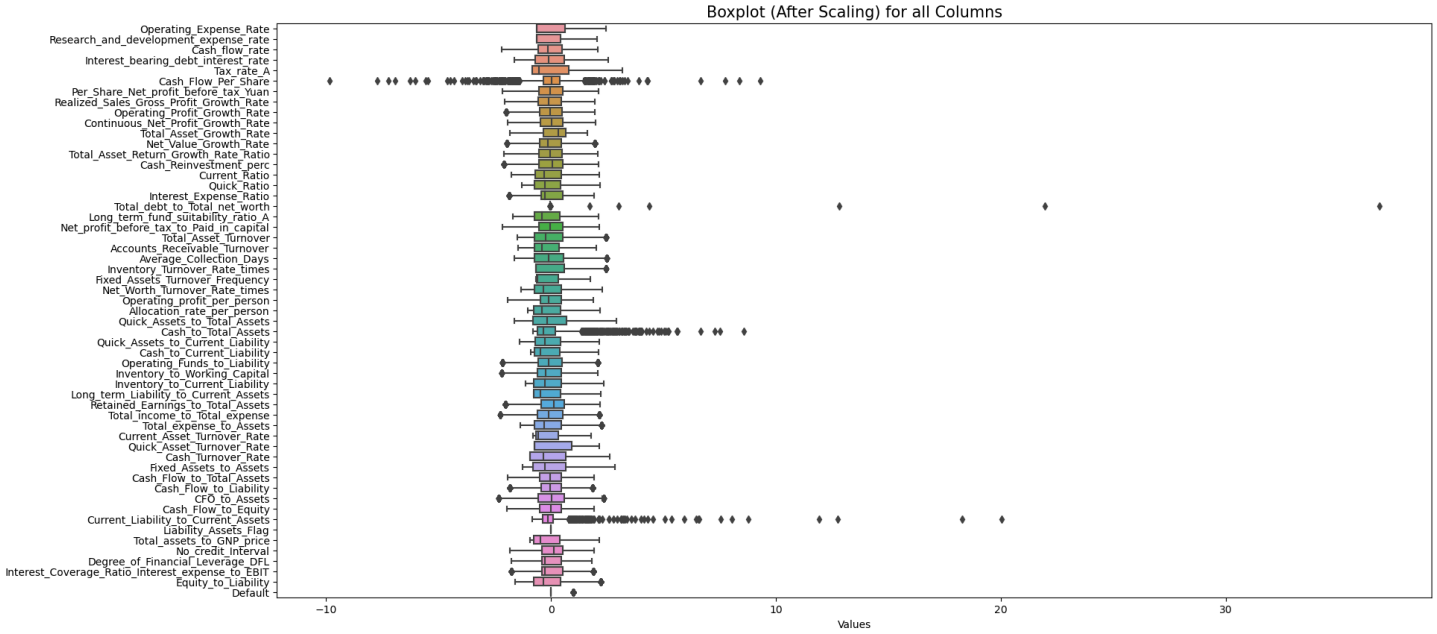

Figure 1.8 - Boxplot (After Scaling) for all Columns

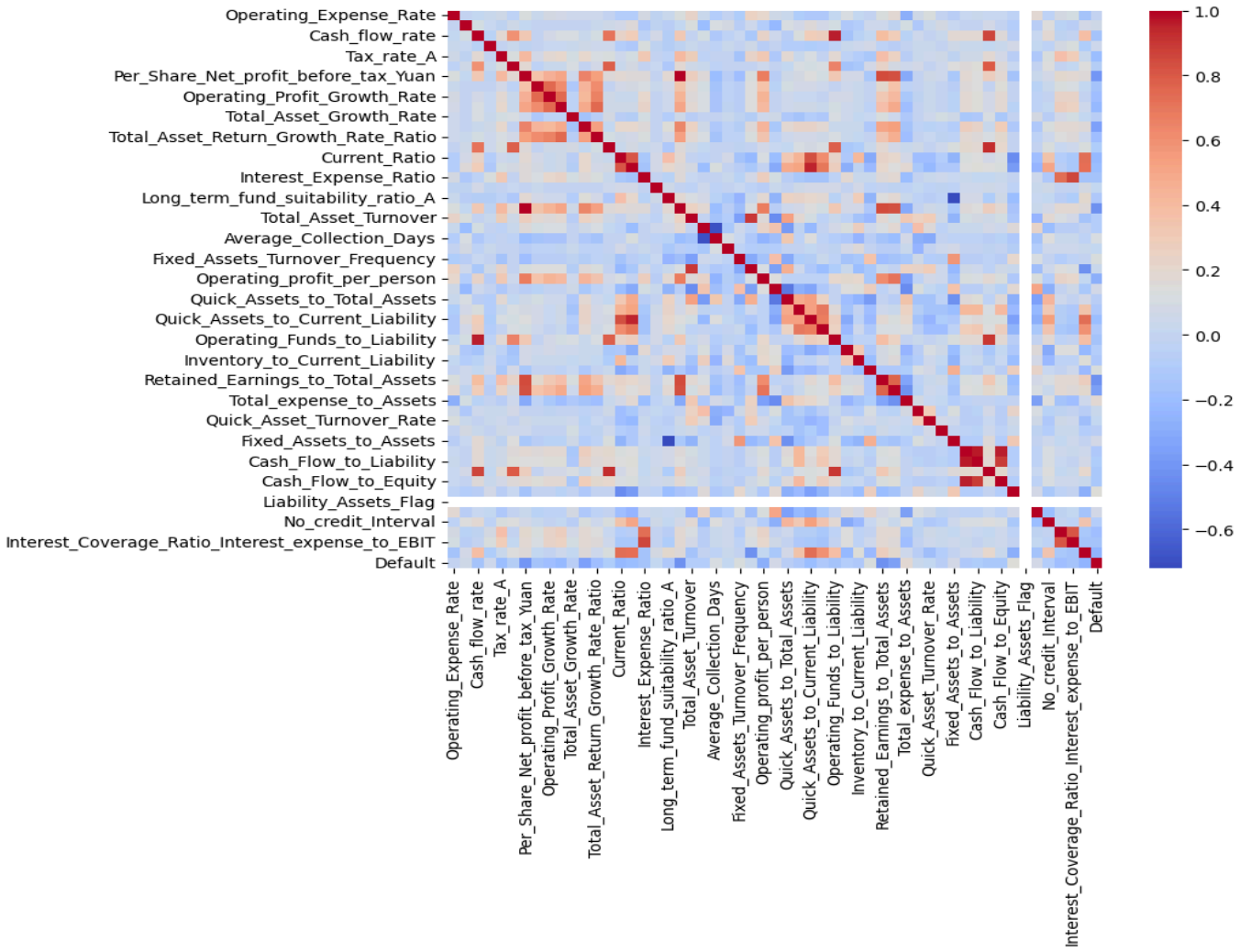

Figure 1.9- sns heatmap before standardization

PART A: Missing Value Treatment

```
Cash_Flow_Per_Share                  167
Total_debt_to_Total_net_worth         21
Cash_to_Total_Assets                  96
Current_Liability_to_Current_Assets   14
dtype: int64
```

Figure 1.10 - Missing Values check

_Cash_Flow_Per_Share: This specific column exhibits the most significant proportion of missing values, representing around 167 instances of missing data in the dataset.
_Total_debt_to_Total_net_worth: Ranking as the third-highest, this column contains roughly 21 instances of missing values.

_Current_Liability_to_Current_Assets: Occupying the second position, this column displays the second-highest proportion of missing values, totaling approximately 96 instances.

Upon identifying these columns with a notable prevalence of missing values (approximately 20%), the decision is to exclude the top three columns. This standard data preprocessing step aims to handle missing data effectively and reduce the dataset's dimensionality when certain columns exhibit a considerable number of missing values.

Figure 1.11- Visually representing the missing values in the data

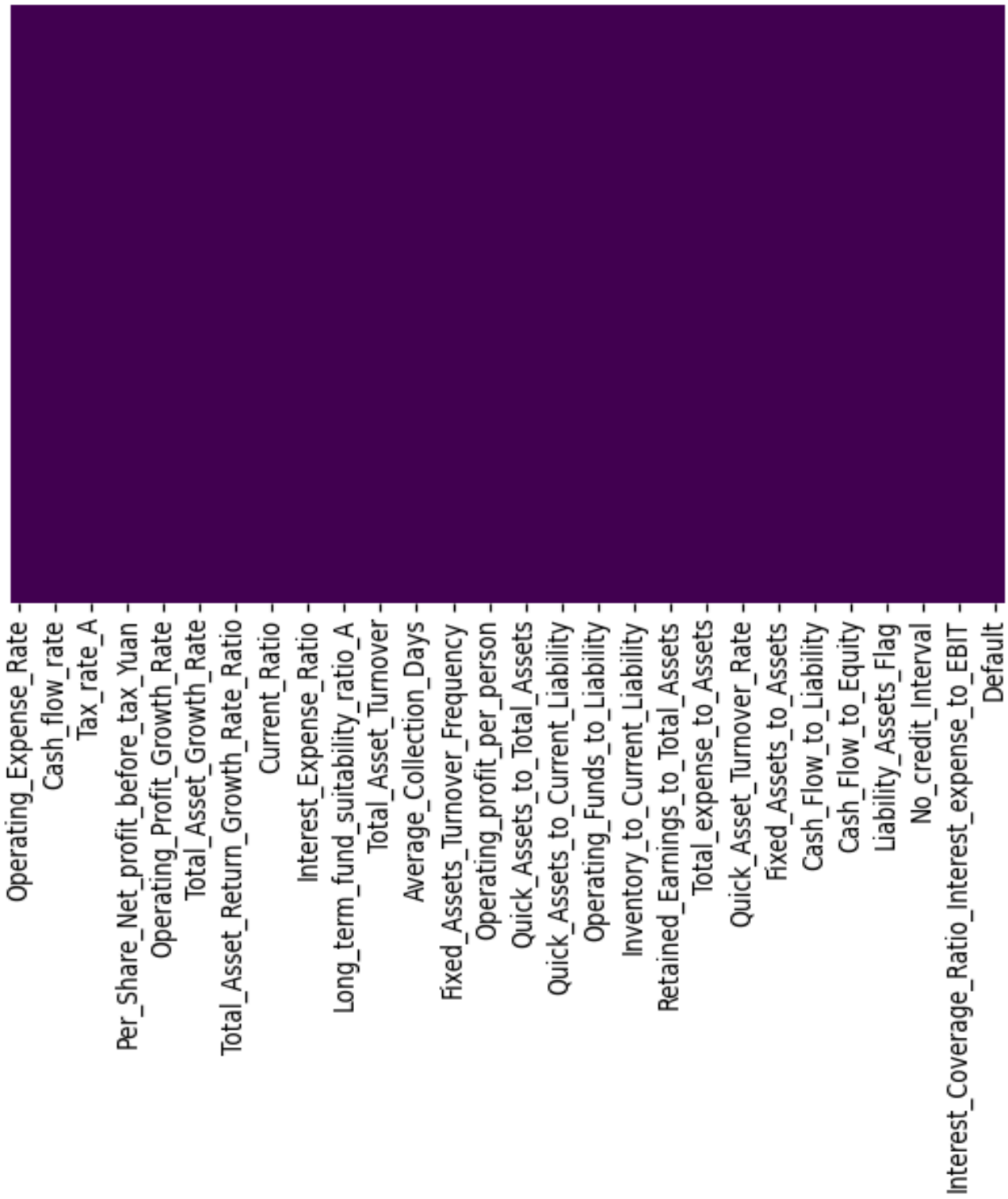

Figure 1.12- Visualising missing values in dataset after imputing

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

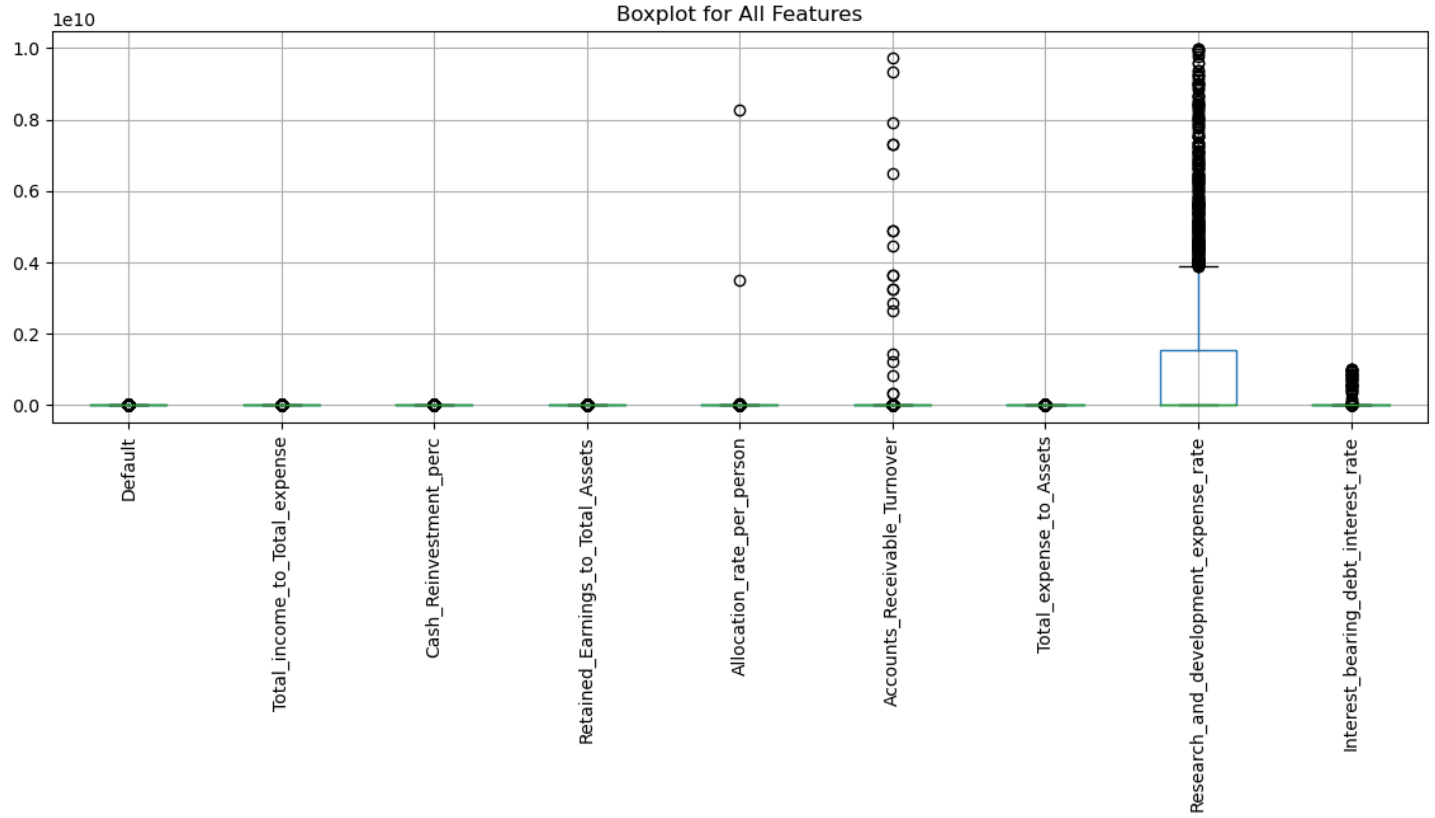

Figure 1.13- Boxplot for All Features

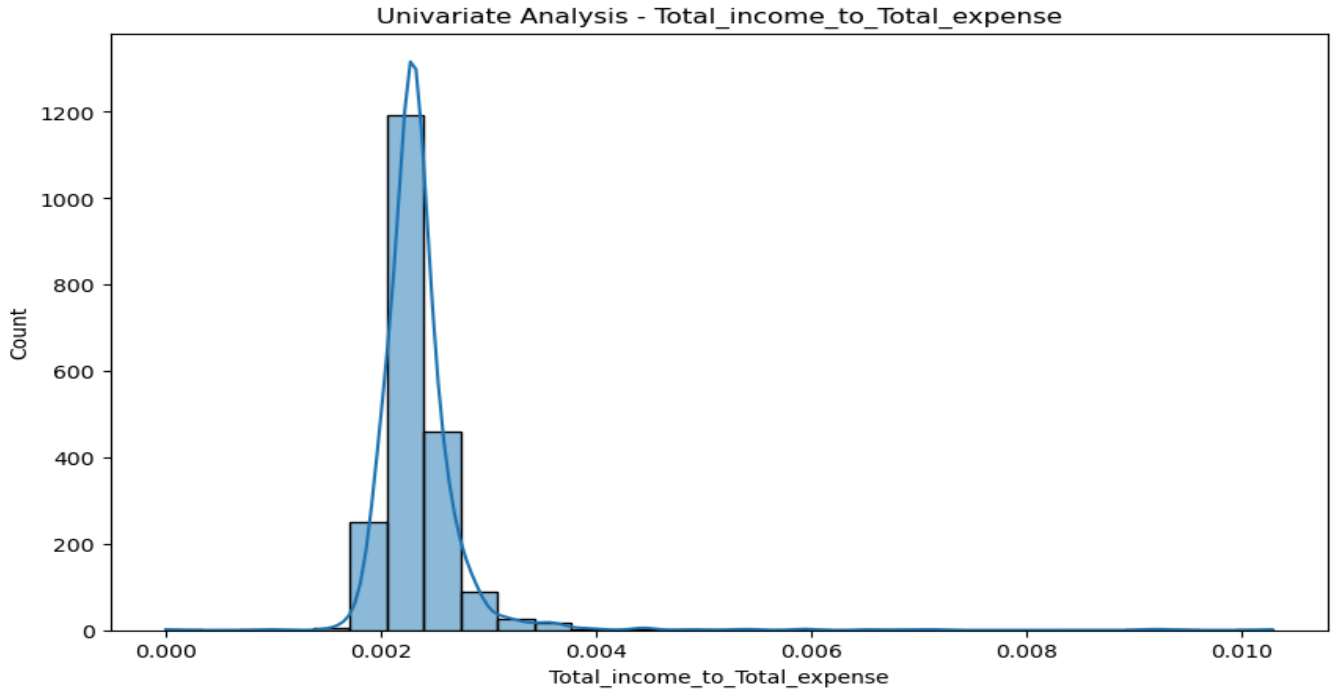

Figure 1.14- Univariate Analysis - Total Income to Total Expense

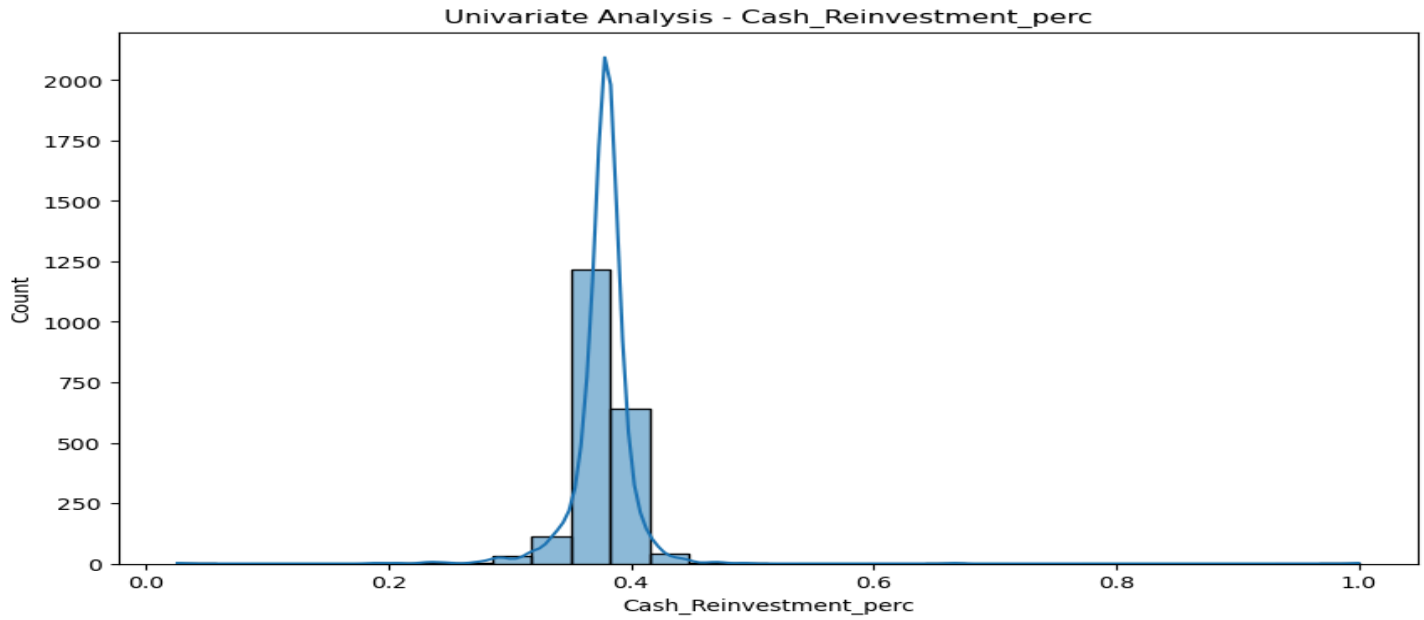

Figure 1.15- Univariate Analysis - Cash Reinvestment perc

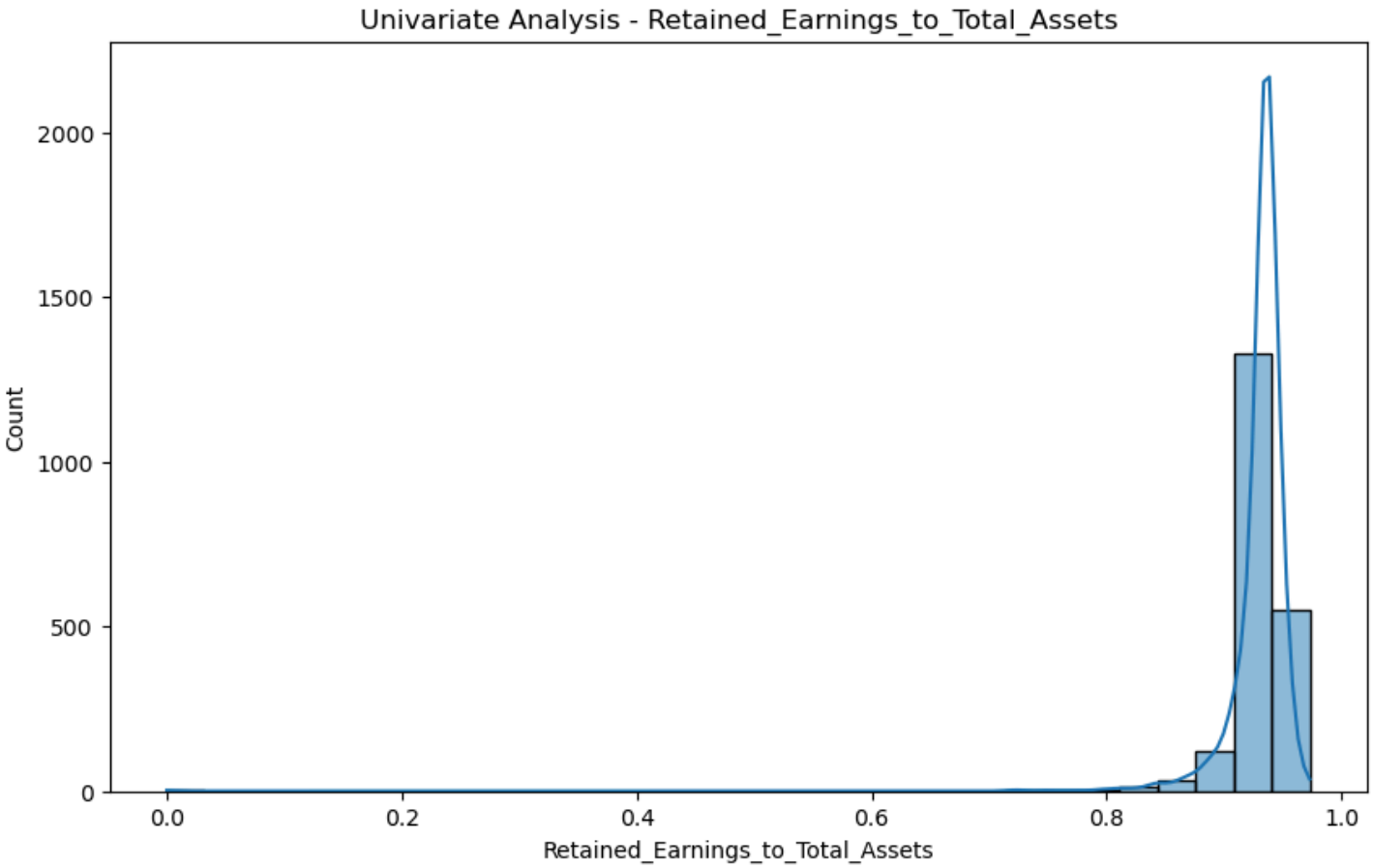

Figure 1.16- Univariate Analysis - Retained Earnings to total assets

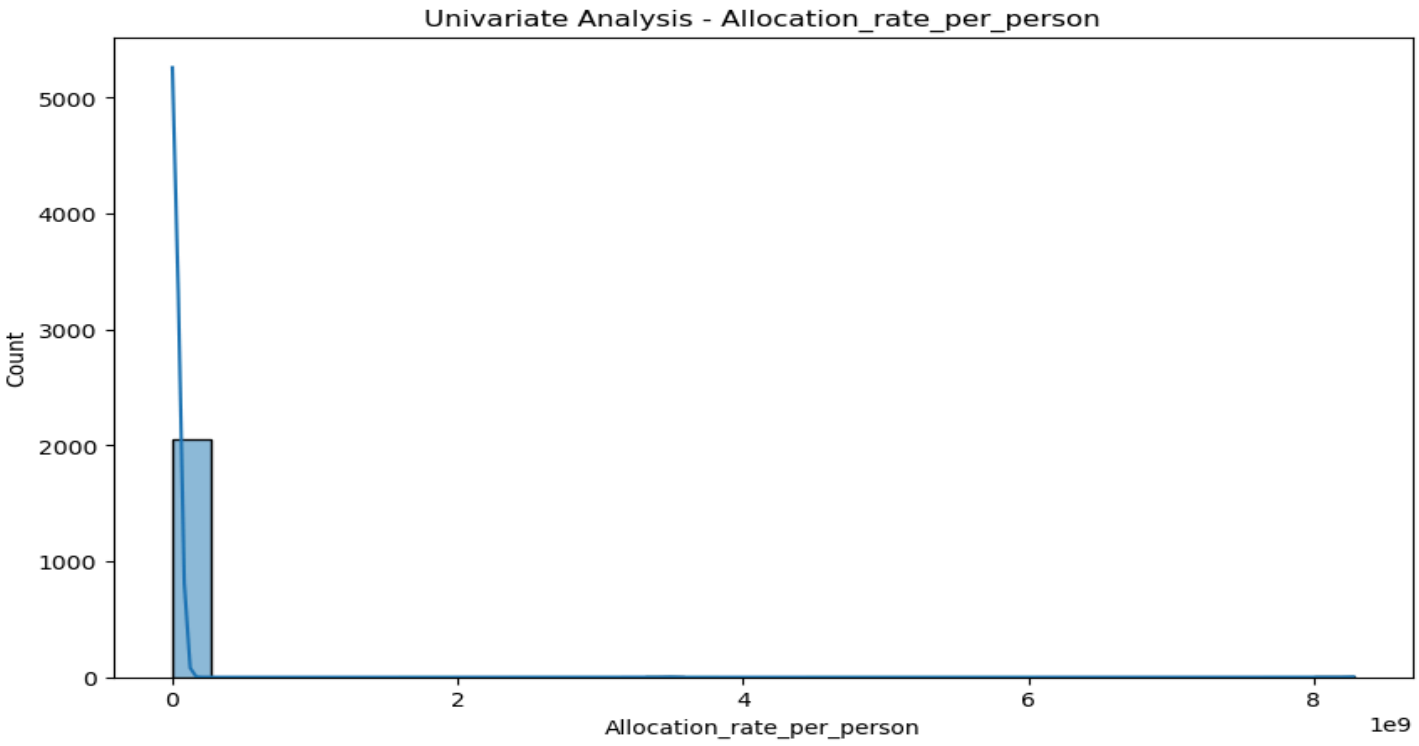

Figure 1.17- Univariate Analysis - Allocation rate per person

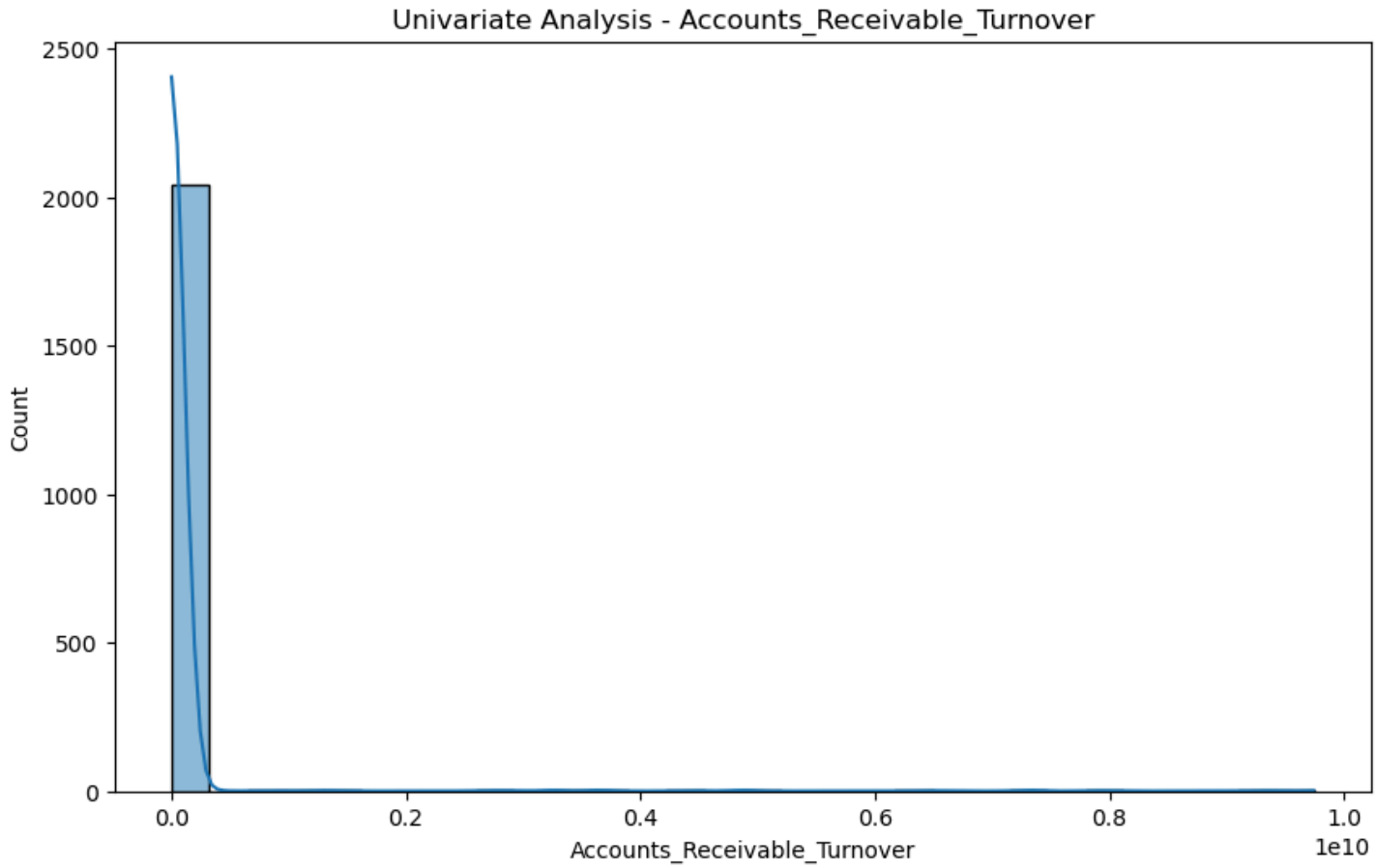

Figure 1.18- Univariate Analysis - Account Receivable Turnover

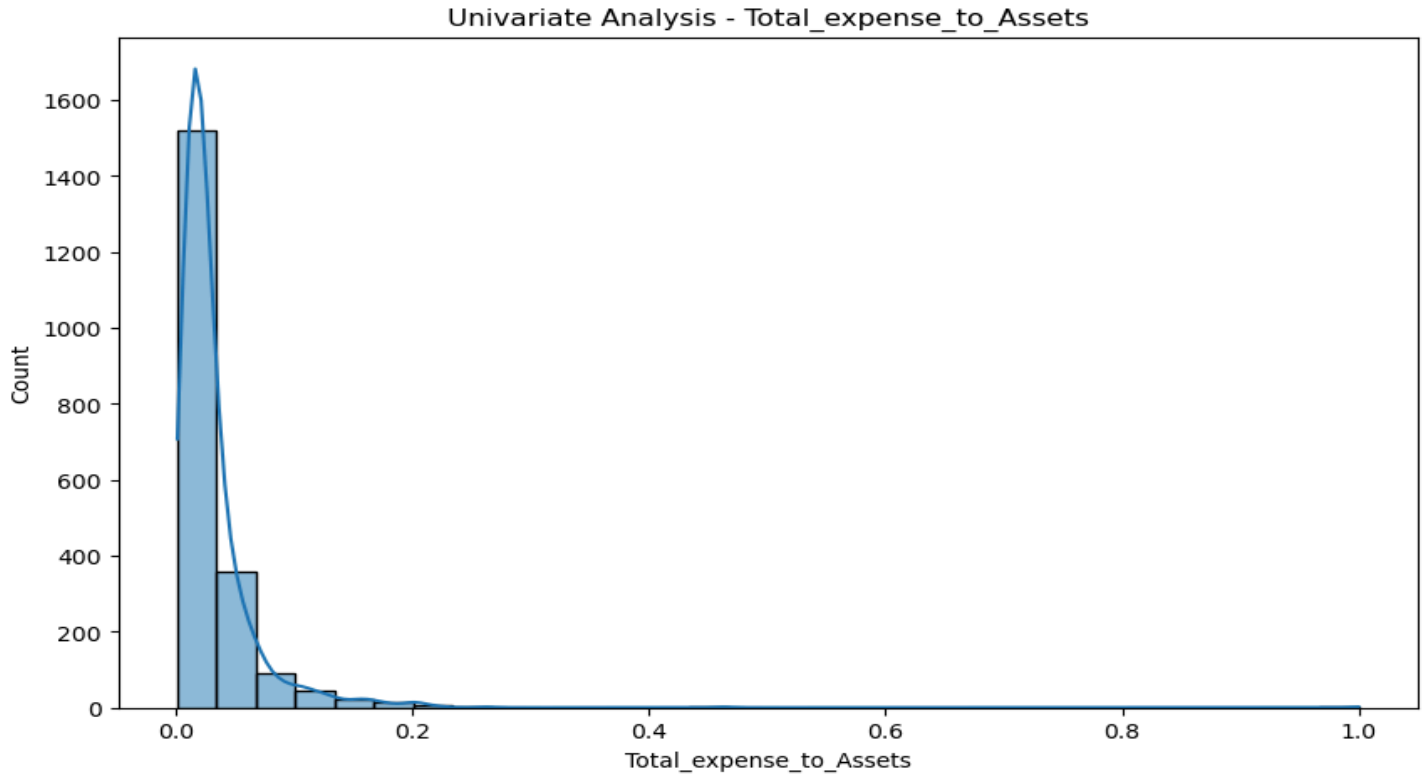

Figure 1.19- Univariate Analysis - Total Expense to Assets

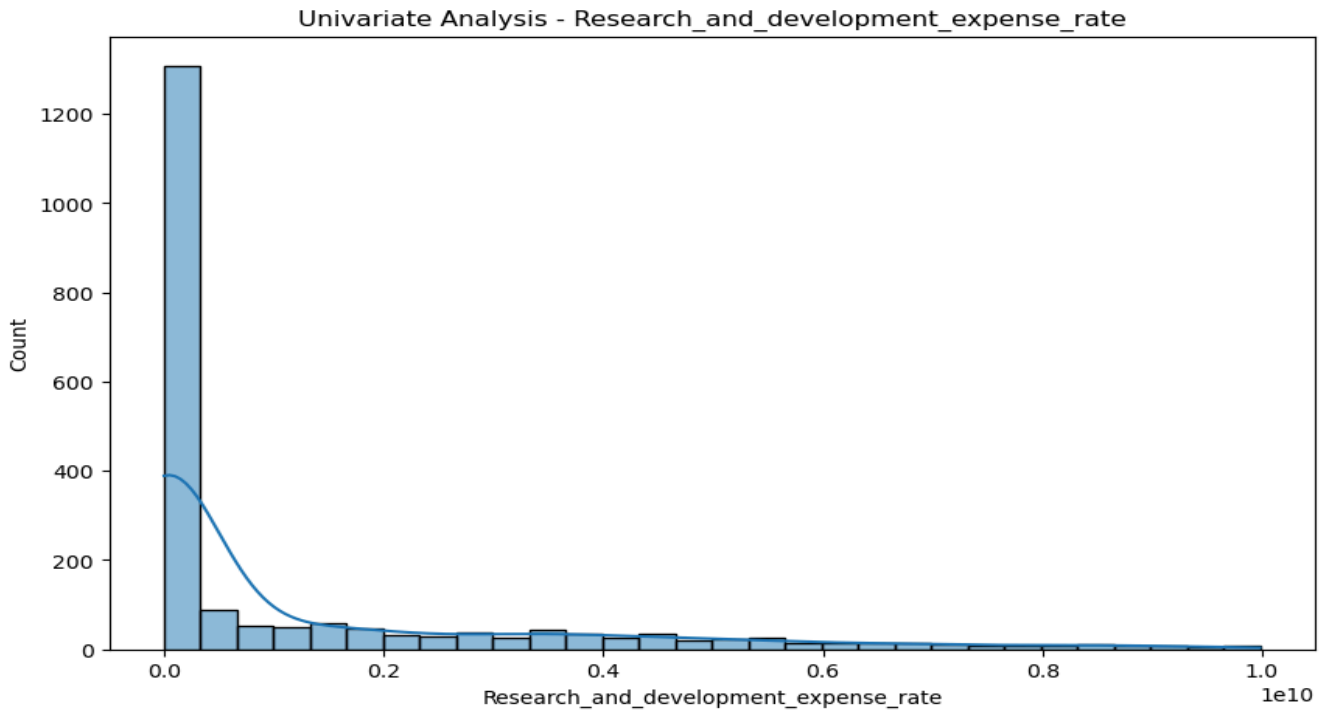

Figure 1.20- Univariate Analysis - Research and development expense rate

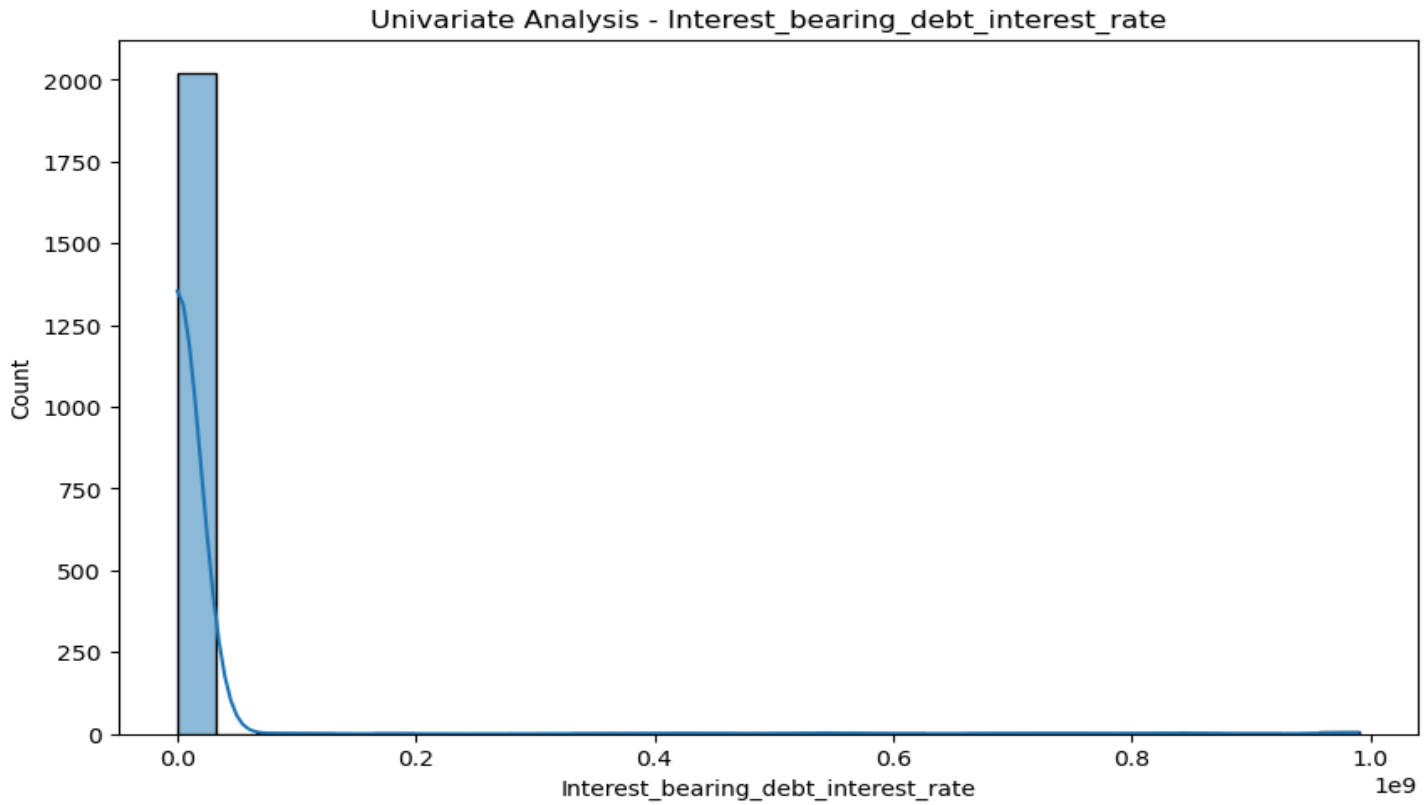

Figure 1.21- Univariate Analysis - Interest bearing debt interest rate

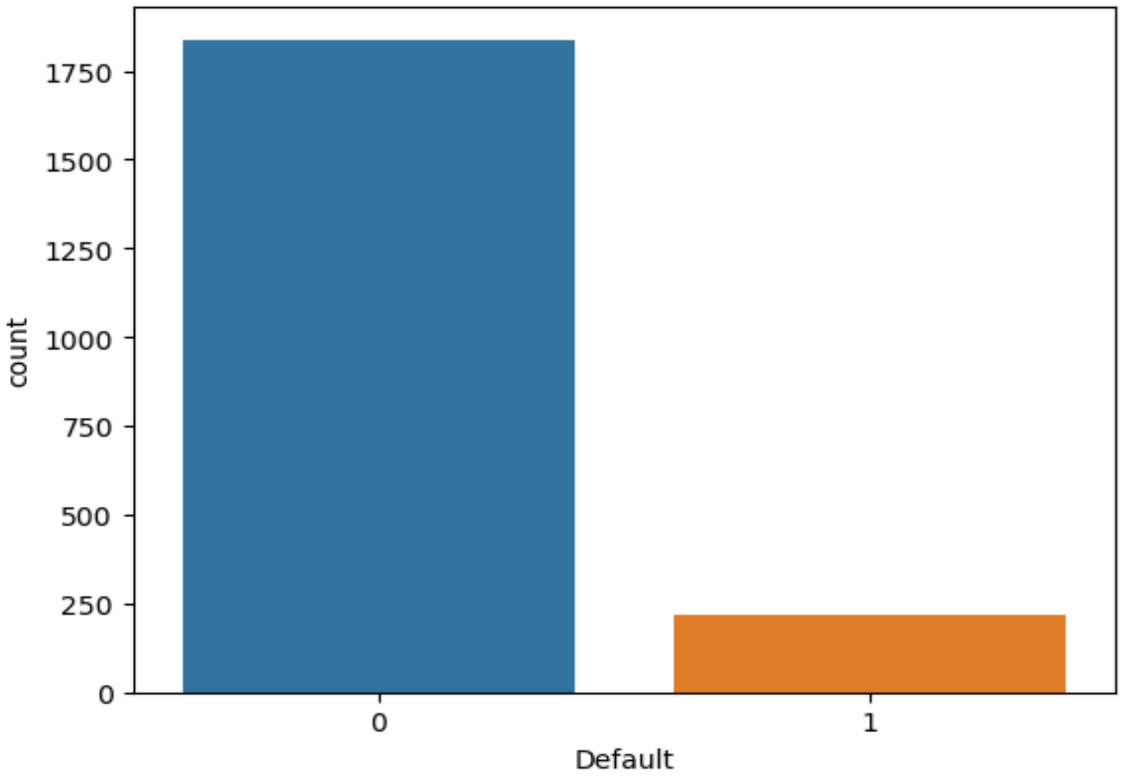

Figure 1.22- Countplot of Default Variable

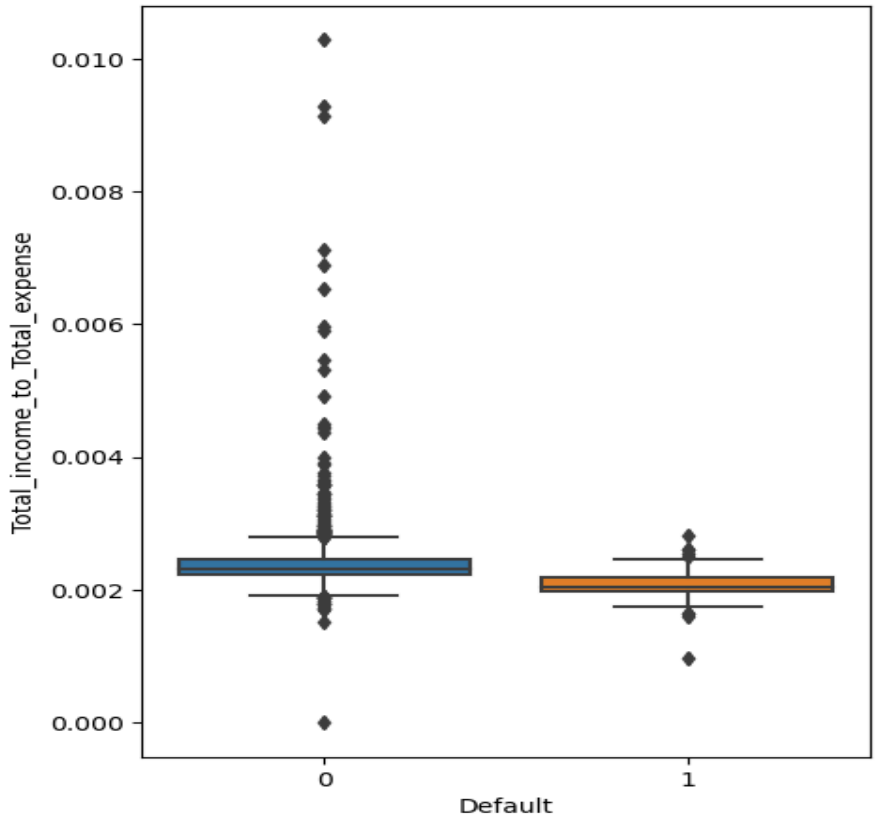

Figure 1.23- Boxplot of Total_income_to_Total_expense

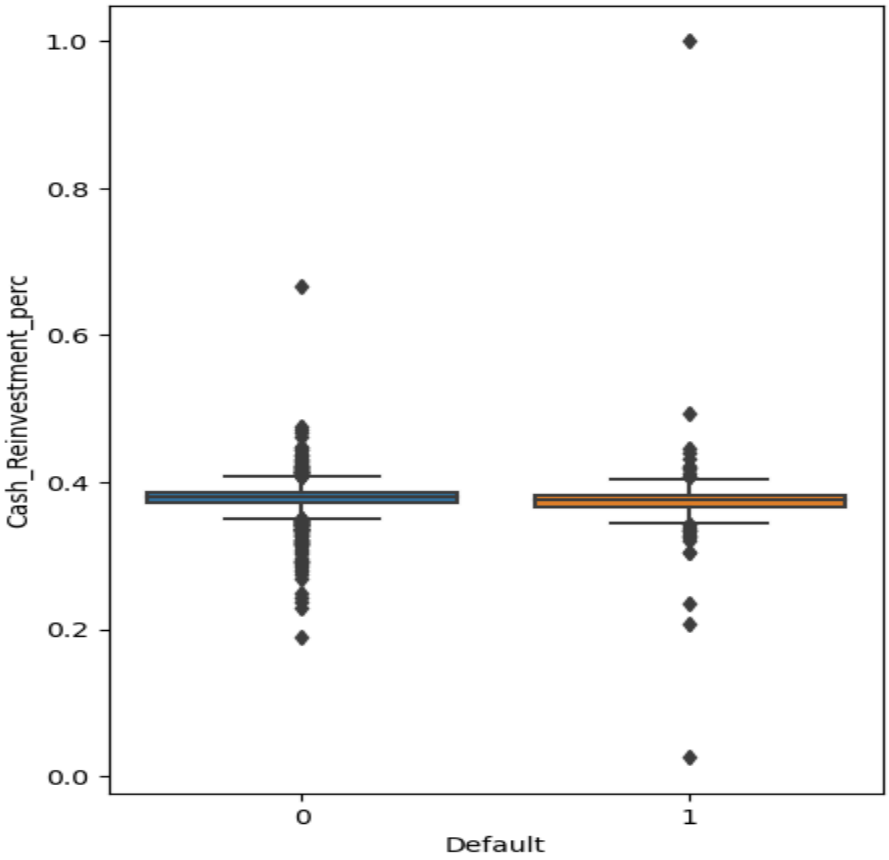

Figure 1.24- Boxplot of Cash_Reinvestment_perc

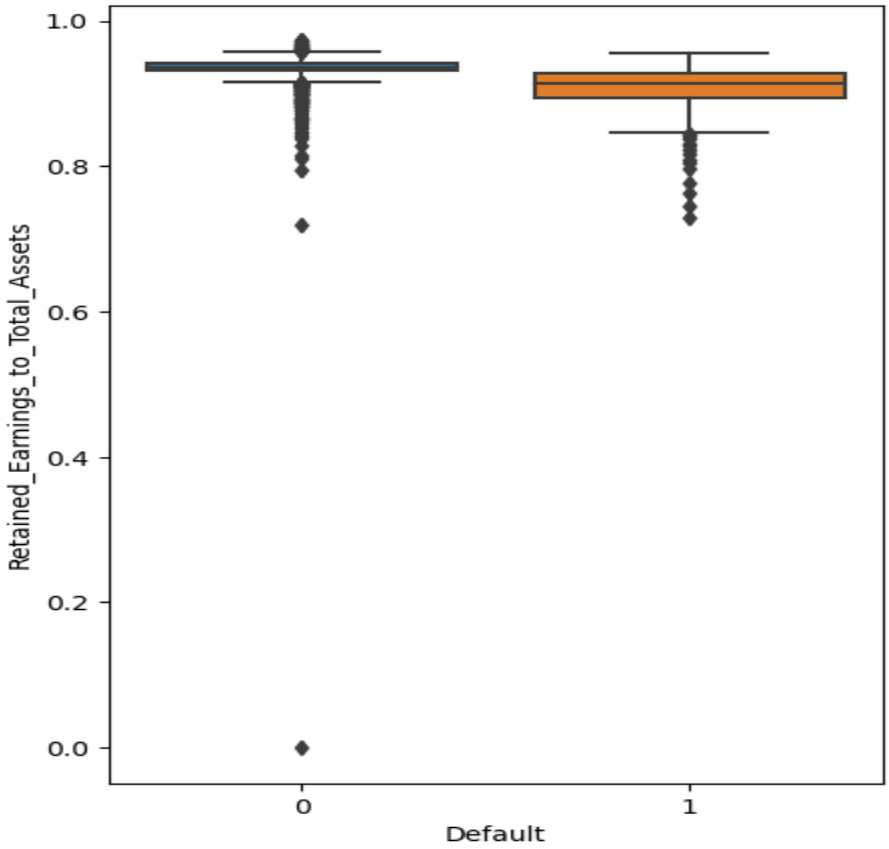

Figure 1.24- Boxplot of Retained_Earnings_to_Total_Assets

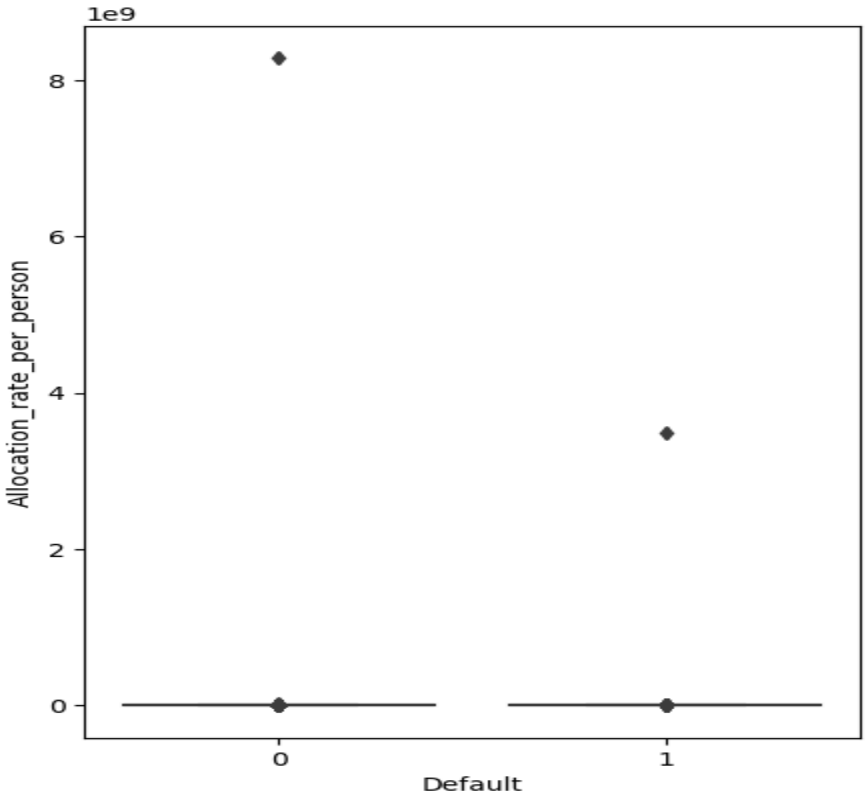

Figure 1.26- Boxplot of Allocation_rate_per_person

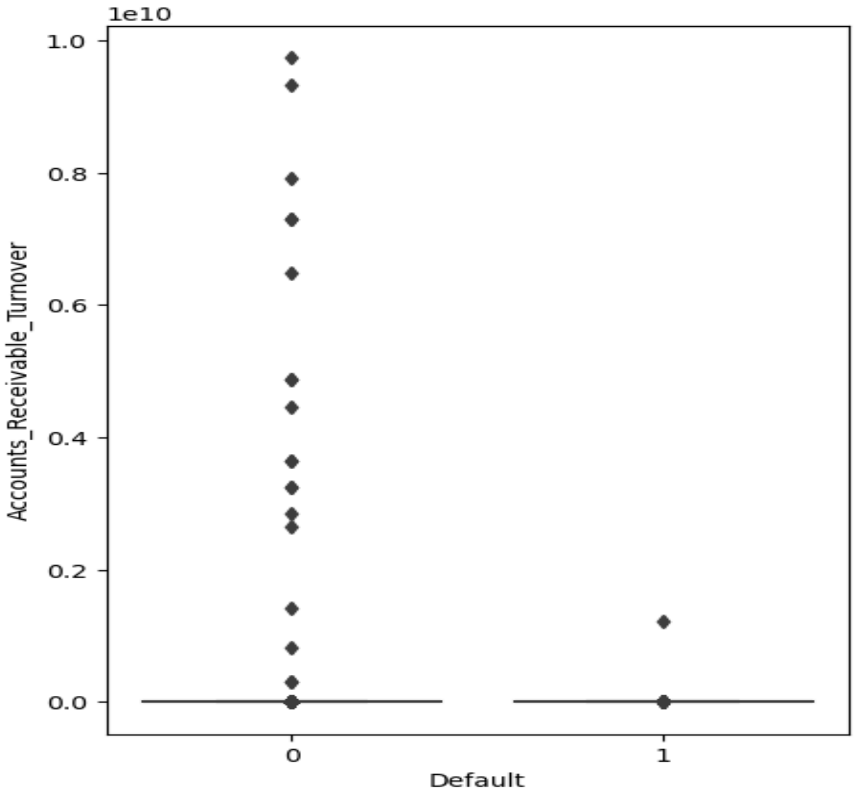

Figure 1.27- Boxplot of Accounts_Receivable_Turnover

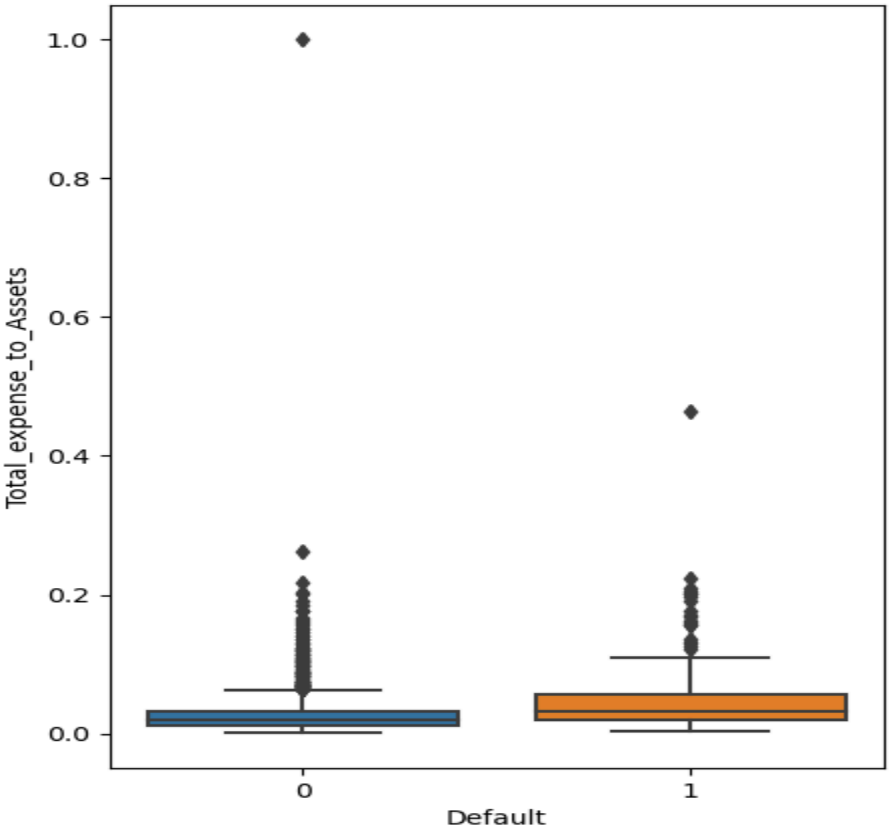

Figure 1.28- Boxplot of Total_expense_to_Assets

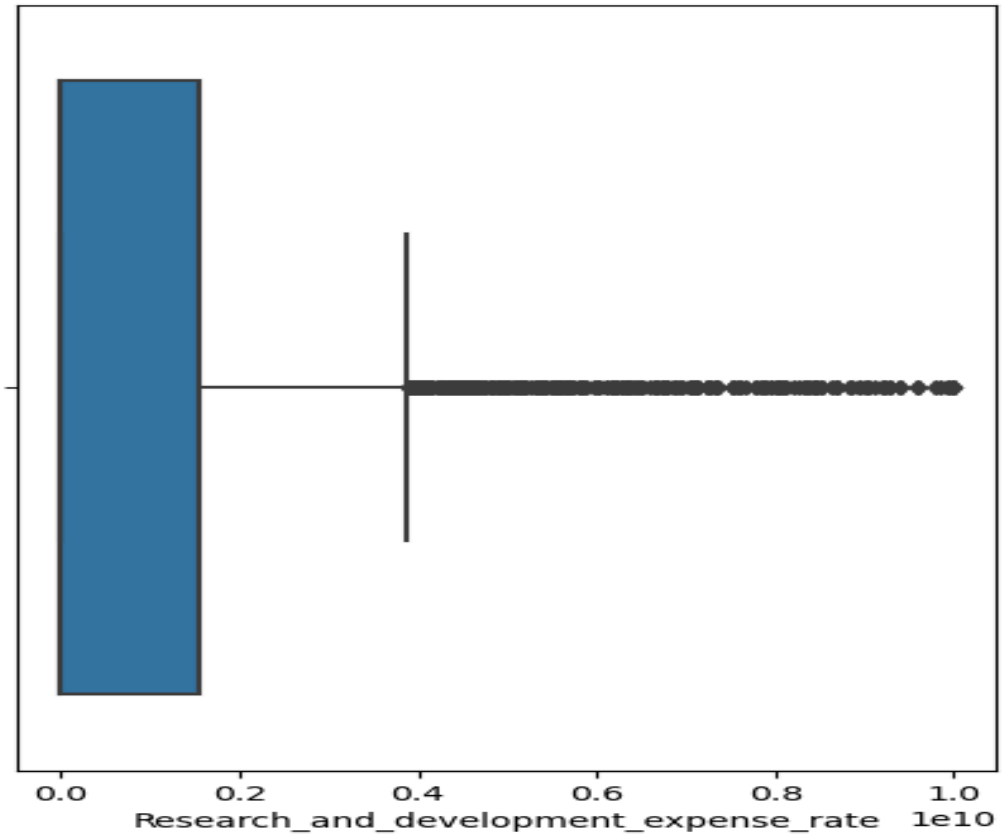

Figure 1.29- Boxplot of Research_and_development_expense_rate

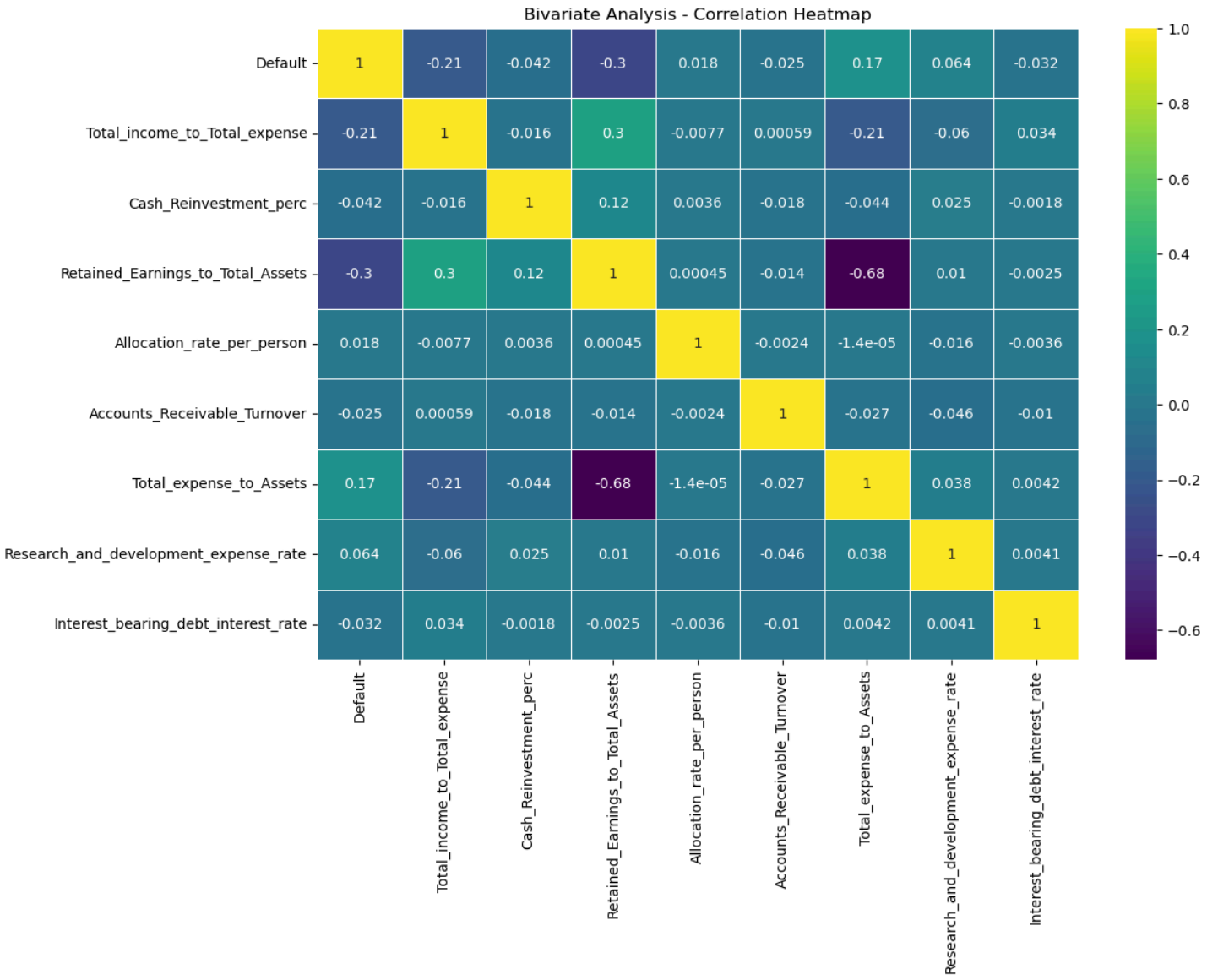

Figure 1.30 - Bivariate Analysis - Correlation Heatmap

- Outliers are noticeable across several attributes.
- The provided passage describes a method to establish the upper and lower limits for outliers in each column through the utilization of the **Interquartile Range (IQR).**
- Data points surpassing these established limits are then substituted with NaN values. This method is a prevalent strategy for handling outliers, shielding the analysis or modeling process from the disproportionate impact of extreme values.
- It's crucial to understand that this approach doesn't completely eradicate outliers; rather, it masks them by substituting their values with NaN.

PART A : Train Test Split

	count	mean	std	min	25%	50%	75%	max
Operating_Expense_Rate	2058.00	0.00	1.00	-0.63	-0.63	-0.63	0.63	2.44
Research_and_development_expense_rate	2058.00	0.00	1.00	-0.65	-0.65	-0.65	0.43	2.04
Cash_flow_rate	2058.00	0.00	1.00	-2.18	-0.58	-0.13	0.49	2.09
Interest_bearing_debt_interest_rate	2058.00	0.00	1.00	-1.63	-0.70	-0.10	0.60	2.56
Tax_rate_A	2058.00	-0.00	1.00	-0.82	-0.82	-0.54	0.78	3.19
Cash_Flow_Per_Share	2058.00	-0.00	0.97	-9.84	-0.32	0.05	0.37	9.30
Per_Share_Net_profit_before_tax_Yuan	2058.00	0.00	1.00	-2.15	-0.54	-0.04	0.53	2.13
Realized_Sales_Gross_Profit_Growth_Rate	2058.00	-0.00	1.00	-2.05	-0.55	-0.10	0.45	1.96
Operating_Profit_Growth_Rate	2058.00	-0.00	1.00	-1.98	-0.50	-0.05	0.48	1.96
Continuous_Net_Profit_Growth_Rate	2058.00	0.00	1.00	-1.91	-0.45	0.01	0.52	1.98
Total_Asset_Growth_Rate	2058.00	0.00	1.00	-1.82	-0.33	0.32	0.66	1.61
Net_Value_Growth_Rate	2058.00	-0.00	1.00	-1.97	-0.51	-0.15	0.47	1.94
Total_Asset_Return_Growth_Rate_Ratio	2058.00	-0.00	1.00	-2.09	-0.53	-0.02	0.51	2.07
Cash_Reinvestment_perc	2058.00	-0.00	1.00	-2.09	-0.52	0.07	0.53	2.10
Current_Ratio	2058.00	0.00	1.00	-1.77	-0.69	-0.30	0.45	2.17
Quick_Ratio	2058.00	-0.00	1.00	-1.31	-0.73	-0.28	0.43	2.18
Interest_Expense_Ratio	2058.00	-0.00	1.00	-1.84	-0.43	-0.27	0.51	1.93
Total_debt_to_Total_net_worth	2058.00	-0.00	1.00	-0.04	-0.04	-0.04	-0.04	36.83
Long_term_fund_suitability_ratio_A	2058.00	0.00	1.00	-1.68	-0.74	-0.41	0.41	2.12
Net_profit_before_tax_to_Paid_in_capital	2058.00	0.00	1.00	-2.16	-0.54	-0.04	0.54	2.16
Total_Asset_Turnover	2058.00	0.00	1.00	-1.48	-0.74	-0.24	0.53	2.44
Accounts_Receivable_Turnover	2058.00	-0.00	1.00	-1.46	-0.72	-0.39	0.37	2.02
Average_Collection_Days	2058.00	-0.00	1.00	-1.63	-0.72	-0.11	0.56	2.49
Inventory_Turnover_Rate_times	2058.00	-0.00	1.00	-0.66	-0.66	-0.65	0.58	2.45
Fixed_Assets_Turnover_Frequency	2058.00	-0.00	1.00	-0.66	-0.64	-0.59	0.31	1.74
Net_Worth_Turnover_Rate_times	2058.00	-0.00	1.00	-1.33	-0.74	-0.32	0.47	2.29
Operating_profit_per_person	2058.00	-0.00	1.00	-1.91	-0.48	-0.11	0.47	1.89

Figure 1.31- Description after train test split

PART A: Build Logistic Regression Model (using statsmodels library) on most important variables on train dataset and choose the optimum cut-off. Also showcase your model building approach

Statsmodels offers a linear regression implementation that grants access to comprehensive statistical insights about the model, such as p-values, t-tests, and confidence intervals. This makes it particularly advantageous for in-depth statistical analysis and hypothesis testing.

Linear Regression with scikit-learn:

In contrast, scikit-learn emphasizes linear regression primarily for predictive modeling purposes, providing a uniform interface for various machine learning models. However, scikit-learn doesn't furnish the same level of intricate statistical details as statsmodels.

In summary, when the goal is to perform meticulous statistical analysis and hypothesis testing in the context of linear regression, statsmodels proves to be more suitable. On the other hand, for constructing predictive models and leveraging standard machine learning tools, scikit-learn stands out as the preferred choice.

```
Optimization terminated successfully.
         Current function value: 0.198490
         Iterations 9
```

Logit Regression Results

Dep. Variable:	Default	**No. Observations:**	1378
Model:	Logit	**Df Residuals:**	1366
Method:	MLE	**Df Model:**	11
Date:	Thu, 11 Jan 2024	**Pseudo R-squ.:**	0.4307
Time:	22:52:23	**Log-Likelihood:**	-273.52
converged:	True	**LL-Null:**	-480.46
Covariance Type:	nonrobust	**LLR p-value:**	6.904e-82

	coef	std err	z	P>|z|	[0.025	0.975]
Intercept	-4.1868	0.266	-15.722	0.000	-4.709	-3.665
Total_income_to_Total_expense	-1.0671	0.271	-3.935	0.000	-1.599	-0.536
Quick_Ratio	-0.7482	0.240	-3.116	0.002	-1.219	-0.278
Equity_to_Liability	-1.0776	0.267	-4.033	0.000	-1.601	-0.554
Cash_Reinvestment_perc	-0.3557	0.109	-3.267	0.001	-0.569	-0.142
Retained_Earnings_to_Total_Assets	-0.8801	0.205	-4.298	0.000	-1.281	-0.479
Operating_profit_per_person	0.4480	0.188	2.377	0.017	0.079	0.817
Allocation_rate_per_person	0.7054	0.138	5.108	0.000	0.435	0.976
Accounts_Receivable_Turnover	-0.6219	0.139	-4.482	0.000	-0.894	-0.350
Total_expense_to_Assets	0.4029	0.149	2.708	0.007	0.111	0.695
Research_and_development_expense_rate	0.3895	0.111	3.520	0.000	0.173	0.606
Interest_bearing_debt_interest_rate	0.3878	0.142	2.739	0.006	0.110	0.665

Figure 1.32

Convergence and Success: The optimization process terminated successfully, indicating that the logistic regression model successfully converged to a solution.

Model Characteristics:

- **Dependent Variable:** The dependent variable in the logistic regression model is labeled as "Default."
- **Number of Observations:** The model is based on 1378 observations.
- **Degrees of Freedom (Df):** There are 11 model parameters, resulting in 11 degrees of freedom.
- **Model Method: Maximum Likelihood Estimation (MLE)** was used for model estimation.

Model Performance Metrics:

- **Pseudo R-squared:** The Pseudo R-squared value is 0.4307, indicating the proportion of variance explained by the model.
- **Log-Likelihood:** The log-likelihood of the model is -273.52.
- **LL-Null:** The log-likelihood of a null model (no predictors) is -480.46.
- **Likelihood Ratio Test (LLR) p-value:** The p-value of the likelihood ratio test is extremely low (6.904e-82), suggesting that the model is statistically significant.

Coefficients and Significance:

- The coefficients for each predictor variable along with their standard errors, **z-scores**, and p-values are provided.
- Variables such as **"Total_income_to_Total_expense," "Quick_Ratio," "Equity_to_Liability,"** and others have significant effects on the log-odds of the dependent variable.

Interpretation of Coefficients:

- For instance, the intercept is **-4.1868**, and the coefficient for **"Total_income_to_Total_expense"** is **-1.0671**. These coefficients are in the log-odds scale, and their interpretation depends on the specific characteristics of the data.

PART A : Validate the Model on Test Dataset and state the performance metrics

In **financial credit risk analysis**, a high recall value and accepting a lower precision value can be a reasonable trade-off:

Recall (Sensitivity): It represents the proportion of actual credit defaults correctly identified by the model. High recall captures a large percentage of potential defaults, reducing missed defaults (false negatives). This helps mitigate losses for the lender.

Precision: It represents the proportion of predicted defaults that are actually credit defaults. A lower precision means more false positives, where customers are predicted to default but may not. This conservative approach minimizes the risk of lending to potential defaulters.

The precision and Recall values in both test and train sets are comparable and hence we can say that we have a fairly good model not showing presence of any overfitting and a Recall score of around 80%.

VIF has been used to mitigate the multicollinearity in the model where as the feature selection has been further improved by using the p-value to identify and eliminate any non-significant features from the model.

The ROC (Receiver Operating Characteristic) curve evaluates a classification model's performance by plotting the trade-off between true positive rate (sensitivity) and false positive rate (1 - specificity) as the classification threshold varies. It helps visualize how well the model distinguishes positive and negative instances.

AUC (Area Under the Curve) summarizes overall performance, with higher values indicating better model performance.

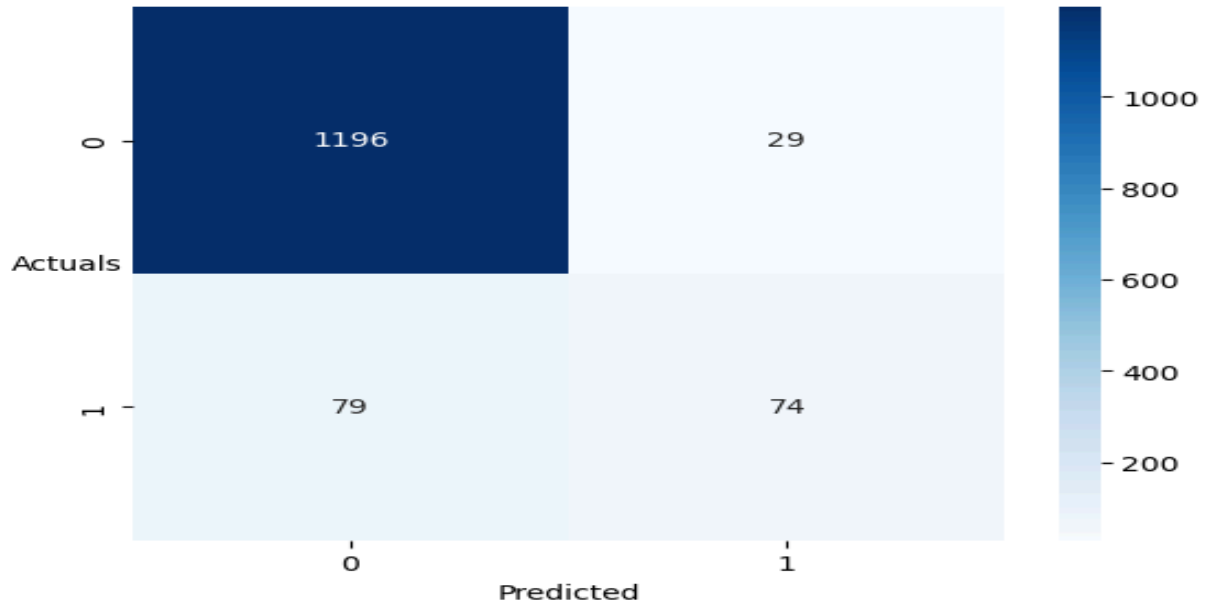

Figure 1.33- MODEL PERFORMANCE through Confusion Matrix

	precision	recall	f1-score	support
0.0	0.938	0.976	0.957	1225
1.0	0.718	0.484	0.578	153
accuracy			0.922	1378
macro avg	0.828	0.730	0.767	1378
weighted avg	0.914	0.922	0.915	1378

Figure 1.34- Precision Matrix

Here are insights derived from the above classification report:

Class 0 (Non-Default) Metrics:

- **Precision**: 93.8% of instances predicted as class 0 were actually class 0.
- **Recall**: 97.6% of actual class 0 instances were correctly predicted as class 0.
- **F1-Score**: The harmonic mean of precision and recall for class 0 is 95.7%.

Class 1 (Default) Metrics:

- **Precision**: 71.8% of instances predicted as class 1 were actually class 1.
- **Recall**: 48.4% of actual class 1 instances were correctly predicted as class 1.
- **F1-Score**: The harmonic mean of precision and recall for class 1 is 57.8%.

Overall Model Performance Metrics:

- **Accuracy**: The overall accuracy of the model is 92.2%, indicating the proportion of correctly predicted instances out of the total.
- **Macro Average**: The macro average of precision, recall, and F1-score is 82.8%, 73.0%, and 76.7%, respectively.
- **Weighted Average:** The weighted average of precision, recall, and F1-score, considering class imbalance, is 91.4%, 92.2%, and 91.5%, respectively.

Interpretation:

- The model performs very well in identifying non-default instances (Class 0) with high precision and recall.
- For default instances (Class 1), the model is less precise and has lower recall, indicating some difficulty in correctly predicting defaults.
- The overall accuracy is relatively high, but it's important to consider the trade-off between precision and recall based on the specific goals and consequences of misclassifications.

Class Imbalance:

- The significant difference in the number of instances between Class 0 and Class 1 may influence the model's performance metrics. Consideration of strategies like resampling or adjusting the classification threshold may be relevant.

Choosing the optimal Threshold using ROC Curve - 0.1070956659347362

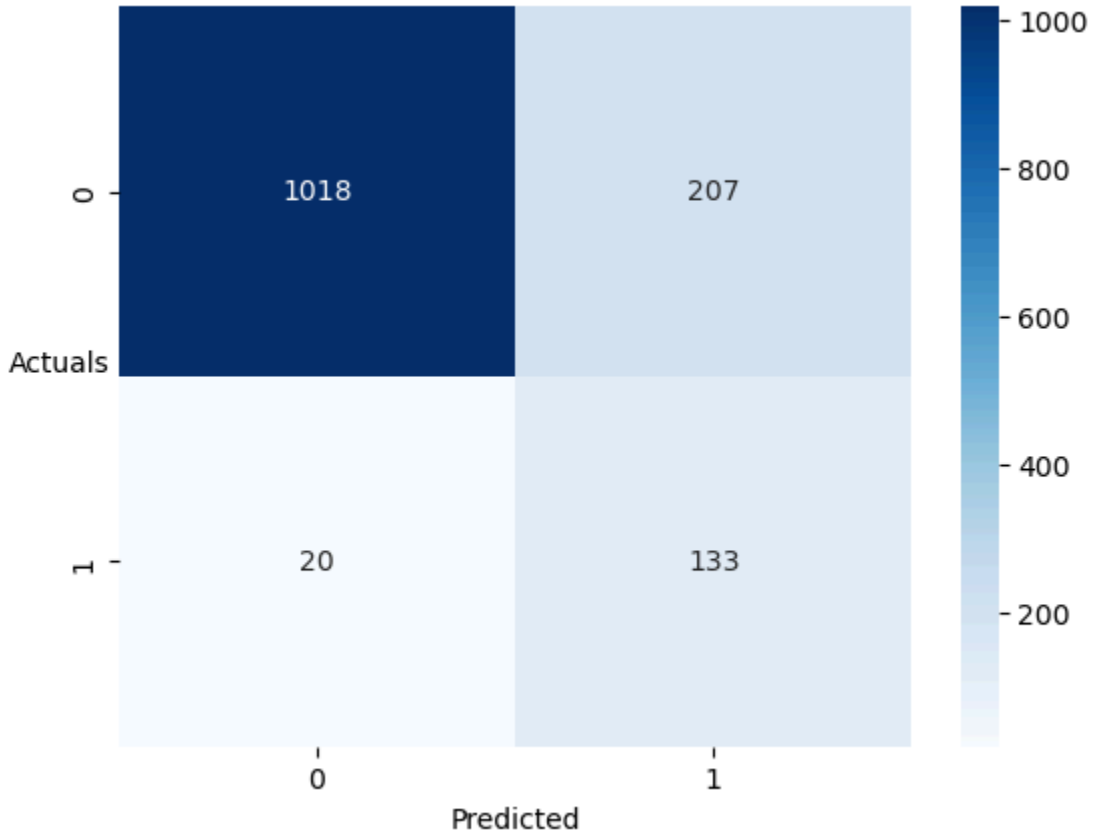

Figure – Prediction after validating on revised threshold

	precision	recall	f1-score	support
0.0	0.981	0.831	0.900	1225
1.0	0.391	0.869	0.540	153
accuracy			0.835	1378
macro avg	0.686	0.850	0.720	1378
weighted avg	0.915	0.835	0.860	1378

Figure 1.35- Confusion Matrix after revised threshold

Here are insights derived from the above classification report:

Class 0 (Non-Default) Metrics:

- **Precision**: 98.1% of instances predicted as class 0 were actually class 0.
- **Recall**: 83.1% of actual class 0 instances were correctly predicted as class 0.
- **F1-Score:** The harmonic mean of precision and recall for class 0 is 90.0%.

Class 1 (Default) Metrics:

- **Precision**: 39.1% of instances predicted as class 1 were actually class 1.
- **Recall**: 86.9% of actual class 1 instances were correctly predicted as class 1.

- **F1-Score**: The harmonic mean of precision and recall for class 1 is 54.0%.

Overall Model Performance Metrics:

- **Accuracy**: The overall accuracy of the model is 83.5%, indicating the proportion of correctly predicted instances out of the total.
- **Macro Average**: The macro average of precision, recall, and F1-score is 68.6%, 85.0%, and 72.0%, respectively.
- **Weighted Average**: The weighted average of precision, recall, and F1-score, considering class imbalance, is 91.5%, 83.5%, and 86.0%, respectively.

Interpretation:

- The model performs exceptionally well in identifying non-default instances (Class 0) with high precision and relatively good recall.
- For default instances (Class 1), the model has lower precision but higher recall, indicating a trade-off between precision and recall.
- The overall accuracy is decent, but the trade-offs between precision and recall should be carefully considered based on the specific goals and consequences of misclassifications.

Class Imbalance:

- The significant difference in the number of instances between Class 0 and Class 1 may influence the model's performance metrics. The trade-offs between precision and recall become crucial in addressing the class imbalance.

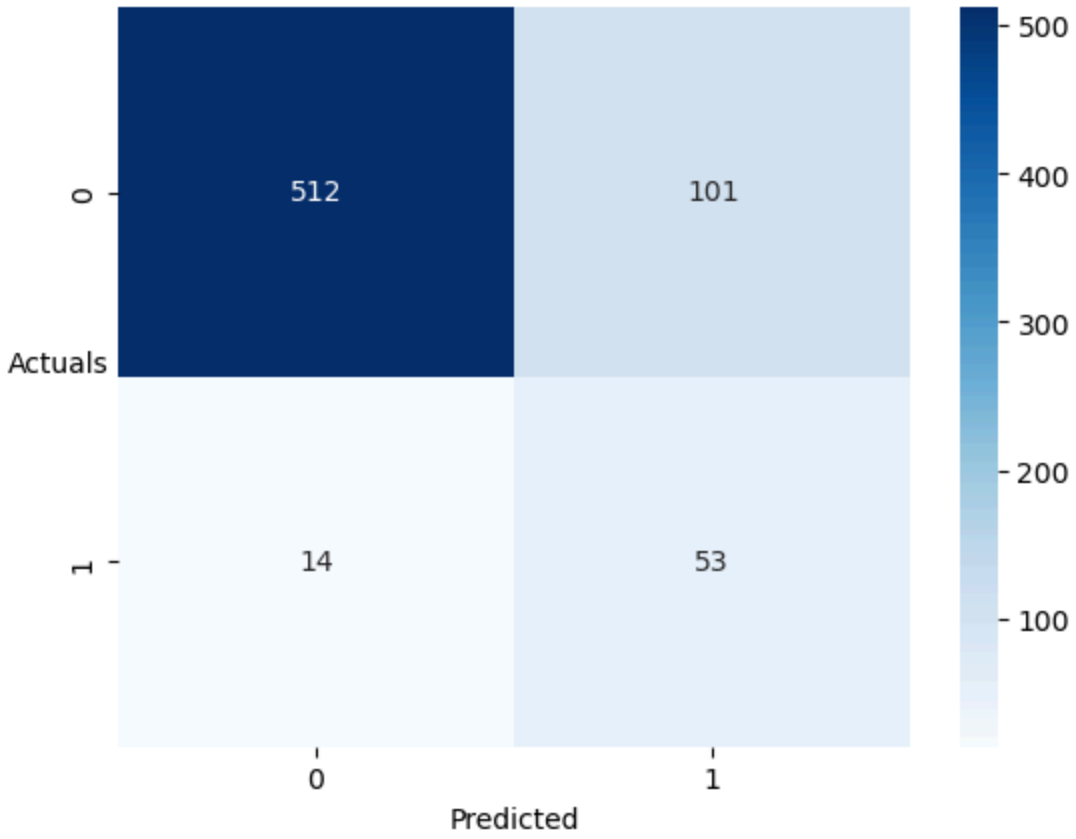

Figure 1.36- Prediction on Test Data

	precision	recall	f1-score	support
0.0	0.973	0.835	0.899	613
1.0	0.344	0.791	0.480	67
accuracy			0.831	680
macro avg	0.659	0.813	0.689	680
weighted avg	0.911	0.831	0.858	680

Figure 1.37- Confusion Matrix for prediction on Test Data

Here are insights derived from the above report:

Class 0 (Non-Default) Metrics:

- **Precision**: 97.3% of instances predicted as class 0 were actually class 0.
- **Recall**: 83.5% of actual class 0 instances were correctly predicted as class 0.
- **F1-Score**: The harmonic mean of precision and recall for class 0 is 89.9%.

Class 1 (Default) Metrics:

- **Precision**: 34.4% of instances predicted as class 1 were actually class 1.
- **Recall**: 79.1% of actual class 1 instances were correctly predicted as class 1.
- **F1-Score**: The harmonic mean of precision and recall for class 1 is 48.0%.

Overall Model Performance Metrics:

- **Accuracy**: The overall accuracy of the model is 83.1%, indicating the proportion of correctly predicted instances out of the total.
- **Macro Average**: The macro average of precision, recall, and F1-score is 65.9%, 81.3%, and 68.9%, respectively.
- **Weighted Average:** The weighted average of precision, recall, and F1-score, considering class imbalance, is 91.1%, 83.1%, and 85.8%, respectively.

Interpretation:

- The model demonstrates strong performance in identifying non-default instances (Class 0) with high precision and decent recall.
- For default instances (Class 1), the model has lower precision but reasonable recall, indicating a trade-off between precision and recall.
- The overall accuracy is good, but the trade-offs between precision and recall should be carefully considered based on the specific goals and consequences of misclassifications.

Class Imbalance:

- The significant difference in the number of instances between Class 0 and Class 1 may influence the model's performance metrics. The trade-offs between precision and recall become crucial in addressing the class imbalance.

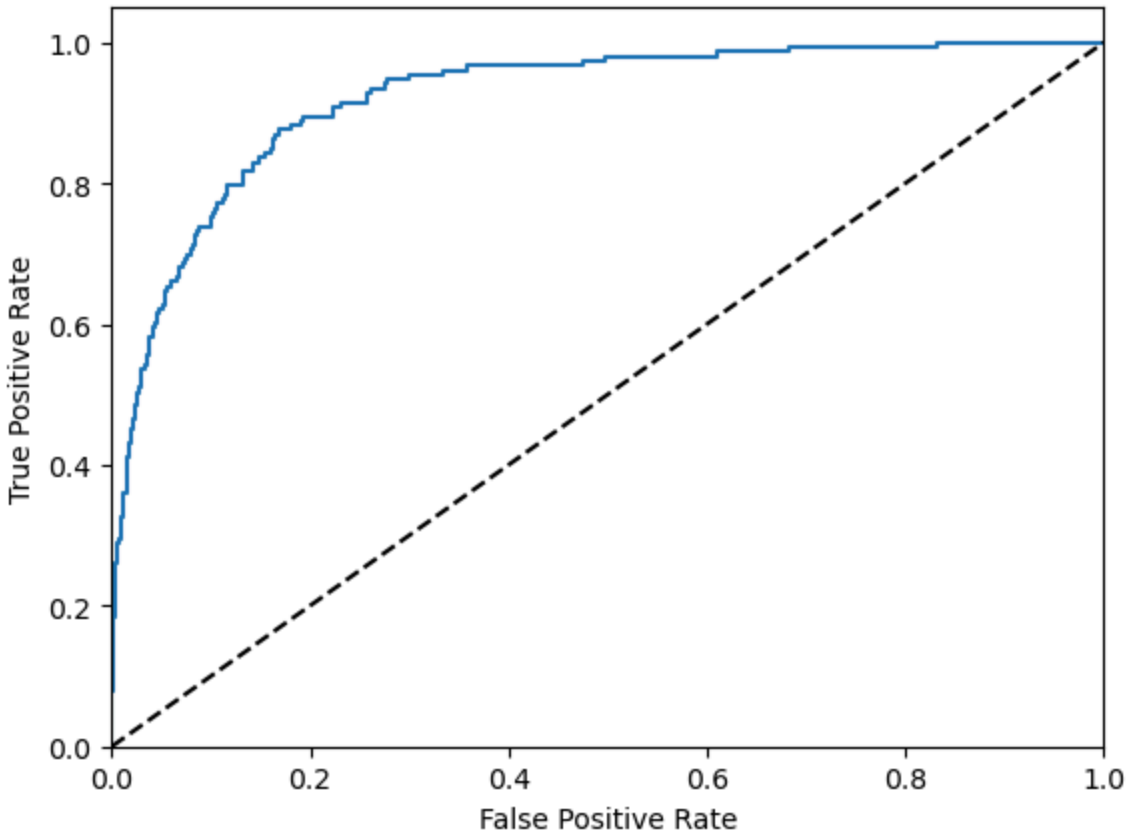

Figure 1.38- ROC curve

The ROC-AUC score for model35 on the test set is 0.90. Here are insights derived from this information:

ROC-AUC Score: **The ROC-AUC (Receiver Operating Characteristic - Area Under the Curve)** score is a metric used to evaluate the performance of a binary classification model. In this case, the score is 0.90, which indicates a high level of discriminative ability.

Model Discrimination: An ROC-AUC score of 0.90 suggests that model35 has a strong ability to distinguish between the positive and negative classes. The higher the ROC-AUC score, the better the model is at ranking true positive instances higher than false positive instances across various probability thresholds.

Performance Evaluation: A ROC-AUC score of 0.90 is generally considered excellent. It implies that the model has a high true positive rate while maintaining a low false positive rate, which is crucial for many classification tasks.

PART A: Build a Random Forest Model on Train Dataset. Also showcase your model building approach

```
GridSearchCV(estimator=RandomForestClassifier(),
             param_grid={'max_depth': [3, 5, 7],
                         'min_samples_leaf': [5, 10, 15],
                         'min_samples_split': [15, 30, 45],
                         'n_estimators': [25, 50]})
```

Figure - Random Forest Classifier

```
{'max_depth': 5,
 'min_samples_leaf': 5,
 'min_samples_split': 30,
 'n_estimators': 25}
```

Figure 1.39- Best Params

max_depth:

- The maximum depth of each tree in the random forest is set to 5. This hyperparameter controls the maximum depth of the decision trees. A lower value may prevent overfitting, while a higher value might lead to a more complex model.

min_samples_leaf:

- The minimum number of samples required to be at a leaf node is set to 5. This hyperparameter controls the minimum number of samples that should be present in a leaf node. A higher value can lead to a smoother decision boundary and may help prevent overfitting.

min_samples_split:

- The minimum number of samples required to split an internal node is set to 30. This hyperparameter controls the minimum number of samples required to split an internal node. A higher value can lead to a more robust model that is less sensitive to noise.

n_estimators:

- The number of trees in the random forest is set to 25. This hyperparameter defines the number of decision trees that will be trained in the ensemble. Increasing the number of trees can improve model performance, up to a certain point.

	precision	recall	f1-score	support
0.0	0.94	0.99	0.96	1225
1.0	0.87	0.49	0.63	153
accuracy			0.94	1378
macro avg	0.91	0.74	0.80	1378
weighted avg	0.93	0.94	0.93	1378

Figure 1.40- Prediction train data

Here are insights derived from the provided above report:

Class 0 (Non-Default) Metrics:

- **Precision**: 94% of instances predicted as class 0 were actually class 0.
- **Recall**: 99% of actual class 0 instances were correctly predicted as class 0.
- **F1-Score:** The harmonic mean of precision and recall for class 0 is 96%.

Class 1 (Default) Metrics:

- **Precision**: 87% of instances predicted as class 1 were actually class 1.
- **Recall**: 49% of actual class 1 instances were correctly predicted as class 1.
- **F1-Score**: The harmonic mean of precision and recall for class 1 is 63%.

Overall Model Performance Metrics:

- **Accuracy**: The overall accuracy of the model is 94%, indicating the proportion of correctly predicted instances out of the total.
- **Macro Average:** The macro average of precision, recall, and F1-score is 91%, 74%, and 80%, respectively.
- **Weighted Average:** The weighted average of precision, recall, and F1-score, considering class imbalance, is 93%, 94%, and 93%, respectively.

Interpretation:

- The model performs exceptionally well in identifying non-default instances (Class 0) with high precision and recall, resulting in a high F1-score.
- For default instances (Class 1), the model has relatively good precision but lower recall, indicating a trade-off between precision and recall.
- The overall accuracy is high, suggesting that the model is effective in making correct predictions on the majority of instances.

	precision	recall	f1-score	support
0.0	0.93	0.98	0.95	613
1.0	0.64	0.34	0.45	67
accuracy			0.92	680
macro avg	0.79	0.66	0.70	680
weighted avg	0.90	0.92	0.90	680

Figure 1.41- Prediction Test Data

Here are insights derived from the provided above report:

Class 0 (Non-Default) Metrics:

- **Precision**: 93% of instances predicted as class 0 were actually class 0.
- **Recall**: 98% of actual class 0 instances were correctly predicted as class 0.
- **F1-Score**: The harmonic mean of precision and recall for class 0 is 95%.

Class 1 (Default) Metrics:

- **Precision**: 64% of instances predicted as class 1 were actually class 1.
- **Recall**: 34% of actual class 1 instances were correctly predicted as class 1.
- **F1-Score**: The harmonic mean of precision and recall for class 1 is 45%.

Overall Model Performance Metrics:

- **Accuracy**: The overall accuracy of the model is 92%, indicating the proportion of correctly predicted instances out of the total.
- **Macro Average**: The macro average of precision, recall, and F1-score is 79%, 66%, and 70%, respectively.
- **Weighted Average**: The weighted average of precision, recall, and F1-score, considering class imbalance, is 90%, 92%, and 90%, respectively.

Interpretation:

- The model performs well in correctly identifying non-default instances (Class 0) with high precision and recall, resulting in a high F1-score.
- For default instances (Class 1), the model has a lower recall and precision, indicating a challenge in correctly predicting defaults.
- The overall accuracy is good, but there is a noticeable imbalance in performance between the two classes.

Class Imbalance:

- The class imbalance is apparent, as there are significantly more instances of Class 0 than Class 1. This can impact the evaluation metrics, especially for the minority class (Class 1).

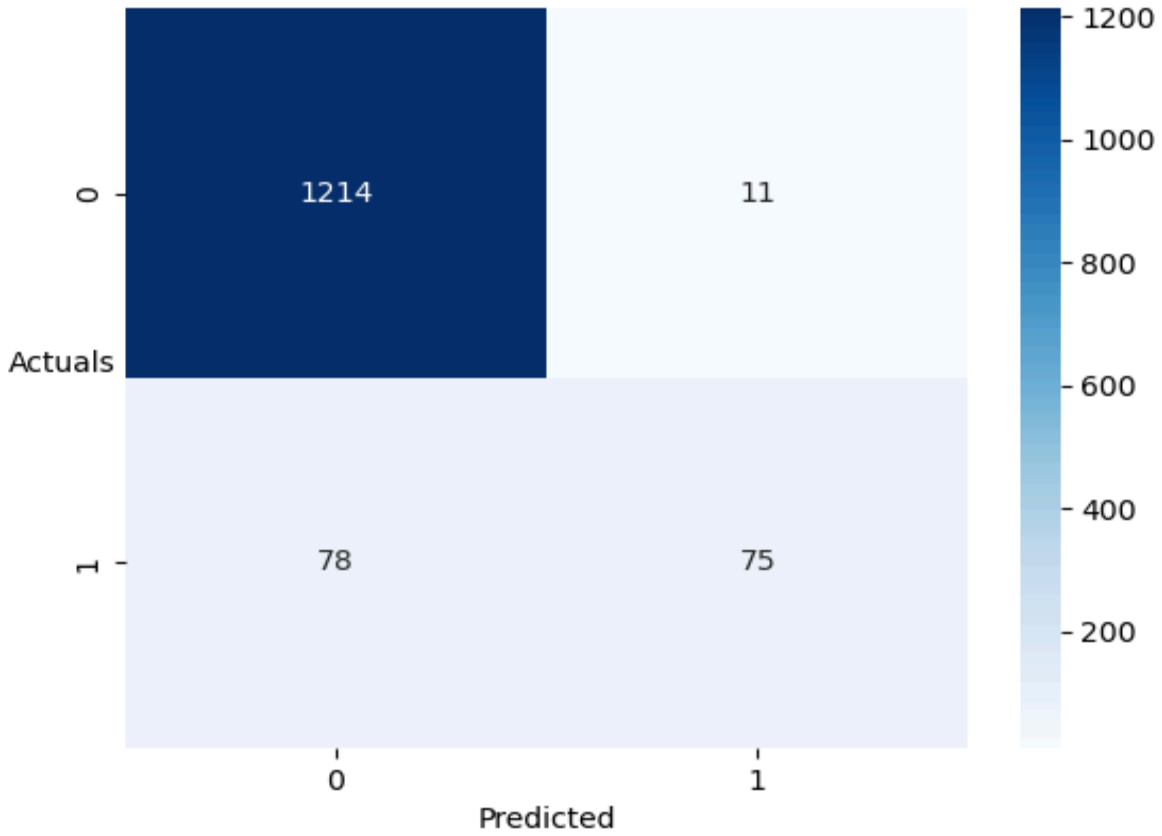

Figure - Prediction train data

	precision	recall	f1-score	support
0.0	0.94	0.99	0.96	1225
1.0	0.87	0.49	0.63	153
accuracy			0.94	1378
macro avg	0.91	0.74	0.80	1378
weighted avg	0.93	0.94	0.93	1378

Figure 1.42- Matrix Prediction for train data

Here is an analysis of the above classification report:

Class 0 (Non-Default) Metrics:

- **Precision**: 94% of instances predicted as class 0 were actually class 0.
- **Recall**: 99% of actual class 0 instances were correctly predicted as class 0.
- **F1-Score**: The harmonic mean of precision and recall for class 0 is 96%.

Class 1 (Default) Metrics:

- **Precision**: 87% of instances predicted as class 1 were actually class 1.

- **Recall**: 49% of actual class 1 instances were correctly predicted as class 1.
- **F1-Score**: The harmonic mean of precision and recall for class 1 is 63%.

Overall Model Performance Metrics:

- **Accuracy**: The overall accuracy of the model is 94%, indicating the proportion of correctly predicted instances out of the total.
- **Macro Average**: The macro average of precision, recall, and F1-score is 91%, 74%, and 80%, respectively.
- **Weighted Average**: The weighted average of precision, recall, and F1-score, considering class imbalance, is 93%, 94%, and 93%, respectively.

Interpretation:

- The model performs exceptionally well in correctly identifying non-default instances (Class 0) with high precision and recall, resulting in a high F1-score.
- For default instances (Class 1), the model has relatively good precision but lower recall, indicating a trade-off between precision and recall.
- The overall accuracy is high, suggesting that the model is effective in making correct predictions on the majority of instances.

Class Imbalance:

- The class imbalance is evident, as there are significantly more instances of Class 0 than Class 1. This can impact the evaluation metrics, and consideration of precision and recall becomes crucial, especially for the minority class

PART A: Validate the Random Forest Model on test Dataset and state the performance metrics. Also state interpretation from the model

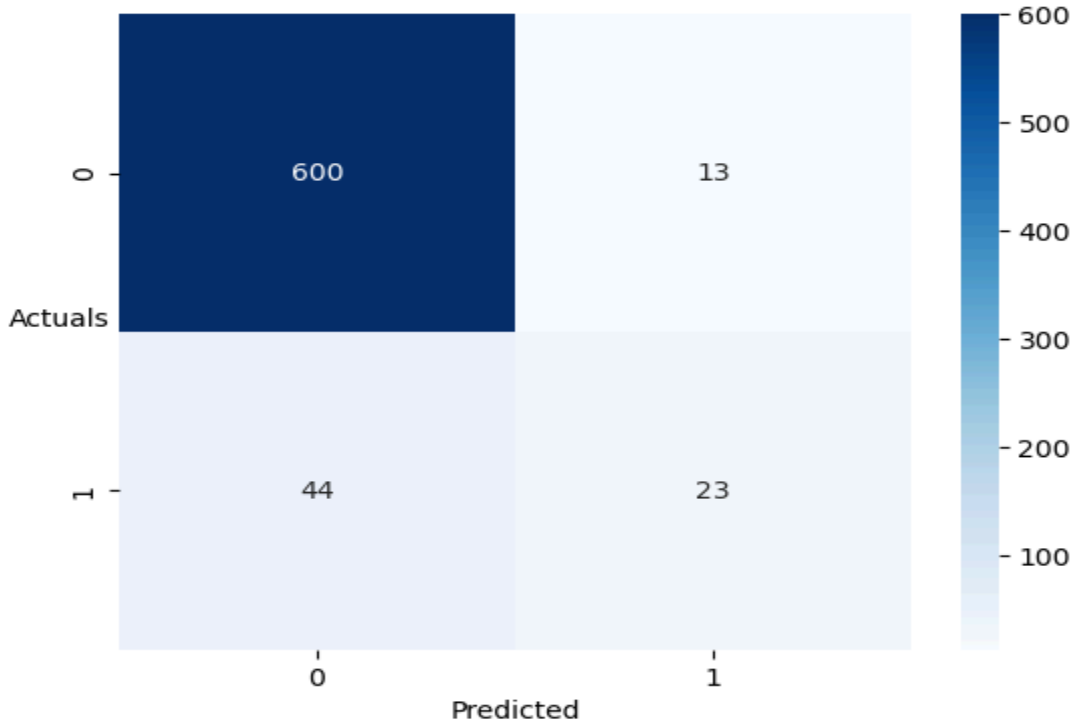

Figure 1.43- Prediction on test data

	precision	recall	f1-score	support
0.0	0.93	0.98	0.95	613
1.0	0.64	0.34	0.45	67
accuracy			0.92	680
macro avg	0.79	0.66	0.70	680
weighted avg	0.90	0.92	0.90	680

Figure 1.44- Matrix for prediction on test data

Here is an analysis of the above classification report:

Class 0 (Non-Default) Metrics:

- **Precision**: 93% of instances predicted as class 0 were actually class 0.
- **Recall**: 98% of actual class 0 instances were correctly predicted as class 0.
- **F1-Score**: The harmonic mean of precision and recall for class 0 is 95%.

Class 1 (Default) Metrics:

- **Precision**: 64% of instances predicted as class 1 were actually class 1.
- **Recall**: 34% of actual class 1 instances were correctly predicted as class 1.
- **F1-Score**: The harmonic mean of precision and recall for class 1 is 45%.

Overall Model Performance Metrics:

- **Accuracy**: The overall accuracy of the model is 92%, indicating the proportion of correctly predicted instances out of the total.
- **Macro Average**: The macro average of precision, recall, and F1-score is 79%, 66%, and 70%, respectively.
- **Weighted Average**: The weighted average of precision, recall, and F1-score, considering class imbalance, is 90%, 92%, and 90%, respectively.

Interpretation:

- The model performs well in correctly identifying non-default instances (Class 0) with high precision and recall, resulting in a high F1-score.
- For default instances (Class 1), the model has a lower recall and precision, indicating a challenge in correctly predicting defaults.
- The overall accuracy is good, but there is a noticeable imbalance in performance between the two classes.

Class Imbalance:

- The class imbalance is apparent, as there are significantly more instances of Class 0 than Class 1. This can impact the evaluation metrics, especially for the minority class (Class 1).

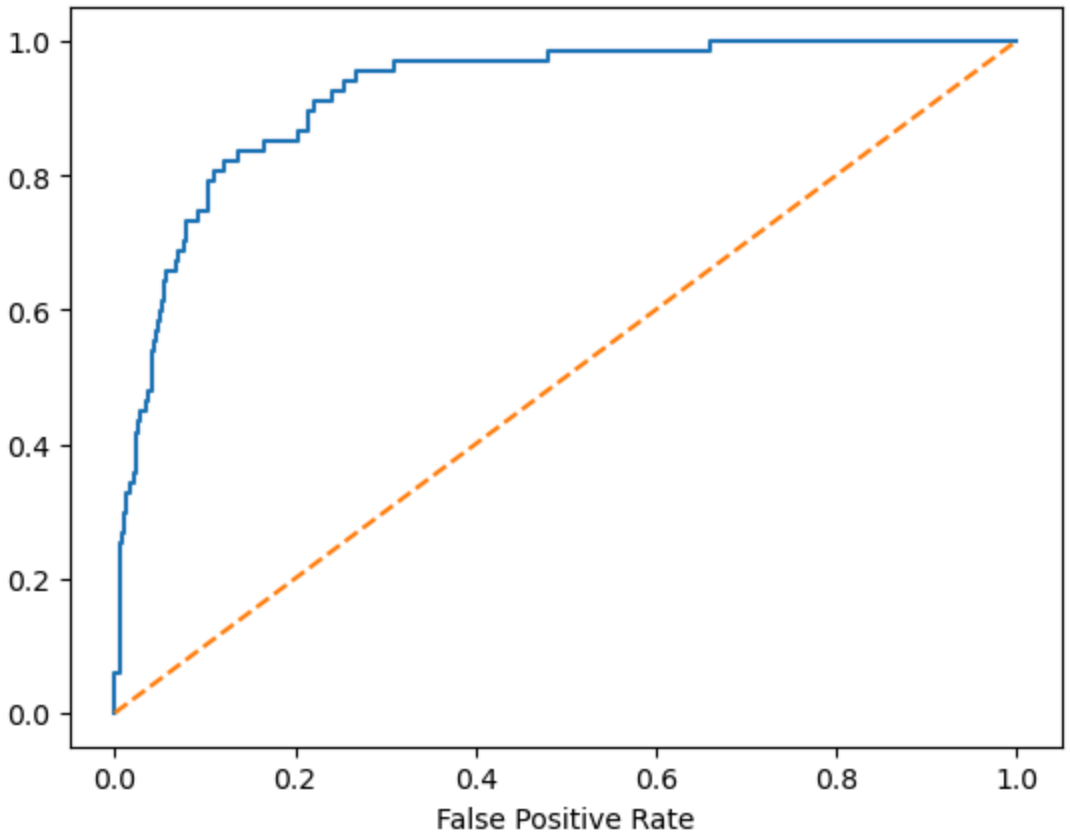

Figure 1.45- ROC Curve for test Data Prediction

PART A: Build a LDA Model on Train Dataset. Also showcase your model building approach

	precision	recall	f1-score	support
0.0	0.95	0.96	0.95	1225
1.0	0.64	0.58	0.60	153
accuracy			0.92	1378
macro avg	0.79	0.77	0.78	1378
weighted avg	0.91	0.92	0.91	1378

Figure 1.46- Matrix for prediction on train data

Here is an analysis of the above classification report:

Class 0 (Non-Default) Metrics:

- **Precision**: 95% of instances predicted as class 0 were actually class 0.
- **Recall**: 96% of actual class 0 instances were correctly predicted as class 0.

- **F1-Score**: The harmonic mean of precision and recall for class 0 is 95%.

Class 1 (Default) Metrics:

- **Precision**: 64% of instances predicted as class 1 were actually class 1.
- **Recall**: 58% of actual class 1 instances were correctly predicted as class 1.
- **F1-Score**: The harmonic mean of precision and recall for class 1 is 60%.

Overall Model Performance Metrics:

- **Accuracy**: The overall accuracy of the model is 92%, indicating the proportion of correctly predicted instances out of the total.
- **Macro Average:** The macro average of precision, recall, and F1-score is 79%, 77%, and 78%, respectively.
- **Weighted Average:** The weighted average of precision, recall, and F1-score, considering class imbalance, is 91%, 92%, and 91%, respectively.

Interpretation:

- The model performs well in correctly identifying non-default instances (Class 0) with high precision and recall, resulting in a high F1-score.
- For default instances (Class 1), the model has reasonable precision and recall, indicating a balanced performance in predicting defaults.
- The overall accuracy is good, and there is a relatively balanced performance between the two classes.

Class Imbalance:

- The class imbalance is present, as there are significantly more instances of Class 0 than Class 1. However, the model seems to handle both classes reasonably well.

PART A: Validate the LDA Model on test Dataset and state the performance metrics. Also state interpretation from the model

	precision	recall	f1-score	support
0.0	0.96	0.94	0.95	613
1.0	0.55	0.63	0.59	67
accuracy			0.91	680
macro avg	0.76	0.79	0.77	680
weighted avg	0.92	0.91	0.92	680

Figure 1.47- matrix for prediction on test Data

Here is an analysis of the above classification report:

Class 0 (Non-Default) Metrics:

- **Precision**: 96% of instances predicted as class 0 were actually class 0.
- **Recall**: 94% of actual class 0 instances were correctly predicted as class 0.
- **F1-Score:** The harmonic mean of precision and recall for class 0 is 95%.

Class 1 (Default) Metrics:

- **Precision**: 55% of instances predicted as class 1 were actually class 1.
- **Recall**: 63% of actual class 1 instances were correctly predicted as class 1.
- **F1-Score**: The harmonic mean of precision and recall for class 1 is 59%.

Overall Model Performance Metrics:

- **Accuracy**: The overall accuracy of the model is 91%, indicating the proportion of correctly predicted instances out of the total.
- **Macro Average:** The macro average of precision, recall, and F1-score is 76%, 79%, and 77%, respectively.
- **Weighted Average**: The weighted average of precision, recall, and F1-score, considering class imbalance, is 92%, 91%, and 92%, respectively.

Interpretation:

- The model performs very well in correctly identifying non-default instances (Class 0) with high precision and recall, resulting in a high F1-score.
- For default instances (Class 1), the model has reasonable precision and recall, indicating a balanced performance in predicting defaults.
- The overall accuracy is good, and there is a relatively balanced performance between the two classes.

Class Imbalance:

- The class imbalance is present, as there are significantly more instances of Class 0 than Class 1. The model seems to handle both classes reasonably well, with a balanced performance in predicting defaults.

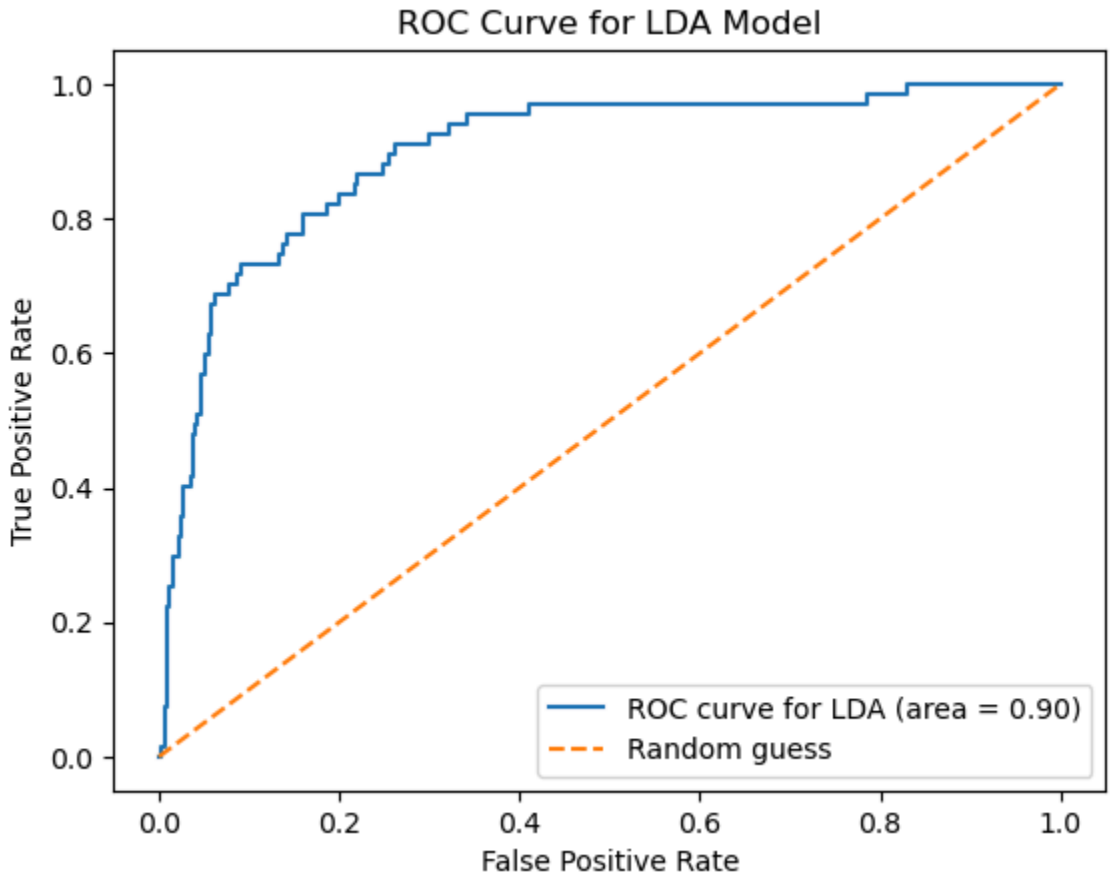

Figure 1.48- ROC Curve for LDA

PART A: Compare the performances of Logistic Regression, Random Forest, and LDA models (include ROC curve)

	Model	Train Accuracy	Test Accuracy	Train Precision	Test Precision	Train Recall	Test Recall	ROC AUC
0	rfcl	0.935414	0.916176	0.872093	0.638889	0.490196	0.343284	0.920552
1	LDA	0.916546	0.913235	0.637681	0.552632	0.575163	0.626866	0.897884

Figure 1.49- Comparison between models

The random forest classifier (rfcl) demonstrates superior performance compared to L**inear Discriminant Analysis (LDA)** across various metrics. It achieves higher accuracy, precision, and ROC AUC on both the training and test datasets. Despite these overall advantages, **LDA excels** in a **crucial aspect** by displaying **superior test recall**. This means that LDA is more adept at correctly identifying companies at risk of default.

In practical terms, this implies that while the random forest model provides better overall predictions, LDA is particularly effective at pinpointing potential defaulters. The decision between these two models should be made based on specific business objectives and priorities. If precision and overall model performance are paramount, the random forest may be preferred. On the other hand, if correctly identifying companies at risk of default is crucial, favoring LDA might be the more prudent choice. It essentially involves striking a balance between overall predictive power and the ability to accurately identify potential defaulters, considering the unique priorities and objectives of the financial risk analysis task at hand.

PART A: Conclusions and Recommendations

Conclusions :

Class Imbalance Impact: The presence of class imbalance in the dataset significantly influences model performance. Utilizing Synthetic Minority Over-sampling Technique (SMOTE) with Logistic Regression and Linear Discriminant Analysis (LDA) helps mitigate this issue by creating synthetic samples for the minority class. This improves recall but may lead to a slight reduction in precision.

Model Selection Considerations: The optimal choice of a model depends on the specific goals of the analysis. Logistic Regression and LDA are linear models, while Random Forest represents a non-linear ensemble model. When selecting a model, take into account the linearity assumption and the complexity of the data.
Hyperparameter Tuning Insights: Despite employing hyperparameter tuning, specifically using Grid Search, for the Random Forest model, no substantial improvement was observed. Experimenting with different hyperparameter values is essential to identify the best settings for the specific dataset.

Recommendations:

Model Ensemble Strategy: Consider leveraging the strengths of individual models through ensembling. For instance, combining Logistic Regression-SMOTE and LDA-SMOTE models could enhance overall classification performance.
Threshold Adjustment: Depending on the business context, explore adjusting the classification threshold to optimize either precision or recall. This allows customization based on specific requirements, such as accurately identifying default cases or minimizing false positives.
Feature Engineering Exploration: Further explore feature engineering techniques to extract more pertinent information from the data, potentially enhancing overall model performance.
Data Collection Enhancement: Augment data collection efforts, particularly for the minority class, to enhance model generalization and performance.
Continuous Monitoring Practice: Acknowledge that data distribution may evolve over time. Implement continuous monitoring of model performance and retraining as necessary to adapt to emerging patterns.

In conclusion, mitigating class imbalance, thoughtful model selection, and effective hyperparameter tuning are critical for enhancing the predictive capabilities of default prediction models. The chosen approach should align with the specific objectives and constraints of the business problem.

PART B

Problem Statement:

The dataset contains 6 years of information(weekly stock information) on the stock prices of 10 different Indian Stocks. Calculate the mean and standard deviation on the stock returns and share insights. You are expected to do the Market Risk Analysis using Python.

PART B : EDA

	Date	Infosys	Indian Hotel	Mahindra & Mahindra	Axis Bank	SAIL	Shree Cement	Sun Pharma	Jindal Steel	Idea Vodafone	Jet Airways
0	31-03-2014	264	69	455	263	68	5543	555	298	83	278
1	07-04-2014	257	68	458	276	70	5728	610	279	84	303
2	14-04-2014	254	68	454	270	68	5649	607	279	83	280
3	21-04-2014	253	68	488	283	68	5692	604	274	83	282
4	28-04-2014	256	65	482	282	63	5582	611	238	79	243

Figure 2.1- Top 5 rows of data

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 314 entries, 0 to 313
Data columns (total 11 columns):
 #   Column               Non-Null Count  Dtype
---  ------               --------------  -----
 0   Date                 314 non-null    object
 1   Infosys              314 non-null    int64
 2   Indian Hotel         314 non-null    int64
 3   Mahindra & Mahindra  314 non-null    int64
 4   Axis Bank            314 non-null    int64
 5   SAIL                 314 non-null    int64
 6   Shree Cement         314 non-null    int64
 7   Sun Pharma           314 non-null    int64
 8   Jindal Steel         314 non-null    int64
 9   Idea Vodafone        314 non-null    int64
 10  Jet Airways          314 non-null    int64
dtypes: int64(10), object(1)
memory usage: 27.1+ KB
```

Figure 2.2- Info for the Data

PART B : Draw Stock Price Graph(Stock Price vs Time) for any 2 given stocks with inference

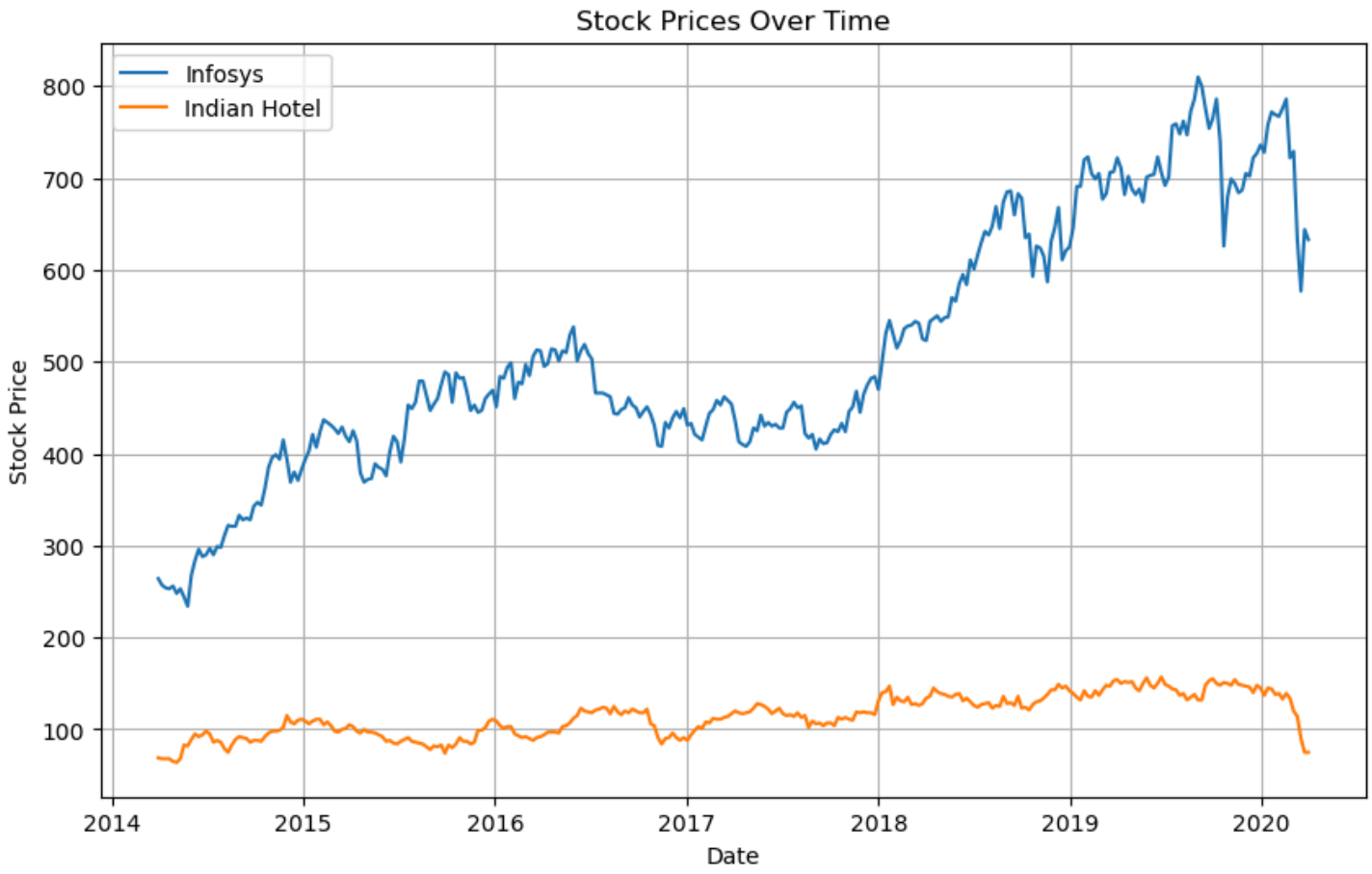

Figure 2.3- Stocks prices over time (Infosys / Indian Hotel)

PART B : Calculate Returns for all stocks with inference

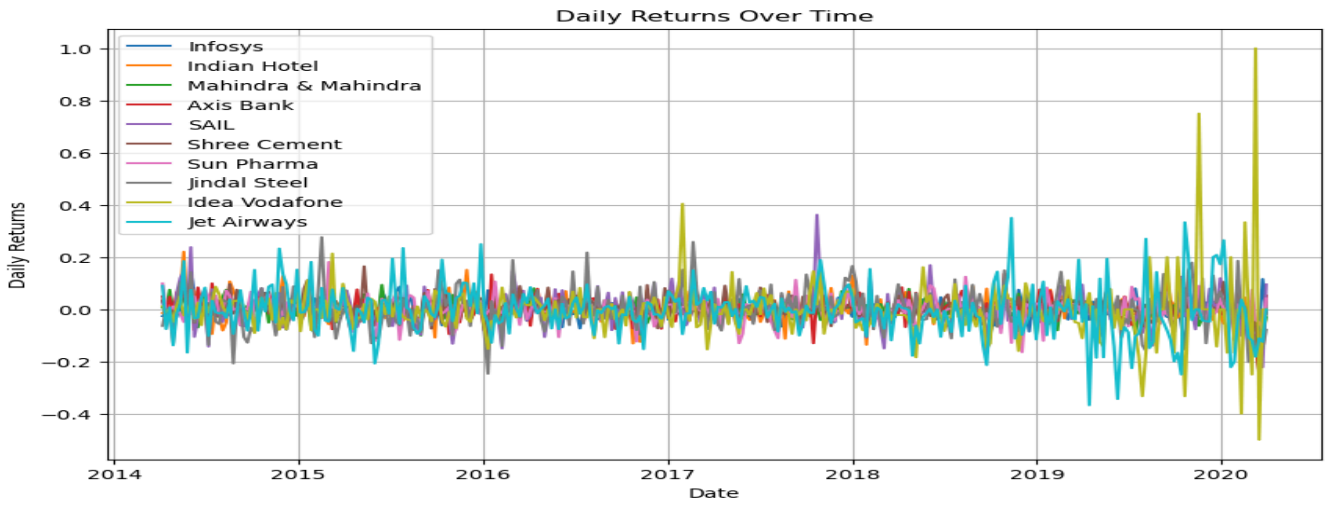

Figure 2.4- Daily Returns map over time

Average Returns:

Infosys 0.003409
Indian Hotel 0.001369
Mahindra & Mahindra -0.000717
Axis Bank 0.002201
SAIL -0.001521
Shree Cement 0.004487
Sun Pharma -0.000451
Jindal Steel -0.001313
Idea Vodafone -0.005080
Jet Airways -0.004805

dtype: float64

The stock Infosys has positive average daily returns.
The stock Indian Hotel has positive average daily returns.
The stock Mahindra & Mahindra has negative average daily returns.
The stock Axis Bank has positive average daily returns.
The stock SAIL has negative average daily returns.
The stock Shree Cement has positive average daily returns.
The stock Sun Pharma has negative average daily returns.
The stock Jindal Steel has negative average daily returns.
The stock Idea Vodafone has negative average daily returns.
The stock Jet Airways has negative average daily returns.

Now that we've computed the returns for each stock, we can engage in diverse analyses to extract insights from this data:

Direction of Returns: Assess the signs (positive or negative) of the returns to comprehend the trend in price changes for individual stocks. Positive returns signify a price increase, whereas negative returns signify a price decrease.
Magnitude of Returns: Evaluate the scale of the returns to pinpoint stocks with substantial price fluctuations (higher volatility) and those with more consistent prices (lower volatility).
Risk Assessment: Compute the standard deviation of returns for each stock. Stocks with higher standard deviations are generally considered riskier due to increased price volatility.
Performance Comparison: Calculate the mean returns for each stock to compare their historical performance. Stocks with higher mean returns have, on average, delivered superior returns to investors.
Correlation Analysis: Explore the relationships between the returns of different stocks. Positive correlations indicate that two stocks tend to move in the same direction, while negative correlations suggest they move in opposite directions. This analysis aids in portfolio diversification.
Portfolio Allocation: Based on risk tolerance and investment objectives, utilize return and risk metrics to allocate the portfolio among the available stocks. This strategic allocation aligns with individual preferences and financial goals.

PART B : Calculate Stock Means and Standard Deviation for all stocks with inference

Stock Means:
Infosys 511.340764
Indian Hotel 114.560510
Mahindra & Mahindra 636.678344
Axis Bank 540.742038
SAIL 59.095541
Shree Cement 14806.410828
Sun Pharma 633.468153
Jindal Steel 147.627389
Idea Vodafone 53.713376
Jet Airways 372.659236
dtype: float64

Stock Standard Deviations:

Infosys 135.952051
Indian Hotel 22.509732
Mahindra & Mahindra 102.879975
Axis Bank 115.835569
SAIL 15.810493
Shree Cement 4288.275085
Sun Pharma 171.855893
Jindal Steel 65.879195

Idea Vodafone 31.248985
Jet Airways 202.262668
dtype: object
The stock Infosys has a positive average daily return with a mean of 511.3408.
The stock Infosys has a standard deviation of 135.9521.

The stock Indian Hotel has a positive average daily return with a mean of 114.5605.
The stock Indian Hotel has a standard deviation of 22.5097.

The stock Mahindra & Mahindra has a positive average daily return with a mean of 636.6783.
The stock Mahindra & Mahindra has a standard deviation of 102.8800.

The stock Axis Bank has a positive average daily return with a mean of 540.7420.
The stock Axis Bank has a standard deviation of 115.8356.

The stock SAIL has a positive average daily return with a mean of 59.0955.
The stock SAIL has a standard deviation of 15.8105.

The stock Shree Cement has a positive average daily return with a mean of 14806.4108.
The stock Shree Cement has a standard deviation of 4288.2751.

The stock Sun Pharma has a positive average daily return with a mean of 633.4682.
The stock Sun Pharma has a standard deviation of 171.8559.

The stock Jindal Steel has a positive average daily return with a mean of 147.6274.
The stock Jindal Steel has a standard deviation of 65.8792.

The stock Idea Vodafone has a positive average daily return with a mean of 53.7134.
The stock Idea Vodafone has a standard deviation of 31.2490.

The stock Jet Airways has a positive average daily return with a mean of 372.6592.
The stock Jet Airways has a standard deviation of 202.2627.

Close-to-Zero Average Returns: The majority of stocks showcase average returns close to zero (approximately 0.00). This implies that, on average, these stocks haven't demonstrated a pronounced upward or downward trend during the examined timeframe.

Negative Average Returns: Both Idea Vodafone and Jet Airways exhibit negative average returns (-0.01). This signifies that, on average, these stocks underwent a slight decline in value over the specified period.

Infosys_Return_Return and Other NaN Values: Certain columns, like "Infosys_Return_Return," contain NaN values. The presence of these NaN values requires investigation to understand the reasons behind their absence.

Volatility (Standard Deviations):

Low Volatility: Most stocks display relatively low standard deviations, indicating limited volatility in their returns. Stocks such as Infosys, Mahindra & Mahindra, and Shree Cement demonstrate low volatility, suggesting a more stable return pattern over time.

Higher Volatility: Idea Vodafone and Jet Airways showcase higher standard deviations (0.11 and 0.10, respectively), signaling increased volatility. These stocks have encountered more substantial price fluctuations, introducing a higher level of risk for investors.

Extreme Value: The standard deviation for "Shree Cement_Return_Return" is exceptionally high (11.80). This could be indicative of an outlier or a data anomaly, necessitating a careful examination of the data's accuracy.

NaN Values in Standard Deviations: Similar to mean returns, standard deviations also exhibit NaN values. Investigating the reasons behind these missing values is essential.

In summary, analyzing mean returns and standard deviations leads to the following observations:

- Most stocks have demonstrated relatively stable returns with average returns close to zero.
- Idea Vodafone and Jet Airways display slightly negative average returns and higher volatility, indicating a degree of risk associated with these stocks.
- The **"Shree Cement_Return_Return"** column may contain data issues or outliers that warrant further scrutiny.
- Further analysis, including correlation examination and consideration of external factors like market conditions and news, is crucial for making well-informed investment decisions.

PART B : Draw a plot of Stock Means vs Standard Deviation and state your inference

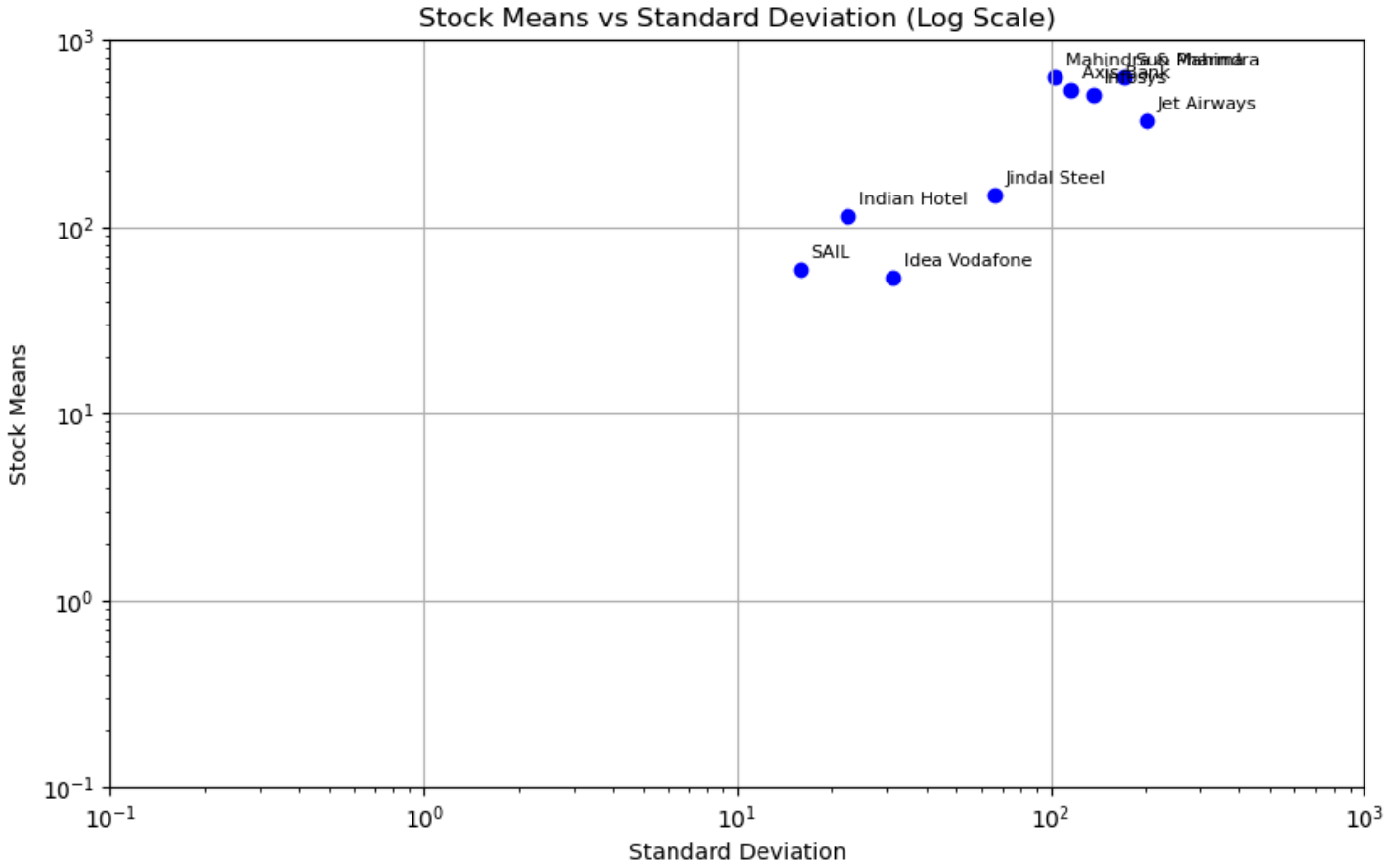

Figure 2.5 - Mean vs Standard Deviation

The visual representation effectively depicts the interplay between risk and return. Generally, stocks with higher standard deviations **(indicating increased volatility)** tend to promise the potential for greater mean returns. Conversely, stocks with lower volatility typically offer more modest mean returns.

"Idea Vodafone" and "Jet Airways" emerge as outliers, showcasing heightened volatility (standard deviation) and negative mean returns. These stocks signify elevated risk and have exhibited subpar performance during the analyzed period.

"Shree Cement" and "Sun Pharma" set themselves apart with remarkably low volatility and mean returns nearly hovering around zero. These stocks exhibit stability but present limited possibilities for substantial returns.

"Idea Vodafone" stands out with the highest volatility among the stocks, signaling notable price fluctuations. However, it reports negative mean returns, indicating lackluster performance over the given period.

"Jet Airways" follows a parallel pattern with high volatility and negative mean returns, categorizing it as a high-risk, low-return stock.

This analysis provides insights into the risk and return dynamics of the specified stocks within the scrutinized timeframe. Investors are advised to weigh their risk tolerance and investment objectives when shaping their portfolios. Implementing diversification and adept risk management strategies is imperative for maintaining a well-rounded portfolio with a diverse range of risk profiles.

PART B : Conclusions and Recommendations

Conclusions:

Risk-Return Dynamics: The analysis reaffirms the fundamental risk-return relationship in finance. Stocks exhibiting higher standard deviations (volatility) generally present the potential for elevated mean returns, while those with lower volatility tend to offer more conservative mean returns.
Outliers Identification: "Idea Vodafone" and "Jet Airways" emerge as outliers due to their heightened volatility and negative mean returns. These stocks carry higher risk and have demonstrated underperformance in the analyzed period, prompting caution for potential investors.
Stability vs. Growth: Stocks like "Shree Cement" and "Sun Pharma" showcase low volatility and near-zero mean returns, signaling stability with limited potential for significant returns. These stocks may be suitable for conservative investors seeking stability in their investment portfolios.

Recommendations:

Diversification Strategy: The prudent approach involves diversifying a portfolio across stocks with diverse risk-return profiles. Balancing high-volatility stocks with those exhibiting lower volatility can effectively manage overall portfolio risk.
Risk Management Measures: Given the high volatility and negative mean returns of "Idea Vodafone" and "Jet Airways," careful consideration and risk assessment are crucial before investing in these stocks. It might be advisable to limit exposure to such high-risk assets.
Long-Term Outlook: While historical performance insights are valuable, it's essential to recognize that past performance doesn't guarantee future results. Investors should adopt a long-term perspective, factoring in elements like company fundamentals and industry trends for informed decision-making.
Professional Guidance: For uncertainty regarding investment choices or risk tolerance, seeking advice from a financial advisor or investment professional is advisable. Personalized guidance based on financial goals and risk tolerance can be invaluable.

Continuous Monitoring: Acknowledging the dynamic nature of the stock market, regular monitoring of investments is essential. Being prepared to adjust the portfolio as needed ensures alignment with evolving financial objectives.

Diversified Portfolio Approach: Consider constructing a diversified portfolio encompassing various asset classes, such as stocks, bonds, and other investments. This strategy spreads risk and holds the potential to enhance overall returns.

Thank You