FINANCE AND RISK ANALYTICS

M.ABINAYA

TABLE OF CONTENTS

PART – A
CREDIT RISK ASSESSMENT

PROBLEM :

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. 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.

DATA DESCRIPTION :

- The given dataset contains 2058 entries with 58 variables.
- It has 57 Numerical and 1 Categorical variables.

```
dups=com.duplicated()
dups.sum()
```

```
0
```

- No Duplicates present in the dataset.
- Default – Dependent variable.

DATA DICTIONARY :

VARIABLES	DESCRIPTION
Co_Code	Company code
Co_Name	Company Name
_Operating_Expense_Rate	Operating Expenses/Net Sales
_Research_and_development_expense_rate	(Research and Development Expenses)/Net Sales
_Cash_flow_rate	Cash Flow from Operating/Current Liabilities
_Interest_bearing_debt_interest_rate	Interest-bearing Debt/Equity
_Tax_rate_A	Effective Tax Rate
_Cash_Flow_Per_Share	After-tax earnings + depreciation on a per-share basis
_Per_Share_Net_profit_before_tax_Yuan_	Pretax Income Per Share
_Realized_Sales_Gross_Profit_Growth_Rate	Realized Sales Gross Profit Growth Rate.
_Operating_Profit_Growth_Rate	Operating Income Growth
_Continuous_Net_Profit_Growth_Rate	Net Income-Excluding Disposal Gain or Loss Growth
_Total_Asset_Growth_Rate	Total Asset Growth
_Net_Value_Growth_Rate	Total Equity Growth
_Total_Asset_Return_Growth_Rate_Ratio	Return on Total Asset Growth

_Cash_Reinvestment_perc	Percentage of annual cash flow that the company invests back into the business as a new investment.
_Current_Ratio	Company's assets / liabilities
_Quick_Ratio	Ability to pay its short-term obligations using only its most liquid assets.
_Interest_Expense_Ratio	Interest Expenses/Total Revenue
_Total_debt_to_Total_net_worth	Total Liability/Equity Ratio
_Long_term_fund_suitability_ratio_A	(Long-term Liability+Equity)/Fixed Assets
_Net_profit_before_tax_to_Paid_in_capital	Pretax Income/Capital
_Total_Asset_Turnover	Net Sales/Average Total Assets
_Accounts_Receivable_Turnover	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.
_Average_Collection_Days	Days Receivable Outstanding
_Inventory_Turnover_Rate_times	Number of days it will take to sell the inventory on hand.
_Fixed_Assets_Turnover_Frequency	(net sales / net fixed assets), calculated over an annual period.
_Net_Worth_Turnover_Rate_times	Equity Turnover
_Operating_profit_per_person	Operation Income Per Employee
_Allocation_rate_per_person	Fixed Assets Per Employee
_Quick_Assets_to_Total_Assets	Quick Assets/Total Assets
_Cash_to_Total_Assets	Cash/Total Assets

_Quick_Assets_to_Current_Liability	Quick Assets/Current Liability
_Cash_to_Current_Liability	Cash/Current Liability
_Operating_Funds_to_Liability	Operating Funds to Liability
_Inventory_to_Working_Capital	Inventory/Working Capital
_Inventory_to_Current_Liability	Inventory/Current Liability
_Long_term_Liability_to_Current_Assets	Long-term Liability to Current Assets
_Retained_Earnings_to_Total_Assets	Retained Earnings to Total Assets
_Total_income_to_Total_expense	Total income/Total expense
_Total_expense_to_Assets	Total expense/Assets
_Current_Asset_Turnover_Rate	Current Assets to Sales
_Quick_Asset_Turnover_Rate	Quick Assets to Sales
_Cash_Turnover_Rate	Cash to Sales
_Fixed_Assets_to_Assets	Fixed Assets to Assets
_Cash_Flow_to_Total_Assets	Cash Flow / Total Assets
_Cash_Flow_to_Liability	current assets (cash or near-cash assets, like notes receivable) - current liabilities (liabilities due during the upcoming accounting period)
_CFO_to_Assets	rate of cash flows to the company assets without being affected by income recognition or income measurements.
_Cash_Flow_to_Equity	cash available to the equity shareholders of a company after all expenses, reinvestment, and debt are paid.
_Current_Liability_to_Current_Assets	Current Liability / Current Assets

_Liability_Assets_Flag	Liability-Assets Flag: 1 if Total Liability exceeds Total Assets, 0 otherwise
_Total_assets_to_GNP_price	Total assets / GNP price.
_No_credit_Interval	_No_credit_Interval
_Degree_of_Financial_Leverage_DFL	sensitivity in fluctuations of a company's overall profitability to the volatility of its operating income caused by changes in its capital structure.
_Interest_Coverage_Ratio_Interest_expense_to_EBIT	Interest coverage ratio is a debt and profitability ratio
_Net_Income_Flag	Net Income Flag: 1 if Net Income is Negative for the last two years, 0 otherwise
_Equity_to_Liability	Equity / Liability.
Default	Whether the Company has Default (Bankrupted) or not? 1 - Defaulted, 0 - Not Defaulted.

DATA SUMMARY :

HEAD :

```
com.head()
```

	Co_Code	Co_Name	_Operating_Expense_Rate	_Research_and_development_expense_rate	_Cash_flow_rate	_Interest_bearing_debt_interest_rate	_Tax_rate_
0	16974	Hind.Cables	8820000000.00	0.00	0.46	0.00	0.0
1	21214	Tata Tele. Mah.	9380000000.00	4230000000.00	0.46	0.00	0.0
2	14852	ABG Shipyard	3800000000.00	815000000.00	0.45	0.00	0.0
3	2439	GTL	6440000000.00	0.00	0.46	0.00	0.0
4	23505	Bharati Defence	3680000000.00	0.00	0.46	0.00	0.4

5 rows × 58 columns

TAIL :

```
com.tail()
```

	Co_Code	Co_Name	_Operating_Expense_Rate	_Research_and_development_expense_rate	_Cash_flow_rate	_Interest_bearing_debt_interest_rate	_Tax_r
2053	2743	Kothari Ferment.	0.00	6490000000.00	0.48	0.00	
2054	21216	Firstobj.Tech.	0.00	0.00	0.47	0.00	
2055	142	Diamines & Chem.	0.00	8370000000.00	0.48	0.00	
2056	18014	IL&FS Engg.	3750000000.00	0.00	0.47	0.00	
2057	43229	Channel Nine	0.00	0.00	0.47	0.00	

5 rows × 58 columns

INFO :

```
<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
```

```
com.shape
```

```
(2058, 58)
```

DESCRIBE :

	Co_Code	_Operating_Expense_Rate	_Research_and_development_expense_rate	_Cash_flow_rate	_Interest_bearing_debt_interest_rate	_Tax_rate_A	_Casl
count	2058.00	2058.00	2058.00	2058.00	2058.00	2058.00	
mean	17572.11	2052388835.76	1208634256.56	0.47	11130223.52	0.11	
std	21892.89	3252623690.29	2144568158.08	0.02	90425949.04	0.15	
min	4.00	0.00	0.00	0.00	0.00	0.00	
25%	3674.00	0.00	0.00	0.46	0.00	0.00	
50%	6240.00	0.00	0.00	0.46	0.00	0.04	
75%	24280.75	4110000000.00	1550000000.00	0.47	0.00	0.22	
max	72493.00	9980000000.00	9980000000.00	1.00	990000000.00	1.00	

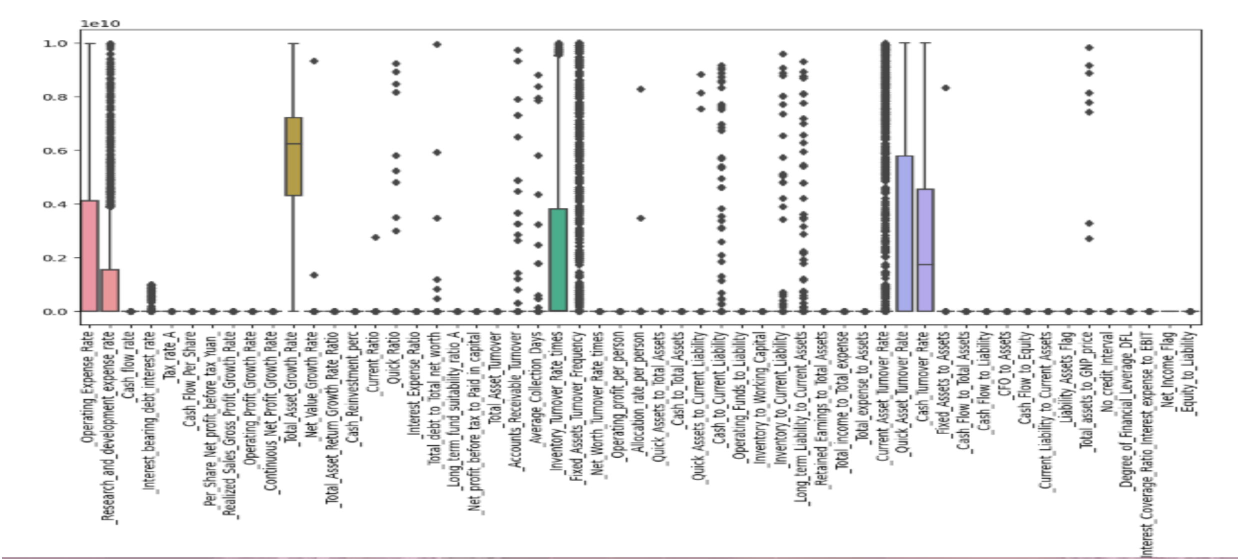
8 rows × 57 columns

SCALING :

```
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
scaled_xcom = pd.DataFrame(scaler.fit_transform(x_com), columns = x_com.columns)
```

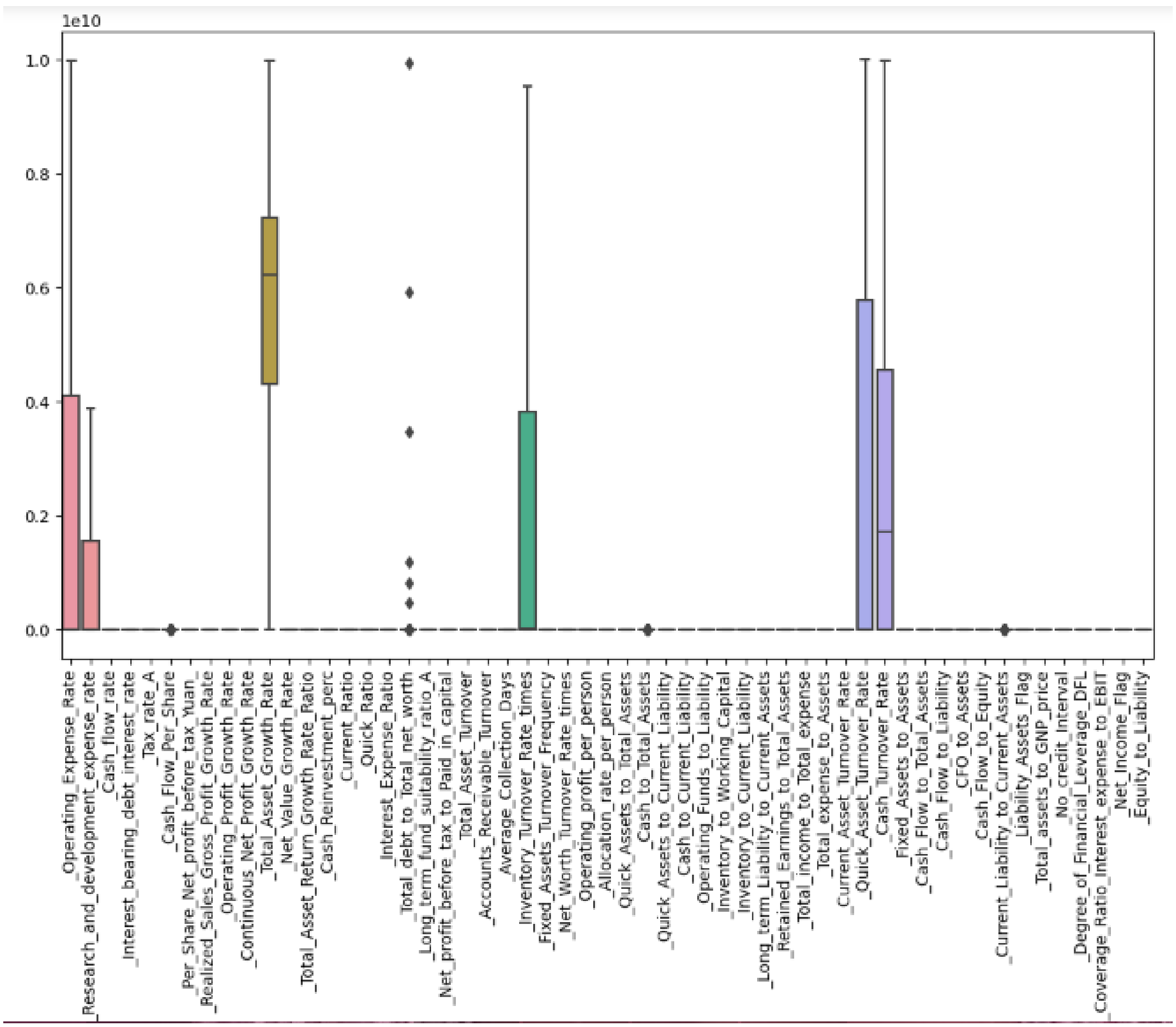
```
xy_com=pd.concat([scaled_xcom,y_com],axis=1)
```

1.1 OUTLIER TREATMENT :



➢ Outliers are present in all column except _operating_expense_rate,_Total_asset_growth_rate, _Quick_asset_turnover_rate,_cash_turnover_rate, _Net_income_flag.

➢ Outliers can be treated using BOXPLOT Method and some outliers still exists because of missing values and scaling done to the dataset.



1.2 MISSING VALUES :

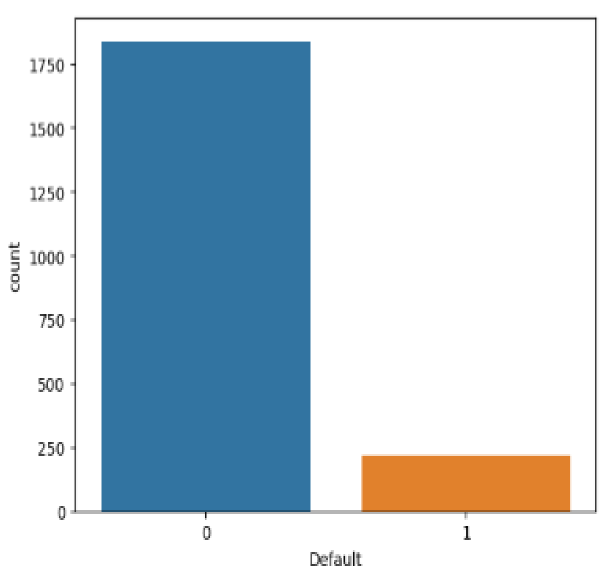
- Presence of missing values in

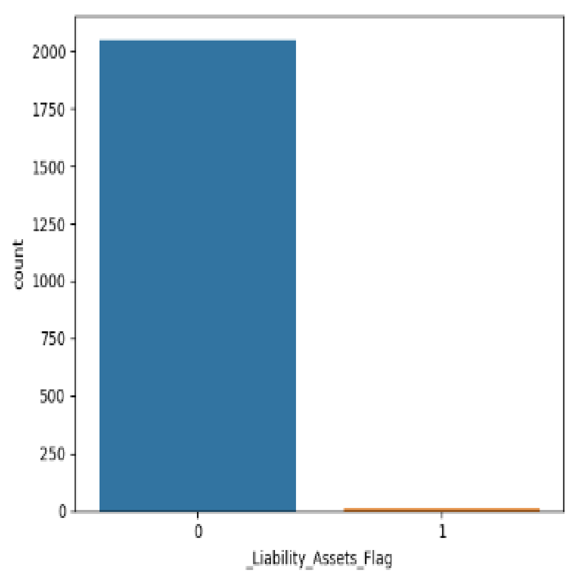
_Cash_Flow_Per_Share	:	167
_Total_debt_to_Total_net_worth	:	21
_Cash_to_Total_Assets	:	96
_Current_Liability_to_Current_Assets	:	14

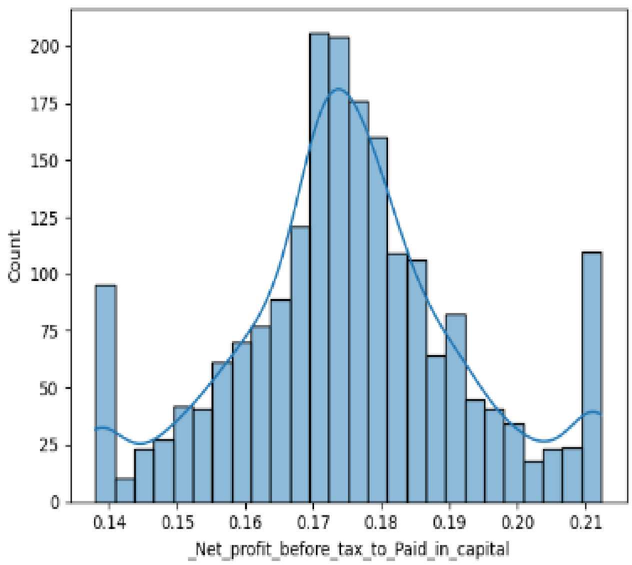
- These missing values has been treated using Median values with the respective variables.

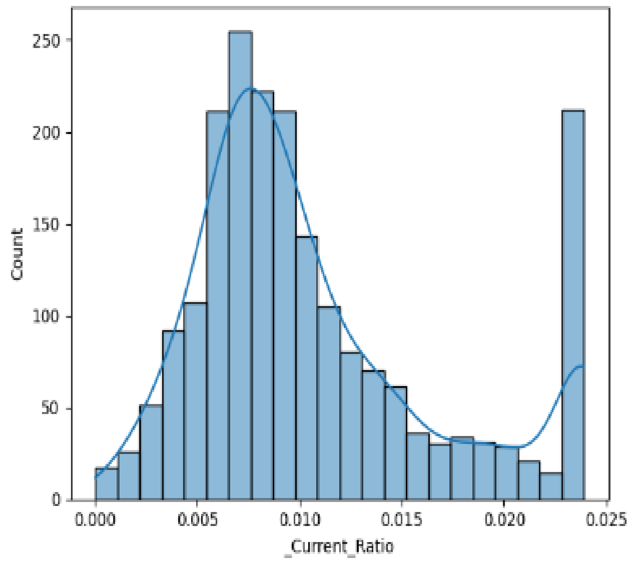
1.3 Univariate (4 marks) & Bivariate (6 marks) analysis with proper interpretation. (You may choose to include only those variables which were significant in the model building).

UNIVARIATE ANALYSIS :

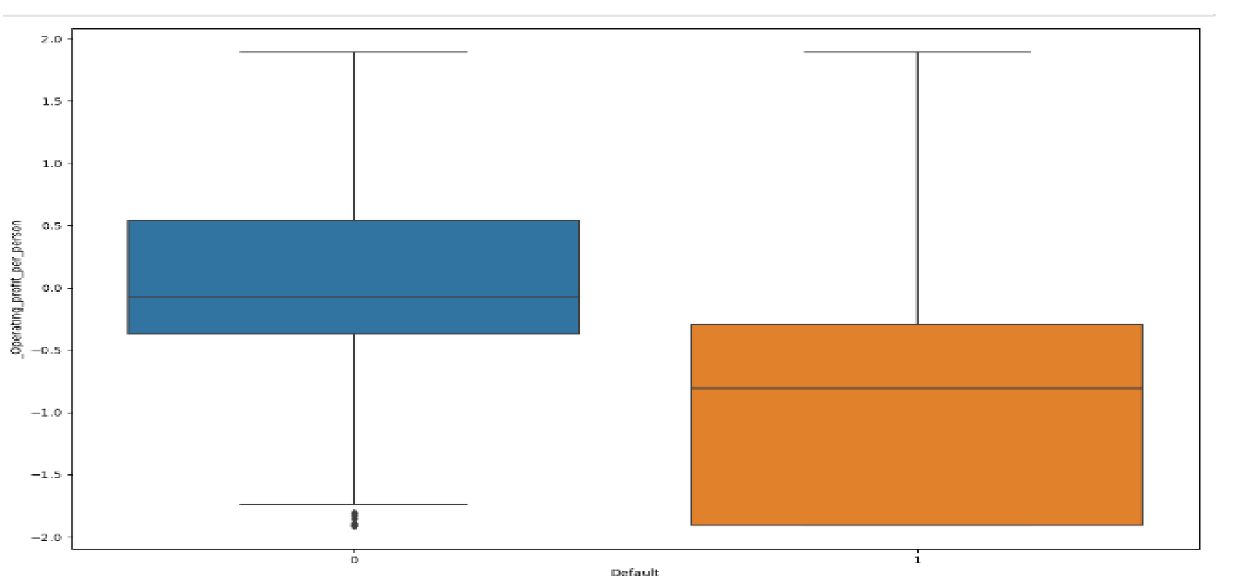


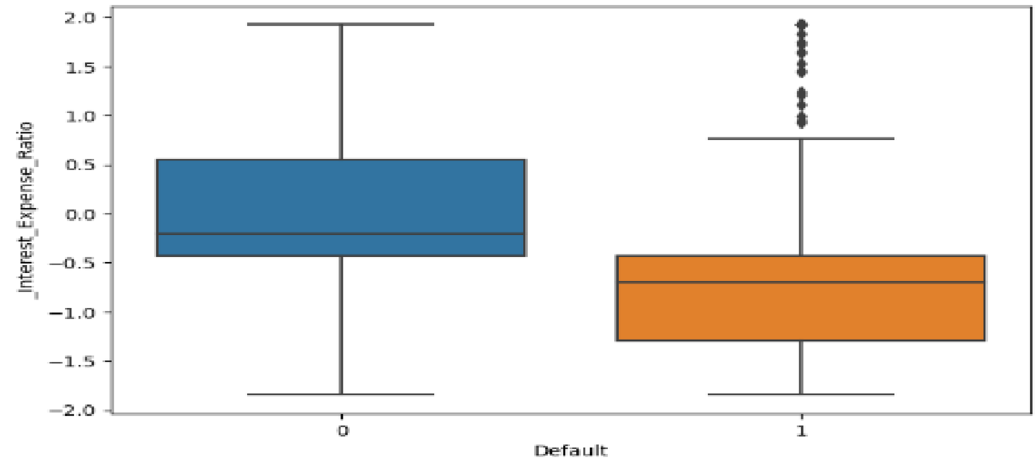




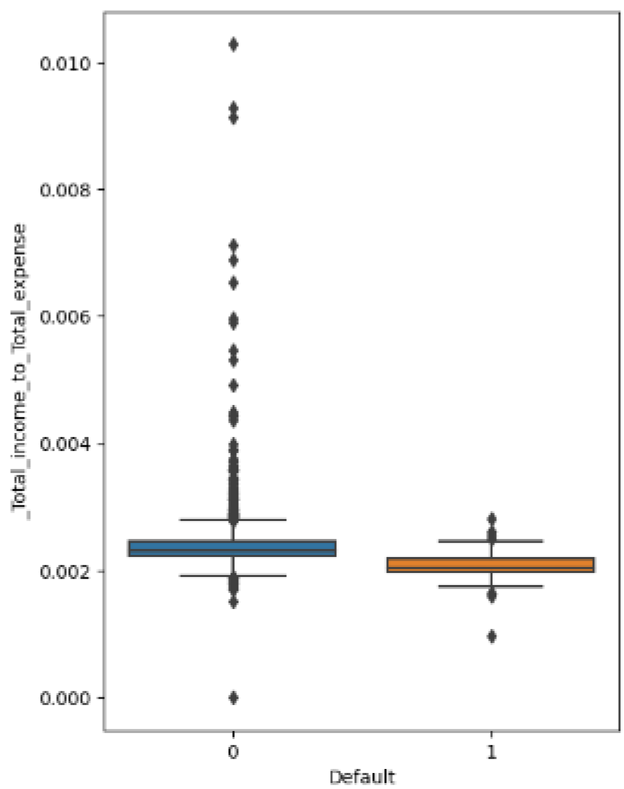


BIVARIATE ANALYSIS :

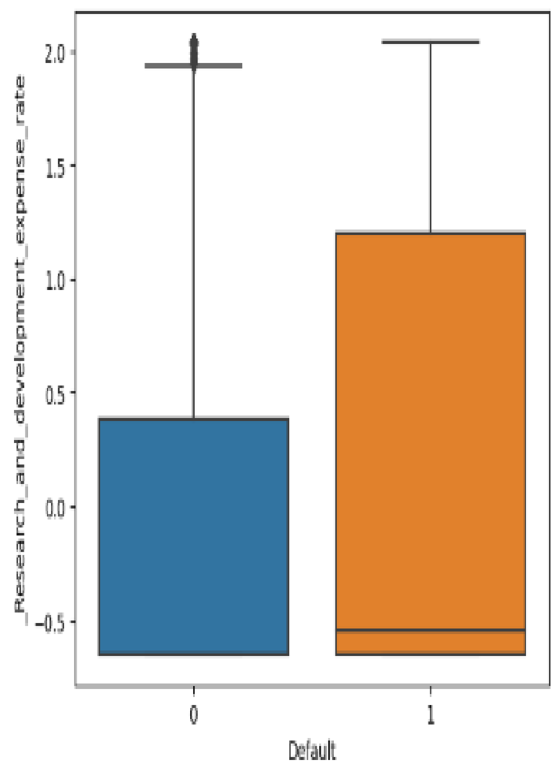




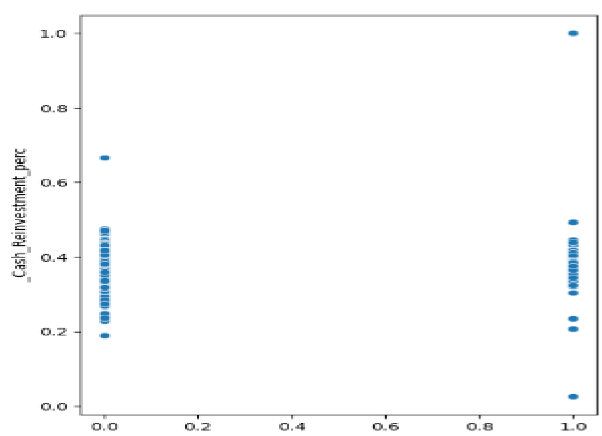
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
_Interest_Expense_Ratio
0
1
Default



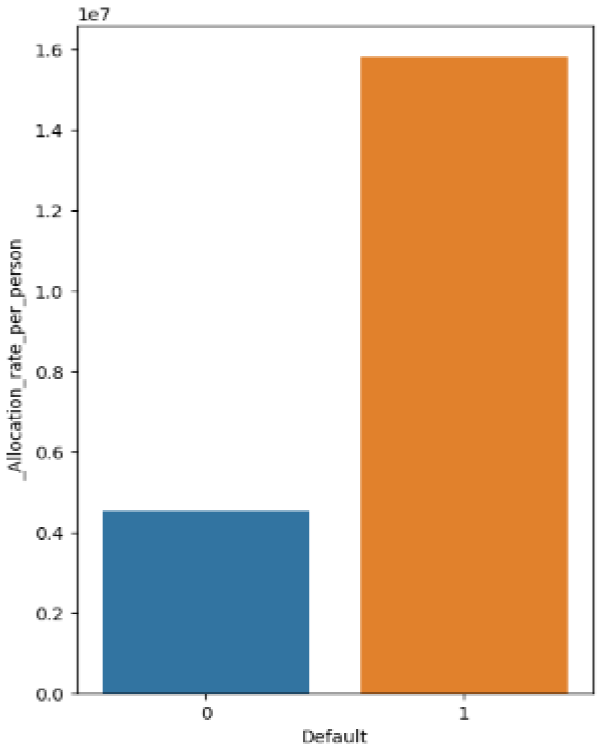
0.010
0.008
0.006
0.004
0.002
0.000
_Total_income_to_Total_expense
0
1
Default

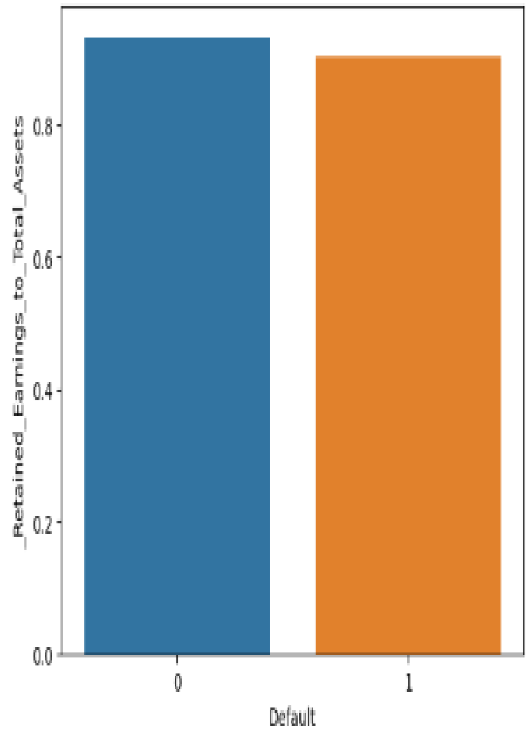


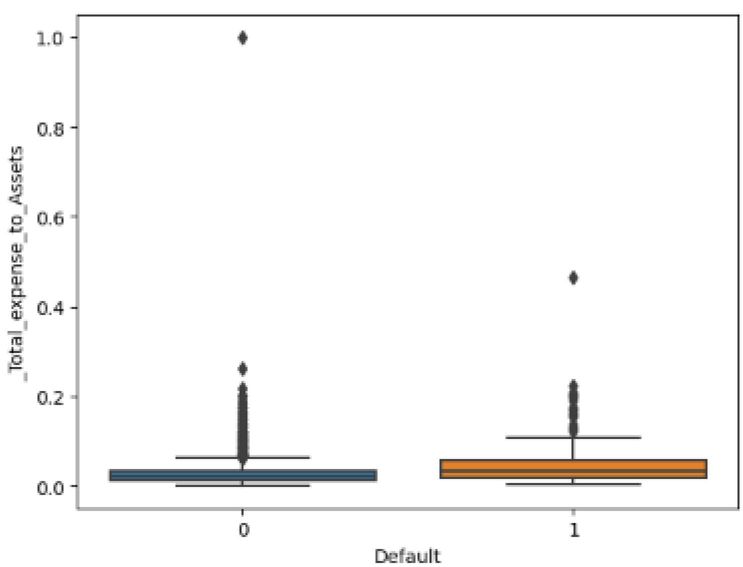
2.0
1.5
1.0
0.5
0.0
−0.5
_Research_and_development_expense_rate
0
1
Default

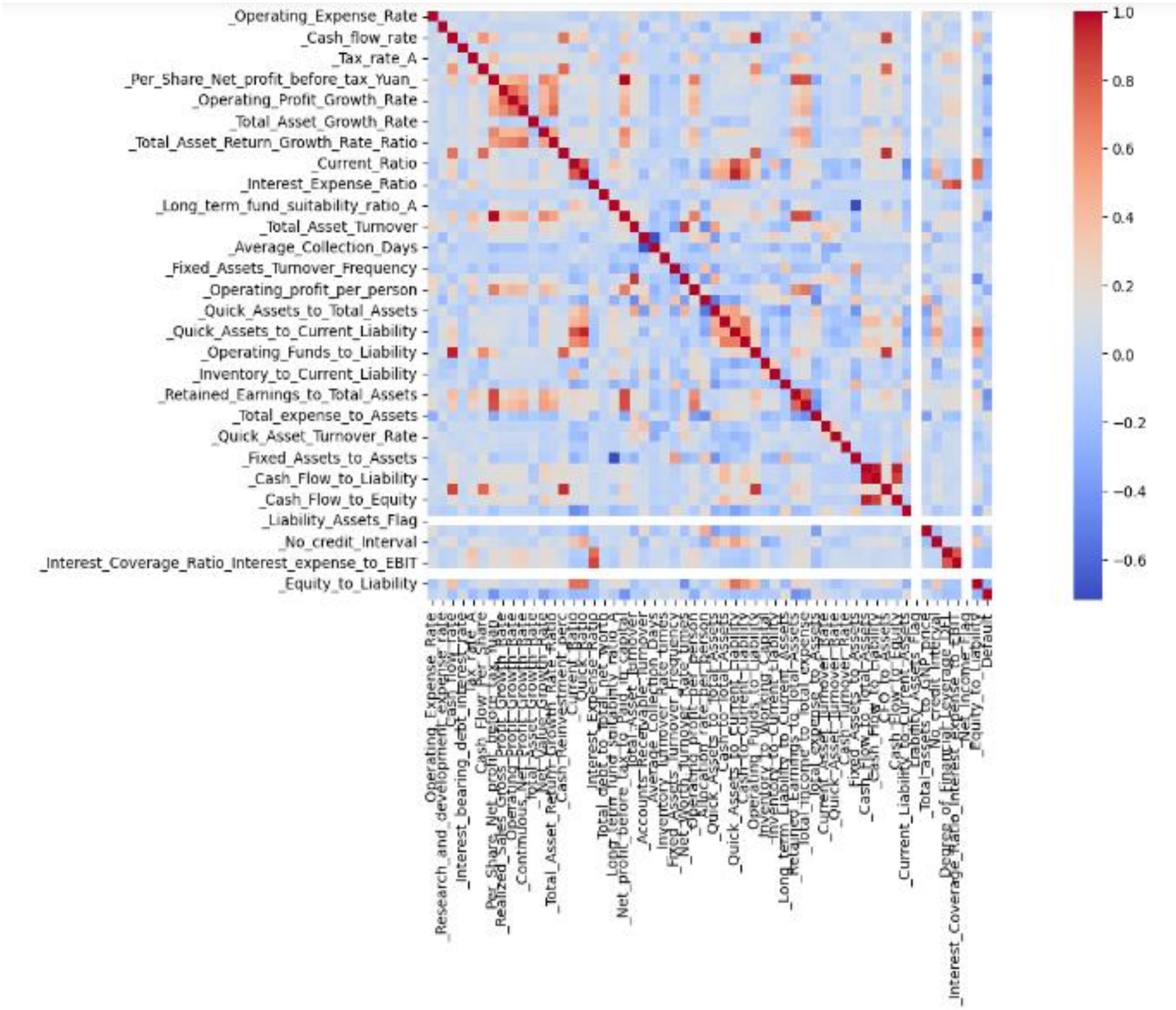


1.0
0.8
0.6
0.4
0.2
0.0
_Cash_Reinvestment_perc
0.0
0.2
0.4
0.6
0.8
1.0
Default


1e7
1.6
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
_Allocation_rate_per_person
0
1
Default


0.8
0.6
0.4
0.2
0.0
_Retained_Earnings_to_Total_Assets
0
1
Default


1.0
0.8
0.6
0.4
0.2
0.0
_Total_expense_to_Assets
0
1
Default

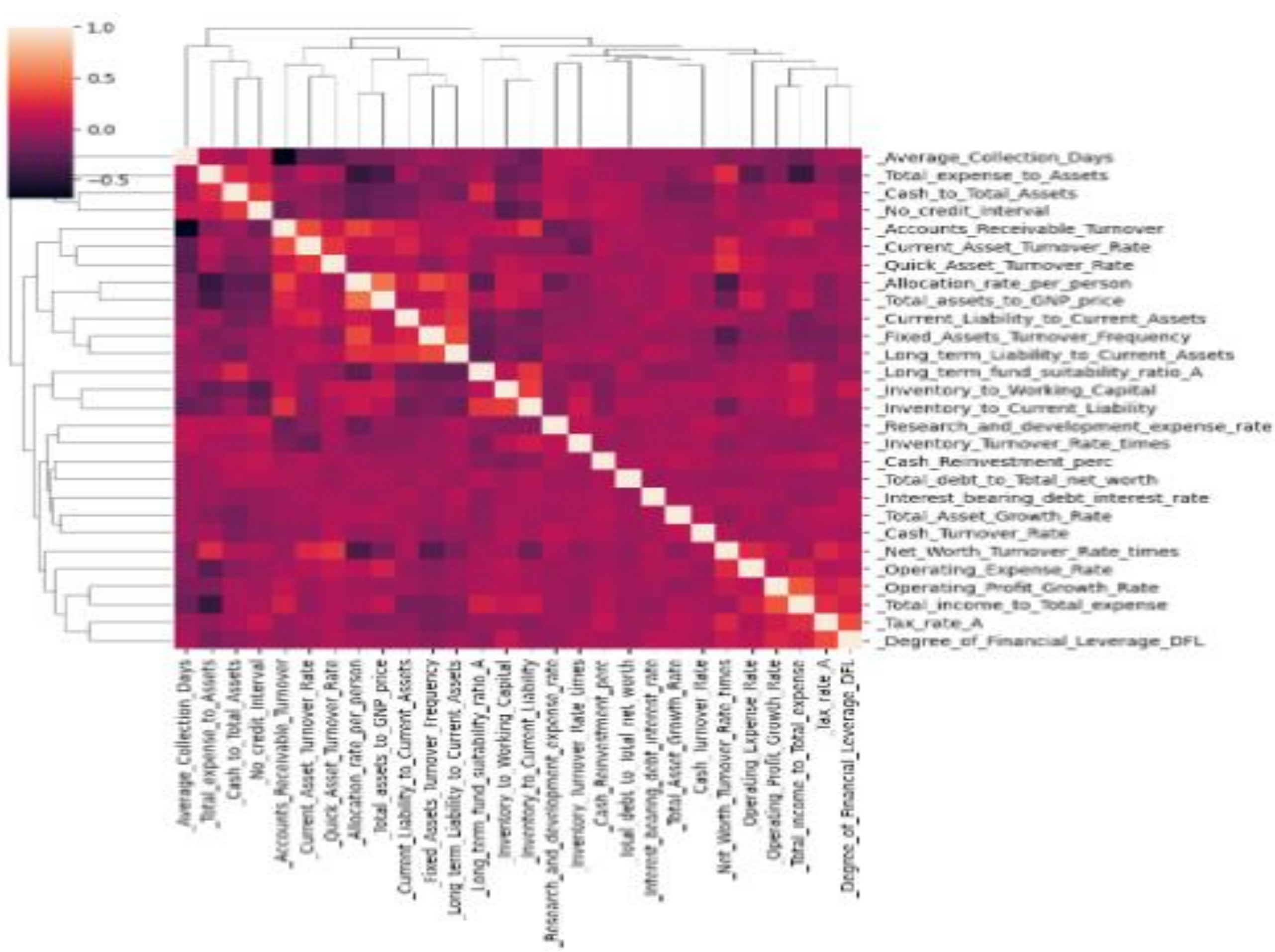


REMOVE MULTI-COLINEARITY :

- Variance Inflation Factor (VIF) is a measure to detect multi collinearity in multiple regression models.

- It quantifies how much the variance of a regression coefficient is inflated due to correlations with other predictors. Variance Inflation Factor tells- How good an independent variable can be explained as a Linear combination of other independent variables (i.e to identify Multicolinearity).

- In current case (with high no of cols, 56),we must remove the redundent columns to avoid multicolinearity before model building
- VIF > 5 is not suitable as it is mostly compensated by other IVs. Hence we use VIF to clean the data of redundent variable.
- After performing VIF, Multi colinearity still exists. So we do cluster mapping to filter variables causing multi collinearity.



➢ After treating with VIF>5 and cluster map, we got 28 variables to perform TRAIN-TEST Split.

```
Index(['_Operating_Expense_Rate', '_Research_and_development_expense_rate',
       '_Interest_bearing_debt_interest_rate', '_Tax_rate_A',
       '_Operating_Profit_Growth_Rate', '_Total_Asset_Growth_Rate',
       '_Cash_Reinvestment_perc', '_Total_debt_to_Total_net_worth',
       '_Long_term_fund_suitability_ratio_A', '_Accounts_Receivable_Turnover',
       '_Average_Collection_Days', '_Inventory_Turnover_Rate_times',
       '_Fixed_Assets_Turnover_Frequency', '_Net_Worth_Turnover_Rate_times',
       '_Allocation_rate_per_person', '_Cash_to_Total_Assets',
       '_Inventory_to_Working_Capital', '_Inventory_to_Current_Liability',
       '_Long_term_Liability_to_Current_Assets',
       '_Total_income_to_Total_expense', '_Total_expense_to_Assets',
       '_Current_Asset_Turnover_Rate', '_Quick_Asset_Turnover_Rate',
       '_Cash_Turnover_Rate', '_Current_Liability_to_Current_Assets',
       '_Total_assets_to_GNP_price', '_No_credit_Interval',
       '_Degree_of_Financial_Leverage_DFL'],
      dtype='object')
```

INFERENCES :

➢ 220 Companies banckrupted out of 2058 (10%)

➢ All the company experienced negative Net Income for last 2 years.

➢ Only 7 companies total Liability exceeds its Asset in critical situation.

➢ On an average, Defaulted bank got 15% of its capital and Non Defaulted bank : 18% of its capital as pre tax income.

- On an average, Non Defaulted company's asset 1496191 and Defaulted – 0.00723 yuan more than its liabilities.

- On an average , Defaulted company's liabilities – 0.7% and Non Defaulted company's liabilities – 1.5% more than its assets.

- Majority of companies reinvest their 38% of income back to the business.

- The operating income per employee differs for both 41% - Non defaulted and 38% defaulted company.

1.4 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*)

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y_com, test_size =0.33, random_state=42)
```

```
X_train.shape
(1378, 28)
y_train.shape
(1378,)
X_test.shape
(680, 28)
y_test.shape
(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

LOGISTIC REGRESSION MODEL :

```
import statsmodels.formula.api as SM
```

```
model_1=SM.logit(formula=f1, data=test_XY).fit()
```

```
Warning: Maximum number of iterations has been exceeded.
         Current function value: 0.206294
         Iterations: 35
```

- We observe many features having p-value > 0.05.
- These features must be removed as they are not statistically significant in the current case.
- Removing the features, one at a time (starting with highest p-value) and checking the result after each step and finally we got 7 important variables.

Logit Regression Results

Dep. Variable:	Default	No. Observations:	680
Model:	Logit	Df Residuals:	672
Method:	MLE	Df Model:	7
Date:	Sun, 24 Sep 2023	Pseudo R-squ.:	0.3444
Time:	13:47:50	Log-Likelihood:	-143.47
converged:	True	LL-Null:	-218.85
Covariance Type:	nonrobust	LLR p-value:	2.822e-29

	coef	std err	z	P>|z|	[0.025	0.975]
Intercept	-3.6768	0.299	-12.316	0.000	-4.262	-3.092
_Accounts_Receivable_Turnover	-0.5488	0.191	-2.880	0.004	-0.922	-0.175
_Net_Worth_Turnover_Rate_times	0.3104	0.142	2.190	0.029	0.033	0.588
_Cash_to_Total_Assets	-0.5902	0.222	-2.653	0.008	-1.026	-0.154
_Total_income_to_Total_expense	-1.7976	0.238	-7.541	0.000	-2.265	-1.330
_Total_expense_to_Assets	0.4403	0.158	2.791	0.005	0.131	0.750
_Cash_Turnover_Rate	-0.4657	0.197	-2.363	0.018	-0.852	-0.080
_Total_assets_to_GNP_price	0.4888	0.163	2.992	0.003	0.169	0.809

```
print(metrics.classification_report(train_XY['Default'],y_pred, digits=2))
```

```
              precision    recall  f1-score   support

           0       0.91      0.98      0.95      1225
           1       0.61      0.27      0.37       153

    accuracy                           0.90      1378
   macro avg       0.76      0.62      0.66      1378
weighted avg       0.88      0.90      0.88      1378
```

To improve RECALL We use optimization technique.

```
fpr,tpr,thresholds = roc_curve(train_XY['Default'],y_prob_pred_train)
```

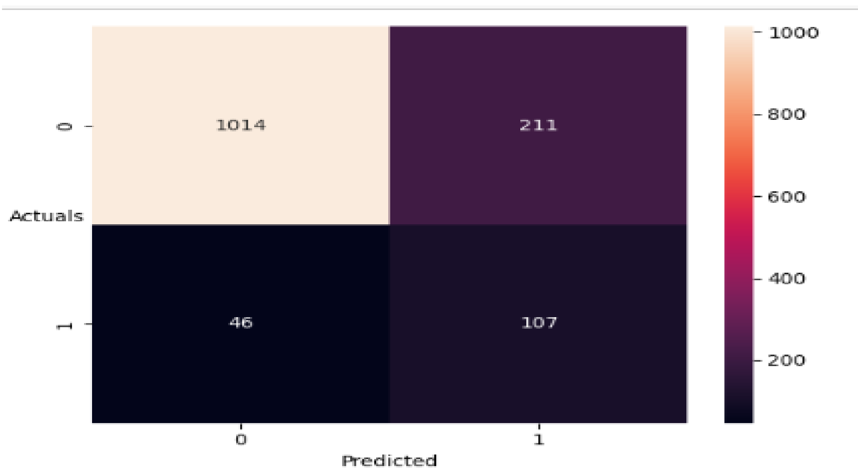
```
optimal_idx = np.argmax(tpr-fpr)
optimal_threshold = thresholds[optimal_idx]
optimal_threshold.round(5)
```

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

PERFORMANCE MATRIX :

TRAIN DATA SET :

CONFUSION MATRIX :

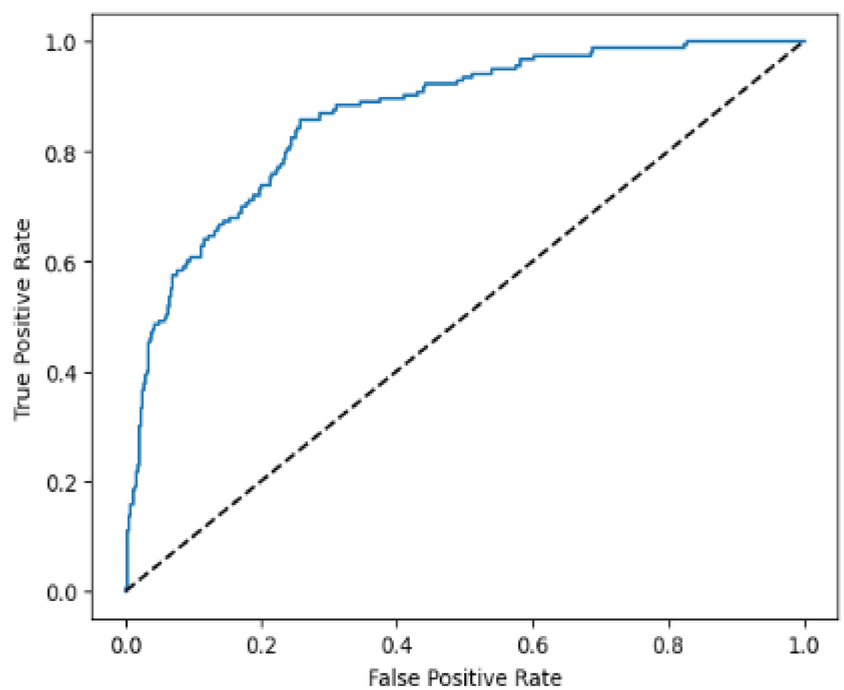


CLASSIFICATION REPORT :

	precision	recall	f1-score	support
0	0.96	0.83	0.89	1225
1	0.34	0.70	0.45	153
accuracy			0.81	1378
macro avg	0.65	0.76	0.67	1378
weighted avg	0.89	0.81	0.84	1378

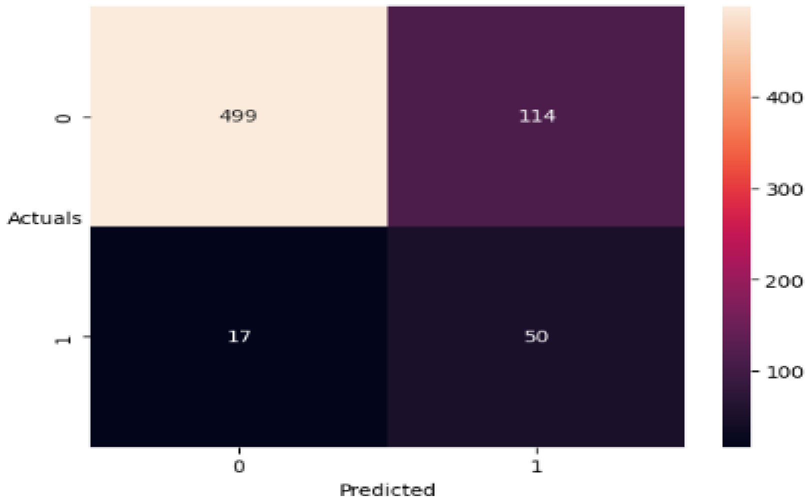
ROC – AUC CURVE :

ROC - AUC SCORE FOR TRAIN DATA : 0.8628304655195411



TEST DATASET :

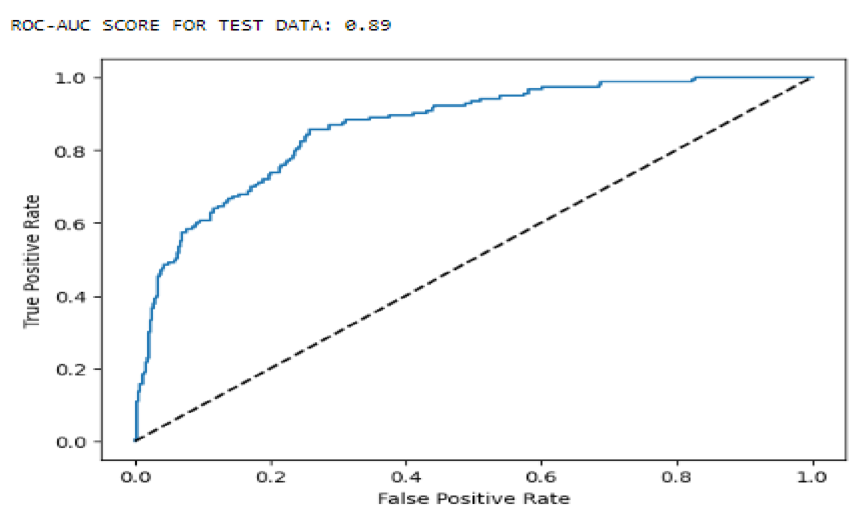
CONFUSION MATRIX :



CLASSIFICATION REPORT :

	precision	recall	f1-score	support
0	0.97	0.81	0.88	613
1	0.30	0.75	0.43	67
accuracy			0.81	680
macro avg	0.64	0.78	0.66	680
weighted avg	0.90	0.81	0.84	680

ROC-AUC CURVE :



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

Random forest model has been performed using Grid search approach.

```python
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import GridSearchCV

param_grid = {
    'max_depth':[3,5,7],
    'min_samples_leaf':[5,10,15],
    'min_samples_split':[15,30,45],
    'n_estimators':[25,50]
}

rfcl=RandomForestClassifier()
grid_search= GridSearchCV(estimator=rfcl,param_grid=param_grid)
```

```python
grid_search.fit(X_train, y_train)
```

```
GridSearchCV
estimator: RandomForestClassifier
RandomForestClassifier
```

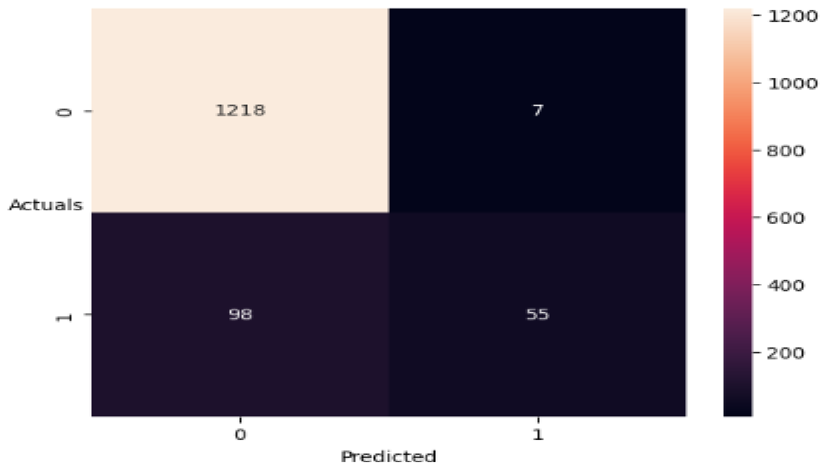
```python
grid_search.best_params_
```

```
{'max_depth': 7,
 'min_samples_leaf': 15,
 'min_samples_split': 15,
 'n_estimators': 25}
```

PERFORMANCE MATRIX :

TRAIN DATASET :

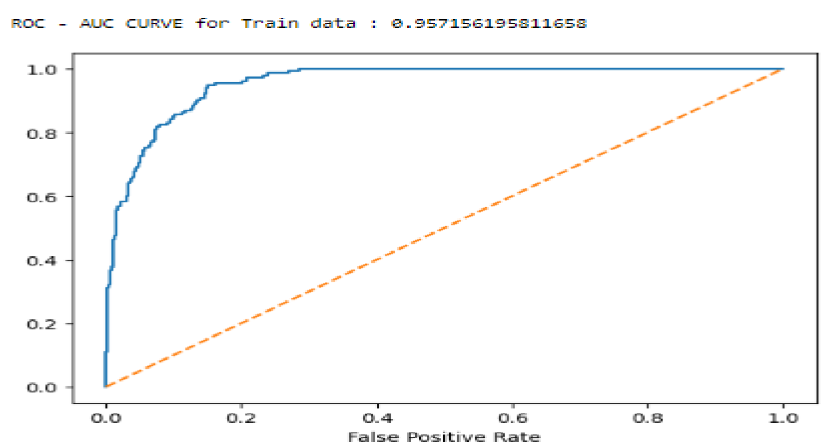
CONFUSION MATRIX :



CLASSIFICATION REPORT :

	precision	recall	f1-score	support
0	0.93	0.99	0.96	1225
1	0.89	0.36	0.51	153
accuracy			0.92	1378
macro avg	0.91	0.68	0.74	1378
weighted avg	0.92	0.92	0.91	1378

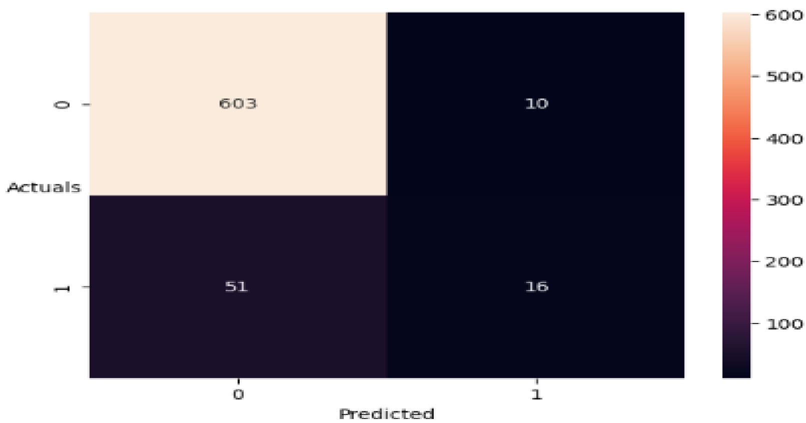
ROC – AUC CURVE :



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

TEST DATASET PERFORMANCE METRICS :

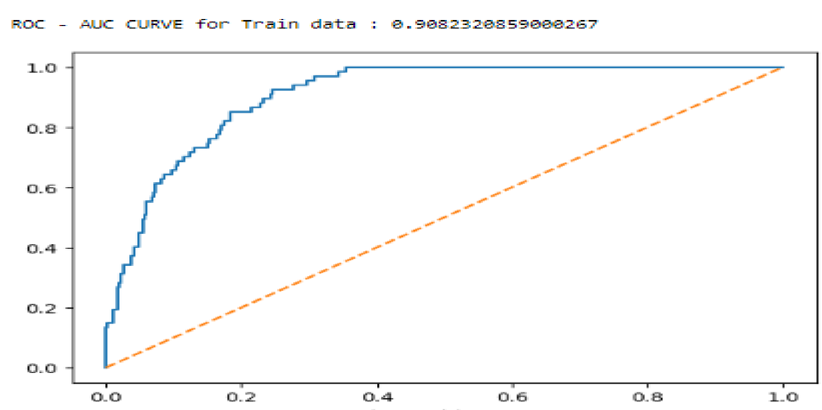
CONFUSION MATRIX :



CLASSIFICATION REPORT :

	precision	recall	f1-score	support
0	0.92	0.98	0.95	613
1	0.62	0.24	0.34	67
accuracy			0.91	680
macro avg	0.77	0.61	0.65	680
weighted avg	0.89	0.91	0.89	680

ROC – AUC CURVE :



- The accuracy score obtained from Random forest model on Test dataset is 91%.

- 62% precision is obtained for predicting those companies are Backrupt .

- ROC AUC score obtained for test dataset is 0.9082.

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

LDA MODEL :

```
LDA = LinearDiscriminantAnalysis()
LDA
```

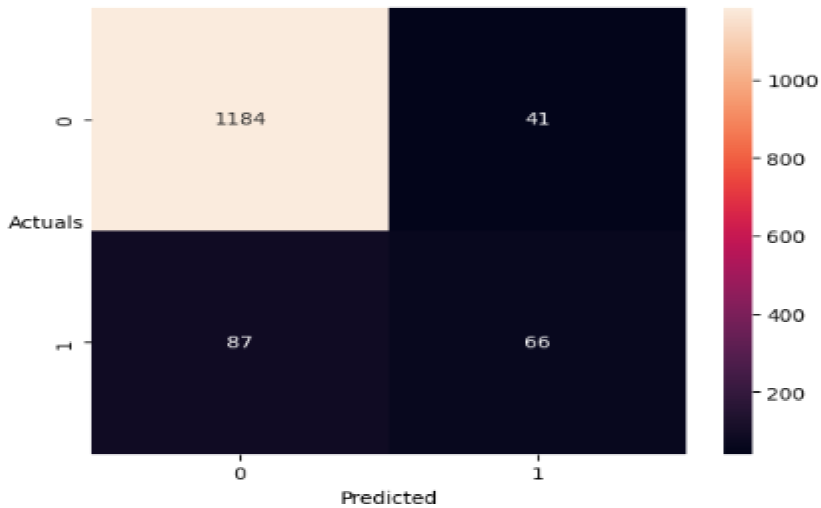
```
▾ LinearDiscriminantAnalysis
LinearDiscriminantAnalysis()
```

```
lda=LDA.fit(X_train,y_train)
```

```
pred_train_lda = lda.predict(X_train)
pred_test_lda = lda.predict(X_test)
```

TRAIN DATASET :

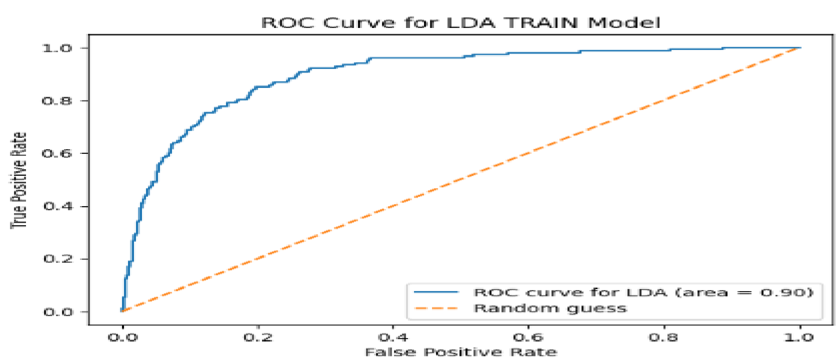
CONFUSION MATRIX :



CLASSIFICATION REPORT :

	precision	recall	f1-score	support
0	0.93	0.97	0.95	1225
1	0.62	0.43	0.51	153
accuracy			0.91	1378
macro avg	0.77	0.70	0.73	1378
weighted avg	0.90	0.91	0.90	1378

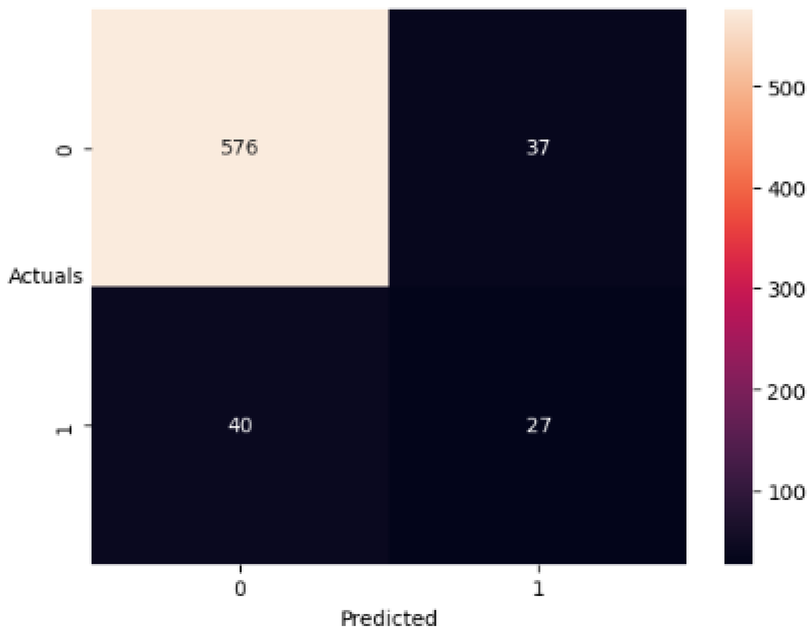
ROC AUC CURVE :



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

TEST DATASET :

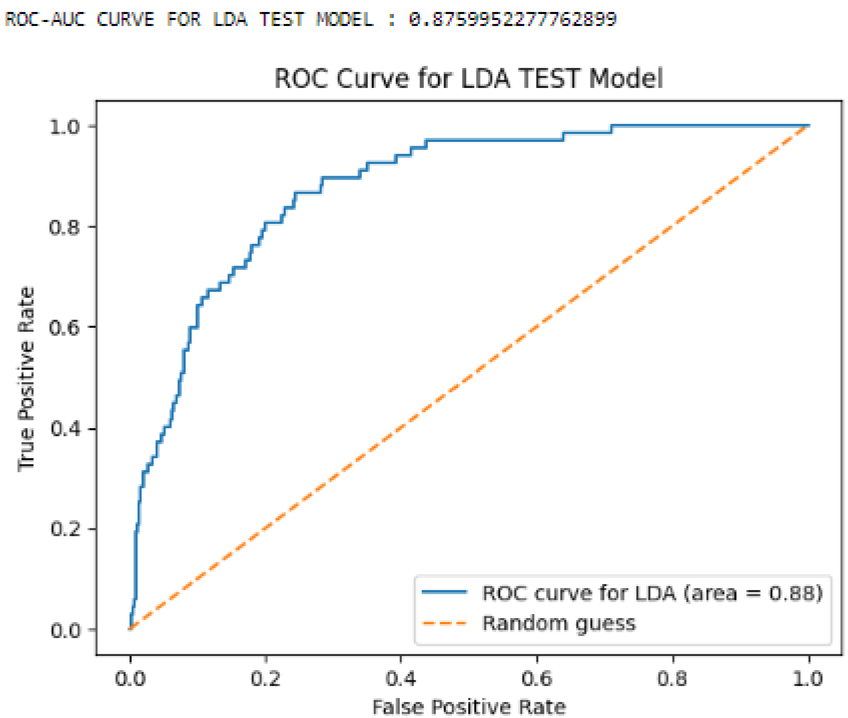
CONFUSION MATRICS :



CLASSIFICATION REPORT :

	precision	recall	f1-score	support
0	0.94	0.94	0.94	613
1	0.42	0.40	0.41	67
accuracy			0.89	680
macro avg	0.68	0.67	0.67	680
weighted avg	0.88	0.89	0.89	680

ROC AUC CURVE :



- The accuracy score for Test dataset is 89%
- It shows 94 % precision while predicting Non bankrupt company.
- The ROC AUC score will be 0.8759.

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

- The **random forest classifier (rfcl)** outperforms Linear Discriminant Analysis (LDA) in several metrics, achieving higher accuracy, precision, and ROC AUC on both training and test data.
- However, **Logistic Regression** shows better performance in identifying potential defaulters based on higher test recall, making it more effective at correctly classifying companies at risk of default (Hence preferred for current case)

	Model	Train Accuracy	Test Accuracy	Train Precision	Test Precision	Train Recall	Test Recall	ROC AUC
0	LOG	0.813498	0.807353	0.336478	0.304878	0.699346	0.746269	0.86283
1	rfcl	0.923803	0.910294	0.887097	0.615385	0.359477	0.238806	0.908232
2	LDA	0.907112	0.886765	0.616822	0.421875	0.431373	0.402985	0.875995

Conclusions :

- In the given dataset , only 220 companies are backrupted and 1351 companies are at risk

- From its financial statement, we know that defaulted company provide previously higher employer profit without earning which is obtained from Fixed asset allocation per employee variable.

- Defaulted company for the 2 years spend 5 % more than its asset.

- Defaulted company has only 5% of its asset as cash.

- Defaulted company has spend more for Research & Development when compared to Non Defaulted.

- Defaulted company spends 2% more than its income.

- Both the companies given 63% of its revenue as interest.

- Defaulted companies having excessive debt than its asset worth.

- Misallocation of resources leads to the company's backrupt.

- Running out of cash occurs in defaulted company.

RECOMMENDATIONS :

- Investors should focus on financial statement of a company while investing
- Investors focus on company which donot misallocate its resources.
- Investors prefer companies whose Total debt is lesser than its asset.
- Investors can invest in financial institutions because it is governed by central bank.
- Company can reduce its expenses to escape from banckruptancy.
- Company can focus on innovation to increase its revenue
- Company have to enhance its supply chain management to perform better in society leads to increase in CASH FLOW RATE.
- Companies can compete with other companies in the market to get better place in the market.

PART – B
MARKET RISK ANALYSIS

PROBLEM :

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

DATA DESCRIPTION :

- The given dataset contains 314 entries and 11 variables.
- It has 10 Numerical and 1 categorical variable.
- Presence of no duplicates in the given dataset
- Presence of No null values in the dataset

HEAD :

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

TAIL :

	Date	Infosys	Indian Hotel	Mahindra & Mahindra	Axis Bank	SAIL	Shree Cement	Sun Pharma	Jindal Steel	Idea Vodafone	Jet Airways
309	02-03-2020	729	120	469	658	33	23110	401	146	3	22
310	09-03-2020	634	114	427	569	30	21308	384	121	6	18
311	16-03-2020	577	90	321	428	27	18904	365	105	3	16
312	23-03-2020	644	75	293	360	21	17666	338	89	3	14
313	30-03-2020	633	75	284	379	23	17546	352	82	3	14

DATA INFO :

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

SHAPE :

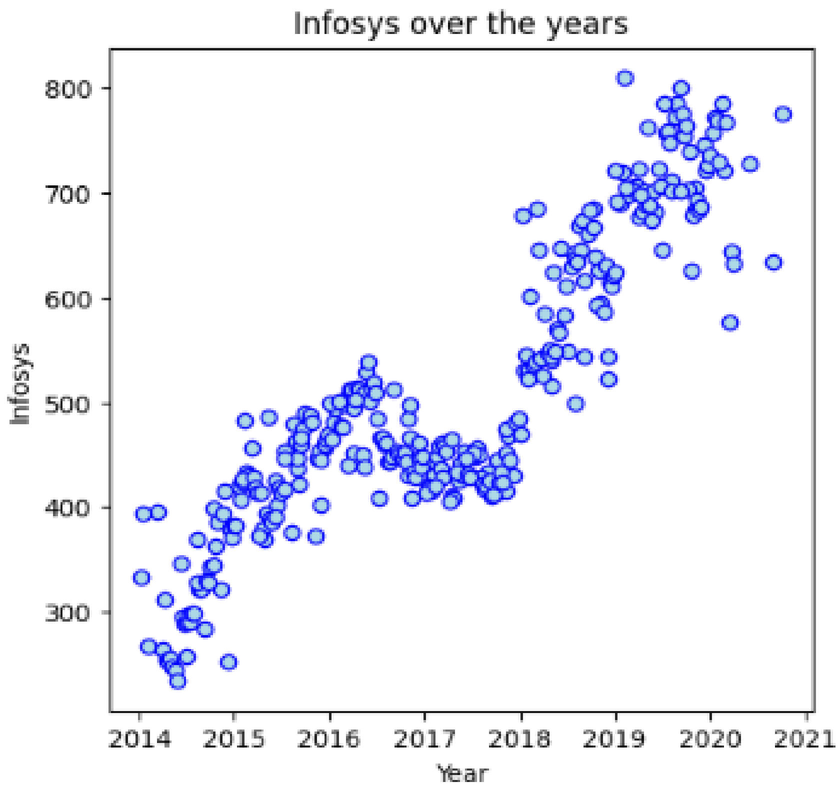
(314, 11)

DATA SUMMARY :

	Infosys	Indian Hotel	Mahindra & Mahindra	Axis Bank	SAIL	Shree Cement	Sun Pharma	Jindal Steel	Idea Vodafone	Jet Airways
count	314.00	314.00	314.00	314.00	314.00	314.00	314.00	314.00	314.00	314.00
mean	511.34	114.56	636.68	540.74	59.10	14806.41	633.47	147.63	53.71	372.66
std	135.95	22.51	102.88	115.84	15.81	4288.28	171.86	65.88	31.25	202.26
min	234.00	64.00	284.00	263.00	21.00	5543.00	338.00	53.00	3.00	14.00
25%	424.00	96.00	572.00	470.50	47.00	10952.25	478.50	88.25	25.25	243.25
50%	466.50	115.00	625.00	528.00	57.00	16018.50	614.00	142.50	53.00	376.00
75%	630.75	134.00	678.00	605.25	71.75	17773.25	785.00	182.75	82.00	534.00
max	810.00	157.00	956.00	808.00	104.00	24806.00	1089.00	338.00	117.00	871.00

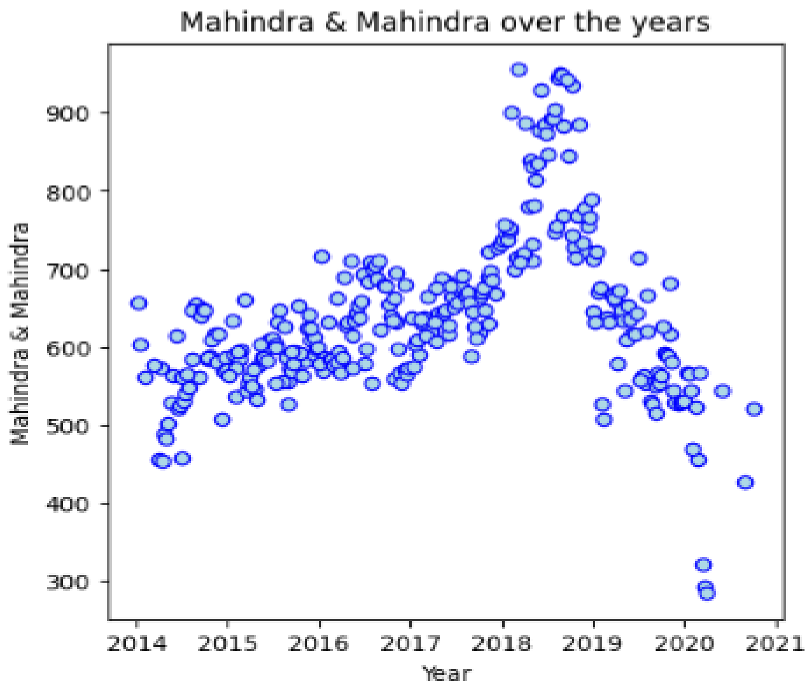
Draw Stock Price Graph(Stock Price vs Time) for any 2 given stocks with inference

INFOSYS :



- The stock price of Infosys is increasing gradually from 2014 .
- Its base price of each stock increases from 300 to 800.
- It faces some decrease in price from 2016 to 2017 during that time investors invest more.

MAHINDRA & MAHINDRA :



- The stock price of Mahindra & Mahindra increase slowly from 2016 to 2018
- It started falling from 2018 to 2021 because of covid pandemic.

Calculate Returns for all stocks with inference :

```
stock_returns = np.log(sk.drop(['Date','date'],axis=1)).diff(axis = 0)
```

	Infosys	Indian Hotel	Mahindra & Mahindra	Axis Bank	SAIL	Shree Cement	Sun Pharma	Jindal Steel	Idea Vodafone	Jet Airways
0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1	-0.03	-0.01	0.01	0.05	0.03	0.03	0.09	-0.07	0.01	0.09
2	-0.01	0.00	-0.01	-0.02	-0.03	-0.01	-0.00	0.00	-0.01	-0.08
3	-0.00	0.00	0.07	0.05	0.00	0.01	-0.00	-0.02	0.00	0.01
4	0.01	-0.05	-0.01	-0.00	-0.08	-0.02	0.01	-0.14	-0.05	-0.15

Idea Vodafone has lowest return whereas Shree cement has higher return.

Calculate Stock Means and Standard Deviation for all stocks with inference

```
stock_means = stock_returns.mean(axis = 0)
stock_means
```

```
Infosys               0.00
Indian Hotel          0.00
Mahindra & Mahindra  -0.00
Axis Bank             0.00
SAIL                 -0.00
Shree Cement          0.00
Sun Pharma           -0.00
Jindal Steel         -0.00
Idea Vodafone        -0.01
Jet Airways          -0.01
dtype: float64
```

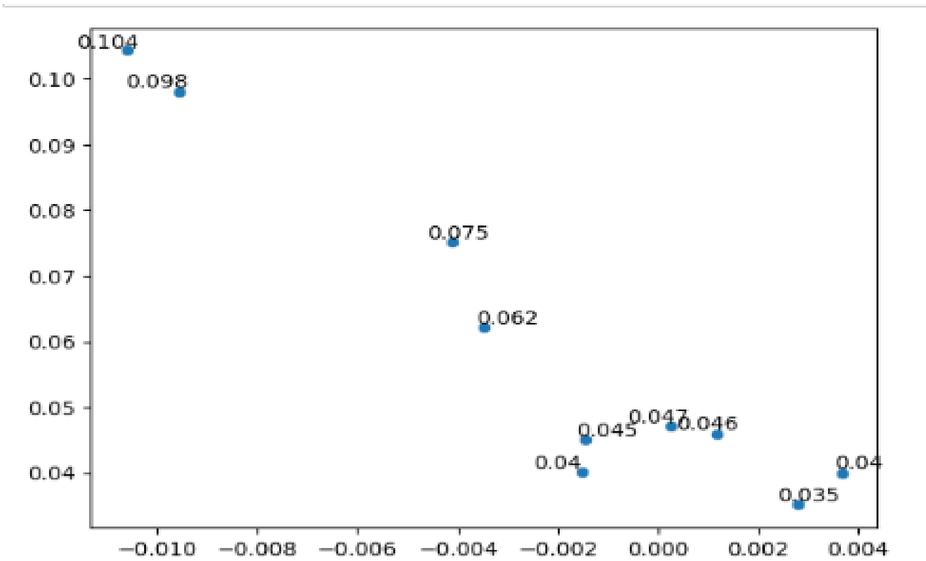
```
stock_sd = stock_returns.std(axis = 0)
stock_sd
```

```
Infosys               0.04
Indian Hotel          0.05
Mahindra & Mahindra   0.04
Axis Bank             0.05
SAIL                  0.06
Shree Cement          0.04
Sun Pharma            0.05
Jindal Steel          0.08
Idea Vodafone         0.10
Jet Airways           0.10
dtype: float64
```

	Average	Volatility
Infosys	0.00	0.04
Indian Hotel	0.00	0.05
Mahindra & Mahindra	-0.00	0.04
Axis Bank	0.00	0.05
SAIL	-0.00	0.06
Shree Cement	0.00	0.04
Sun Pharma	-0.00	0.05
Jindal Steel	-0.00	0.08
Idea Vodafone	-0.01	0.10
Jet Airways	-0.01	0.10

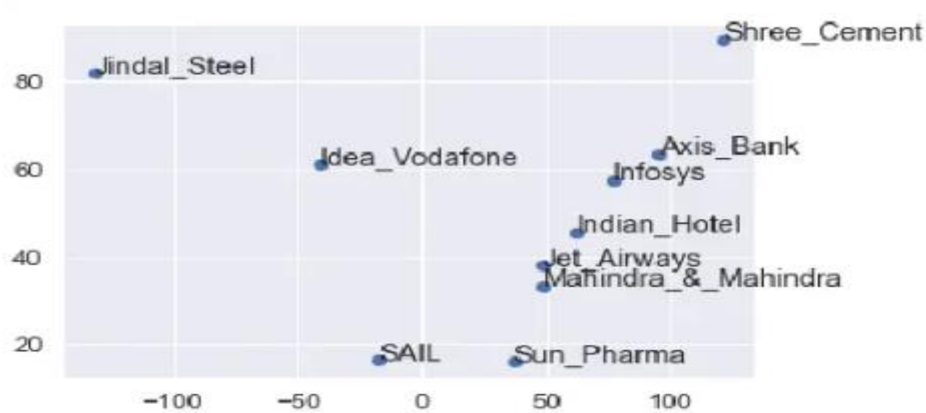
Jet Airways has highest risk for investing while Infosys has least.

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



- Stocks are higher up but on the far left indicate high volatility and low returns , while stocks on the bottom indicate low volatility and high returns.

- This graph is usefull to find balance between risk and reward when investing in different companies.



CONCLUSIONS:

- Jindal steel , idea Vodafone provide high volality and low returns.
- SAIL and Sun pharma provide low volatility and high return
- Jet Airways has highest risk for investing while Infosys has least.

RECOMMENDATIONS :

- Assess the market. Before you add a position, note how the broader market is moving, since research suggests that roughly 75% of stocks move in step with the market.
- Identify a sector.
- Screen for stocks.
- The best time to buy stocks is when the share prices of a given stock are at a low. There is always a chance that they will drop even further, but buying at a low price is significantly safer than buying at a high price where the price of the stock is unlikely to climb much higher.
- Invest in volatile stock is profitable but its not without risk.
- Set up stop-loss orders to limit potential loss.

THANK YOU