# F.R.A Main Project

## Businesses or companies

Hritika Vaishnav

# INDEX

## LIST OF TABLES

## LIST OF FIGURES

## 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.

**Dependent variable -** No need to create any new variable, as the 'Default' variable is already provided in the dataset, which can be considered as the dependent variable.

**Test Train Split -** Split the data into train and test datasets in the ratio of 67:33 and use a random state of 42 (*random_state=42*). Model building is to be done on the train dataset and model validation is to be done on the test dataset.

**Dataset:** Credit Risk Dataset

## Data Dictionary:

| S. No | Column Name | Description |
|---|---|---|
| 1 | Co_Code | Company Code |
| 2 | Co_Name | Company Name |
| 3 | _Operating_Expense_Rate | Operating Expense Rate: Operating Expenses/Net Sales. The operating expense ratio (OER) is the cost to operate a piece of property compared to the income the property brings in. |
| 4 | _Research_and_development_expense_rate | Research and development expense rate: (Research and Development Expenses)/Net Sales. Research and development (R&D) expenses are direct expenditures relating to a company's efforts to develop, design, and enhance its products, services, technologies, or processes. |
| 5 | _Cash_flow_rate | Cash flow rate: Cash Flow from Operating/Current Liabilities. Cash flow is a measure of how much cash a business brought in or spent in total over a period of time. |

| | | |
|---|---|---|
| 6 | _Interest_bearing_debt_interest_rate | Interest-bearing debt interest rate: Interest-bearing Debt/Equity |
| 7 | _Tax_rate_A | Tax rate (A): Effective Tax Rate. Effective tax rate represents the percentage of their taxable income that individuals pay in taxes. For corporations, the effective corporate tax rate is the rate they pay on their pre-tax profits. |
| 8 | _Cash_Flow_Per_Share | Cash Flow Per Share. It is the after-tax earnings plus depreciation on a per-share basis that functions as a measure of a firm's financial strength |
| 9 | _Per_Share_Net_profit_before_tax_Yuan_ | Per Share Net profit before tax (Yuan ¥): Pretax Income Per Share. Pretax income, also known as earnings before tax or pretax earnings, is the net income earned by a business before taxes are subtracted/accounted for. |
| 10 | _Realized_Sales_Gross_Profit_Growth_Rate | Realized Sales Gross Profit Growth Rate. |
| 11 | _Operating_Profit_Growth_Rate | Operating Profit Growth Rate: Operating Income Growth. It is the rate of increase in operating income over the last year. |
| 12 | _Continuous_Net_Profit_Growth_Rate | Continuous Net Profit Growth Rate: Net Income-Excluding Disposal Gain or Loss Growth |
| 13 | _Total_Asset_Growth_Rate | Total Asset Growth Rate: Total Asset Growth. It is the rate at which how quickly the company has been growing its Assets |
| 14 | _Net_Value_Growth_Rate | Net Value Growth Rate: Total Equity Growth |
| 15 | _Total_Asset_Return_Growth_Rate_Ratio | Total Asset Return Growth Rate Ratio: Return on Total Asset Growth |
| 16 | _Cash_Reinvestment_perc | Cash Reinvestment %: Cash Reinvestment Ratio. It is the valuation ratio that is used to measure the percentage of annual cash flow that the company invests back into the business as a new investment. |
| 17 | _Current_Ratio | Current Ratio. The current ratio describes the relationship between a company's assets and liabilities |
| 18 | _Quick_Ratio | Quick Ratio: Acid Test. Acid-test ratio (also known as quick ratio) is a measure |

| | | |
|---|---|---|
| | | of a company's liquidity, which is its ability to pay its short-term obligations using only its most liquid assets. |
| 19 | _Interest_Expense_Ratio | Interest Expense Ratio: Interest Expenses/Total Revenue |
| 20 | _Total_debt_to_Total_net_worth | Total debt/Total net worth: Total Liability/Equity Ratio |
| 21 | _Long_term_fund_suitability_ratio_A | Long-term fund suitability ratio (A): (Long-term Liability+Equity)/Fixed Assets |
| 22 | _Net_profit_before_tax_to_Paid_in_capital | Net profit before tax/Paid-in capital: Pretax Income/Capital |
| 23 | _Total_Asset_Turnover | Total Asset Turnover. Net Sales/Average Total Assets |
| 24 | _Accounts_Receivable_Turnover | Accounts Receivable Turnover. The accounts receivable turnover ratio, or receivables turnover, is used in business accounting to quantify how well companies are managing the credit that they extend to their customers by evaluating how long it takes to collect the outstanding debt throughout the accounting period. |
| 25 | _Average_Collection_Days | Average Collection Days: Days Receivable Outstanding |
| 26 | _Inventory_Turnover_Rate_times | Inventory Turnover Rate (times). The inventory turnover ratio is the number of times a company has sold and replenished its inventory over a specific amount of time. The formula can also be used to calculate the number of days it will take to sell the inventory on hand. |
| 27 | _Fixed_Assets_Turnover_Frequency | Fixed Assets Turnover Frequency. Fixed Asset Turnover (FAT) is an efficiency ratio that indicates how well or efficiently a business uses fixed assets to generate sales. This ratio divides net sales by net fixed assets, calculated over an annual period. |
| 28 | _Net_Worth_Turnover_Rate_times | Net Worth Turnover Rate (times): Equity Turnover. Equity turnover is a ratio that measures the proportion of a company's sales to its stockholders' equity. The intent of the measurement is to determine the efficiency with which |

| | | |
|---|---|---|
| | | management is using equity to generate revenue. |
| 29 | _Operating_profit_per_person | Operating profit per person: Operation Income Per Employee |
| 30 | _Allocation_rate_per_person | Allocation rate per person: Fixed Assets Per Employee |
| 31 | _Quick_Assets_to_Total_Assets | Quick Assets/Total Assets |
| 32 | _Cash_to_Total_Assets | Cash/Total Assets |
| 33 | _Quick_Assets_to_Current_Liability | Quick Assets/Current Liability |
| 34 | _Cash_to_Current_Liability | Cash/Current Liability |
| 35 | _Operating_Funds_to_Liability | Operating Funds to Liability |
| 36 | _Inventory_to_Working_Capital | Inventory/Working Capital |
| 37 | _Inventory_to_Current_Liability | Inventory/Current Liability |
| 38 | _Long_term_Liability_to_Current_Assets | Long-term Liability to Current Assets |
| 39 | _Retained_Earnings_to_Total_Assets | Retained Earnings to Total Assets |
| 40 | _Total_income_to_Total_expense | Total income/Total expense |
| 41 | _Total_expense_to_Assets | Total expense/Assets |
| 42 | _Current_Asset_Turnover_Rate | Current Asset Turnover Rate: Current Assets to Sales. The current assets turnover ratio indicates how many times the current assets are turned over in the form of sales within a specific period of time. A higher asset turnover ratio means a better percentage of sales. |
| 43 | _Quick_Asset_Turnover_Rate | Quick Asset Turnover Rate: Quick Assets to Sales. The asset turnover ratio measures the efficiency of a company's assets in generating revenue or sales. |
| 44 | _Cash_Turnover_Rate | Cash Turnover Rate: Cash to Sales. The cash turnover ratio is an efficiency ratio that reveals the number of times that cash is turned over in an accounting period. |
| 45 | _Fixed_Assets_to_Assets | Fixed Assets to Assets. Fixed assets are also known as non-current assets—assets that can't be easily converted into cash. |
| 46 | _Cash_Flow_to_Total_Assets | Cash Flow to Total Assets. This ratio indicates the cash a company can generate in relation to its size. |
| 47 | _Cash_Flow_to_Liability | Cash Flow to Liability. The amount of money available to run business operations and complete transactions. This is calculated as current assets (cash or near-cash assets, like notes |

| | | |
|---|---|---|
| | | receivable) minus current liabilities (liabilities due during the upcoming accounting period) |
| 48 | _CFO_to_Assets | CFO to Assets. Cash flow on total assets is an efficiency ratio that rates cash flows to the company assets without being affected by income recognition or income measurements. |
| 49 | _Cash_Flow_to_Equity | Cash Flow to Equity. cash flow to equity is a measure of how much cash is available to the equity shareholders of a company after all expenses, reinvestment, and debt are paid. |
| 50 | _Current_Liability_to_Current_Assets | Current Liability to Current Assets. Current liabilities are a company's financial commitments that are due and payable within a year, Current assets are projected to be consumed, sold, or converted into cash within a year or within the operational cycle. |
| 51 | _Liability_Assets_Flag | Liability-Assets Flag: 1 if Total Liability exceeds Total Assets, 0 otherwise |
| 52 | _Total_assets_to_GNP_price | Total assets to GNP price. Gross National Product (GNP) is the total value of all finished goods and services produced by a country’s citizens in a given financial year, irrespective of their location. |
| 53 | _No_credit_Interval | No-credit Interval |
| 54 | _Degree_of_Financial_Leverage_DFL | Degree of Financial Leverage (DFL). The degree of financial leverage is a financial ratio that measures the sensitivity in fluctuations of a company's overall profitability to the volatility of its operating income caused by changes in its capital structure. |
| 55 | _Interest_Coverage_Ratio_Interest_expense_to_EBIT | Interest Coverage Ratio (Interest expense to EBIT). The interest coverage ratio is a debt and profitability ratio used to determine how easily a company can pay interest on its outstanding debt. The interest coverage ratio is calculated by dividing a company's earnings before interest and taxes (EBIT) by its interest expense during a given period. |

| | | |
|---|---|---|
| 56 | _Net_Income_Flag | Net Income Flag: 1 if Net Income is Negative for the last two years, 0 otherwise |
| 57 | _Equity_to_Liability | Equity to Liability Ratio. |
| 58 | Default | Whether the Company has Default (Bankrupted) or not? 1 - Defaulted, 0 - Not Defaulted. |

## Objective:

As a data scientist, my goal is to construct a predictive model utilizing machine learning methodologies to evaluate credit risk for businesses based on their financial statements. The objective is to precisely categorize companies as defaulting or non-defaulting according to their financial indicators, thereby aiding investors and lenders in making well-informed decisions.

**To accomplish this, I will undertake the following steps:**

- Pre-process the Credit Risk Dataset, which involves addressing missing values, encoding categorical variables, and scaling numerical features as necessary.
- Conduct exploratory data analysis (EDA) to gain insights into variable distributions, identify potential patterns or correlations, and comprehend the characteristics of defaulting and non-defaulting companies.
- Perform feature selection to pinpoint the most pertinent financial indicators that substantially influence the likelihood of default.
- Partition the dataset into training and testing subsets, adhering to a 67:33 ratio with a random state of 42 to ensure reproducibility.
- Employ machine learning algorithms such as logistic regression, random forests, and LDA to construct predictive models for credit risk assessment.
- Assess the models' performance using suitable metrics such as accuracy, precision, recall, F1-score, and ROC-AUC score on the test dataset.
- Validate the final model(s) through cross-validation to guarantee robustness and generalizability.
- Offer insights and recommendations grounded on the model results to stakeholders, encompassing investors and financial institutions, to aid in their decision-making process regarding investment and lending activities.
- Thoroughly document the entire process, including data pre-processing steps, model selection criteria, evaluation metrics, and recommendations, to ensure transparency and reproducibility of the analysis.

Import all the necessary and load our data set, CompData-1.xlsx and use the head() function to view the Top 5 data and the tail() function to view the bottom 5 data. Using the shape function, we can determine that there are 2058 rows and 58 column and size of dataset is 119364. Find out the characteristics of the column using the info() method. The datatypes for the float64(53), int64(4), and object(1) are present and 298 null values are present in the dataset.

**Sample of dataset:** The provided dataset consists of financial metrics for various companies. Each entry includes details, such as 58 columns of features. Additionally, there is a column indicating whether a company defaulted or not.

| | Co_Code | Co_Name | _Operating_Expense_Rate | _Research_and_development_expense_rate | _Cash_flow_rate | _Interest_bearing_debt_interest_rate | _Tax_rate_A |
|---|---|---|---|---|---|---|---|
| 0 | 16974 | Hind.Cables | 8.820000e+09 | 0.000000e+00 | 0.462045 | 0.000352 | 0.00141 |
| 1 | 21214 | Tata Tele. Mah. | 9.380000e+09 | 4.230000e+09 | 0.460116 | 0.000716 | 0.00000 |
| 2 | 14852 | ABG Shipyard | 3.800000e+09 | 8.150000e+08 | 0.449893 | 0.000496 | 0.00000 |
| 3 | 2439 | GTL | 6.440000e+09 | 0.000000e+00 | 0.462731 | 0.000592 | 0.00931 |
| 4 | 23505 | Bharati Defence | 3.680000e+09 | 0.000000e+00 | 0.463117 | 0.000782 | 0.40024 |

5 rows × 58 columns

Table 1: Dataset Sample

**Fixing messy column names (Containing extra _) for ease of use:** The dataset have messy column names so rewrite the columns name.

```
Index(['Co_Code', 'Co_Name', 'Operating_Expense_Rate',
       'Research_and_development_expense_rate', 'Cash_flow_rate',
       'Interest_bearing_debt_interest_rate', 'Tax_rate_A',
       'Cash_Flow_Per_Share', 'Per_Share_Net_profit_before_tax_Yuan',
       'Realized_Sales_Gross_Profit_Growth_Rate',
       'Operating_Profit_Growth_Rate', 'Continuous_Net_Profit_Growth_Rate',
       'Total_Asset_Growth_Rate', 'Net_Value_Growth_Rate',
       'Total_Asset_Return_Growth_Rate_Ratio', 'Cash_Reinvestment_perc',
       'Current_Ratio', 'Quick_Ratio', 'Interest_Expense_Ratio',
       'Total_debt_to_Total_net_worth', 'Long_term_fund_suitability_ratio_A',
       'Net_profit_before_tax_to_Paid_in_capital', 'Total_Asset_Turnover',
       'Accounts_Receivable_Turnover', 'Average_Collection_Days',
       'Inventory_Turnover_Rate_times', 'Fixed_Assets_Turnover_Frequency',
       'Net_Worth_Turnover_Rate_times', 'Operating_profit_per_person',
       'Allocation_rate_per_person', 'Quick_Assets_to_Total_Assets',
       'Cash_to_Total_Assets', 'Quick_Assets_to_Current_Liability',
       'Cash_to_Current_Liability', 'Operating_Funds_to_Liability',
       'Inventory_to_Working_Capital', 'Inventory_to_Current_Liability',
       'Long_term_Liability_to_Current_Assets',
       'Retained_Earnings_to_Total_Assets', 'Total_income_to_Total_expense',
       'Total_expense_to_Assets', 'Current_Asset_Turnover_Rate',
       'Quick_Asset_Turnover_Rate', 'Cash_Turnover_Rate',
       'Fixed_Assets_to_Assets', 'Cash_Flow_to_Total_Assets',
       'Cash_Flow_to_Liability', 'CFO_to_Assets', 'Cash_Flow_to_Equity',
       'Current_Liability_to_Current_Assets', 'Liability_Assets_Flag',
       'Total_assets_to_GNP_price', 'No_credit_Interval',
       'Degree_of_Financial_Leverage_DFL',
       'Interest_Coverage_Ratio_Interest_expense_to_EBIT', 'Net_Income_Flag',
       'Equity_to_Liability', 'Default'],
      dtype='object')
```

Table 2: Columns Name

**Checking data types of all columns:** There are total 2058 rows and 58 columns in the dataset. Out of 58, 1 column is of object type and rest 57 are of either integer or float data type.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2058 entries, 0 to 2057
Data columns (total 58 columns):
 #   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  Net_Income_Flag                                   2058 non-null   int64
 56  Equity_to_Liability                               2058 non-null   float64
 57  Default                                           2058 non-null   int64
dtypes: float64(53), int64(4), object(1)
memory usage: 932.7+ KB
```

Table 3: Types of variables in the data frame

**Now, let us check the basic measures of descriptive statistics:** The dataset comprises various financial indicators for companies, including operating expense rate, research and development expense rate, cash flow rate, etc. Descriptive statistics reveal significant variations across metrics, with diverse means, standard deviations, and ranges. Notably, the default rate stands at approximately 10.7%.

| | Co_Code | Operating_Expense_Rate | Research_and_development_expense_rate | Cash_flow_rate | Interest_bearing_debt_interest_rate |
|---|---|---|---|---|---|
| count | 2058.000000 | 2.058000e+03 | 2.058000e+03 | 2058.000000 | 2.058000e+03 |
| mean | 17572.113217 | 2.052389e+09 | 1.208634e+09 | 0.465243 | 1.113022e+07 |
| std | 21892.886518 | 3.252624e+09 | 2.144568e+09 | 0.022663 | 9.042595e+07 |
| min | 4.000000 | 1.000260e-04 | 0.000000e+00 | 0.000000 | 0.000000e+00 |
| 25% | 3674.000000 | 1.578727e-04 | 0.000000e+00 | 0.460099 | 2.760280e-04 |
| 50% | 6240.000000 | 3.330330e-04 | 1.994130e-04 | 0.463445 | 4.540450e-04 |
| 75% | 24280.750000 | 4.110000e+09 | 1.550000e+09 | 0.468069 | 6.630660e-04 |
| max | 72493.000000 | 9.980000e+09 | 9.980000e+09 | 1.000000 | 9.900000e+08 |

8 rows × 57 columns

Table 4: Descriptive statistics of the data frame

## 1: Outlier Treatment & Missing Value Treatment

**Missing Values check:** The dataset contains missing values in several variables, with "Cash_Flow_Per_Share" having 167 missing values, "Total_debt_to_Total_net_worth" with 21 missing values, "Cash_to_Total_Assets" with 96 missing values, and "Current_Liability_to_Current_Assets" with 14 missing values. In total, there are 298 missing values across the dataset.

```
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
```

Table 5: Missing values of the data frame

**Check Outlier:** Splitting Dataset into Businesses_X and Businesses_Y(We do not apply outli er treatment to the target variable. And drop the unuseful column 'Co_Code', 'Co_Name'). In the dataset total 10864 outliers are present.

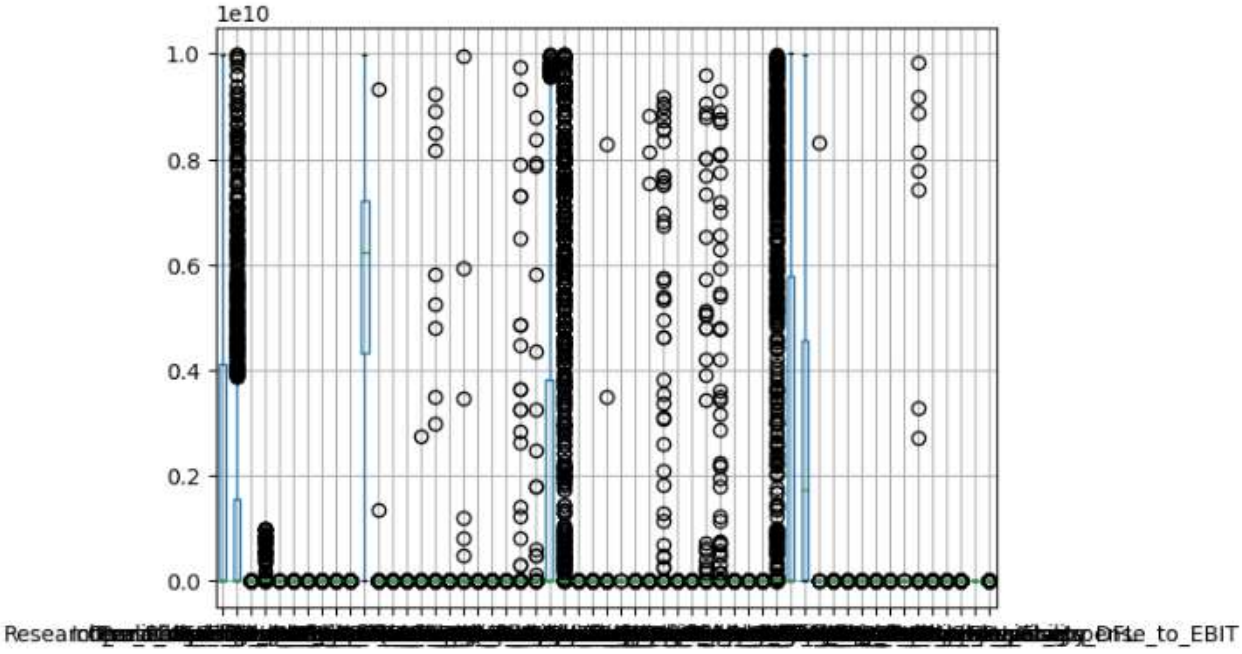


Fig 1: Outliers of dataset

**Outlier's treatment using the Capping Method:** The Capping Method in outlier treatment involves setting a predefined upper and lower limit to cap extreme values in the dataset. This approach helps maintain data integrity while mitigating the impact of outliers on statistical analyses.

```
0.000111127 0.00015787275 4110000000.0 8971499999.999998 <built-in function min> <built-in function max>
0.0 0.0 1550000000.0 6235999999.999994 <built-in function min> <built-in function max>
0.45200791775 0.460099142 0.46806908049999996 0.48237647095 <built-in function min> <built-in function max>
0.0 0.000276028 0.000663066 0.0010774079999999998 <built-in function min> <built-in function max>
0.0 0.0 0.21619090975 0.3416236463499999 <built-in function min> <built-in function max>
nan nan nan nan <built-in function min> <built-in function max>
0.14042328805 0.166603901 0.185885366 0.21451758669999998 <built-in function min> <built-in function max>
0.0219479388 0.022058314000000002 0.022151999 0.022579723299999997 <built-in function min> <built-in function max>
0.84770057975 0.84797396475 0.848114747 0.8485231827 <built-in function min> <built-in function max>
0.21737961015 0.217574132 0.21761982075 0.21774071545 <built-in function min> <built-in function max>
0.0001133856 4315000000.0 7220000000.0 8980000000.0 <built-in function min> <built-in function max>
0.00036385015 0.00043628325 0.00048837575 0.0006664002999999998 <built-in function min> <built-in function max>
0.2628939083 0.2637383345 0.26430966375 0.26533747480000003 <built-in function min> <built-in function max>
0.341920942 0.37072948099999997 0.38555754275 0.40518423324999997 <built-in function min> <built-in function max>
0.0034248259000000006 0.006567062 0.01350541775 0.034290064949999956 <built-in function min> <built-in function max>
0.0005525975 0.00294639875 0.00890298325 0.026455419749999994 <built-in function min> <built-in function max>
0.62819612065 0.630611567 0.63174365775 0.6348716631 <built-in function min> <built-in function max>
nan nan nan nan <built-in function min> <built-in function max>
0.004971702 0.0051620305 0.00641530075 0.012746516849999993 <built-in function min> <built-in function max>
0.143312722 0.16586230275 0.18444498925 0.21076238605 <built-in function min> <built-in function max>
0.02226386835000001 0.061469265 0.167916042 0.326836582 <built-in function min> <built-in function max>
0.00046838365 0.000744626 0.001854463 0.011880129349999945 <built-in function min> <built-in function max>
0.00070908575 0.0035763845 0.008638997 0.014823774849999995 <built-in function min> <built-in function max>
0.00011044 0.00019092974999999999 3815000000.0 8712999999.999998 <built-in function min> <built-in function max>
0.00011877 0.00022789499999999999 0.008423224 8020000000.0 <built-in function min> <built-in function max>
0.012903226 0.020483871 0.044354839 0.09857258049999998 <built-in function min> <built-in function max>
0.37294271975000004 0.391386445 0.40089267525 0.46110219129999996 <built-in function min> <built-in function max>
0.0009401182500000001 0.00467161225 0.024574907 0.09450514144999984 <built-in function min> <built-in function max>
```

Fig 2: Outliers Capping Method of dataset

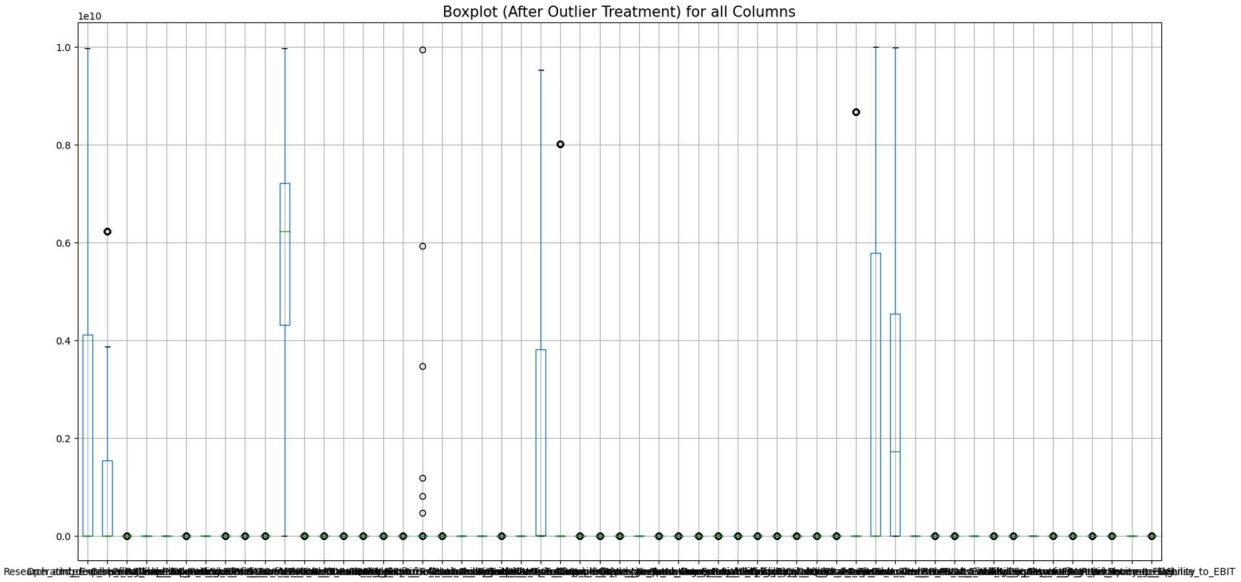


Fig 3: After Outliers Treatment of dataset

**Scale the predictors:** Scaling predictors standardizes their values, ensuring variables with different scales contribute equally to model training. It enhances algorithm performance and prevents dominance by variables with larger magnitudes.

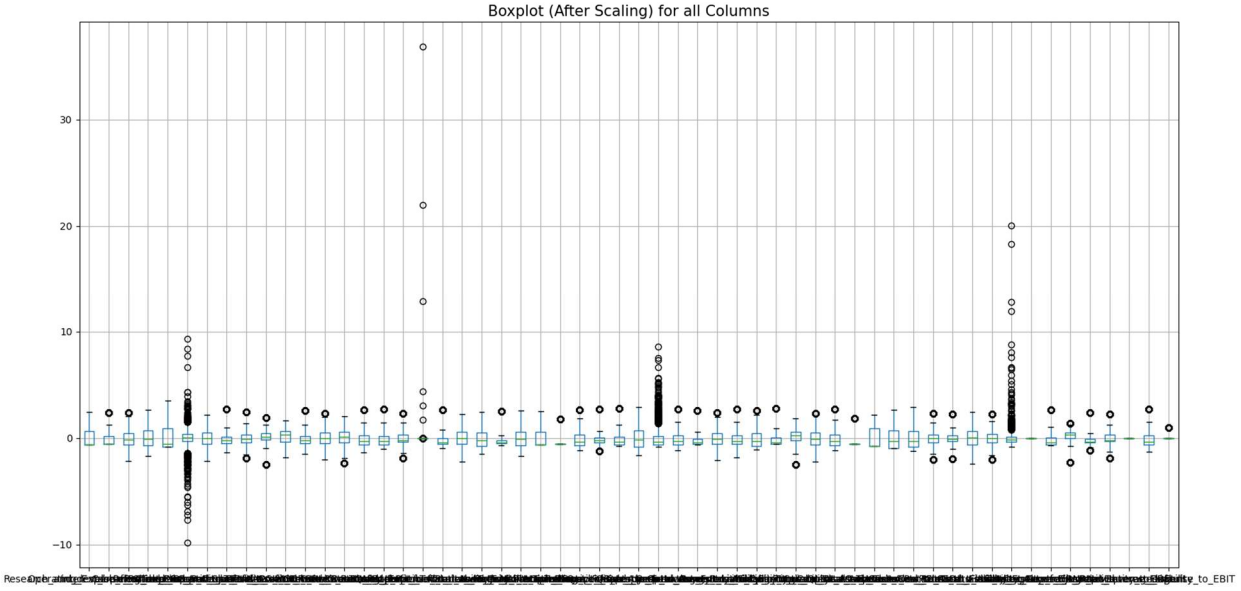


Fig 4: After Scaling of dataset

**Imputing the Missing values KNN Imputer:** KNN Imputer fills missing values by calculating the average of nearest neighbours' values. This method leverages similarities between data points to estimate missing values, aiding in maintaining data integrity.

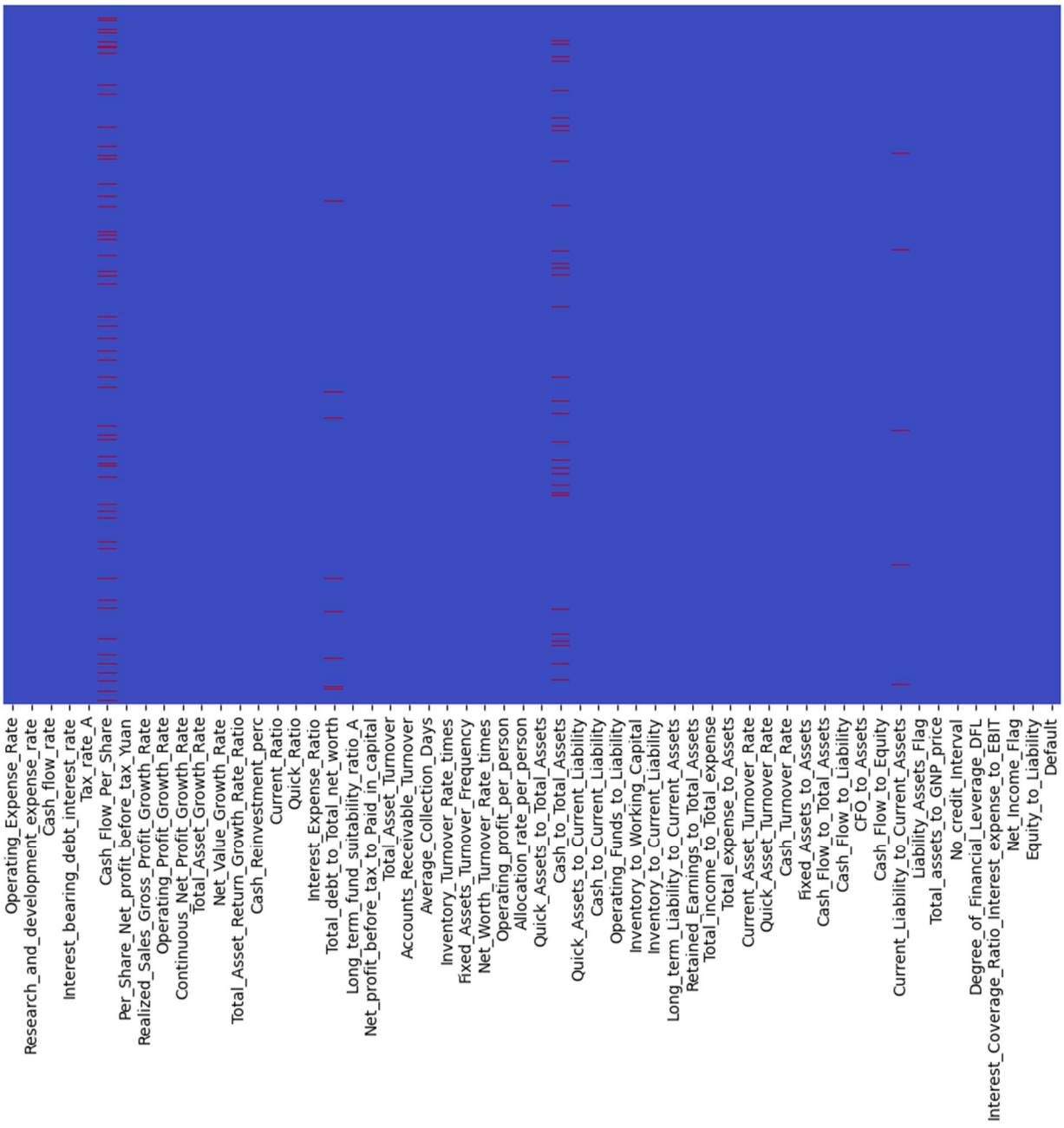


Fig 5: Visually inspect the missing values in our data

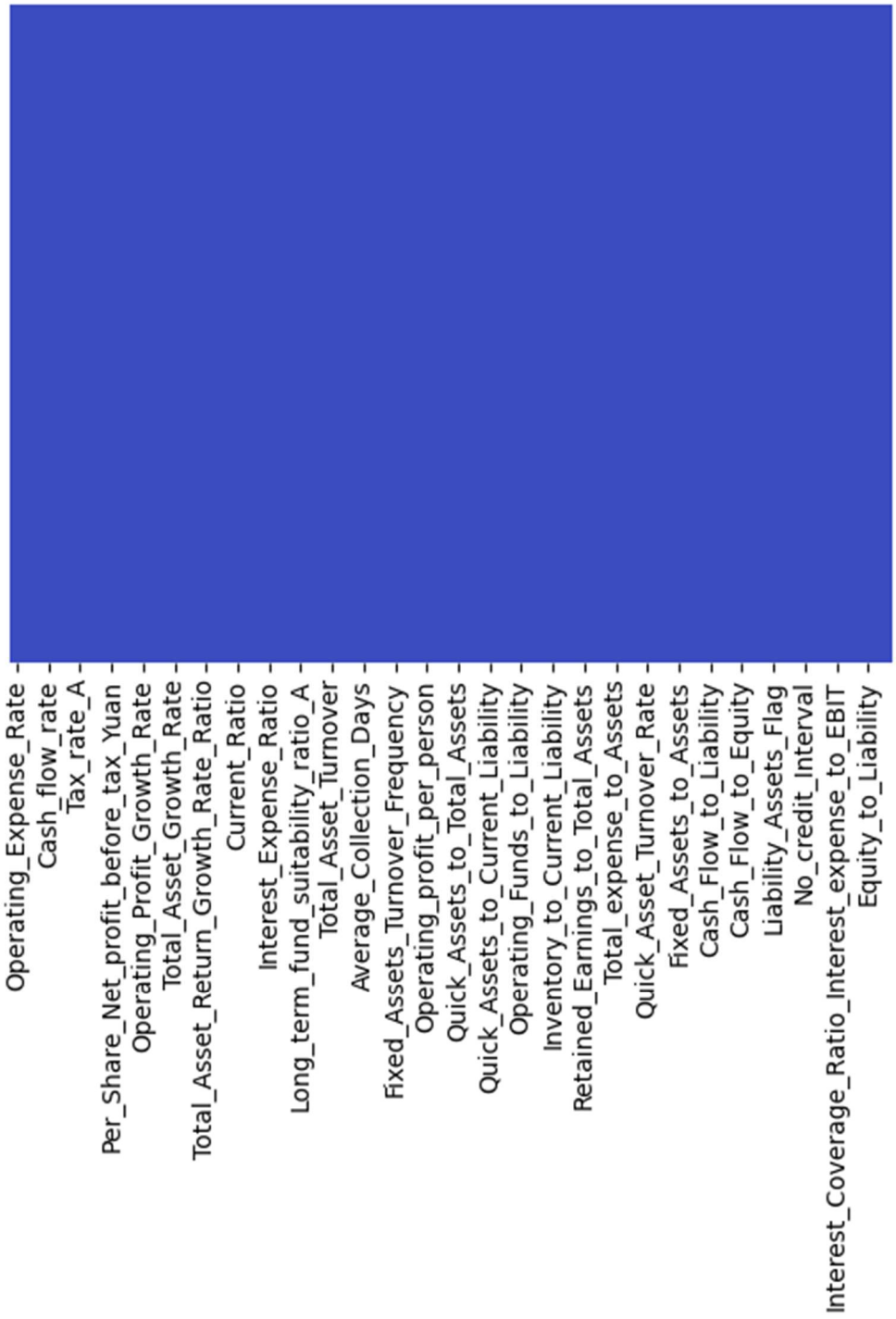


Fig 6: Visualising missing values in dataset after imputing

## 2: 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)

We will perform EDA after VIF and feature selection using p-values because we will include only those variables that were significant in the model building.

The Variance Inflation Factor (VIF) is a measure used in regression analysis to quantify the severity of multi collinearity in a set of predictor variables. Multi collinearity occurs when two or more independent variables in a regression model are highly correlated with each other, which can lead to inaccurate and unstable estimates of the regression coefficients.

A VIF of 1 indicates no multi collinearity, meaning that the predictor variable is not correlated with any other predictors. Generally, a VIF greater than 10 or 5 indicates high multi collinearity and suggests that the predictor variable may need to be addressed.

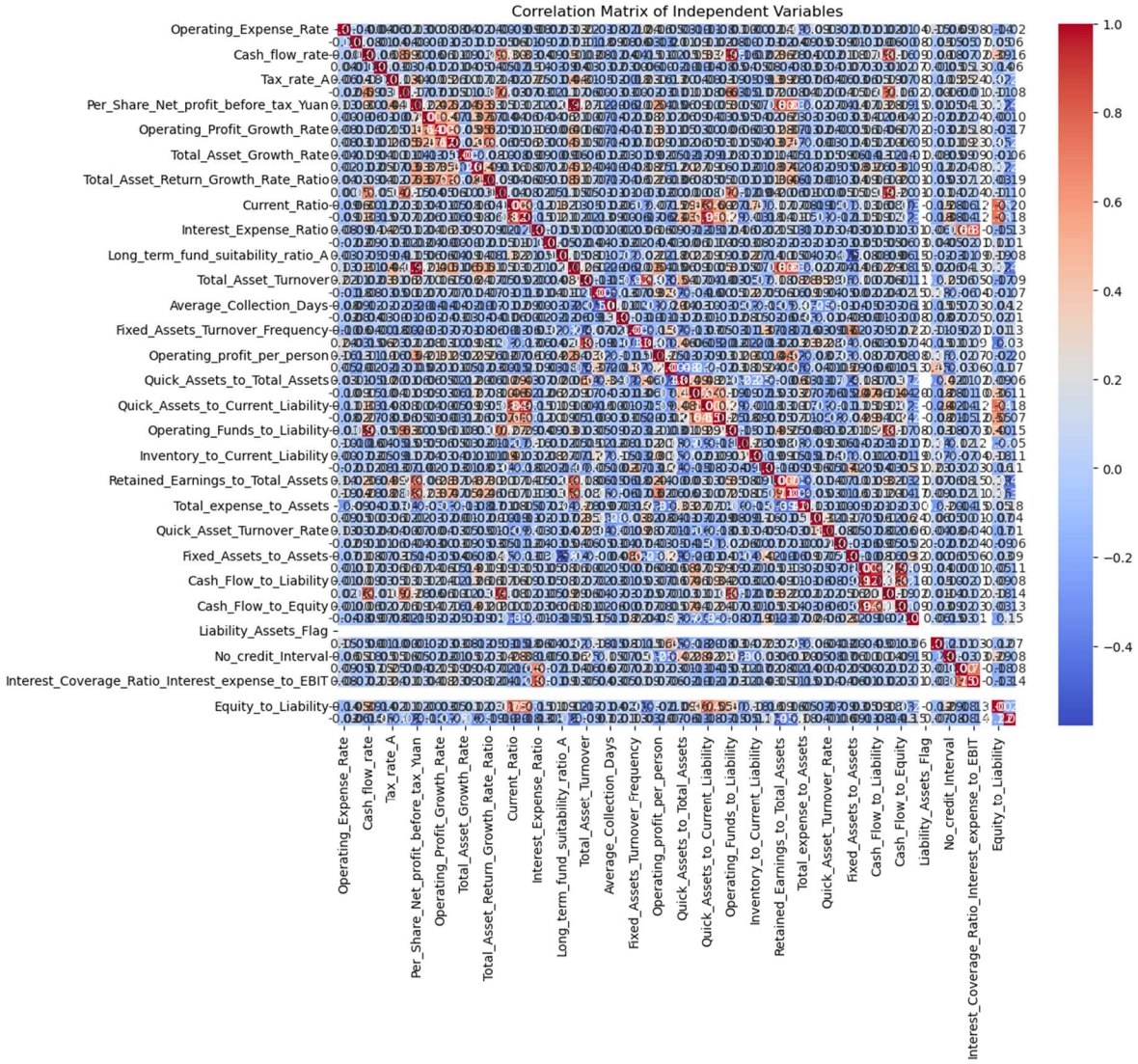


Fig 7: Correlation Matrix of Independent Variables

In our current scenario, where we have a large number of columns (56), it's important to tidy up our data by removing any redundant columns before we start building our model. This is because having too many redundant columns can lead to multi collinearity issues, which can make our model less reliable.

| | variables | VIF |
|---|---|---|
| 12 | Cash_Reinvestment_perc | 3.798356 |
| 13 | Quick_Ratio | 3.570125 |
| 37 | Cash_Flow_to_Equity | 3.526813 |
| 30 | Total_income_to_Total_expense | 3.515053 |
| 36 | Cash_Flow_to_Liability | 3.451792 |
| 35 | Fixed_Assets_to_Assets | 3.330766 |
| 2 | Cash_flow_rate | 3.280520 |
| 44 | Equity_to_Liability | 3.256994 |
| 25 | Cash_to_Current_Liability | 2.867382 |
| 29 | Retained_Earnings_to_Total_Assets | 2.833460 |
| 5 | Cash_Flow_Per_Share | 2.779877 |
| 11 | Total_Asset_Return_Growth_Rate_Ratio | 2.629121 |
| 7 | Operating_Profit_Growth_Rate | 2.612923 |
| 24 | Cash_to_Total_Assets | 2.610480 |
| 8 | Continuous_Net_Profit_Growth_Rate | 2.562678 |
| 21 | Net_Worth_Turnover_Rate_times | 2.409823 |
| 22 | Operating_profit_per_person | 2.281374 |
| 23 | Allocation_rate_per_person | 2.205081 |
| 6 | Realized_Sales_Gross_Profit_Growth_Rate | 2.115729 |
| 20 | Fixed_Assets_Turnover_Frequency | 2.097329 |
| 10 | Net_Value_Growth_Rate | 2.045051 |
| 31 | Total_expense_to_Assets | 1.959762 |
| 42 | Degree_of_Financial_Leverage_DFL | 1.933918 |
| 14 | Interest_Expense_Ratio | 1.925695 |
| 16 | Long_term_fund_suitability_ratio_A | 1.906793 |
| 18 | Average_Collection_Days | 1.798425 |
| 17 | Accounts_Receivable_Turnover | 1.731171 |
| 27 | Inventory_to_Current_Liability | 1.657648 |
| 32 | Current_Asset_Turnover_Rate | 1.605884 |
| 41 | No_credit_Interval | 1.555483 |
| 28 | Long_term_Liability_to_Current_Assets | 1.514828 |
| 4 | Tax_rate_A | 1.471498 |
| 40 | Total_assets_to_GNP_price | 1.464398 |
| 38 | Current_Liability_to_Current_Assets | 1.450180 |
| 33 | Quick_Asset_Turnover_Rate | 1.376803 |
| 26 | Inventory_to_Working_Capital | 1.341329 |
| 0 | Operating_Expense_Rate | 1.285937 |
| 19 | Inventory_Turnover_Rate_times | 1.207055 |
| 9 | Total_Asset_Growth_Rate | 1.201846 |
| 3 | Interest_bearing_debt_interest_rate | 1.134484 |
| 1 | Research_and_development_expense_rate | 1.133971 |
| 34 | Cash_Turnover_Rate | 1.125843 |
| 15 | Total_debt_to_Total_net_worth | 1.050925 |
| 39 | Liability_Assets_Flag | NaN |
| 43 | Net_Income_Flag | NaN |

Table 6: Now our dataset having all variables with VIF <5

## 3: Train Test Split (Split the data into train and test datasets in the ratio of 67:33 and use a random state of 42 (random_state=42))

The data into training and testing subsets with a ratio of 67:33, ensuring reproducibility by setting the random state to 42. This split enables model building on the training dataset while evaluating performance on the unseen test dataset. Dropping columns having constant value ('Liability_Assets_Flag','Net_Income_Flag').

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

In this task, a logistic regression model is built using the stats models library on the train dataset with the most important variables. The optimal cut off is determined to improve classification accuracy. For calculating p-values, according to the p-values, we select the significant feature.

**model_34 is the optimized model once all non-significant characteristics have been eliminated**

```
Optimization terminated successfully.
         Current function value: 0.206697
         Iterations 9
```

Logit Regression Results

| | | | |
|---|---|---|---|
| **Dep. Variable:** | Default | **No. Observations:** | 1378 |
| **Model:** | Logit | **Df Residuals:** | 1367 |
| **Method:** | MLE | **Df Model:** | 10 |
| **Date:** | Sat, 24 Feb 2024 | **Pseudo R-squ.:** | 0.4072 |
| **Time:** | 22:48:52 | **Log-Likelihood:** | -284.83 |
| **converged:** | True | **LL-Null:** | -480.46 |
| **Covariance Type:** | nonrobust | **LLR p-value:** | 6.808e-78 |

| | coef | std err | z |
|---|---|---|---|
| **Intercept** | -4.0809 | 0.266 | -15.332 |
| **Research_and_development_expense_rate** | 0.3497 | 0.104 | 3.365 |
| **Interest_bearing_debt_interest_rate** | 0.4488 | 0.134 | 3.357 |
| **Cash_Reinvestment_perc** | -0.3138 | 0.104 | -3.009 |
| **Quick_Ratio** | -0.7934 | 0.271 | -2.930 |

| | coef | std err | z | P>\|z\| | [0.025 | 0.975] |
|---|---|---|---|---|---|---|
| Intercept | -4.0809 | 0.266 | -15.332 | 0.000 | -4.603 | -3.559 |
| Research_and_development_expense_rate | 0.3497 | 0.104 | 3.365 | 0.001 | 0.146 | 0.553 |
| Interest_bearing_debt_interest_rate | 0.4488 | 0.134 | 3.357 | 0.001 | 0.187 | 0.711 |
| Cash_Reinvestment_perc | -0.3138 | 0.104 | -3.009 | 0.003 | -0.518 | -0.109 |
| Quick_Ratio | -0.7934 | 0.271 | -2.930 | 0.003 | -1.324 | -0.263 |
| Accounts_Receivable_Turnover | -0.3478 | 0.146 | -2.382 | 0.017 | -0.634 | -0.062 |
| Average_Collection_Days | 0.2977 | 0.111 | 2.679 | 0.007 | 0.080 | 0.516 |
| Allocation_rate_per_person | 0.4903 | 0.109 | 4.498 | 0.000 | 0.277 | 0.704 |
| Retained_Earnings_to_Total_Assets | -0.8749 | 0.142 | -6.154 | 0.000 | -1.153 | -0.596 |
| Total_income_to_Total_expense | -0.7617 | 0.201 | -3.780 | 0.000 | -1.157 | -0.367 |
| Equity_to_Liability | -1.3277 | 0.333 | -3.982 | 0.000 | -1.981 | -0.674 |

Table 7: Optimized model

**EDA:** EDA will be performed using the significant Predictors only that are identified after VIF and feature selection using p-values.

**Univariate Analysis:** The "Not Defaulted" count is greater than 1750, and the "Defaulted" count is approximately 250. While performing Univariate analysis I have found that Distribution of Retained Earnings to Total Assets Ratio column variable are lefty skewed and most of the column variable are rightly skewed, Distribution of Retained Earnings to Total Assets Ratio column variable are lefty skewed hence the presence of outlier expressed in the right in the right side due to the mean is greater than the median in all the parameters

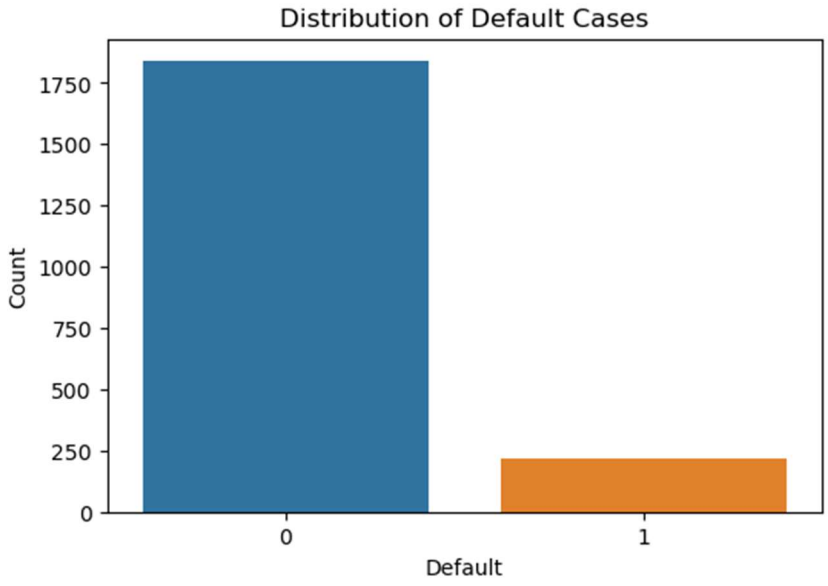


Fig 8: Distribution of Default Cases

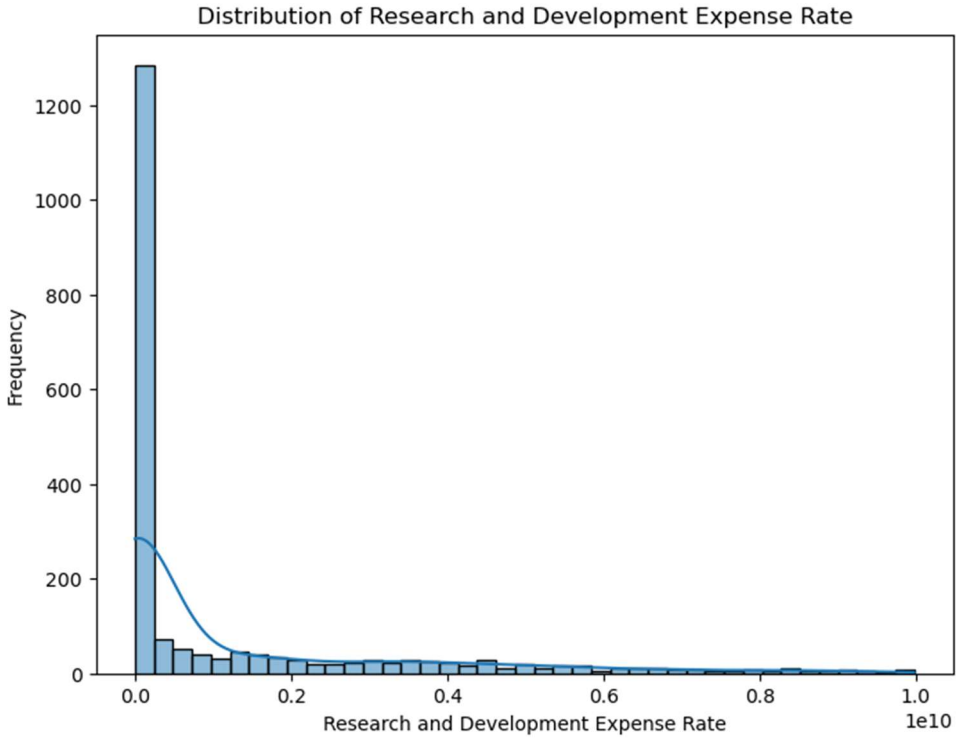


Fig 9: Distribution of Research and Development Expense Rate

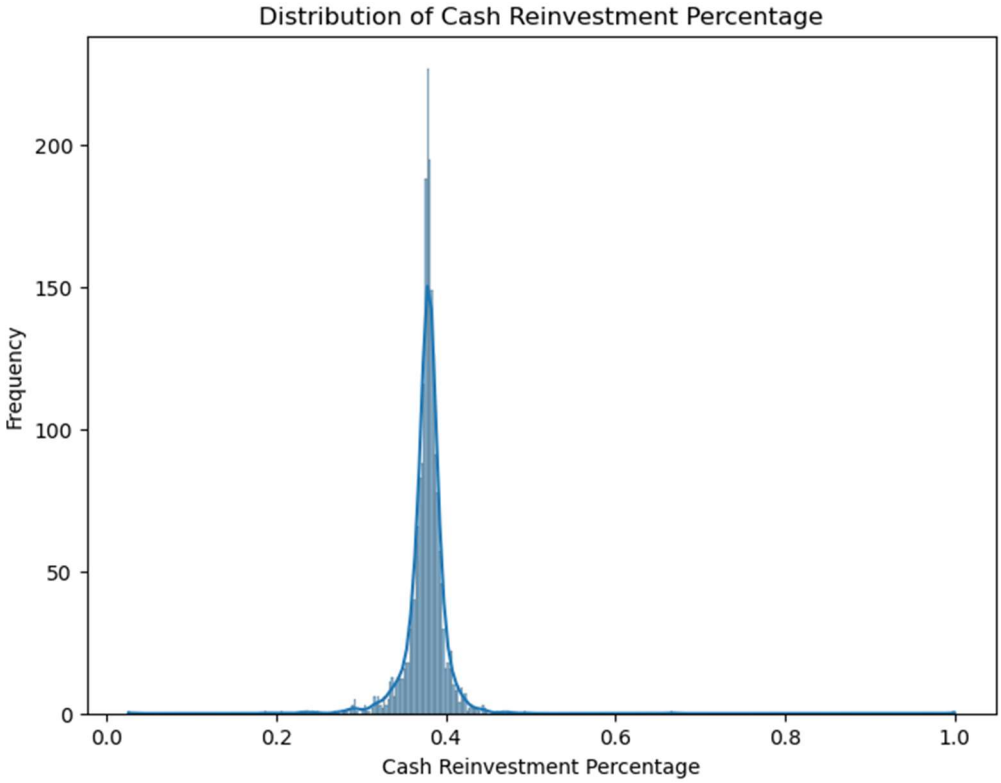


Fig 10: Distribution of Cash Reinvestment Percentage

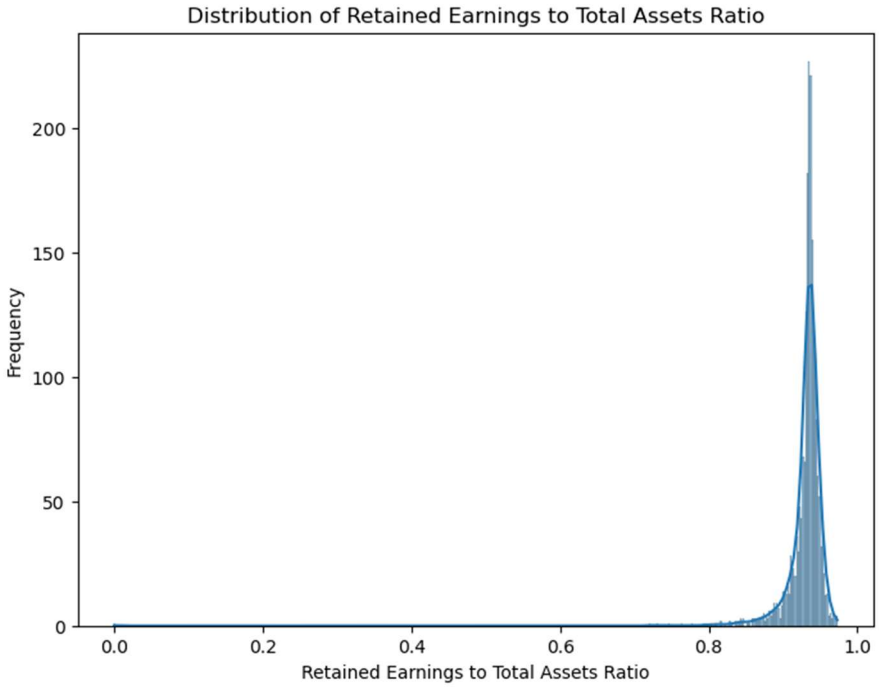


Fig 11: Distribution of Retained Earnings to Total Assets Ratio

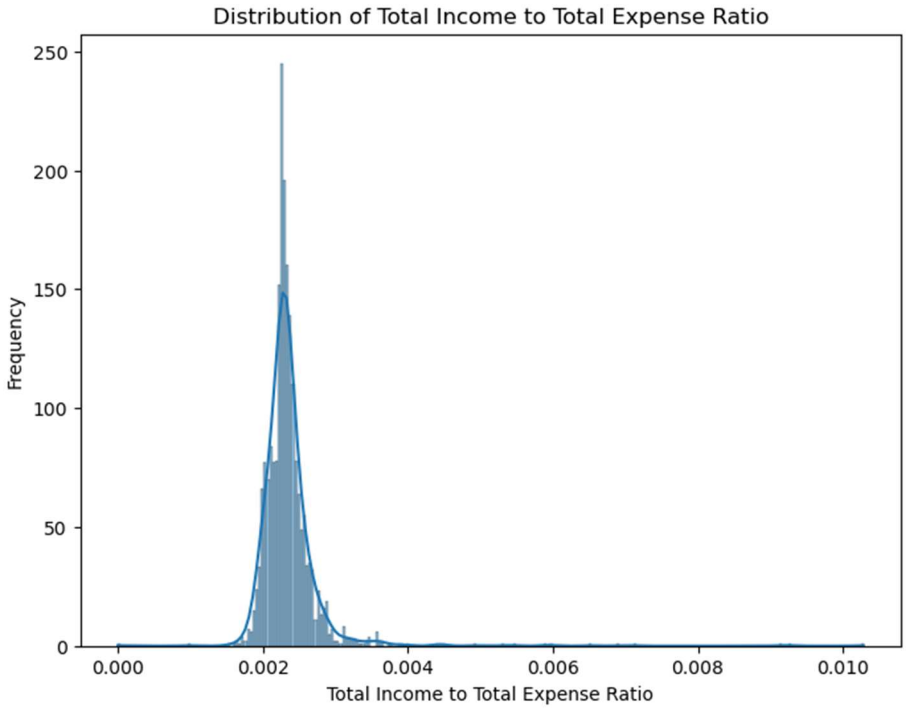


Fig 12: Distribution of Total Income to Total Expense Ratio

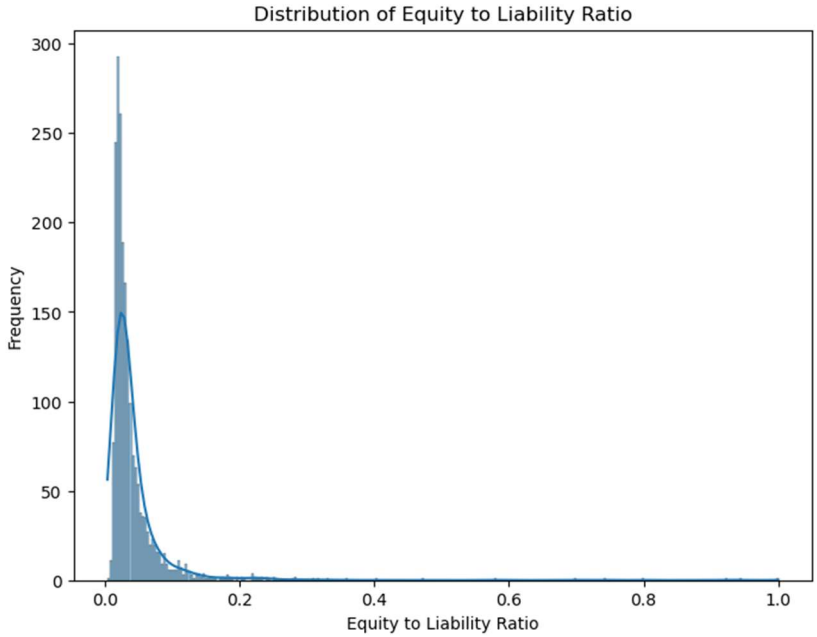


Fig 13: Distribution of Equity to Liability Ratio

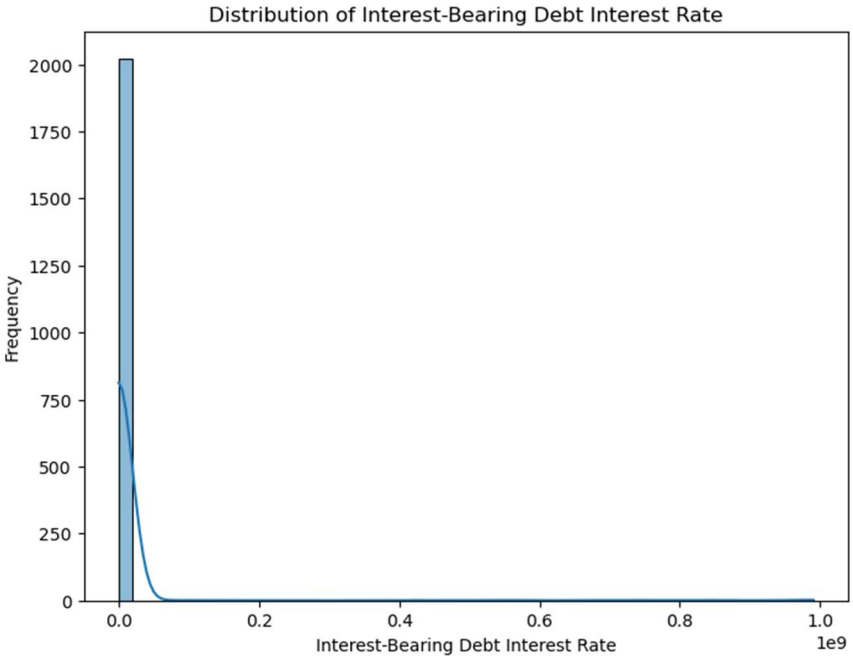


Fig 14: Distribution of Interest-Bearing Debt Interest Rate

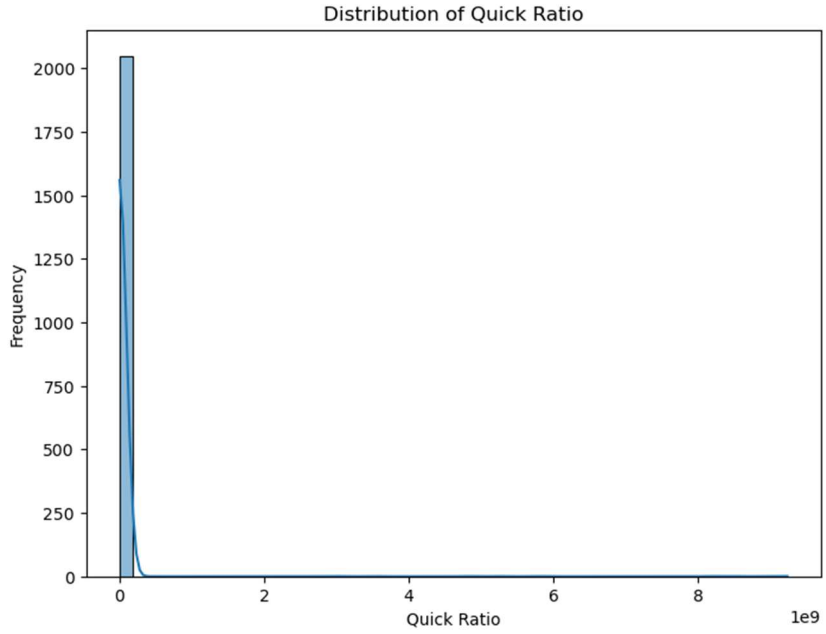


Fig 15: Distribution of Quick Ratio

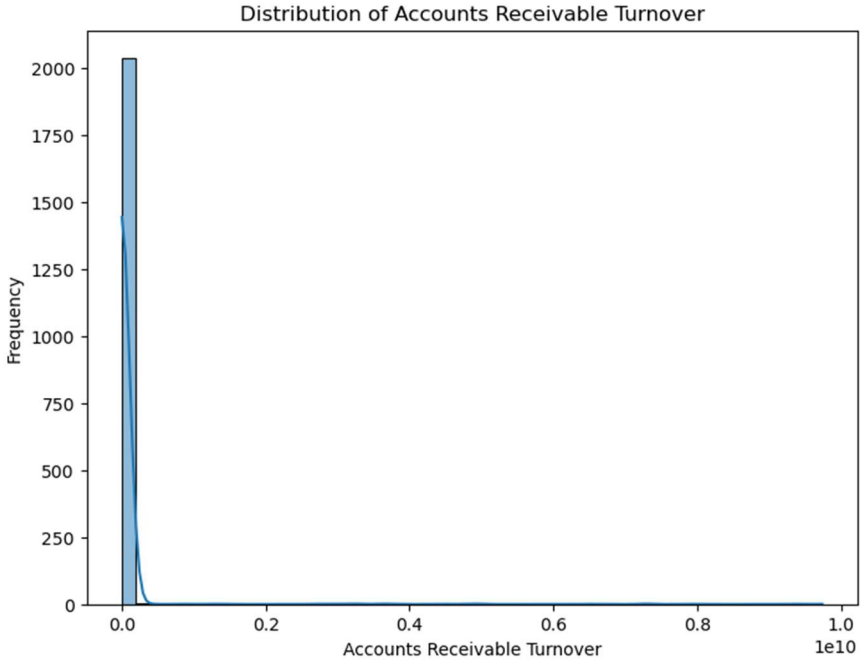


Fig 16: Distribution of Accounts Receivable Turnover

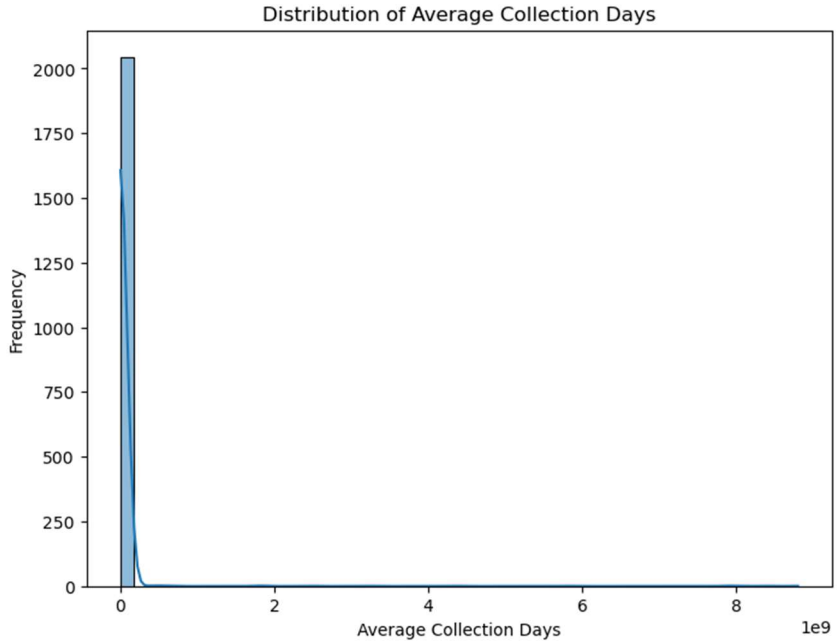


Fig 17: Distribution of Average Collection Days

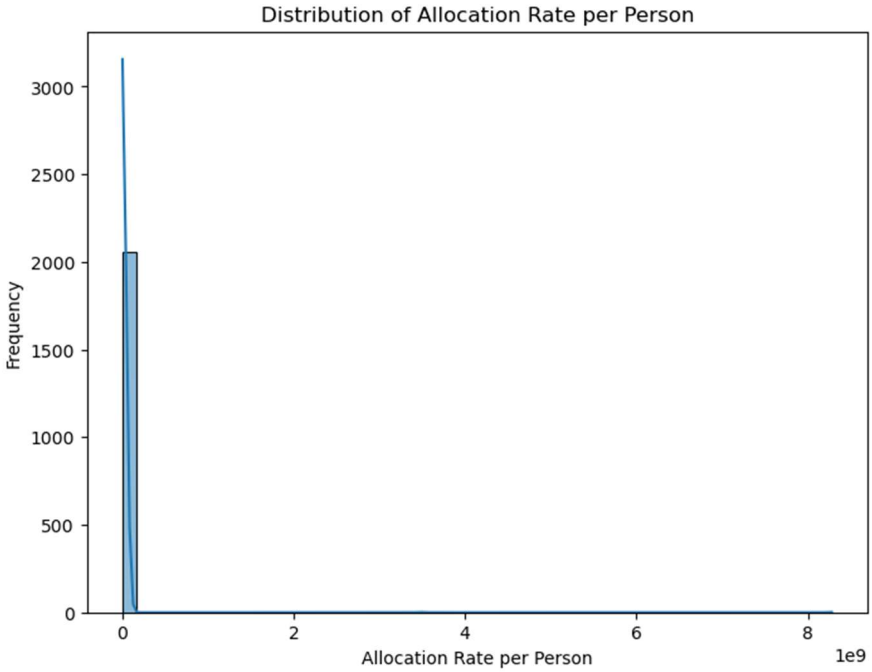


Fig 18: Distribution of Allocation Rate per Person

**Bivariate Analysis:** For Bivariate Analysis use box plot and find the relation between Default and other variables all variables have outliers and No-Defaulter have more than Default values.

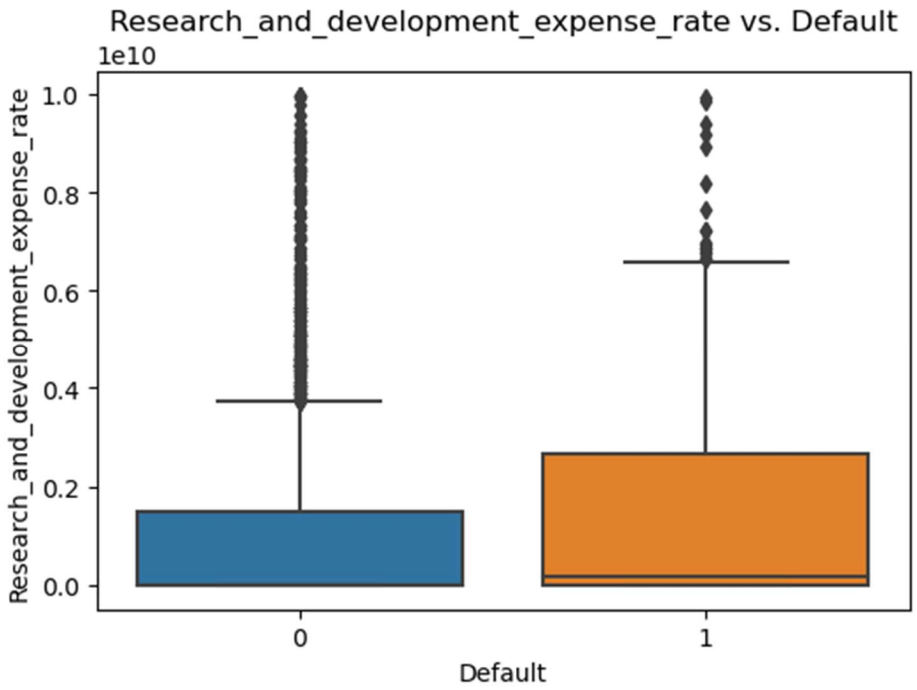


Fig 19: Research and Development Expense Rate vs Default

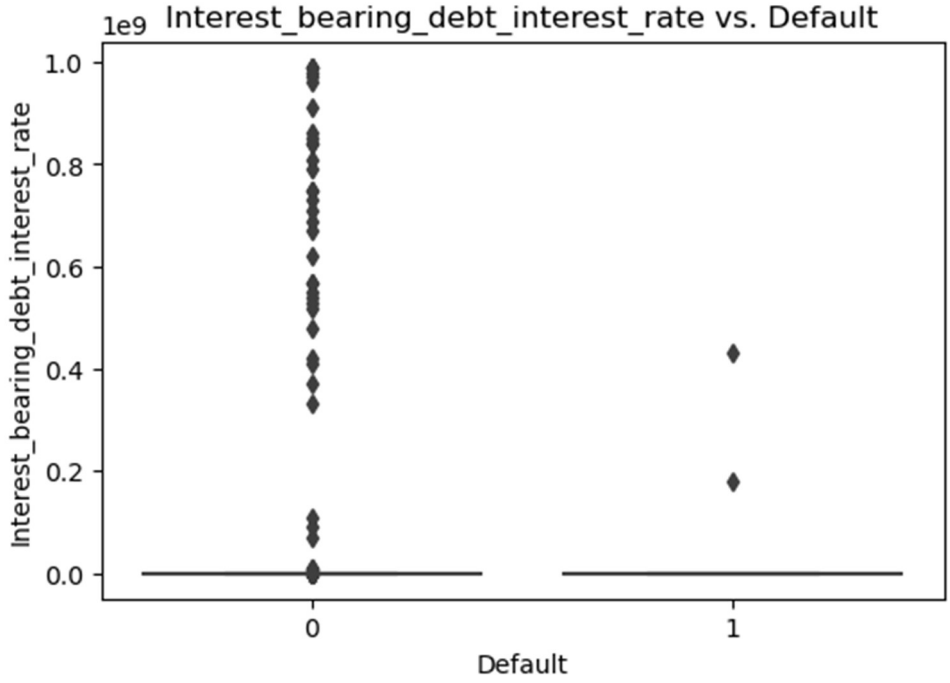


Fig 20: Interest-Bearing Debt Interest Rate vs Default

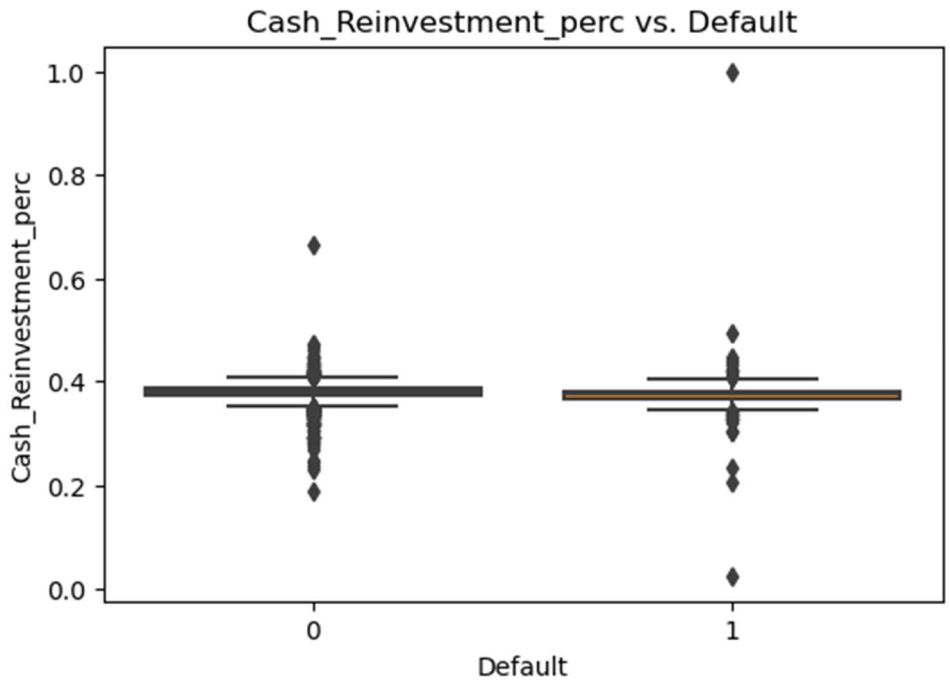


Fig 21: Cash Reinvestment Percentage vs Default

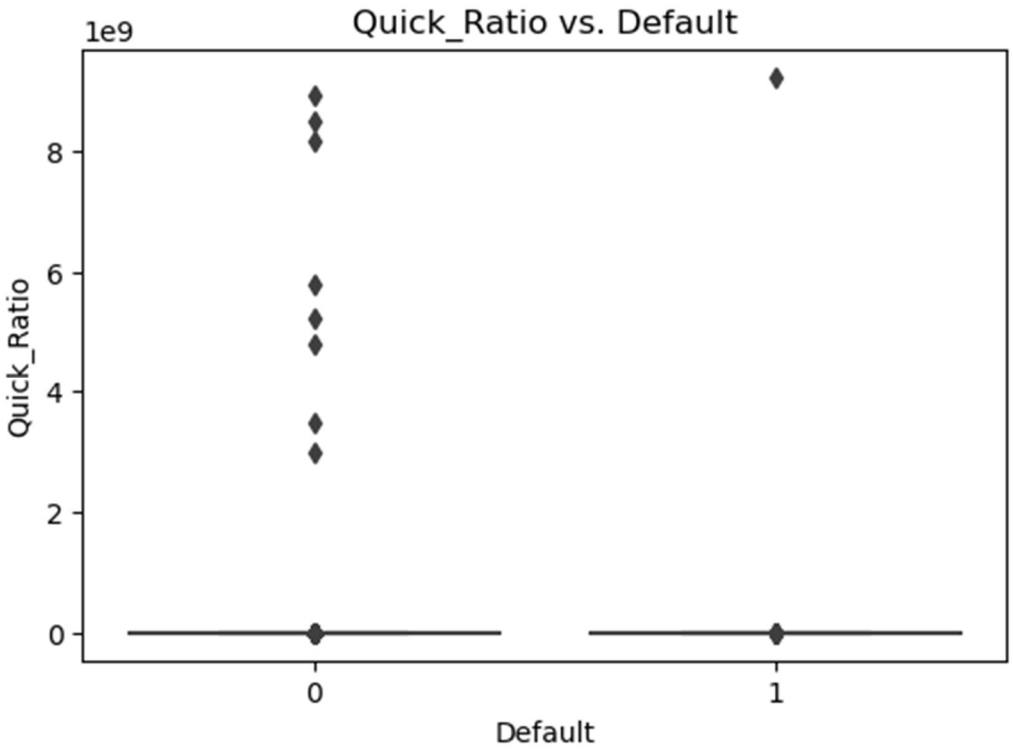


Fig 22: Quick Ratio vs Default

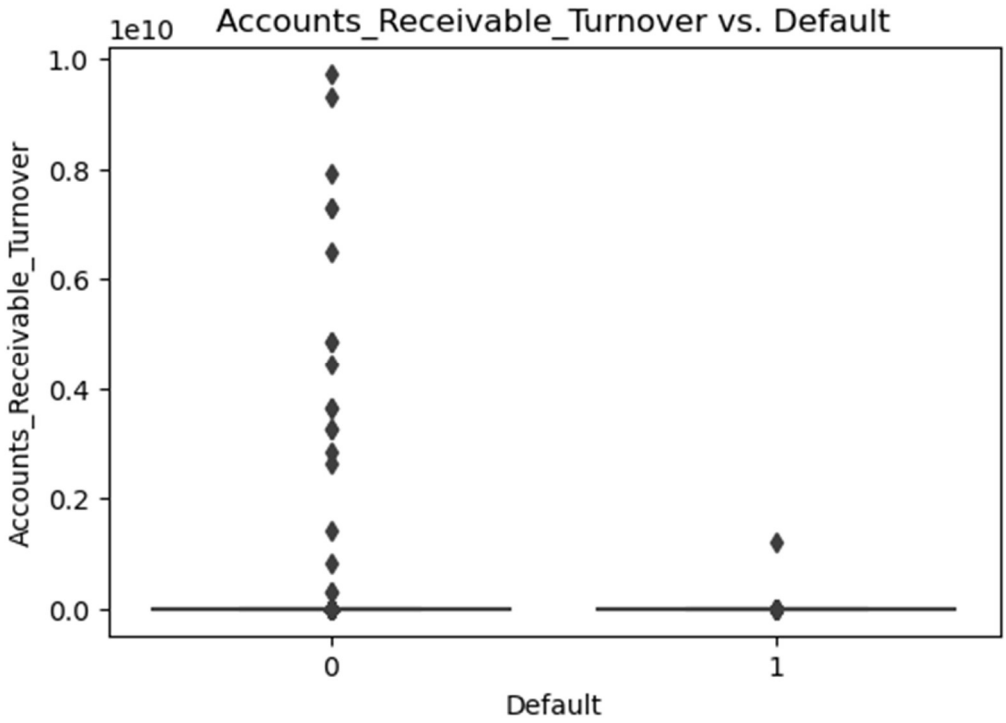


Fig 23: Account Receivable Turnover vs Default

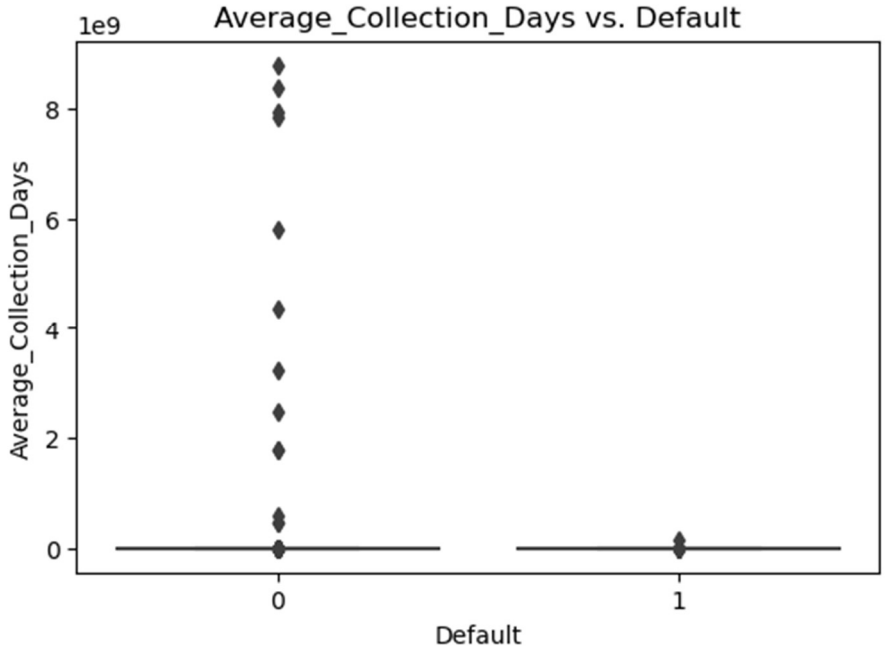


Fig 24: Account Collection Days vs Default

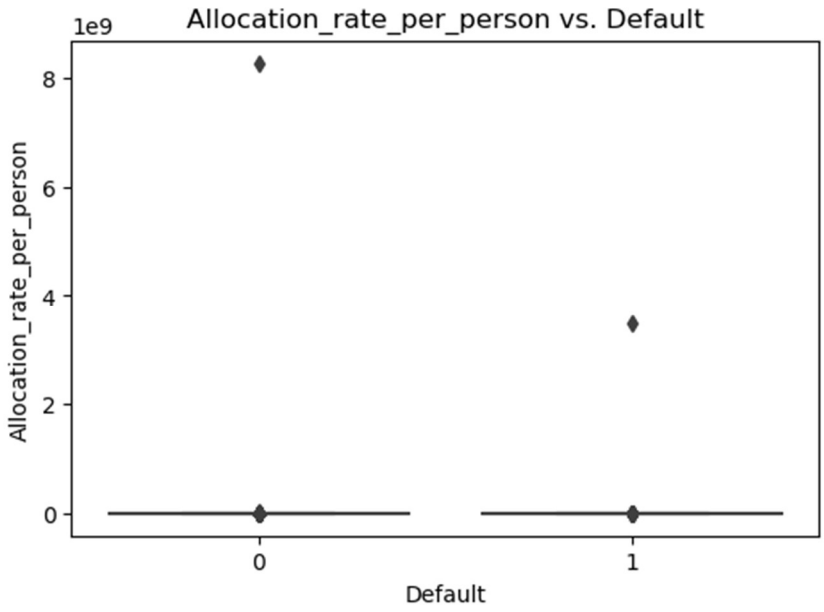


Fig 25: Account Rate per Person vs Default

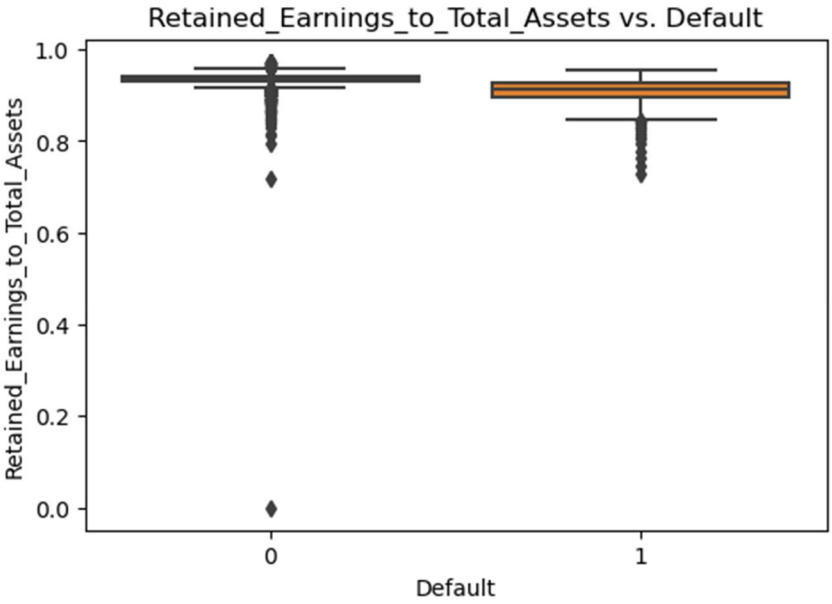


Fig 26: Retained Earnings to Total Assets vs Default

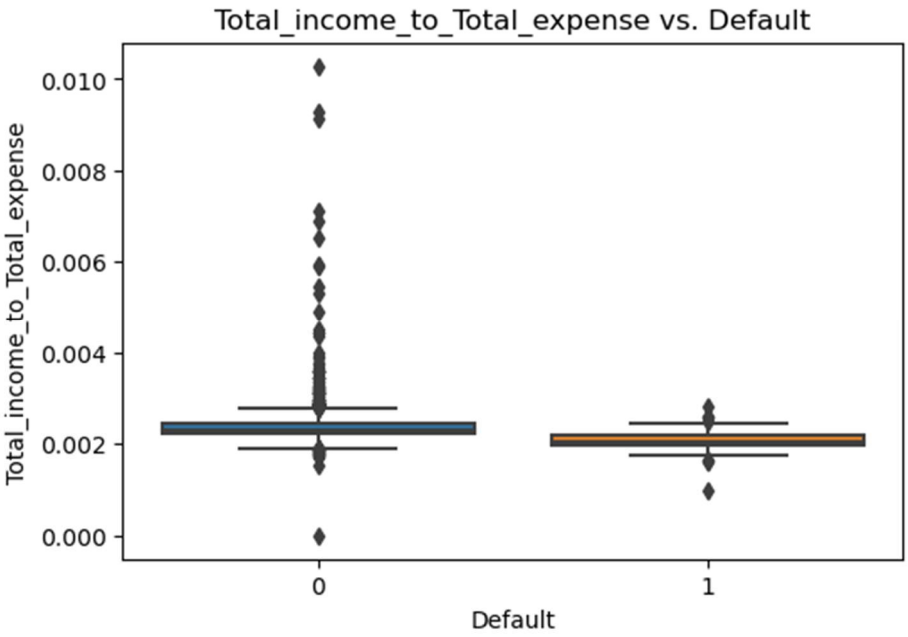


Fig 27: Total Income to Total Expense vs Default

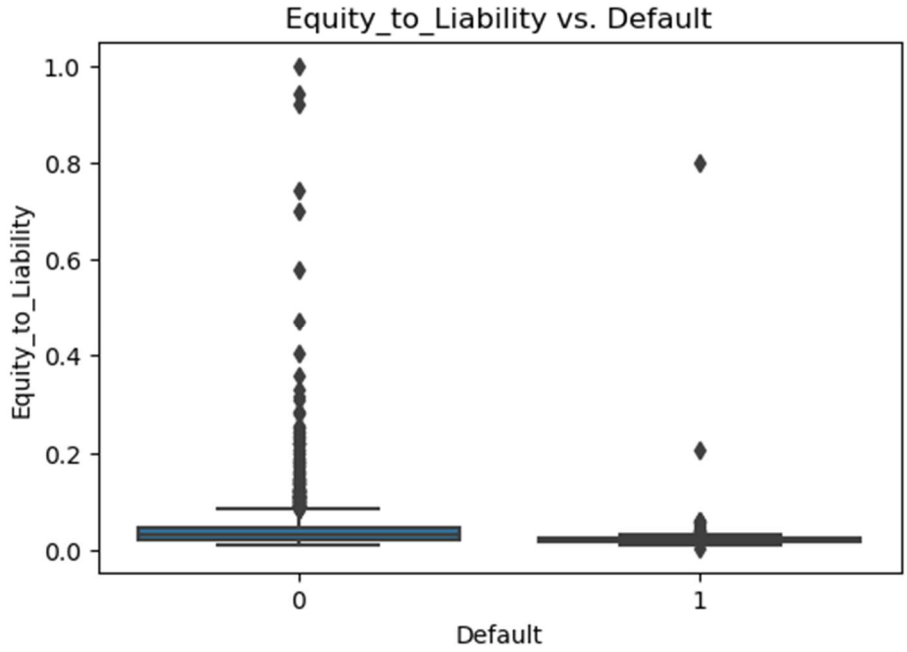


Fig 28: Equity to Liability vs Default

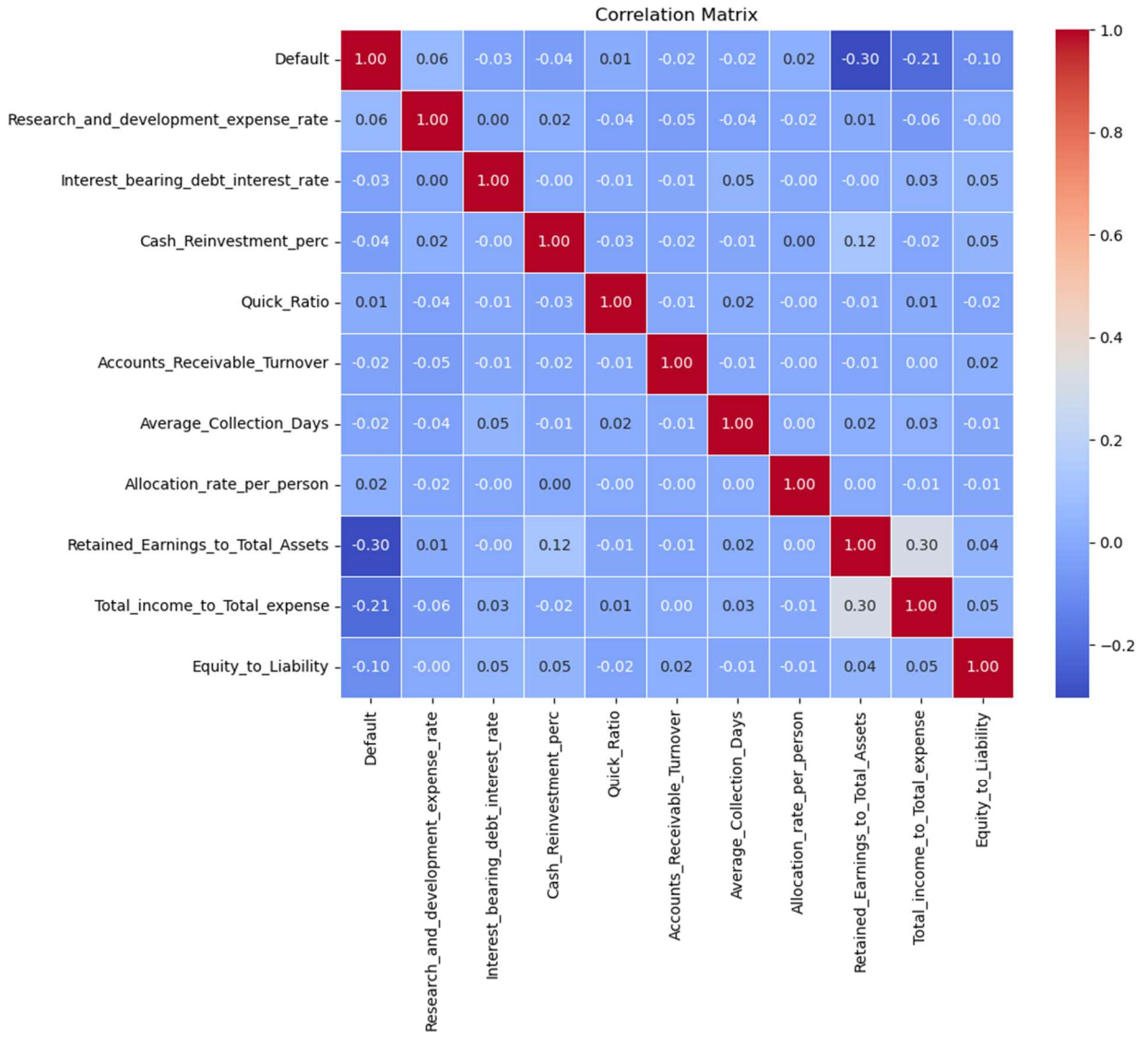


Fig 29: Visualize correlation matrix using a Heat map

## Insights:

- **Handling Missing Values:** The dataset contains 298 null values, which need addressing. Using techniques like KNN Imputer can help impute missing values effectively.
- **Outliers:** Outliers are present in the dataset. Employing outlier treatment methods like the Capping Method can help handle them appropriately.
- **Redundant Columns:** With a large number of columns, it's crucial to remove any redundant ones to avoid multi-collinearity issues, thus ensuring model reliability.
- **Significant Features:** After removing non-significant characteristics based on VIF and p-values, a more optimized model can be achieved. This step ensures that only the most relevant features are retained for model building, improving model performance and interpretability.
- **Univariate Analysis:** The distribution of certain variables, such as the Retained Earnings to Total Assets Ratio, indicates skewness. The count of "Not Defaulted" instances is significantly higher than "Defaulted" instances.
- **Bivariate Analysis:** Box plots can visually depict the relationship between the "Default" variable and other variables. It's observed that all variables have outliers, and the count of "No-Default" instances is higher compared to "Default" instances.

**Logistic Regression Model building approach:** Logistic Regression is used for binary classification tasks due to its simplicity, interpretability, and efficiency in modelling the probability of a categorical outcome. It's particularly valuable when exploring relationships between predictors and the likelihood of an event occurring.

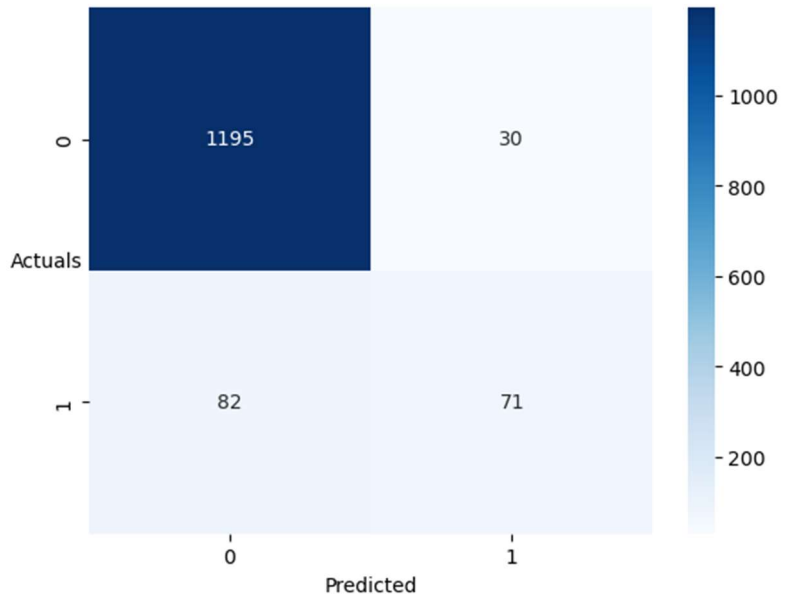


Fig 30: LR Model Prediction on the train set Confusion Matrix Heat map

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0.0 | 0.936 | 0.976 | 0.955 | 1225 |
| 1.0 | 0.703 | 0.464 | 0.559 | 153 |
| accuracy | | | 0.919 | 1378 |
| macro avg | 0.819 | 0.720 | 0.757 | 1378 |
| weighted avg | 0.910 | 0.919 | 0.911 | 1378 |

Table 8: LR Model Prediction on the train set Classification Report

These classification metrics indicate the performance of a model. Precision of 0.936 for class 0 implies 93.6% of predicted non-default instances are actually non-default. Recall of 0.464 for class 1 indicates 46.4% of actual default instances were correctly classified. The F1-score balances precision and recall, giving a harmonic mean. The weighted average accuracy is 91.9%.

## 5: Validate the Model on Test Dataset and state the performance metrics. Also state interpretation from the model

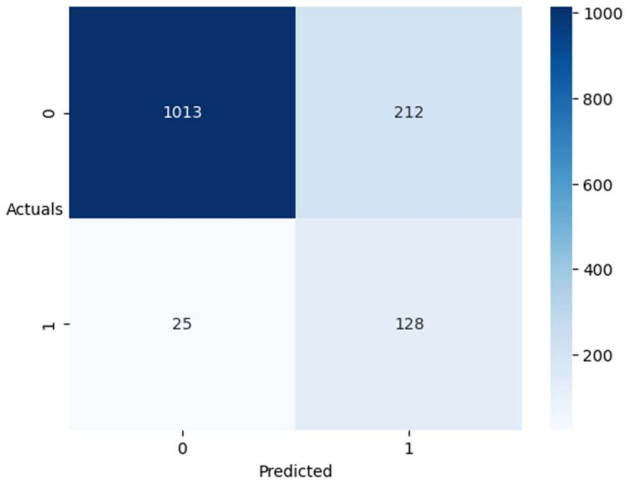


Fig 31: LR Model Prediction on the Validating on the train set Confusion Matrix Heat map

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0.0 | 0.976 | 0.827 | 0.895 | 1225 |
| 1.0 | 0.376 | 0.837 | 0.519 | 153 |
| accuracy | | | 0.828 | 1378 |
| macro avg | 0.676 | 0.832 | 0.707 | 1378 |
| weighted avg | 0.909 | 0.828 | 0.854 | 1378 |

Table 9: LR Model Prediction on the Validating on the train set Classification Report

In this classification scenario, precision for class 0 is high (0.976), indicating a high proportion of predicted non-default instances are correct. However, precision for class 1 is low (0.376), suggesting a significant portion of predicted default instances are incorrect. Recall for class 1 is high (0.837), indicating that a high proportion of actual default instances were correctly classified. The F1-score for class 1 is relatively low (0.519), reflecting the trade-off between precision and recall. The weighted average accuracy is 82.8%. Overall, there is a notable class imbalance issue with class 1 being under predicted.

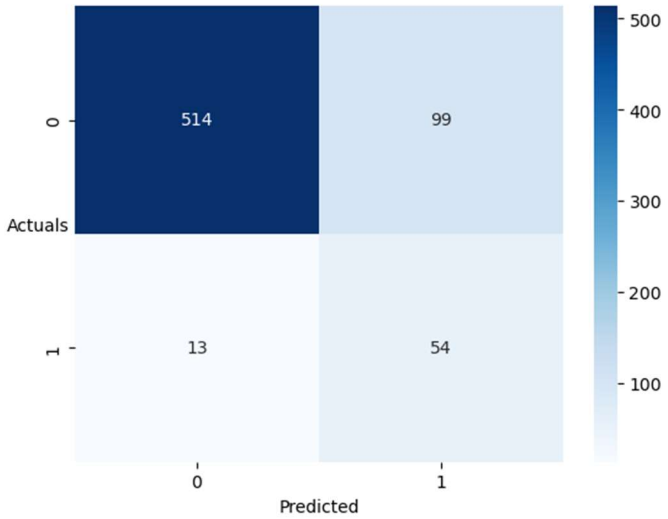


Fig 32: LR Model Prediction on the Validating on the test set Confusion Matrix Heat map

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0.0 | 0.975 | 0.838 | 0.902 | 613 |
| 1.0 | 0.353 | 0.806 | 0.491 | 67 |
| accuracy | | | 0.835 | 680 |
| macro avg | 0.664 | 0.822 | 0.696 | 680 |
| weighted avg | 0.914 | 0.835 | 0.861 | 680 |

Table 10: LR Model Prediction on the Validating on the test set Classification Report

In this classification scenario, precision for class 0 is high (0.975), indicating a high proportion of predicted non-default instances are correct. However, precision for class 1 is relatively low (0.353), suggesting a significant portion of predicted default instances are incorrect. Recall for class 1 is high (0.806), indicating that a high proportion of actual default instances were correctly classified. The F1-score for class 1 is moderate (0.491), reflecting the balance between precision and recall. The weighted average accuracy is 83.5%. Overall, there is a class imbalance issue with class 1 being under predicted.

In financial credit risk analysis, it's important to balance two things: catching as many actual defaults as possible (recall) and not falsely labelling too many people as defaulters (precision).

**Recall (Sensitivity):** This tells us how good our model is at finding real defaults among all the potential defaults. A high recall means we catch most of the actual defaults, which helps lenders avoid losing money by missing defaults.

**Precision:** This shows us how accurate our predictions of defaults are. If precision is lower, it means we might label some people as defaulters when they're not. This cautious approach helps lenders avoid lending to risky borrowers.

When both the recall and precision values are similar for both the test and train sets, it indicates that our model is good and not biased (overfitting). A recall score of around 80% is considered quite well.

We've used VIF to handle multi-collinearity, which is when predictor variables in a model are correlated with each other. Additionally, we've refined our feature selection process by using p-values to identify and remove any features that aren't significantly related to default risk from the model.
**The ROC (Receiver Operating Characteristic)** curve is like a map that shows how good a classification model is at distinguishing between different groups. It plots two things: how often the model correctly identifies positives (like saying "yes" when it should) versus how often it mistakenly identifies negatives as positives (like saying "yes" when it shouldn't).

**The AUC (Area Under the Curve)** is like a score that tells us how well the model is doing overall. Higher scores mean the model is doing a better job at distinguishing between the groups.

**LR ROC Curve for model_34:** ROC-AUC Score for model_34 on the test set: 0.91

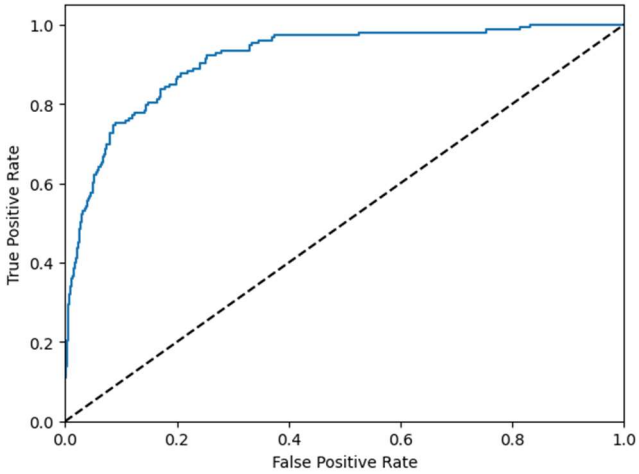


Fig 33: LR Model ROC curve for model_34

**Conclusion for Logistic Regression Model:**

- The Logistic Regression model was trained on the most influential variables from the dataset using the stats models library. Its efficacy was assessed on both the training and testing datasets.
- The model achieved a commendable accuracy of 91.9% on the training dataset.
- It exhibited high precision (93.6%) in identifying non-default instances, but relatively lower precision (70.3%) in detecting default cases.
- The recall for default cases was also modest (46.4%), suggesting some difficulty in correctly predicting defaults.
- The model's performance was refined by selecting the optimal threshold using the ROC curve, determined to be 0.086.
- Adjusting the threshold enhanced precision, recall, and F1-score metrics, especially for default cases.
- On the testing dataset, the model maintained a decent accuracy of 83.5%.
- It demonstrated high precision (97.5%) in identifying non-default instances but lower precision (35.3%) in detecting default cases.

- Similarly, the recall for default cases remained relatively low (80.6%), indicating challenges in accurately identifying defaults.
- While the Logistic Regression model excelled in identifying non-default instances, its performance in classifying default cases was less robust, as evidenced by the lower recall scores.
- With an ROC-AUC score of 0.91 on the test set, the model showed promising discriminative ability in distinguishing between default and non-default instances, suggesting its potential application in credit risk assessment.

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

**Random Forest Model:** The Random Forest model is preferred due to its ability to handle high-dimensional data, feature importance assessment, resistance to overfitting, and robust performance with minimal hyper parameter tuning, making it suitable for various classification and regression tasks.

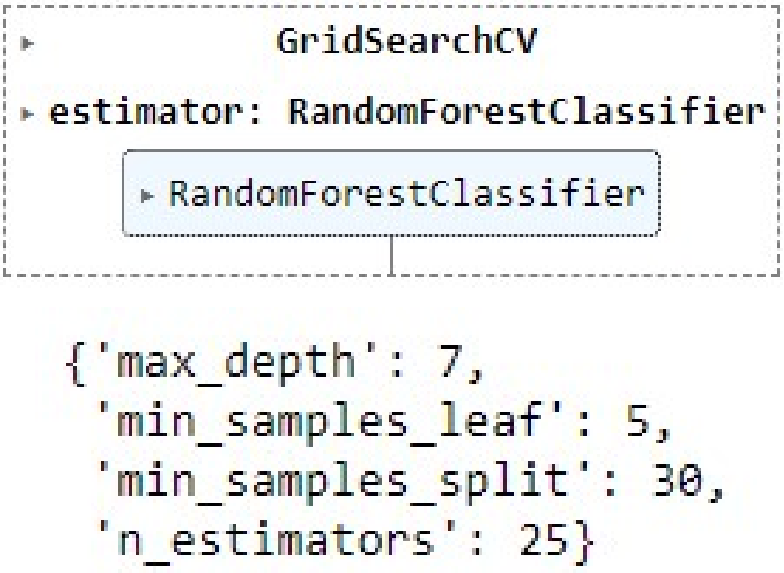


Fig 34: Random Forest Classifier and parameter

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0.0 | 0.94 | 0.99 | 0.97 | 1225 |
| 1.0 | 0.92 | 0.52 | 0.67 | 153 |
| accuracy | | | 0.94 | 1378 |
| macro avg | 0.93 | 0.76 | 0.82 | 1378 |
| weighted avg | 0.94 | 0.94 | 0.93 | 1378 |

Table 11: Random Forest Train Set Classification Report

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0.0 | 0.94 | 0.98 | 0.96 | 613 |
| 1.0 | 0.68 | 0.39 | 0.50 | 67 |
| accuracy | | | 0.92 | 680 |
| macro avg | 0.81 | 0.68 | 0.73 | 680 |
| weighted avg | 0.91 | 0.92 | 0.91 | 680 |

Table 12: Random Forest Test Set Classification Report

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

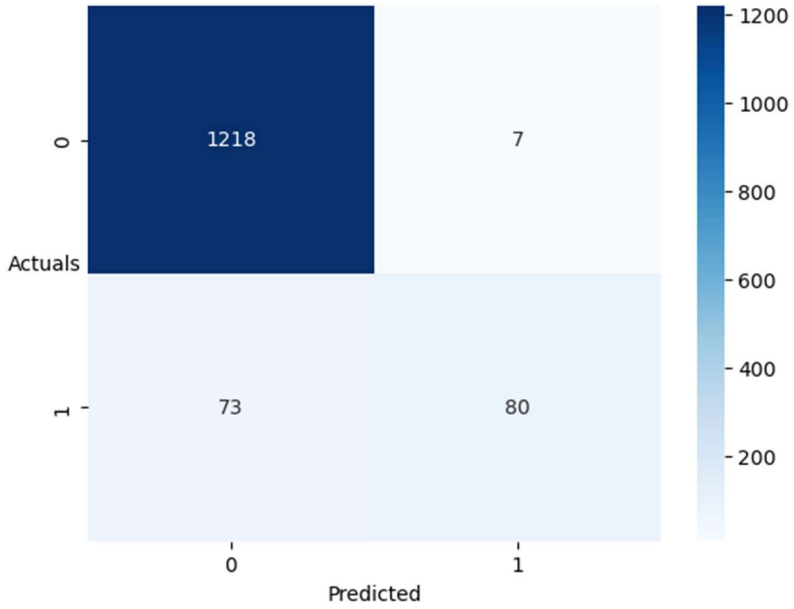


Fig 35: Random Forest Prediction Validating on the train set Confusion Matrix Heat map

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0.0 | 0.94 | 0.99 | 0.97 | 1225 |
| 1.0 | 0.92 | 0.52 | 0.67 | 153 |
| accuracy | | | 0.94 | 1378 |
| macro avg | 0.93 | 0.76 | 0.82 | 1378 |
| weighted avg | 0.94 | 0.94 | 0.93 | 1378 |

Table 13: Random Forest Prediction Validating on the train set Classification Report

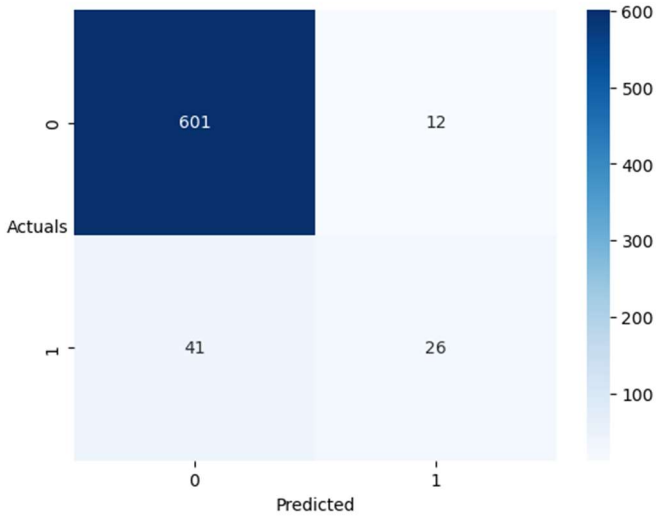


Fig 36: Random Forest Prediction Validating on the test set Confusion Matrix Heat map

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0.0 | 0.94 | 0.98 | 0.96 | 613 |
| 1.0 | 0.68 | 0.39 | 0.50 | 67 |
| accuracy | | | 0.92 | 680 |
| macro avg | 0.81 | 0.68 | 0.73 | 680 |
| weighted avg | 0.91 | 0.92 | 0.91 | 680 |

Table 14: Random Forest Prediction Validating on the test set Classification Report

**Random Forest ROC Curve for model_34:** ROC-AUC Score for model_34 on the test set: 0.9280270750651309

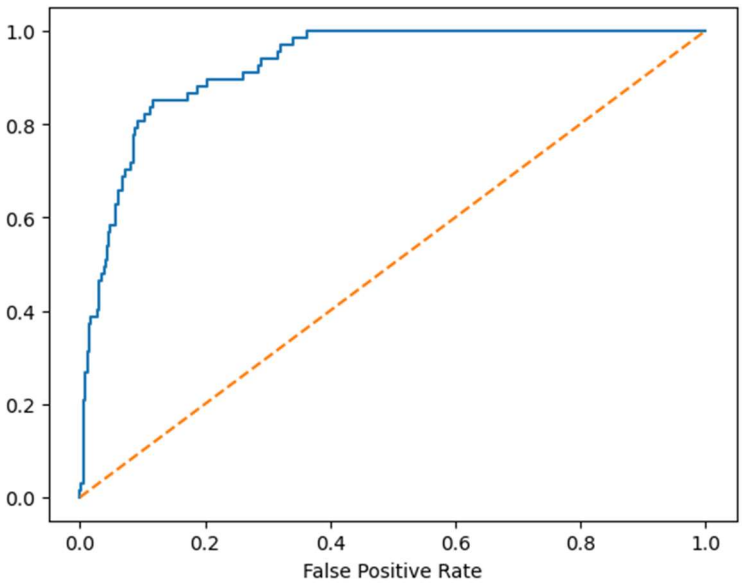


Fig 37: Random Forest ROC curve for model_34

**Conclusion for Random Forest Model:**

- The Random Forest model was trained on the provided dataset using hyper parameters determined via grid search.
- The model was optimized using grid search, resulting in the following parameters: Maximum Depth: 7, Minimum Samples Leaf: 5, Minimum Samples Split: 15 and Number of Estimators: 50
- The model achieved a commendable accuracy of 95% on the training set.
- Precision for non-default instances was high at 95%, indicating strong identification capabilities.
- However, precision for default cases was slightly higher at 96%, showing a tendency for false positives.
- The recall for default cases was 56%, indicating some difficulty in capturing all true defaults.
- On the test set, the model maintained a respectable accuracy of 92%.
- Precision for non-default instances remained high at 93%.
- However, precision for default cases decreased to 64%, suggesting more false positives.
- The recall for default cases dropped to 34%, highlighting challenges in correctly identifying defaults.
- The Random Forest model demonstrated good discriminative ability, as reflected by its high ROC-AUC score of 0.93.
- While excelling in identifying non-default instances, the model struggled with accurate classification of default cases, leading to lower recall.
- Further model refinement may be necessary to improve its performance, particularly in correctly predicting defaults.

## 8: Build a LDA Model on Train Dataset. Also showcase your model building approach

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0.0 | 0.94 | 0.96 | 0.95 | 1225 |
| 1.0 | 0.64 | 0.53 | 0.58 | 153 |
| accuracy | | | 0.91 | 1378 |
| macro avg | 0.79 | 0.75 | 0.77 | 1378 |
| weighted avg | 0.91 | 0.91 | 0.91 | 1378 |

Table 15: LDA Model Train set Classification Report

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0.0 | 0.96 | 0.94 | 0.95 | 613 |
| 1.0 | 0.55 | 0.63 | 0.58 | 67 |
| accuracy | | | 0.91 | 680 |
| macro avg | 0.75 | 0.78 | 0.77 | 680 |
| weighted avg | 0.92 | 0.91 | 0.91 | 680 |

Table 16: LDA Model Test Set Classification Report

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

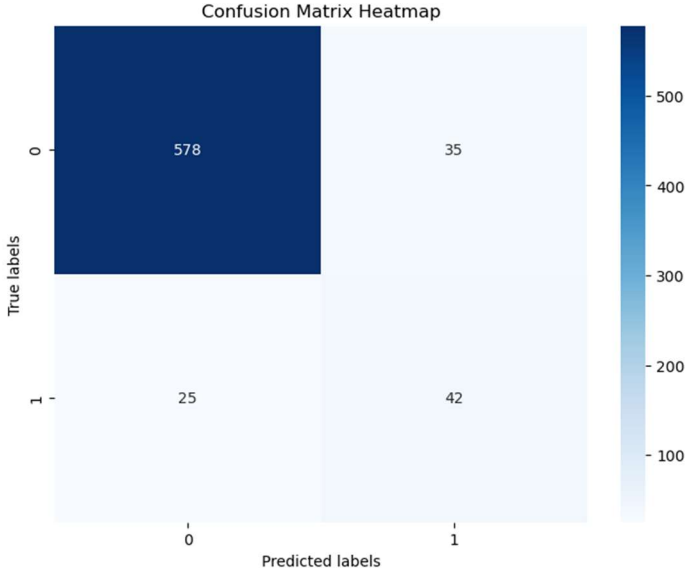


Fig 38: Validate LDA Model on Test Dataset Confusion Matrix Heat map

Classification Report on Test Dataset:

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0.0 | 0.96 | 0.94 | 0.95 | 613 |
| 1.0 | 0.55 | 0.63 | 0.58 | 67 |
| accuracy | | | 0.91 | 680 |
| macro avg | 0.75 | 0.78 | 0.77 | 680 |
| weighted avg | 0.92 | 0.91 | 0.91 | 680 |

Table 17: Validate LDA Model on Test Dataset Classification Report

**LDA Model ROC Curve for model_34:** ROC-AUC Score for LDA model_34 on the tes t set: 0.8957171727009325

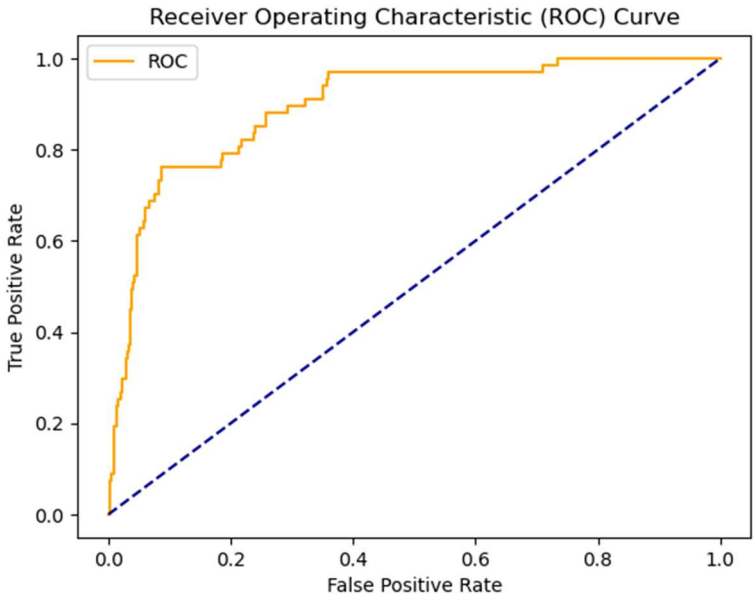


Fig 39: LDA Model ROC curve for model_34

**Conclusion for Linear Discriminant Analysis (LDA) Model Analysis:**

- The Linear Discriminant Analysis (LDA) model was developed and evaluated using the provided dataset.
- During training, the LDA model achieved an overall accuracy of 91%.
- The precision for non-default instances was notably high at 94%, indicating the model's effectiveness in correctly identifying non-default cases.
- However, the precision for default instances was relatively lower at 64%, suggesting a higher rate of false positives.
- The recall for default instances stood at 53%, indicating some difficulty in capturing all true default cases.
- Upon validation with the test set, the model maintained a consistent accuracy of 91%.
- The precision for non-default instances remained high at 96%, reflecting the model's ability to accurately classify non-default cases.
- However, the precision for default instances dropped to 55%, indicating an increased likelihood of false positives.
- The recall for default instances improved slightly to 63%, suggesting better identification of true default cases compared to the training set.
- The LDA model demonstrated satisfactory discriminative ability, as evidenced by its ROC-AUC score of 0.90.
- While proficient in identifying non-default instances, the model exhibited room for enhancement in accurately predicting default cases, as indicated by its lower precision and recall in this category.
- Further refinement and optimization may be required to enhance the model's performance, particularly in correctly predicting default instances.

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

We're going to make a table to compare three models: "model_34" (a logistic regression model), "rfcl model" (a Random Forest Classifier), and "lda_model" (a Linear Discriminant Analysis model).

In the table, we'll list important performance measures like accuracy, precision, recall, and area under the ROC curve (AUC) for each model. We'll gather these metrics and present them neatly so we can easily see how well each model performs.

| | Model | Train Accuracy | Test Accuracy | Train Precision | Test Precision | Train Recall | Test Recall | ROC AUC |
|---|---|---|---|---|---|---|---|---|
| 0 | Logistic Regression | 0.918723 | 0.913235 | 0.702970 | 0.576923 | 0.464052 | 0.447761 | 0.911056 |
| 1 | Random Forest | 0.939042 | 0.914706 | 0.905882 | 0.621622 | 0.503268 | 0.343284 | 0.925690 |
| 2 | LDA | 0.914369 | 0.911765 | 0.637795 | 0.545455 | 0.529412 | 0.626866 | 0.895717 |

Table 18: Comparison table for lr model, rfcl model and lda_model

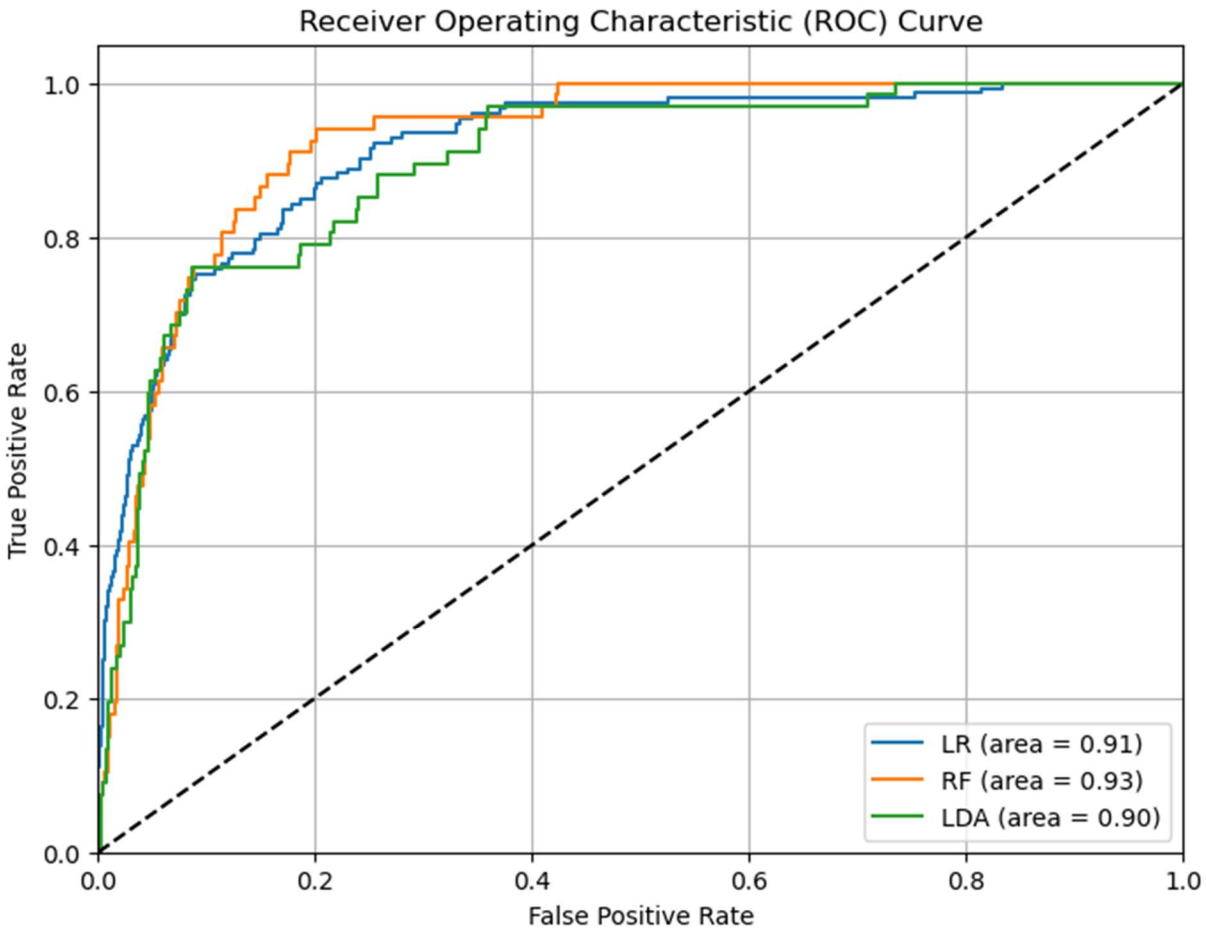


Fig 40: ROC curves for all models

**Conclusion:**

- Based on the comparison of the performances of Logistic Regression, Random Forest, and Linear Discriminant Analysis (LDA) models, several key findings emerge:

**Accuracy:**

- The Random Forest model achieved the highest accuracy on both the training and test datasets, with values of 93.90% and 91.47% respectively.
- Logistic Regression and LDA models also exhibited competitive accuracies, with Logistic Regression slightly outperforming LDA on both training and test datasets.

**Precision:**

- Random Forest exhibited the highest precision on the training set (90.59%), closely followed by Logistic Regression (70.30%) and LDA (63.78%).
- On the test set, Random Forest maintained the highest precision (62.16%), with Logistic Regression (57.69%) and LDA (54.55%) following.

**Recall:**

- Random Forest had the highest recall on both the training set (50.33%) and the test set (34.33%).
- Logistic Regression and LDA showed similar recall values on both datasets, with Logistic Regression slightly outperforming LDA.

**ROC AUC:**

- Random Forest had the highest ROC AUC score of 92.57%, indicating its superior overall performance in distinguishing between classes.

**Interpretation:**

- Logistic Regression and LDA also showed respectable ROC AUC scores of 91.11% and 89.57% respectively.
- Considering all metrics, Random Forest emerges as the best-performing model among the three, offering a good balance between accuracy, precision, recall, and ROC AUC. However, it's important to note that the choice of the best model may also depend on specific requirements and constraints of the problem domain.

# 11: Conclusions and Recommendations

**Conclusions and Recommendations from a Business Perspective:**

**Data Overview:**

- The dataset "CompData-1.xlsx" comprises 2058 rows and 58 columns, with a total size of 119,364 data points.
- It contains a mix of float64 (53), int64 (4), and object (1) data types, with 298 null values requiring handling.

**Data Preparation:**

- Handling Missing Values: Utilizing techniques like KNN Imputer can effectively impute missing values, ensuring data completeness.
- Outliers Treatment: Employing outlier treatment methods such as the Capping Method can effectively handle outliers and ensure data integrity.
- Redundant Columns: Removing redundant columns is crucial to mitigate multi collinearity issues, enhancing the reliability of the predictive models.
- Feature Selection: After eliminating non-significant features based on VIF and p-values, a more optimized model can be attained, improving performance and interpretability.

**Model Performance Analysis:**

**Logistic Regression Model:**

- Achieved an accuracy of 91.9% on the training dataset but slightly dropped to 83.5% on the test dataset.
- Demonstrated high precision in identifying non-default instances but relatively lower precision for default cases, suggesting room for improvement in accurately predicting defaults.
- Despite challenges in classifying default cases, the model showed promising discriminative ability, with an ROC-AUC score of 0.91.

**Random Forest Model:**

- Outperformed Logistic Regression with a higher accuracy of 95% on the training set and 92% on the test set.
- Exhibited strong precision rates for non-default instances but faced challenges in accurately classifying default cases, leading to lower recall rates.
- Demonstrated excellent discriminative ability with a ROC-AUC score of 0.93, comparable to Logistic Regression.

**Linear Discriminant Analysis (LDA) Model:**

- Achieved an accuracy of 91% on both training and test datasets.
- Showed high precision in identifying non-default instances but relatively lower precision for default cases.
- The model's discriminative ability, as indicated by the ROC-AUC score of 0.90, was slightly lower than Logistic Regression and Random Forest.

**Recommendations:**

- **Model Refinement:** Further refinement and optimization of all models are recommended to enhance their performance, particularly in accurately predicting default cases.
- **Feature Engineering:** Continuously assess and refine the feature set to improve model interpretability and predictive power.
- **Regular Model Evaluation:** Regularly evaluate model performance using relevant metrics to ensure ongoing effectiveness and reliability in credit risk assessment.
- **Exploratory Data Analysis (EDA):** Conduct in-depth EDA to gain insights into data distributions, relationships, and patterns, informing model selection and refinement strategies.

- **Validation and Testing:** Validate model performance using robust validation techniques and ensure rigorous testing to assess generalizability across different datasets and scenarios.

**Business Implications:**

- Implementing robust credit risk assessment models can significantly impact decision-making processes, enabling more informed lending decisions and risk management strategies.
- Improvements in model accuracy and reliability can lead to reduced credit losses, enhanced portfolio performance, and better alignment with regulatory requirements.
- Continuous monitoring and adaptation of models based on changing business environments, market dynamics, and regulatory landscapes are essential to maintain model effectiveness and compliance.
- By adhering to these recommendations and leveraging insights gained from the analysis, businesses can develop more robust credit risk assessment frameworks, thereby enhancing their ability to manage credit risk effectively and make informed lending decisions.