PROJECT
ON
FINANCE AND RISK ANALYTICS
(PART - A)

By SHAJIL FERNANDEZ

25-02-2024

Table of Contents:

Problem 1

List of Figures:

List of Tables:

Problem 1

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.

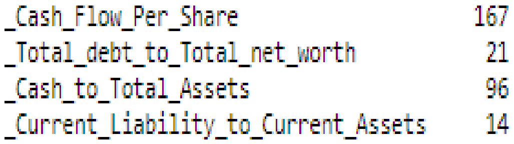
1.1 Outlier Treatment.

Ans:

Basic info:

- There are 2058 rows and 58 columns in the given data.
- There are 53 float64, 4 int64 and 1 object datatypes.
- There are no duplicates.
- There are 4 variables with missing values in the given data.

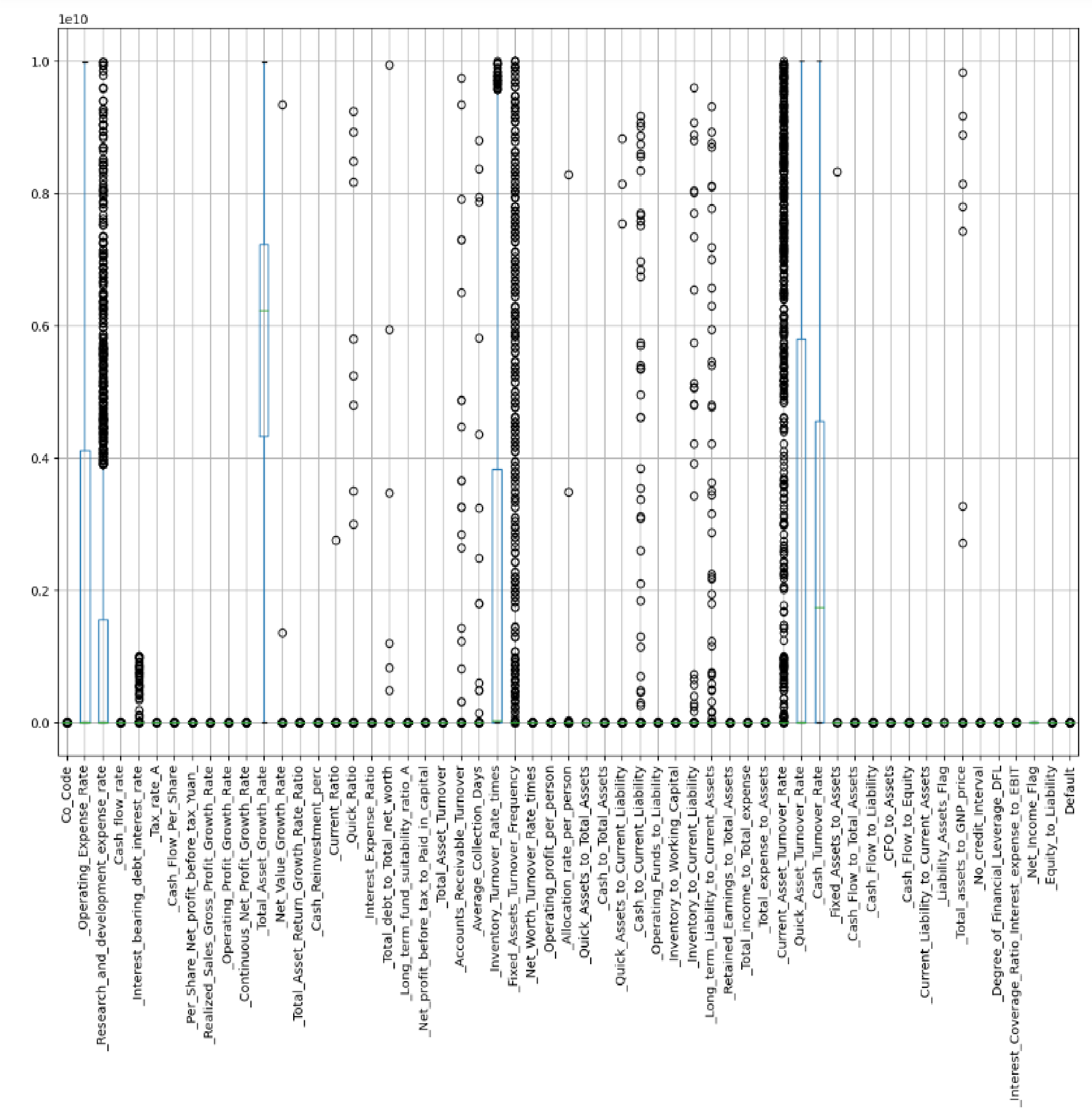
Table 1: Number of missing values in each variable



```
_Cash_Flow_Per_Share                    167
_Total_debt_to_Total_net_worth           21
_Cash_to_Total_Assets                    96
_Current_Liability_to_Current_Assets     14
```

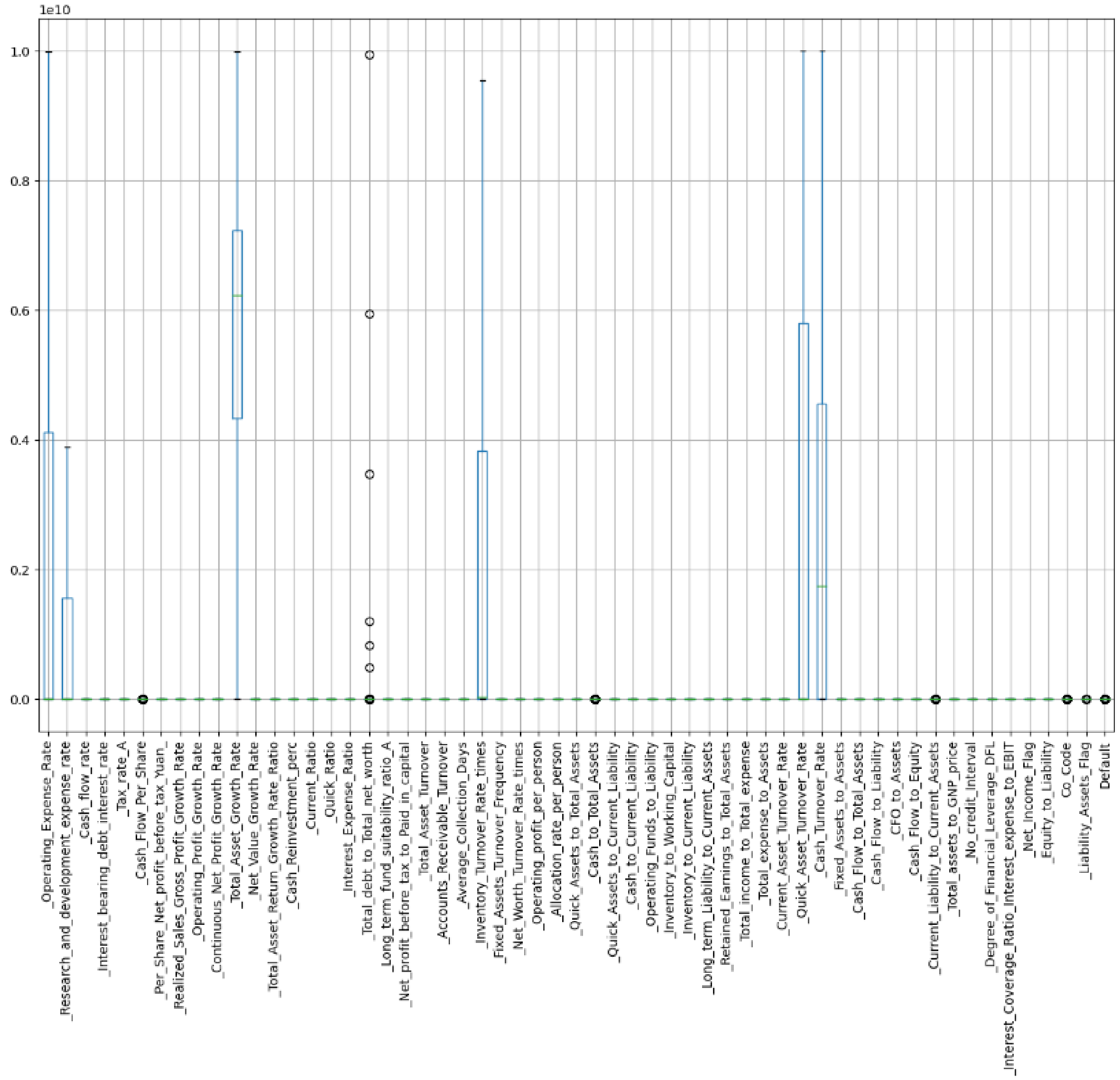
<u>Outlier treatment</u>:

Figure 1: Boxplot to check outliers



- There are outliers for most of the variables.
- We are treating outliers based on the percentile method.

Figure 2: Boxplot after treating outliers



- We can observe that most of the outliers are been treated.

1.2 Missing Value Treatment.

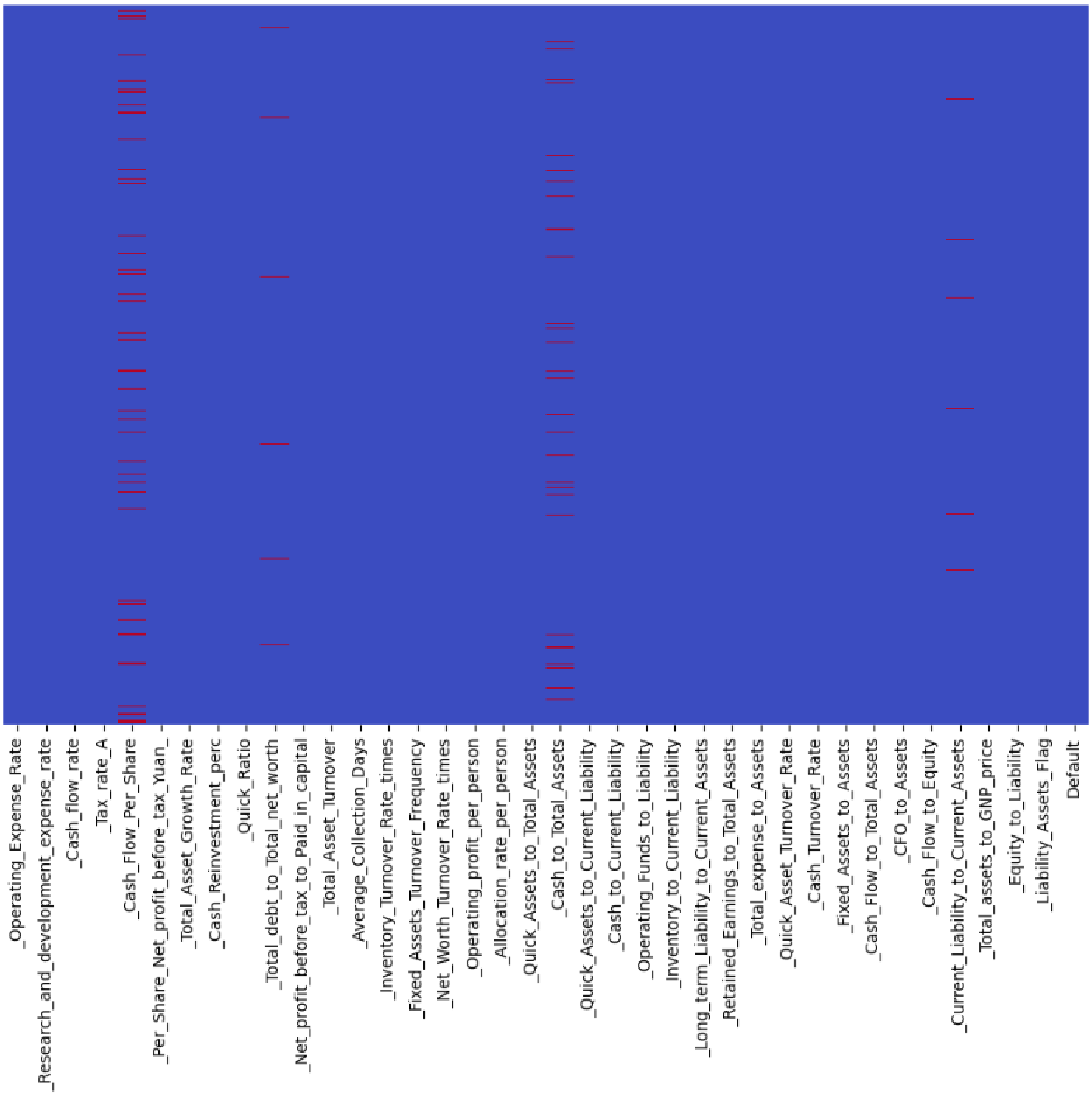
Ans:

- Before missing value treatment, we are removing few constant variables which does not contribute much on the model building.
- Constant variables are variables on which majority of the values are same or nearly same. After removing constant variables, data now consists of 2058 rows and 38 columns.

<u>Treating missing values</u>:

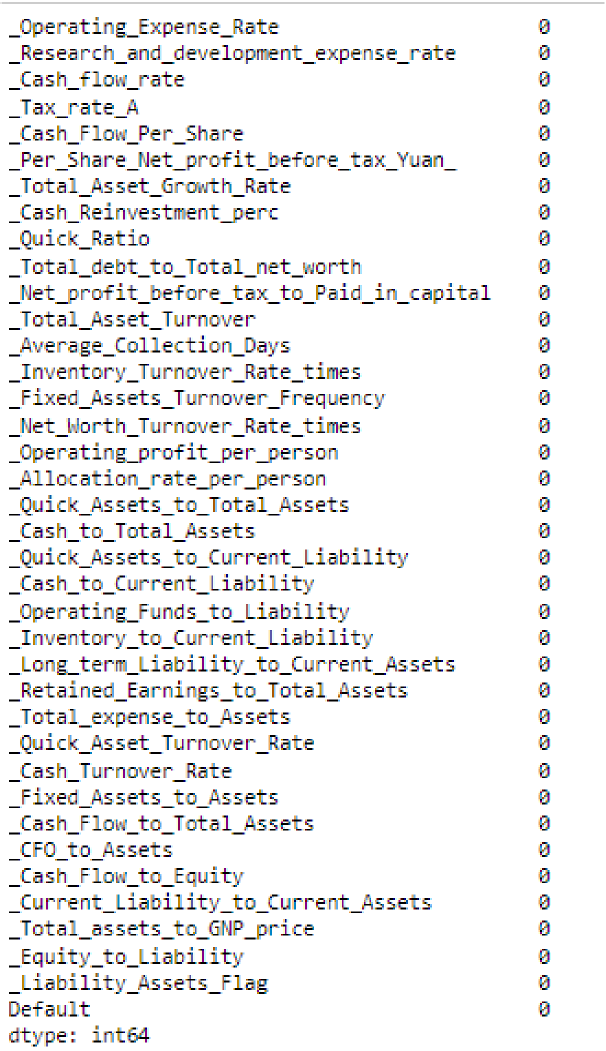
- About 0.25% of missing values were present in the entire dataset, which increased to 0.38% after removing constant variables.
- The proportion of missing values of the 4 variables are 8%, 5%, 1% and 1%, so we can proceed without dropping columns as it is below 30%.

Figure 3: Boxplot showing presence of null values



- Scaling the data and imputing null values based on KNN imputer method.

Table 2: After treatment of null values



```
_Operating_Expense_Rate                        0
_Research_and_development_expense_rate         0
_Cash_flow_rate                                0
_Tax_rate_A                                    0
_Cash_Flow_Per_Share                           0
_Per_Share_Net_profit_before_tax_Yuan_         0
_Total_Asset_Growth_Rate                       0
_Cash_Reinvestment_perc                        0
_Quick_Ratio                                   0
_Total_debt_to_Total_net_worth                 0
_Net_profit_before_tax_to_Paid_in_capital      0
_Total_Asset_Turnover                          0
_Average_Collection_Days                       0
_Inventory_Turnover_Rate_times                 0
_Fixed_Assets_Turnover_Frequency               0
_Net_Worth_Turnover_Rate_times                 0
_Operating_profit_per_person                   0
_Allocation_rate_per_person                    0
_Quick_Assets_to_Total_Assets                  0
_Cash_to_Total_Assets                          0
_Quick_Assets_to_Current_Liability             0
_Cash_to_Current_Liability                     0
_Operating_Funds_to_Liability                  0
_Inventory_to_Current_Liability                0
_Long_term_Liability_to_Current_Assets         0
_Retained_Earnings_to_Total_Assets             0
_Total_expense_to_Assets                       0
_Quick_Asset_Turnover_Rate                     0
_Cash_Turnover_Rate                            0
_Fixed_Assets_to_Assets                        0
_Cash_Flow_to_Total_Assets                     0
_CFO_to_Assets                                 0
_Cash_Flow_to_Equity                           0
_Current_Liability_to_Current_Assets           0
_Total_assets_to_GNP_price                     0
_Equity_to_Liability                           0
_Liability_Assets_Flag                         0
Default                                        0
dtype: int64
```

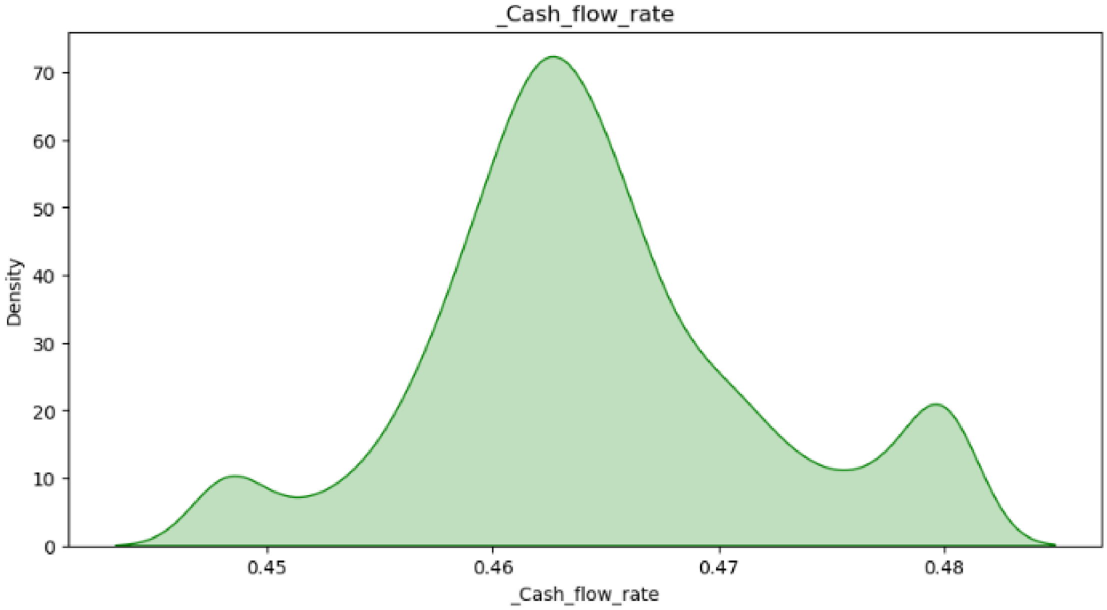
1.3 Univariate & Bivariate analysis with proper interpretation. (You may choose to include only those variables which were significant in the model building).

Ans:

Univariate & Bivariate analysis:

i) Cash_flow_rate:

Figure 4: Cash_flow_rate



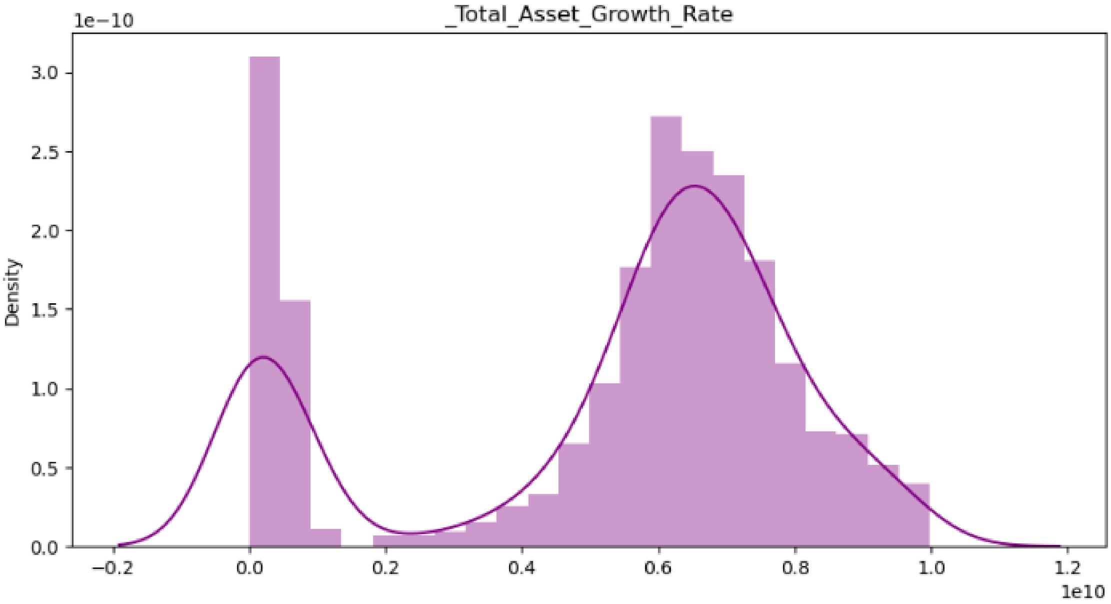
Observations

- Mean Cash Flow rate is 0.47 and median is 0.46.

- About 75% of the companies have cash flow rate below 0.47.

ii) <u>Total_Asset_Growth_Rate:</u>

Figure 5: Total_Asset_Growth_Rate



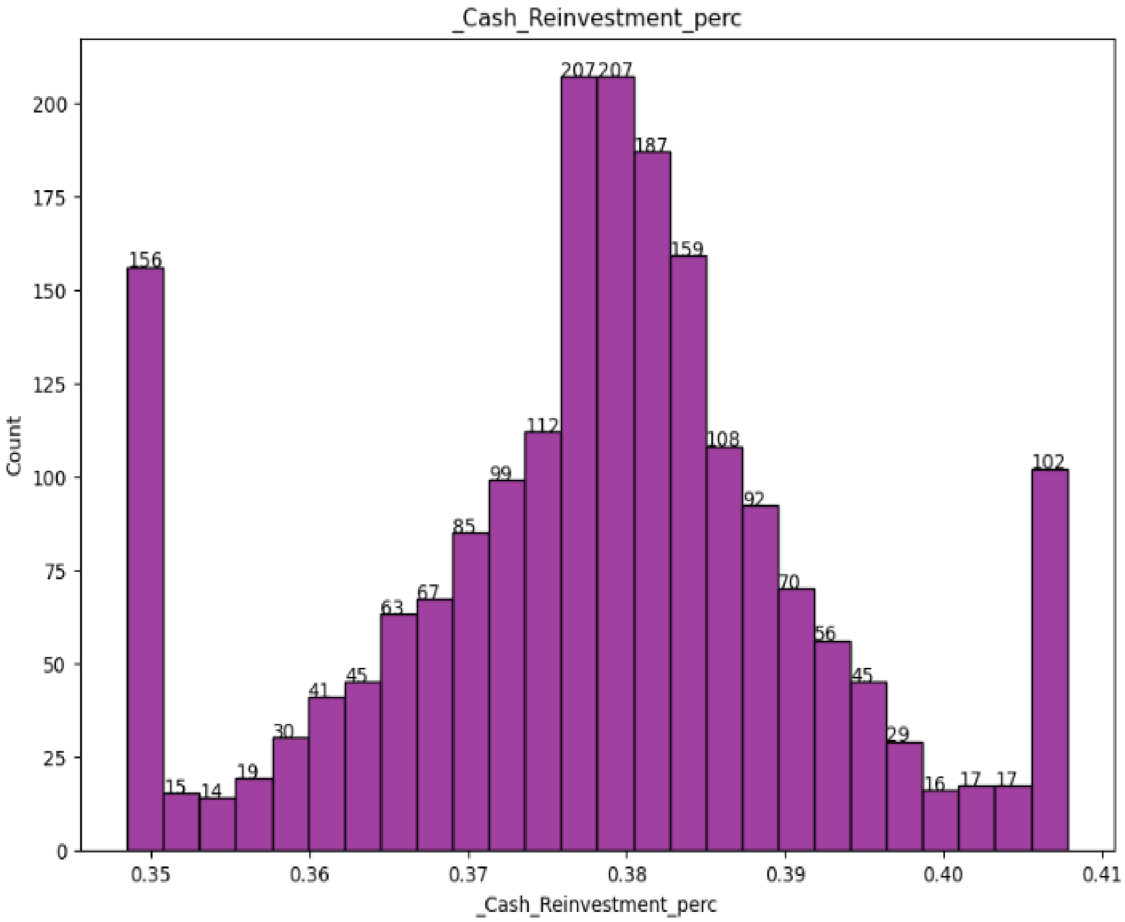
Observations

- Total asset growth rate ranges from 0 to 9980000000.

- About 50% of the total asset growth rate is below 6225000000.

iii) Cash_Reinvestment_perc:

Figure 6: Cash_Reinvestment_perc



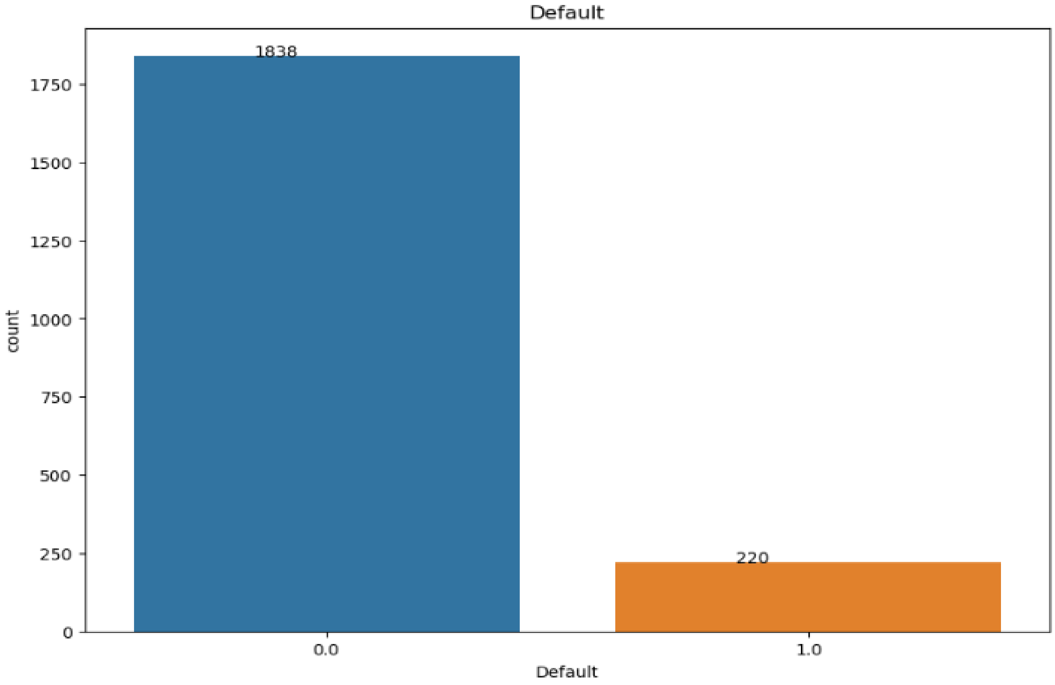
Observations

- Cash reinvestment percentage ranges from 0.03 to 1 across companies.

- Mean cash reinvestment percentage is 0.38 and about 75% of the companies have reinvestment percentage below 0.39.

iv) Default:

Figure 7: Default

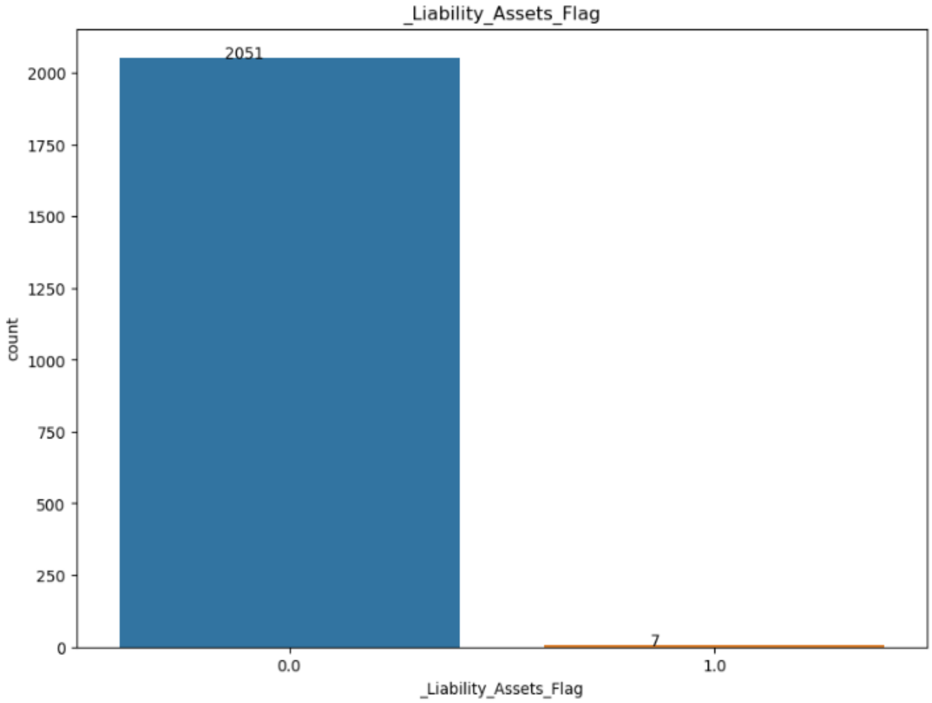


Observation

- 220 companies have defaulted and 1838 companies are non-defaulters.

v) Liability_Assets_Flag:

Figure 8: Liability_Assets_Flag



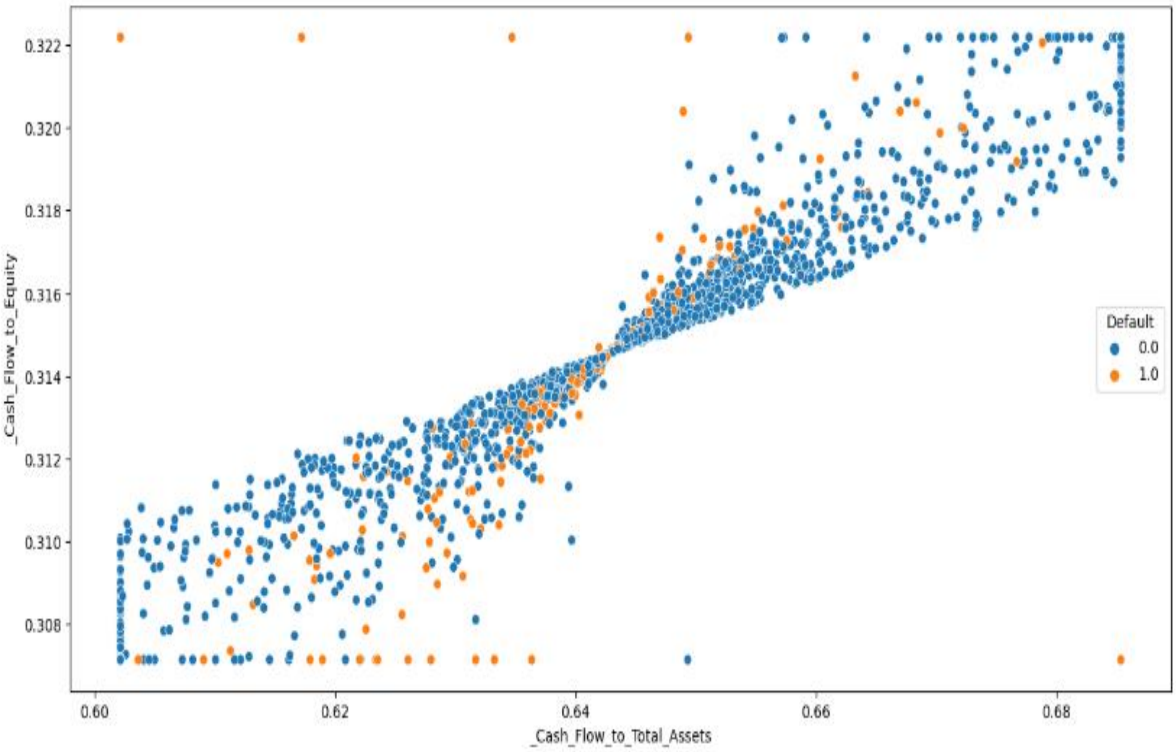
Observations

- 2051 companies have total assets more than total liabilities.

- Only 7 companies have more total liabilities than their total assets.

vi) Cash_Flow_to_Total_Assets, _Cash_Flow_to_Equity and Default:

Figure 9: Cash_Flow_to_Total_Assets, Cash_Flow_to_Equity and Default



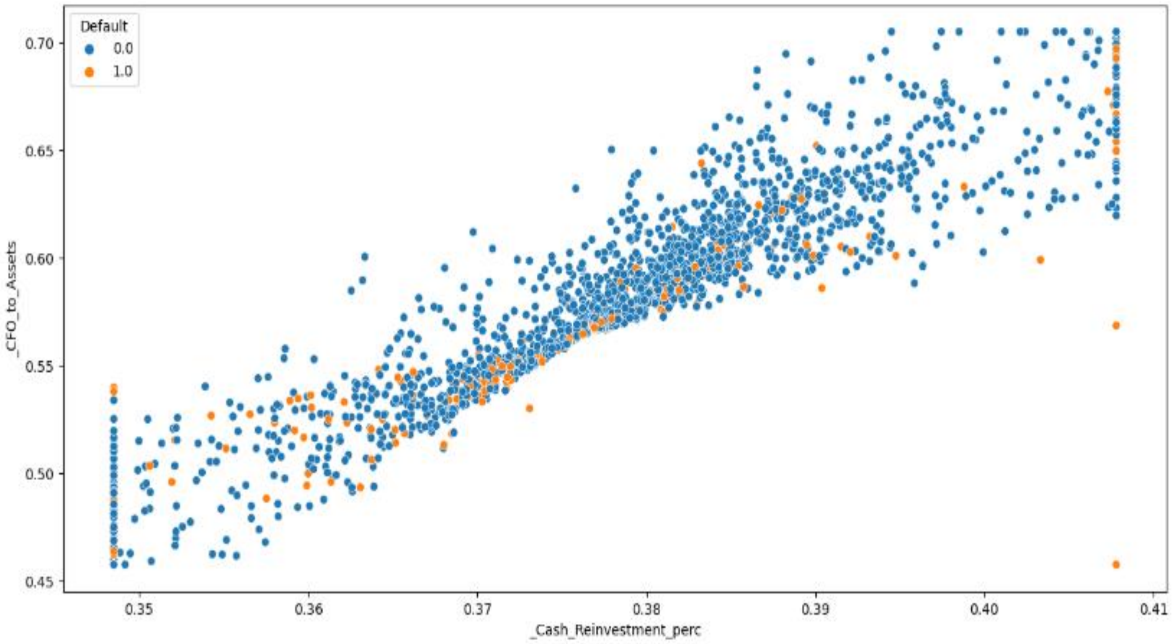
Observations

- We can say that ‘cash flow to total assets’ and ‘cash flow to equity’ are positively correlated.

- Most of the defaulters approximately have ‘cash flow to total assets’ below 0.645 and ‘cash flow to equity’ below 0.315.

vii) Cash_Reinvestment_perc, _CFO_to_Assets and Default:

Figure 10: Cash_Reinvestment_perc, CFO_to_Assets and Default

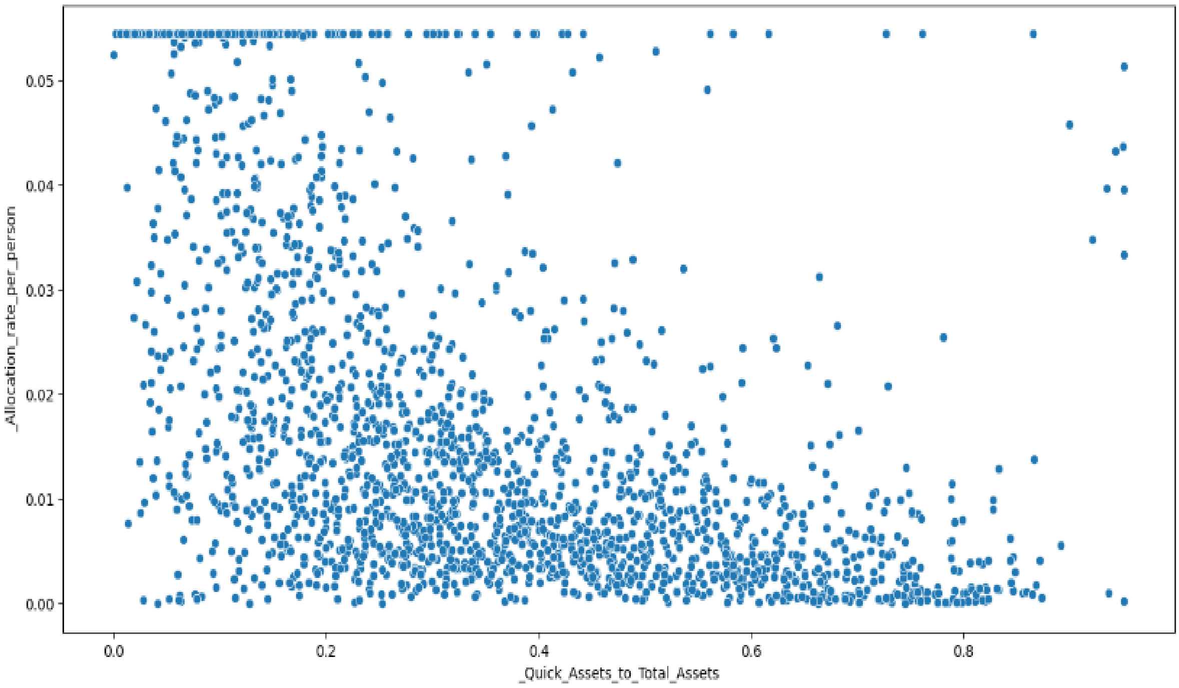


Observation

- 'Cash reinvestment percentage' and 'CFO to assets' is highly correlated.

viii) Quick_Assets_to_Total_Assets and Allocation_rate_per_person:

Figure 11: Quick_Assets_to_Total_Assets and Allocation_rate_per_person



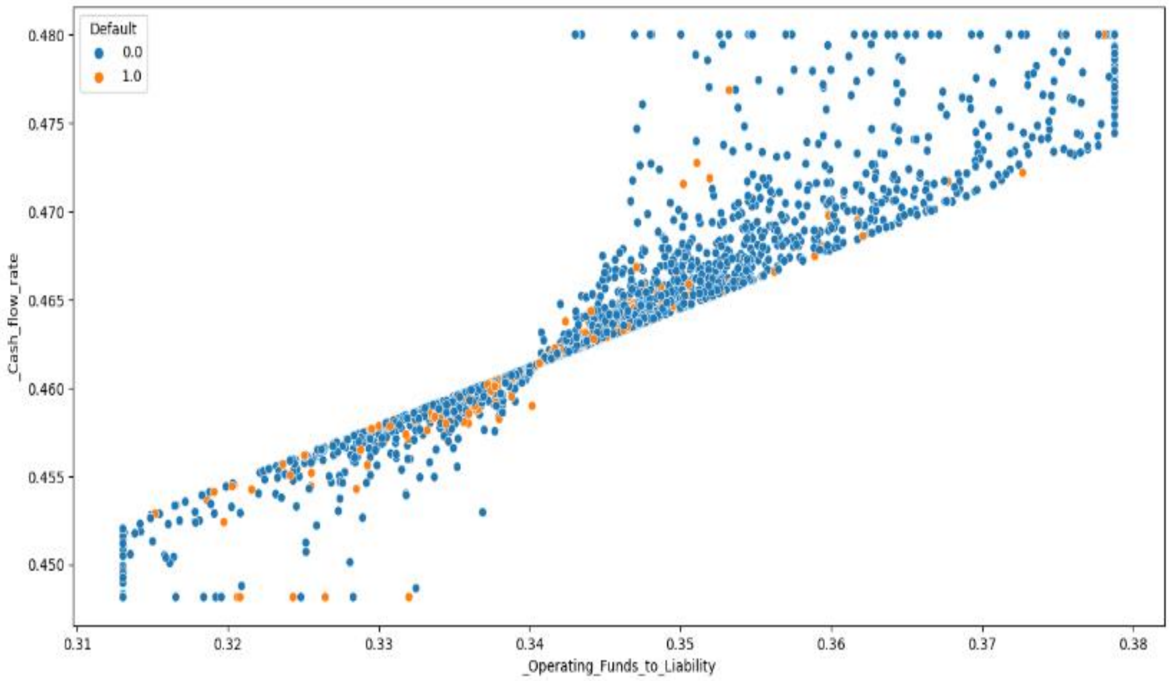
Observations

- We can note that ‘Quick assets to total assets’ and ‘Allocation rate per person’ are uncorrelated.

- Scatters in the plot are widely spread showing no correlation between the variables.

ix) Operating_Funds_to_Liability, _Cash_flow_rate and Default:

Figure 12: Operating_Funds_to_Liability, Cash_flow_rate and Default



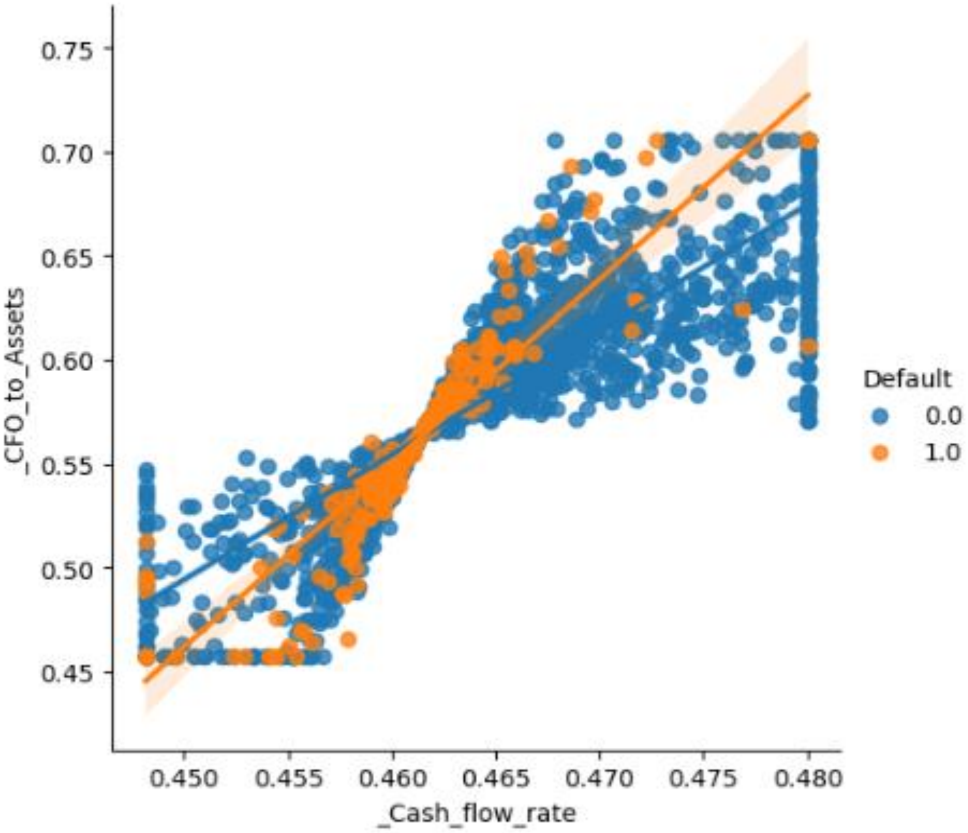
Observations

- ‘Operating funds to liability’ and ‘cash flow rate’ variables are highly correlated between each other, i.e., these two variables have strong correlation of 0.96.

- We can also note that most of the defaulters are in the lower half of the plot.

x) Cash_flow_rate, _CFO_to_Assets and Default:

Figure 13: Cash_flow_rate, CFO_to_Assets and Default



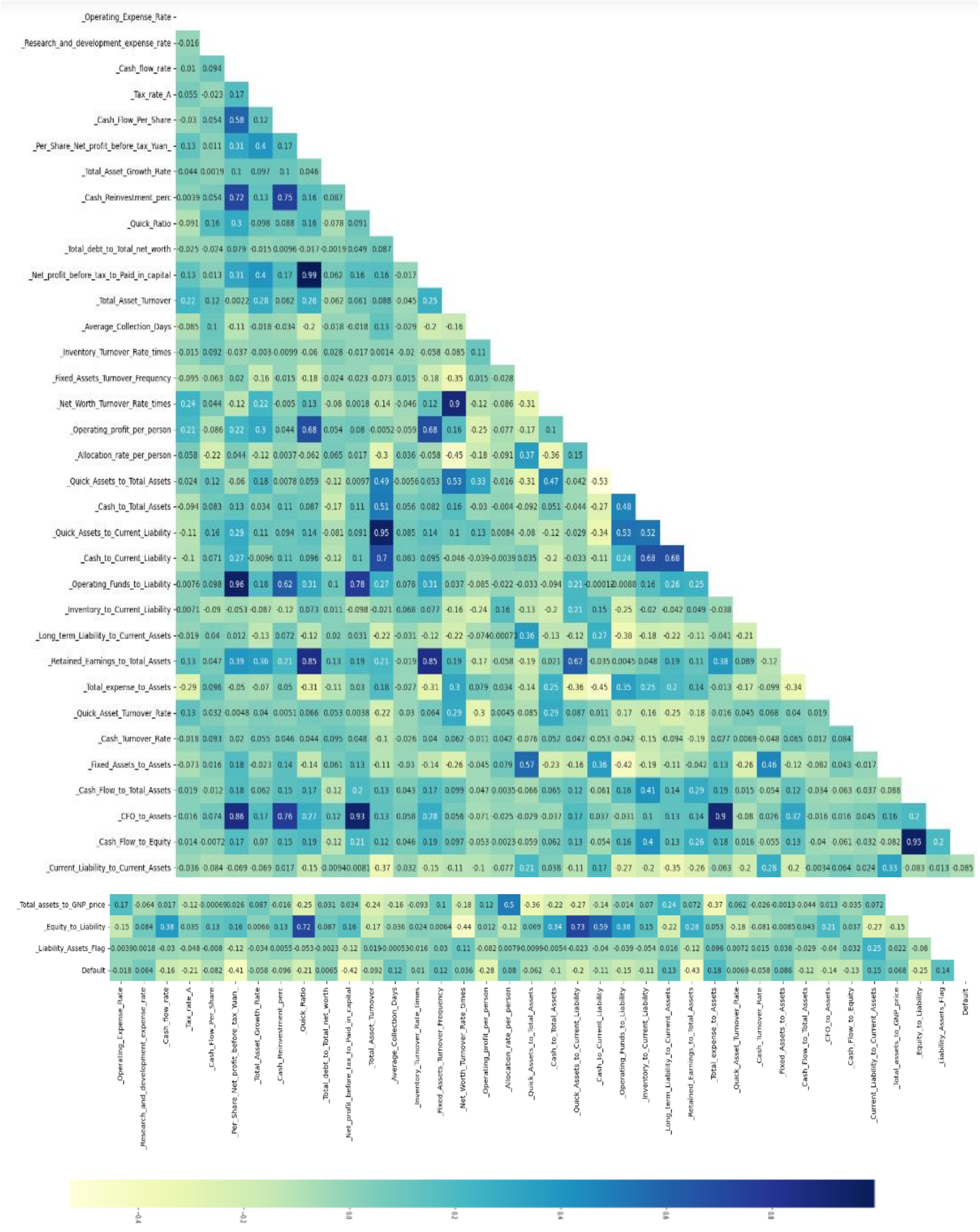
Observations

- ‘CFO to assets’ and ‘Cash flow rate’ are positively correlated (0.86).

- Most of the non-defaulters have high ‘CFO to assets’ ratio and ‘cash flow rate’.

xi) Correlation:

Figure 14: Heatmap to check correlation between variables



Observations

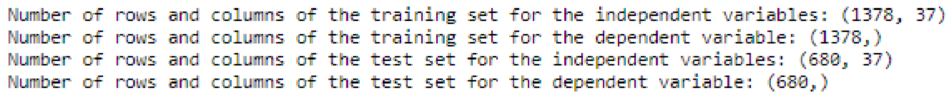
- The highest correlation of 0.99 is between the variables ‘Net profit before tax to Paid in capital’ and ‘Per Share Net profit before tax Yuan’.

- The lowest correlation of -0.53 is between the variables ‘Quick Assets to Total Assets’ and ‘Allocation rate per person’.

1.4 Train Test Split

Ans:

- Splitting the data into Train and Test in the ratio 67:33 after removing constant variables.
- Random state of 42 is been used while splitting the data.
- Train data consists of 1378 rows and 37 columns.
- Test data consists of 680 rows and 37 columns.



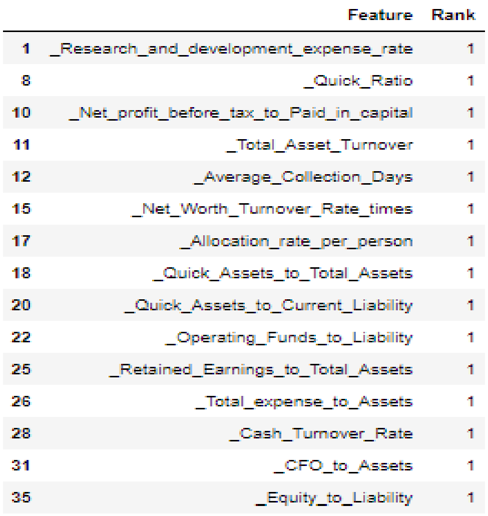
```
Number of rows and columns of the training set for the independent variables: (1378, 37)
Number of rows and columns of the training set for the dependent variable: (1378,)
Number of rows and columns of the test set for the independent variables: (680, 37)
Number of rows and columns of the test set for the dependent variable: (680,)
```

1.5 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.

Ans:

- Initializing and Fitting the **Logistic Regression Model.**
- Choosing 15 number of features and ranking them.

Table 3: Top ranked features

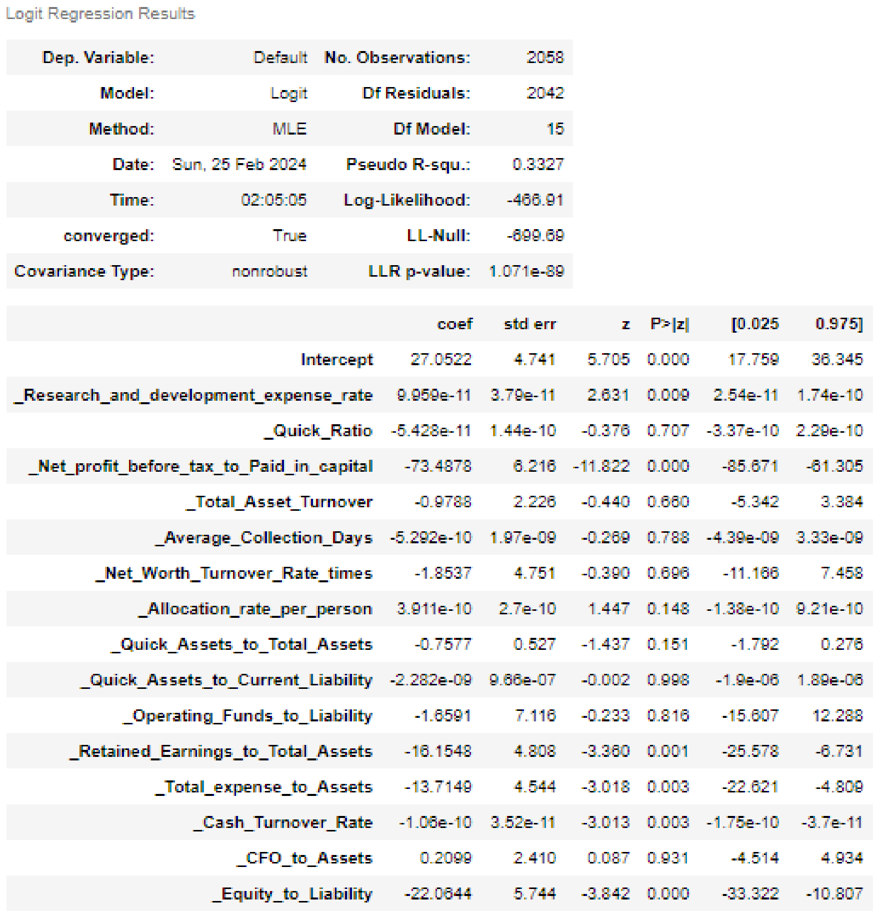


	Feature	Rank
1	_Research_and_development_expense_rate	1
8	_Quick_Ratio	1
10	_Net_profit_before_tax_to_Paid_in_capital	1
11	_Total_Asset_Turnover	1
12	_Average_Collection_Days	1
15	_Net_Worth_Turnover_Rate_times	1
17	_Allocation_rate_per_person	1
18	_Quick_Assets_to_Total_Assets	1
20	_Quick_Assets_to_Current_Liability	1
22	_Operating_Funds_to_Liability	1
25	_Retained_Earnings_to_Total_Assets	1
26	_Total_expense_to_Assets	1
28	_Cash_Turnover_Rate	1
31	_CFO_to_Assets	1
35	_Equity_to_Liability	1

- With the top ranked features, we build models using Logistic regression for ‘Probability at Default’.

- Describe the ‘Default’ variable with all independent variables and checking the model summary.

Table 4: Logistic regression - model 1 summary



Logit Regression Results

Dep. Variable:	Default	No. Observations:	2058
Model:	Logit	Df Residuals:	2042
Method:	MLE	Df Model:	15
Date:	Sun, 25 Feb 2024	Pseudo R-squ.:	0.3327
Time:	02:05:05	Log-Likelihood:	-466.91
converged:	True	LL-Null:	-699.69
Covariance Type:	nonrobust	LLR p-value:	1.071e-89

	coef	std err	z	P>|z|	[0.025	0.975]
Intercept	27.0522	4.741	5.705	0.000	17.759	36.345
_Research_and_development_expense_rate	9.959e-11	3.79e-11	2.631	0.009	2.54e-11	1.74e-10
_Quick_Ratio	-5.428e-11	1.44e-10	-0.376	0.707	-3.37e-10	2.29e-10
_Net_profit_before_tax_to_Paid_in_capital	-73.4878	6.216	-11.822	0.000	-85.671	-61.305
_Total_Asset_Turnover	-0.9788	2.226	-0.440	0.660	-5.342	3.384
_Average_Collection_Days	-5.292e-10	1.97e-09	-0.269	0.788	-4.39e-09	3.33e-09
_Net_Worth_Turnover_Rate_times	-1.8537	4.751	-0.390	0.696	-11.166	7.458
_Allocation_rate_per_person	3.911e-10	2.7e-10	1.447	0.148	-1.38e-10	9.21e-10
_Quick_Assets_to_Total_Assets	-0.7577	0.527	-1.437	0.151	-1.792	0.276
_Quick_Assets_to_Current_Liability	-2.282e-09	9.66e-07	-0.002	0.998	-1.9e-06	1.89e-06
_Operating_Funds_to_Liability	-1.6591	7.116	-0.233	0.816	-15.607	12.288
_Retained_Earnings_to_Total_Assets	-16.1548	4.808	-3.360	0.001	-25.578	-6.731
_Total_expense_to_Assets	-13.7149	4.544	-3.018	0.003	-22.621	-4.809
_Cash_Turnover_Rate	-1.06e-10	3.52e-11	-3.013	0.003	-1.75e-10	-3.7e-11
_CFO_to_Assets	0.2099	2.410	0.087	0.931	-4.514	4.934
_Equity_to_Liability	-22.0644	5.744	-3.842	0.000	-33.322	-10.807

Observation

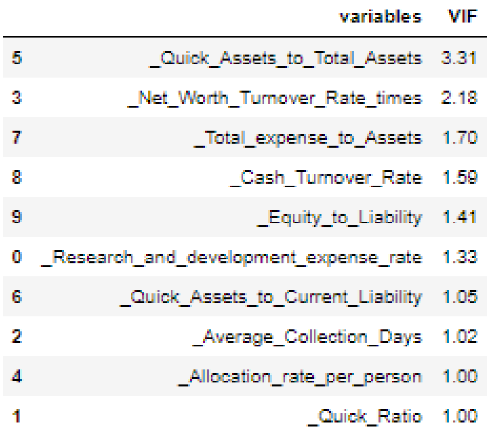
- We can observe that there are few variables with p-values greater than 0.05, which means those are insignificant so we can go ahead and drop them.

- **Variance Inflation factor:**

  - VIF method is used to treat multicollinearity between the variables, i.e., by dropping the features with VIF above 5.

  - Dropping all the insignificant features one by one which have VIF above 5.

Table 5: Features after treating multicollinearity



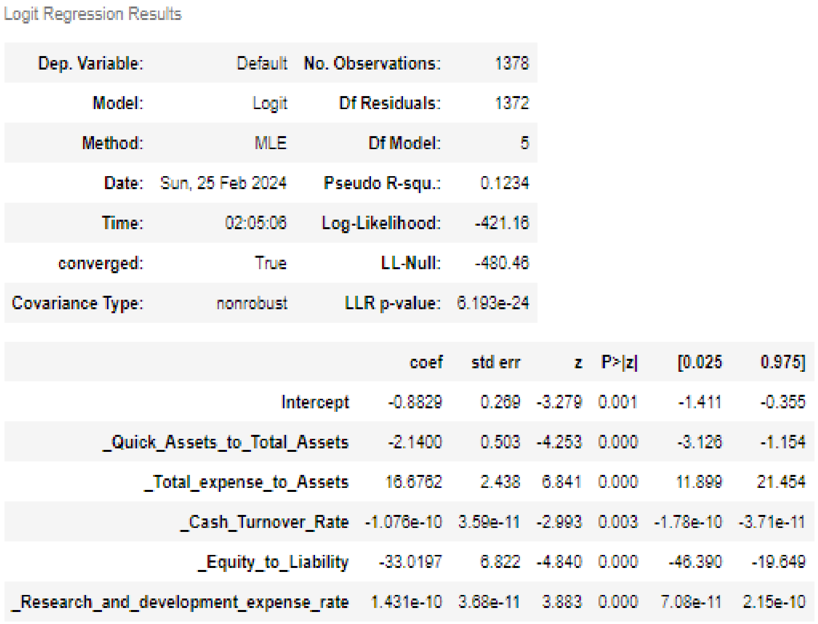
	variables	VIF
5	_Quick_Assets_to_Total_Assets	3.31
3	_Net_Worth_Turnover_Rate_times	2.18
7	_Total_expense_to_Assets	1.70
8	_Cash_Turnover_Rate	1.59
9	_Equity_to_Liability	1.41
0	_Research_and_development_expense_rate	1.33
6	_Quick_Assets_to_Current_Liability	1.05
2	_Average_Collection_Days	1.02
4	_Allocation_rate_per_person	1.00
1	_Quick_Ratio	1.00

- **Train and Test Split:**

  - Splitting the data into Train and Test at 67:33 ratio.

- Removing features from the model summary until all the p-values are less than 0.05.

Table 6: Logistic regression - model 7 summary



Logit Regression Results

Dep. Variable:	Default	No. Observations:	1378
Model:	Logit	Df Residuals:	1372
Method:	MLE	Df Model:	5
Date:	Sun, 25 Feb 2024	Pseudo R-squ.:	0.1234
Time:	02:05:06	Log-Likelihood:	-421.16
converged:	True	LL-Null:	-480.46
Covariance Type:	nonrobust	LLR p-value:	6.193e-24

	coef	std err	z	P>|z|	[0.025	0.975]
Intercept	-0.8829	0.269	-3.279	0.001	-1.411	-0.355
_Quick_Assets_to_Total_Assets	-2.1400	0.503	-4.253	0.000	-3.126	-1.154
_Total_expense_to_Assets	16.6762	2.438	6.841	0.000	11.899	21.454
_Cash_Turnover_Rate	-1.076e-10	3.59e-11	-2.993	0.003	-1.78e-10	-3.71e-11
_Equity_to_Liability	-33.0197	6.822	-4.840	0.000	-46.390	-19.649
_Research_and_development_expense_rate	1.431e-10	3.68e-11	3.883	0.000	7.08e-11	2.15e-10

Observation

- Eliminated all the insignificant features.

- Model_7 summary consists of 5 features, which have p-values less than 0.05 and also does not have multicollinearity issues.

- **Optimal Threshold:**

  - The optimal threshold for the Logistic Regression model is **0.1165**.

- Confusion matrix and Classification report based on the revised threshold of 0.12.

- **<u>Confusion Matrix and Classification report:</u>**

- **Train** data:

Table 7: Logistic regression - Confusion matrix of the Train data

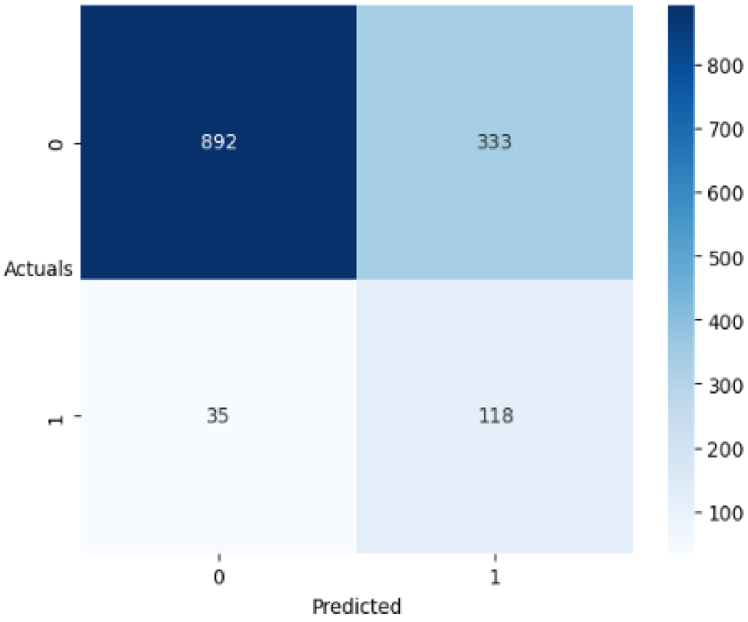
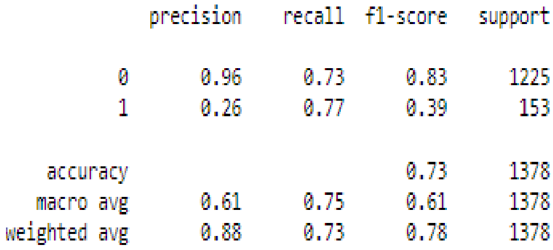


Table 8: Logistic regression - Classification report of the Train data



```
              precision    recall  f1-score   support

           0       0.96      0.73      0.83      1225
           1       0.26      0.77      0.39       153

    accuracy                           0.73      1378
   macro avg       0.61      0.75      0.61      1378
weighted avg       0.88      0.73      0.78      1378
```

<u>1.6</u> Validate the Model on Test Dataset and state the performance metrics. Also state interpretation from the model.

Ans:

- **Confusion Matrix and Classification report:**

- **Test** data:

Table 9: Logistic regression - Confusion matrix of the Test data

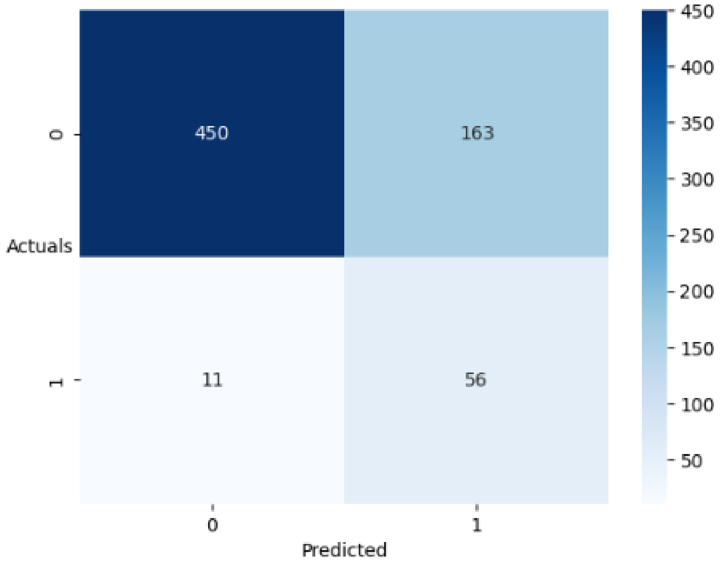
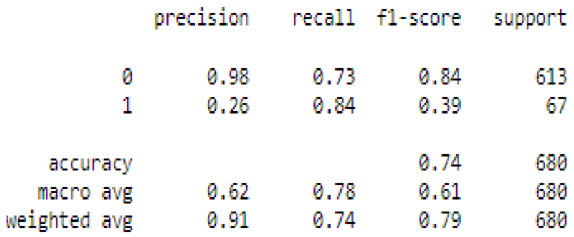


Table 10: Logistic regression - Classification report of the Test data



```
              precision    recall  f1-score   support

           0       0.98      0.73      0.84       613
           1       0.26      0.84      0.39        67

    accuracy                           0.74       680
   macro avg       0.62      0.78      0.61       680
weighted avg       0.91      0.74      0.79       680
```

Observation & Interpretation:

- Accuracy score of Train data is 0.73 and Test data is 0.74.

- Train data have correctly predicted 892 True positives and 118 True negatives.

- Model has good precision and recall score of non-defaulters.

- Recall percentage of defaulters have improved from Train data 0.77 to Test data 0.84.

- Model does not hold good with Precision, as both train and test data for defaulters have precision score of 0.26.

- However, in this case the primary focus is on the recall score of defaulters, i.e., of all actual defaulters how many were predicted as defaulters.

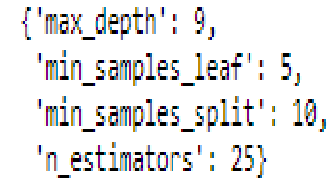
- The main objective of this model is to increase the recall score, as the company or the investor does not want to invest in a company who defaults.

- This model has good recall score of 0.84.

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

Ans:

- Initializing and Fitting the **Random Forest Model.**
- We are performing and considering grid search on Random Forest model, as we can tune hyperparameters.
- Grid search best parameters.



```
{'max_depth': 9,
 'min_samples_leaf': 5,
 'min_samples_split': 10,
 'n_estimators': 25}
```

- **Confusion Matrix and Classification report:**

- **Train** data:

Table 11: Random Forest - Confusion matrix of the Train data

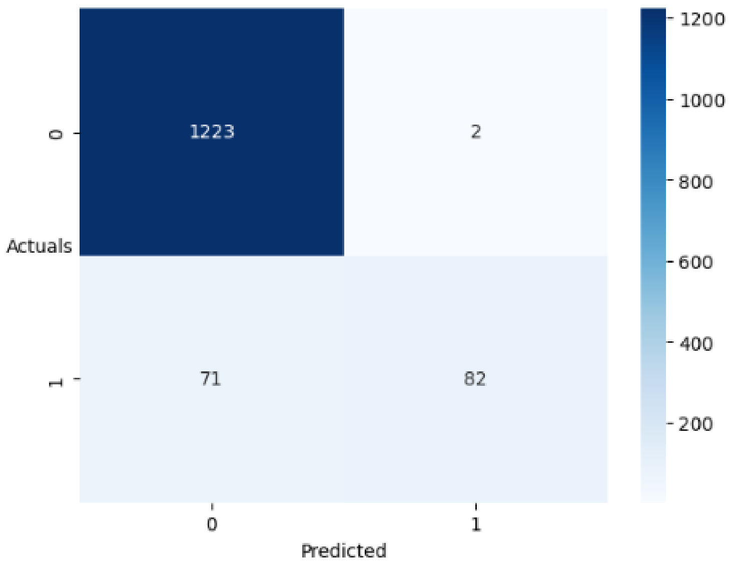
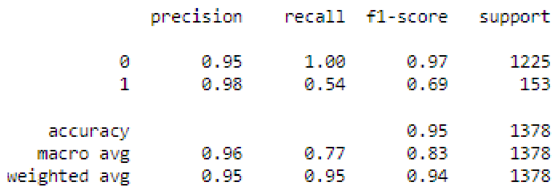


Table 12: Random Forest - Classification report of the Train data



```
              precision    recall  f1-score   support

           0       0.95      1.00      0.97      1225
           1       0.98      0.54      0.69       153

    accuracy                           0.95      1378
   macro avg       0.96      0.77      0.83      1378
weighted avg       0.95      0.95      0.94      1378
```

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

Ans:

- **Confusion Matrix and Classification report:**

- **Test** data:

Table 13: Random Forest - Confusion matrix of the Test data

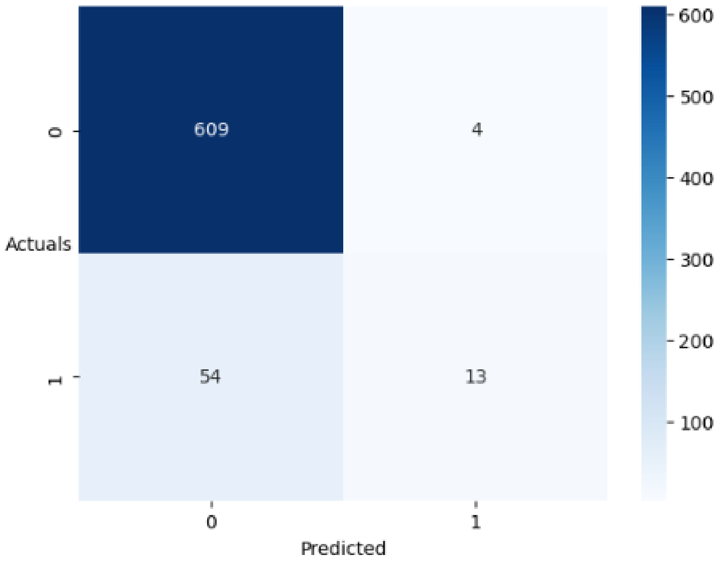
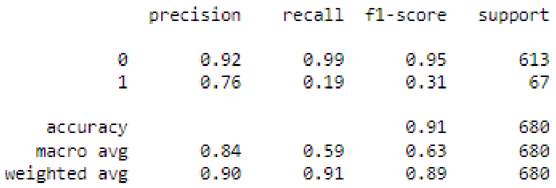


Table 14: Random Forest - Classification report of the Test data



```
              precision    recall  f1-score   support

           0       0.92      0.99      0.95       613
           1       0.76      0.19      0.31        67

    accuracy                           0.91       680
   macro avg       0.84      0.59      0.63       680
weighted avg       0.90      0.91      0.89       680
```

Observation & Interpretation:

- Accuracy score of Train data is 0.95 and Test data is 0.91.

- Train data have correctly predicted 1223 True positives and 82 True negatives.

- Precision score of non-defaulters on Train data is 0.95 and on Test data is 0.92.

- Recall score of non-defaulters on Train data is 1.00 and on Test data is 0.99.

- Recall score of defaulters on Train data is 0.54 which reduced to 0.19 on Test data.

- However, we cannot consider this model because of the lowest recall score on the test data.

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

Ans:

- Initializing and Fitting the **LDA Model.**
- **Optimal Threshold:**

  - The optimal threshold for the LDA model is **0.1052**.

- **Confusion Matrix and Classification report:**

- **Train** data:

Table 15: LDA - Confusion matrix of the Train data

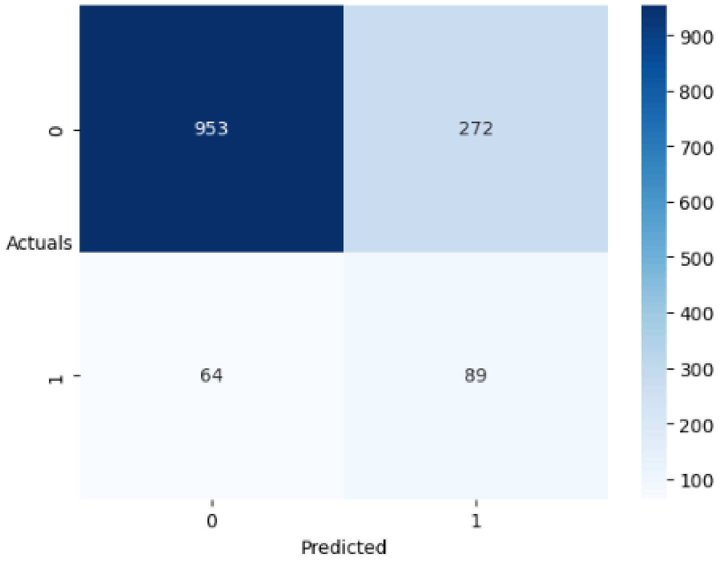
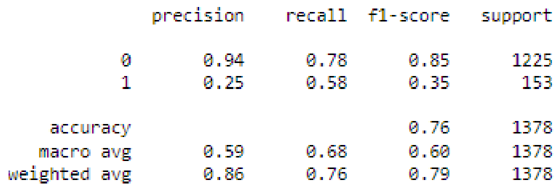


Table 16: LDA - Classification report of the Train data



```
              precision    recall  f1-score   support

           0       0.94      0.78      0.85      1225
           1       0.25      0.58      0.35       153

    accuracy                           0.76      1378
   macro avg       0.59      0.68      0.60      1378
weighted avg       0.86      0.76      0.79      1378
```

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

Ans:

- **Confusion Matrix and Classification report:**

- **Test** data:

Table 17: LDA - Confusion matrix of the Test data

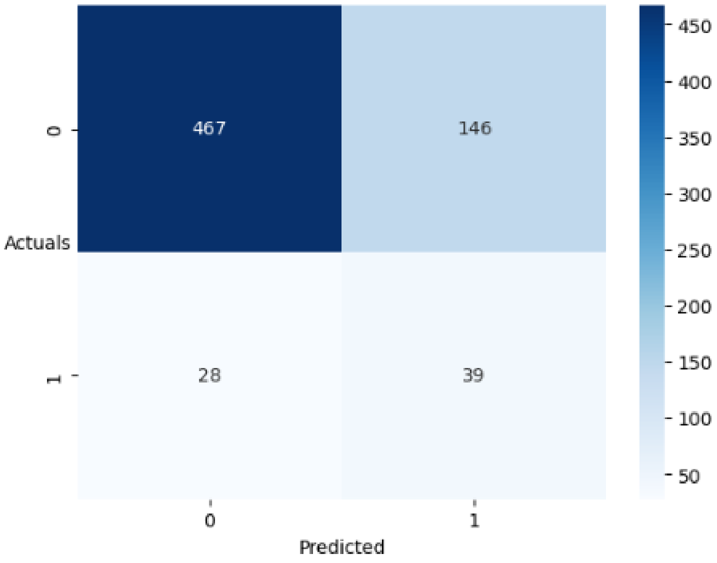
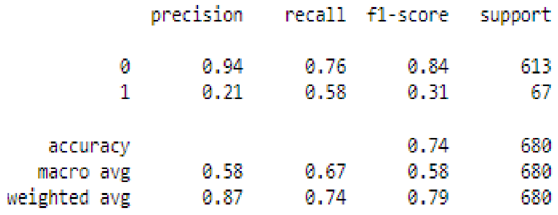


Table 18: LDA - Classification report of the Test data



```
              precision    recall  f1-score   support

           0       0.94      0.76      0.84       613
           1       0.21      0.58      0.31        67

    accuracy                           0.74       680
   macro avg       0.58      0.67      0.58       680
weighted avg       0.87      0.74      0.79       680
```

Observation & Interpretation:

- Accuracy score of Train data is 0.76 and Test data is 0.74.

- Train data have incorrectly predicted 64 False positives and 272 False negatives.

- Test data have incorrectly predicted 28 False positives and 146 False negatives.

- Model has good precision score on non-defaulters.

- Train data have correctly predicted 953 True positives and 89 True negatives.

- Recall percentage of 0.58 on defaulters remains the same on both Train and Test data.

- Model does not hold good with Precision, as both train and test data for defaulters have precision score of 0.25 and 0.21.

<u>1.11</u> Compare the performances of Logistic Regression, Random Forest, and LDA models (include ROC curve).

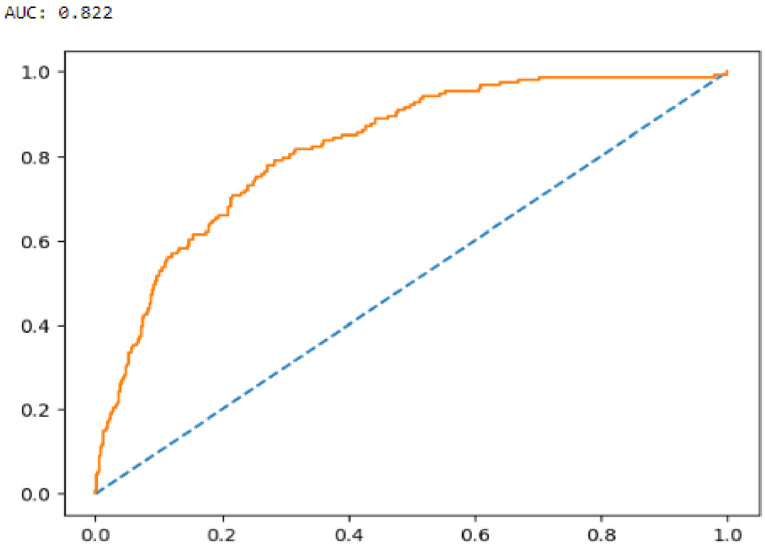
Ans:

i) **<u>Logistic Regression model:</u>**

- **ROC Curve & ROC_AUC score:**

- **<u>Train</u>** <u>data</u>:

Figure 15: Logistic Regression – Train data ROC Curve

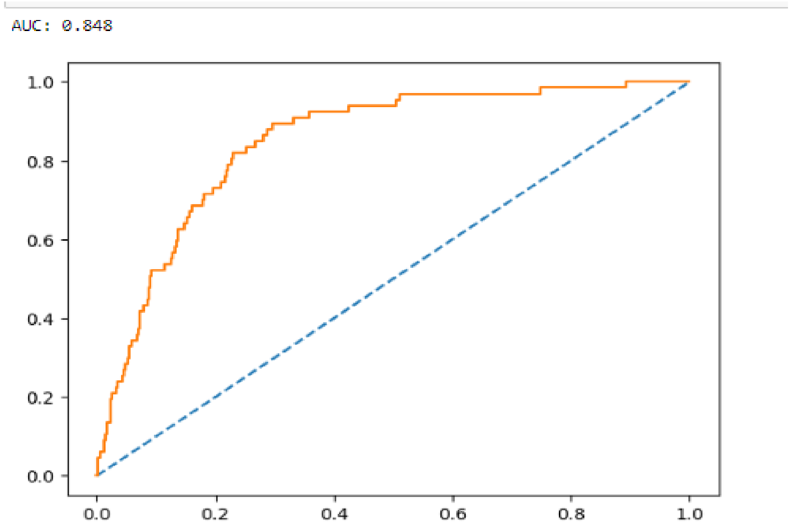


```
AUC: 0.822
```

- **ROC_AUC score for Train data is 0.822**

- **<u>Test</u>** <u>data</u>:

Figure 16: Logistic Regression – Test data ROC Curve



```
AUC: 0.848
```

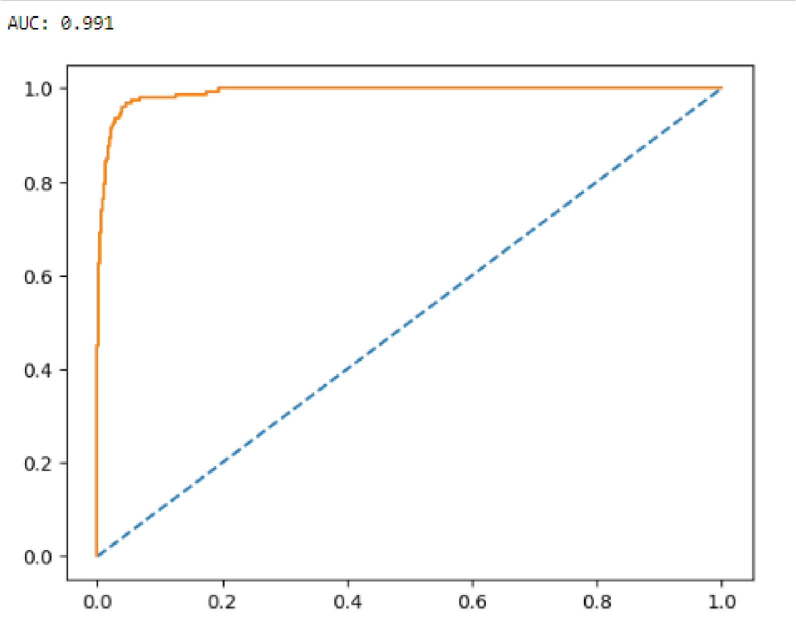
- **ROC_AUC score for Test data is 0.848**

ii) <u>Random Forest model:</u>

- **ROC Curve & ROC_AUC score:**

- **<u>Train</u>** <u>data</u>:

Figure 17: Random Forest – Train data ROC Curve

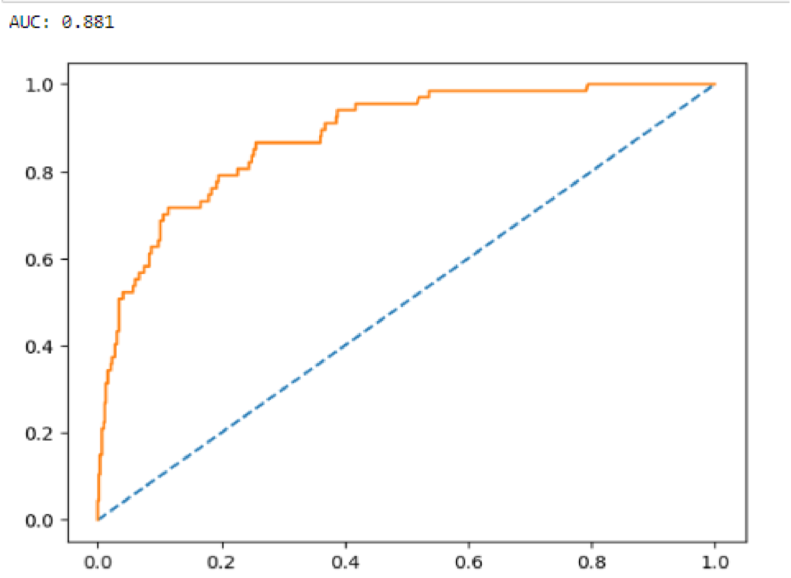


```
AUC: 0.991
```

- **ROC_AUC score for Train data is 0.991**

- **<u>Test</u>** <u>data</u>:

Figure 18: Random Forest – Test data ROC Curve



```
AUC: 0.881
```

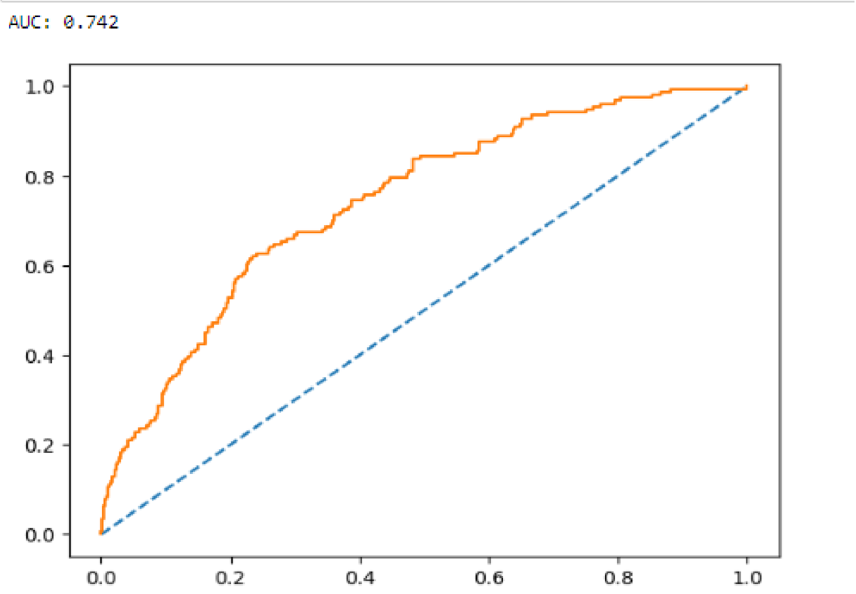
- **ROC_AUC score for Train data is 0.881**

iii) LDA model:

- **ROC Curve & ROC_AUC score:**

- **Train** data:

Figure 19: LDA– Train data ROC Curve

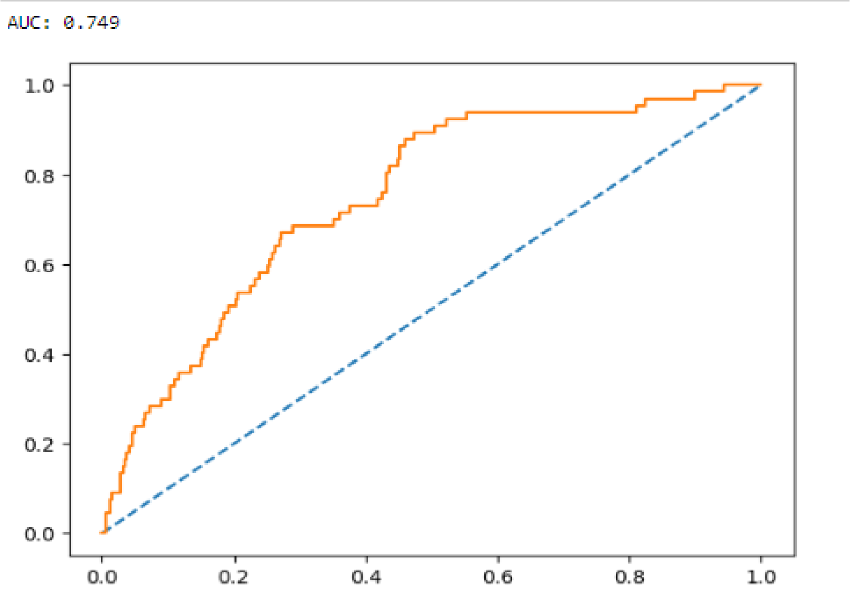


```
AUC: 0.742
```

- **ROC_AUC score for Train data is 0.742**

- **Test** data:

Figure 20: LDA– Test data ROC Curve



```
AUC: 0.749
```

- **ROC_AUC score for Train data is 0.749**

Table 19: Models and model performances

Models	**Model Performances**
a) Logistic Regression	- Model has an accuracy of 73% on Train data and 74% on test data. - Recall increased from 0.77 to 0.84 for conservative data. - Presence of False negatives and False positives is a disadvantage.
b) Random Forest	- Recall percentage on test data is very low i.e.,0.19. - Model has an accuracy of 95% on Train data and 91% on test data.
c) LDA	- Model has an accuracy of 76% on Train data and 74% on test data. - Recall percentage (0.58) remains the same for both Train and Test data.

When we compare all the three models, **Logistic Regression model** seems to be the best one because of the high recall percentage (0.84) on test data.

1.12 Conclusions and Recommendations.

Ans:

Conclusion:

Based on the performance and recall percentage of Logistic Regression, Random Forest and LDA models, we can go ahead with **Logistic regression** because of the high recall percentage on train and test data, i.e., 0.77 and 0.84. Company or investors while making an investment decision would want to know about the companies who default so that they can stay away from those companies. Here, the primary focus of model building was to increase Recall percentage, when we compared the same with all three models, Logistic Regression had a very recall rate than other models.

Recommendations.

- Model performances can still be improved by tuning other hyperparameters on the models. Also, SMOTE can be used to overcome the problem of imbalanced data.
- There can be other models which may perform good and have high recall percentages.
- The investors at times, while making investment decisions should take models as reference and can consider their own expertise or others in determining the company's bankruptcy.
- The provided data helps in identifying the current performance of the company and also helps in determining the probabilities for the companies to Default, but this data may not be sufficient to conclude that the company would actually Default or not.
- There are various other factors such as 'yearly growth', 'Age of the company', 'nature of the business', 'financial performance across the years' which needs to be considered in order to increase the accuracy of the prediction. The current situation or performance of the company may not contribute much in actual prediction, as there are high chances for the companies to be a non-defaulter and can be predicted Defaulter. The companies who are under loss can also be non-defaulters.

---THE END--