|  |  |
| --- | --- |
|  |  |
| FINANCE AND RISK ANALYTICS PROJECT |  |
|  |  |
|  | Submitted by,VIDYA V |
|  | PGPDSBA.O.2023.B 28.01.2024 |

**CONTENTS**

|  |  |
| --- | --- |
| Case 1: Credit Risk Analysis | 4 |
| 1. Outlier Treatment | 4 |
| 1. Missing Value Treatment | 10 |
| 1. 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) | 12 |
| 1. Train Test Split | 39 |
| 1. 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 | 40 |
| 1. Validate the Model on Test Dataset and state the performance metrics. Also state interpretation from the model | 42 |
| 1. Build a Random Forest Model on Train Dataset. Also showcase your model building approach | 44 |
| 1. Validate the Random Forest Model on test Dataset and state the performance metrics. Also state interpretation from the model | 45 |
| 1. Build an LDA Model on Train Dataset. Also showcase your model building approach | 46 |
| 1. Validate the LDA Model on test Dataset and state the performance metrics. Also state interpretation from the model | 47 |
| 1. Compare the performances of Logistic Regression, Random Forest, and LDA models (include ROC curve) | 49 |
| 1. Conclusions and Recommendations |  |
| Case 2: Market Risk Analysis | 51 |
| 1. Draw Stock Price Graph (Stock Price vs Time) for any 2 given stocks with inference | 51 |
| 1. Calculate Returns for all stocks with inference | 56 |
| 1. Calculate Stock Means and Standard Deviation for all stocks with inference | 56 |
| 1. Draw a plot of Stock Means vs Standard Deviation and state your inference | 57 |
| 1. Conclusions and Recommendations | 57 |
|  |  |

**List of Figures**

|  |  |
| --- | --- |
| **Name** | **Page No.** |
| Fig.1.1. Dataset Description before outlier treatment | 5 |
| Fig.1.2. Boxplot of numerical variables before outlier treatment | 6 |
| Fig.1.3. Number of outliers in each field | 7 |
| Fig.1.4. Outlier percentage in each field | 8 |
| Fig.1.5. Boxplot of numerical variables after outlier treatment | 9 |
| Fig.1.6. Description of dataset after outlier treatment | 10 |
| Fig.1.7. Number of missing values in each field in the dataset | 11 |
| Fig.1.8. Univariate Analysis of Numerical Variables | 12-18 |
| Fig.1.9. Univariate Analysis- Categorical Variables | 19 |
| Fig.1.10 Bivariate analysis- Heatmap of numerical variables | 20 |
| Fig.1.11. Bivariate Analysis- Numerical Variables vs Target Variables | 21-37 |
| Fig.1.12. Heatmap of correlation between numerical variables after eliminating variables causing high multicollinearity using VIF | 39 |
| Fig.1.13. Logistic Regression model summary | 41 |
| Fig.1.14. Logistic Regression- Train data performance | 42 |
| Fig.1.15. Logistic Regression- Confusion matrix- Train Data | 42 |
| Fig.1.16. Logistic Regression Model- Test Data Performance | 43 |
| Fig.1.17. Logistic Regression- Confusion Matrix- Test Data | 43 |
| Fig.1.18. Random Forest model- Train Data performance | 44 |
| Fig.1.19. Random Forest model- Confusion Matrix- Train Data | 45 |
| Fig.1.20. Random Forest model- Test data performance | 45 |
| Fig.1.21. Random Forest model- Confusion matrix- test data | 46 |
| Fig.1.22. LDA model- Train data performance | 47 |
| Fig.1.23. LDA model- Confusion Matrix- Train Data | 47 |
| Fig.1.24. LDA model- Test Data Performance | 48 |
| Fig.1.25. LDA model- Confusion Matrix- test data | 48 |
| Fig.2.1. Price Vs time plots for stocks | 51-55 |
| Fig.2.2. Market returns- Data Description | 56 |
| Fig.2.3. Stocks- Returns and Volatility (Std deviation) | 56 |
| Fig.2.4. Returns vs Volatility | 57 |

**List of Tables**

|  |  |
| --- | --- |
| **Name** | **Page No.** |
| Table 1.1. Test Data Performance Comparison | 49 |

# Case 1: Credit Risk Analysis

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. **Outlier Treatment.**

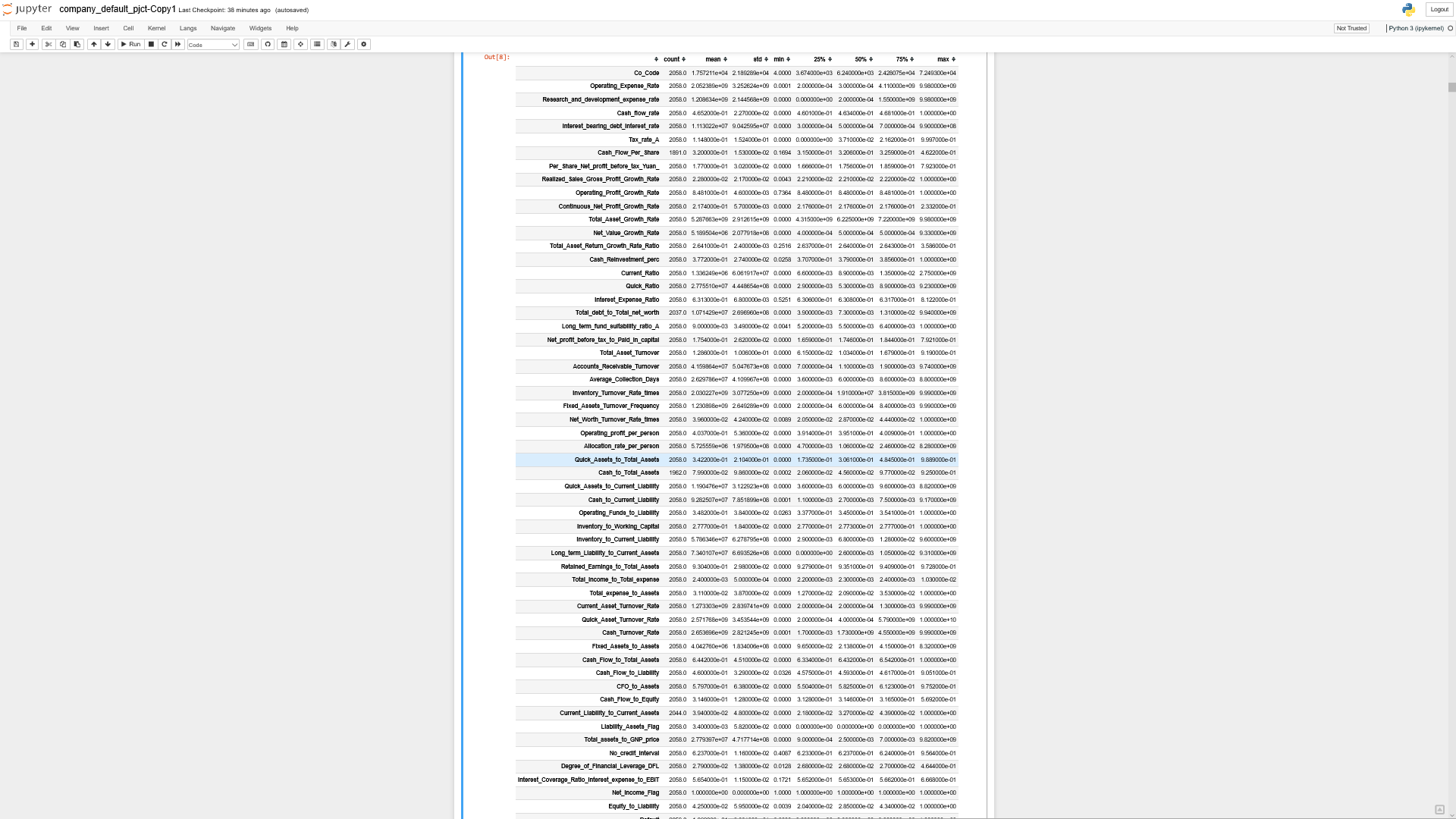


Fig.1.1. Dataset Description before outlier treatment

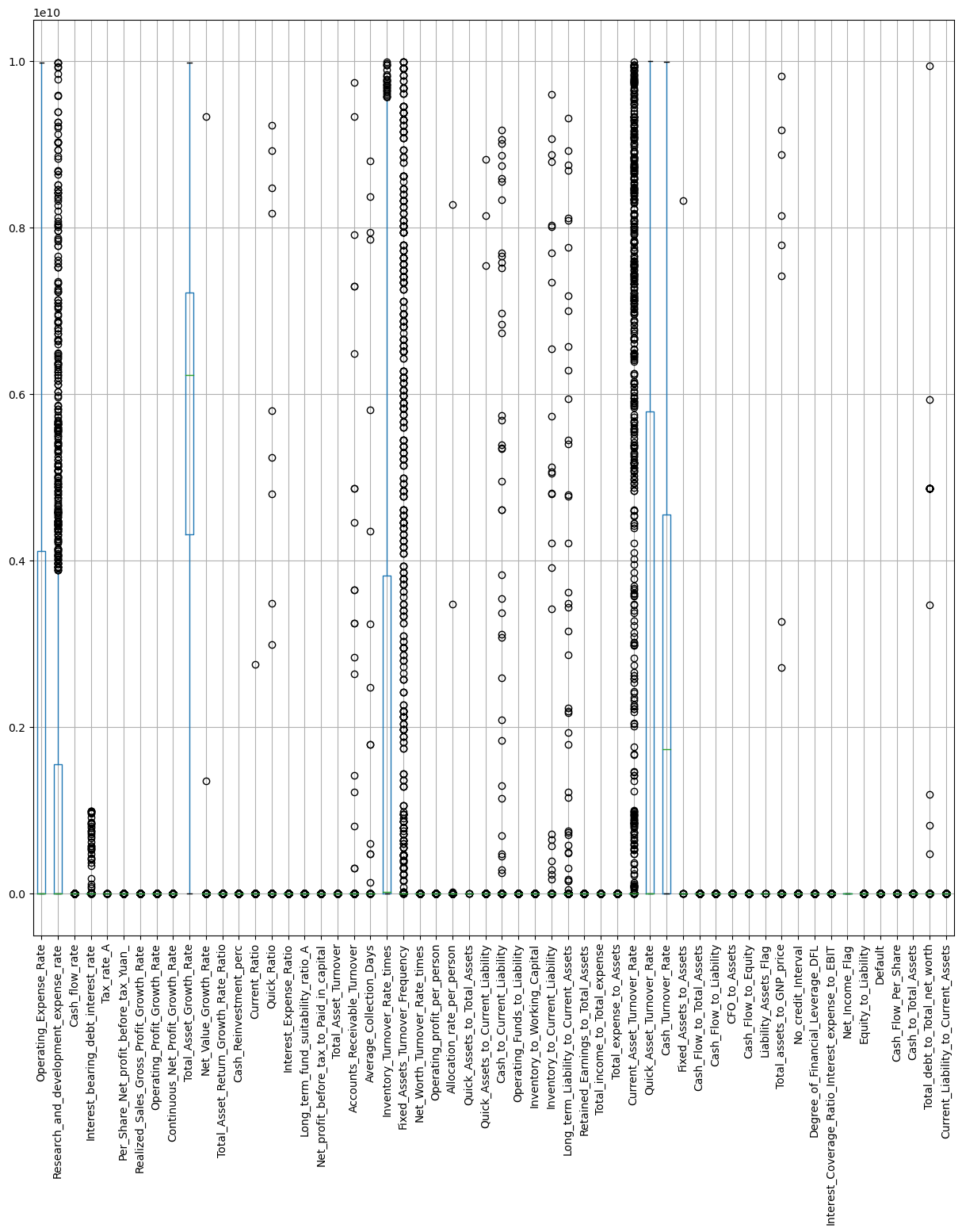


Fig.1.2. Boxplot of numerical variables before outlier treatment

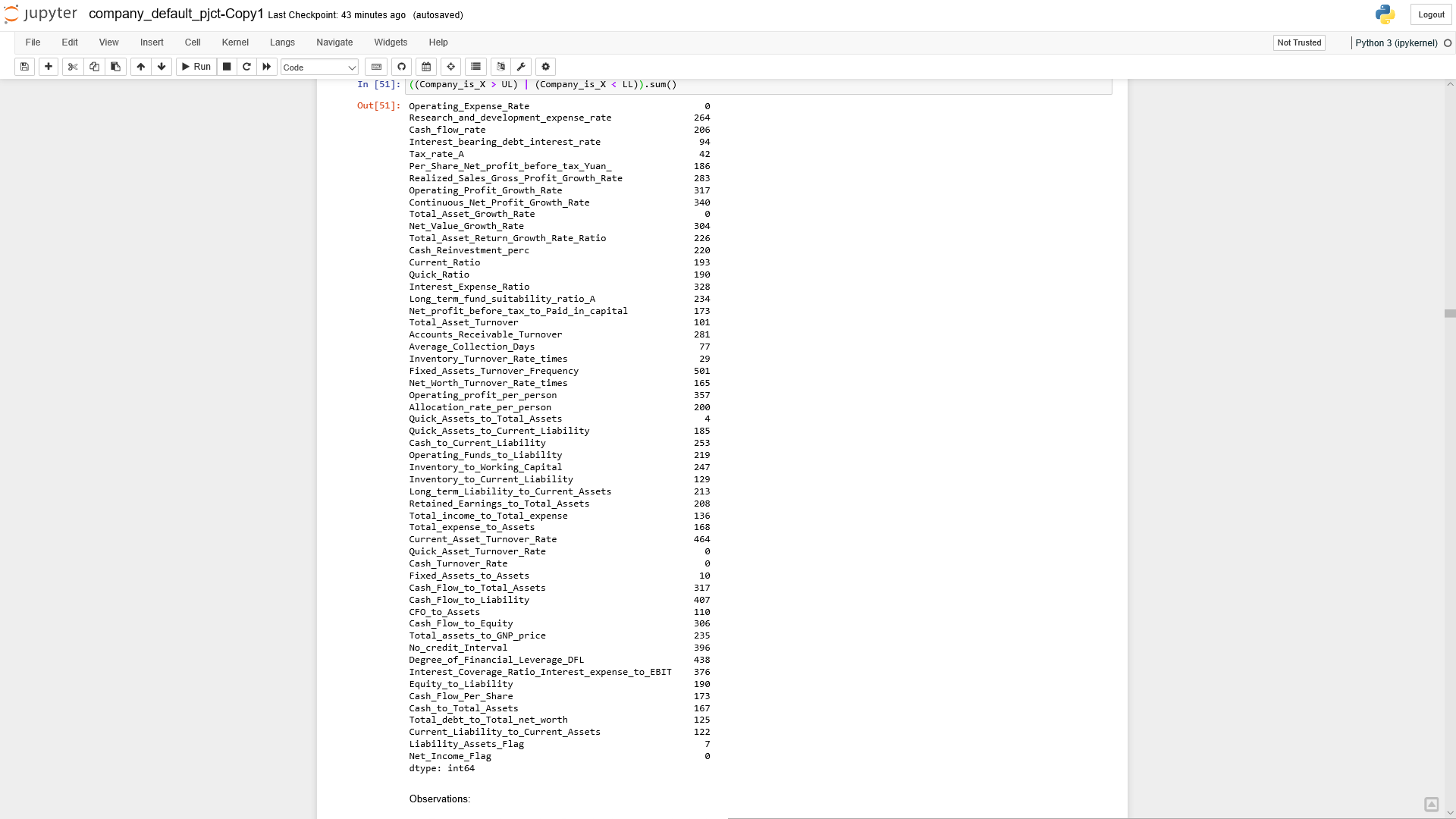


Fig.1.3. Number of outliers in each field

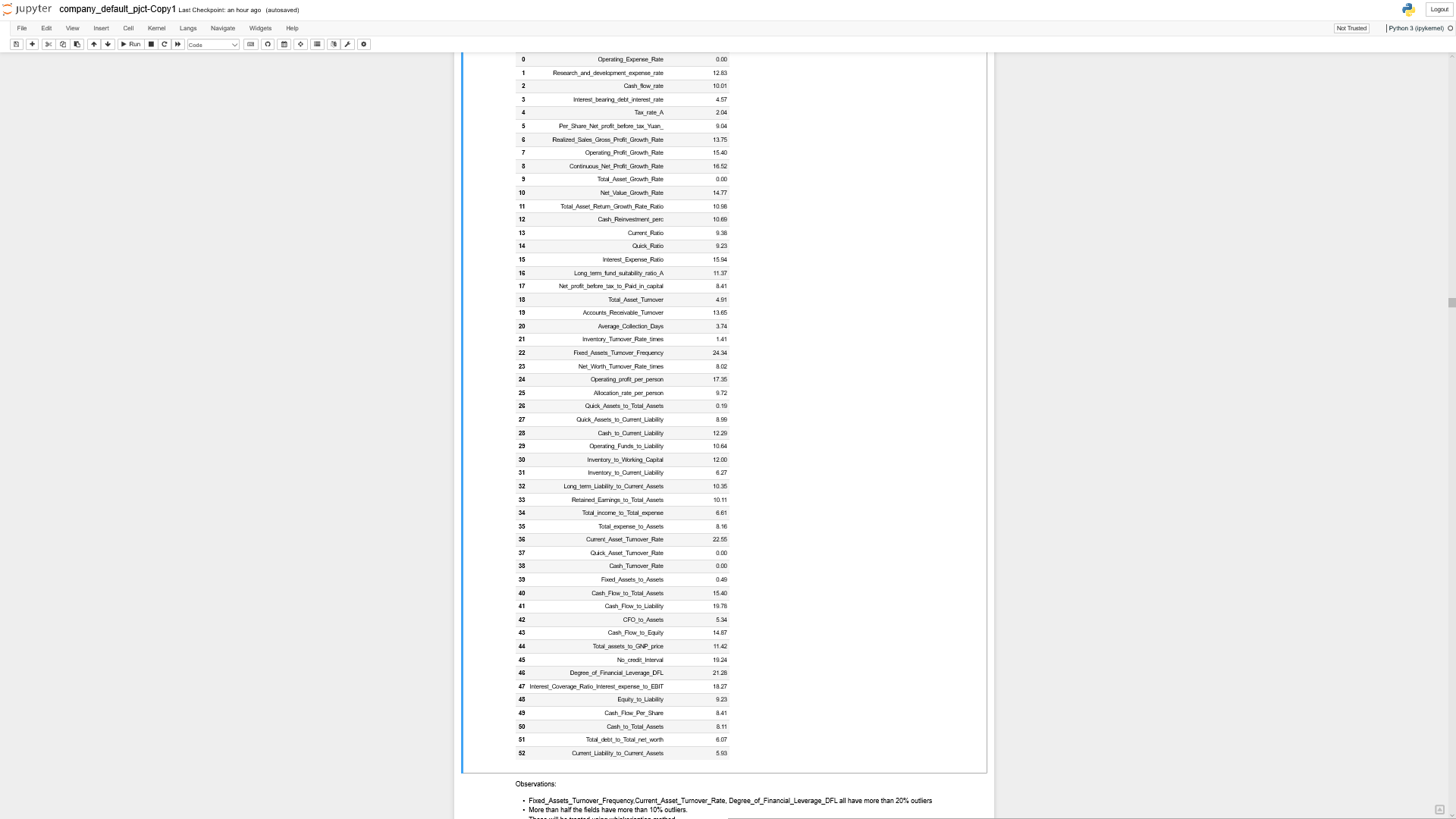


Fig.1.4. Outlier percentage in each field

Observations:

* From the output above, almost all the fields have outliers and need to be cleaned before further analysis
* Fixed\_Assets\_Turnover\_Frequency, Current\_Asset\_Turnover\_Rate, Degree\_of\_Financial\_Leverage\_DFL all have more than 20% outliers
* More than half the fields have more than 10% outliers.
* These are treated by capping the outliers at the lower or upper bounds, determined by the interquartile range
* Also, in case of multicollinearity and elimination of variables, the percentage of outliers also need to be considered while choosing the significant variables and eliminating insignificant variables

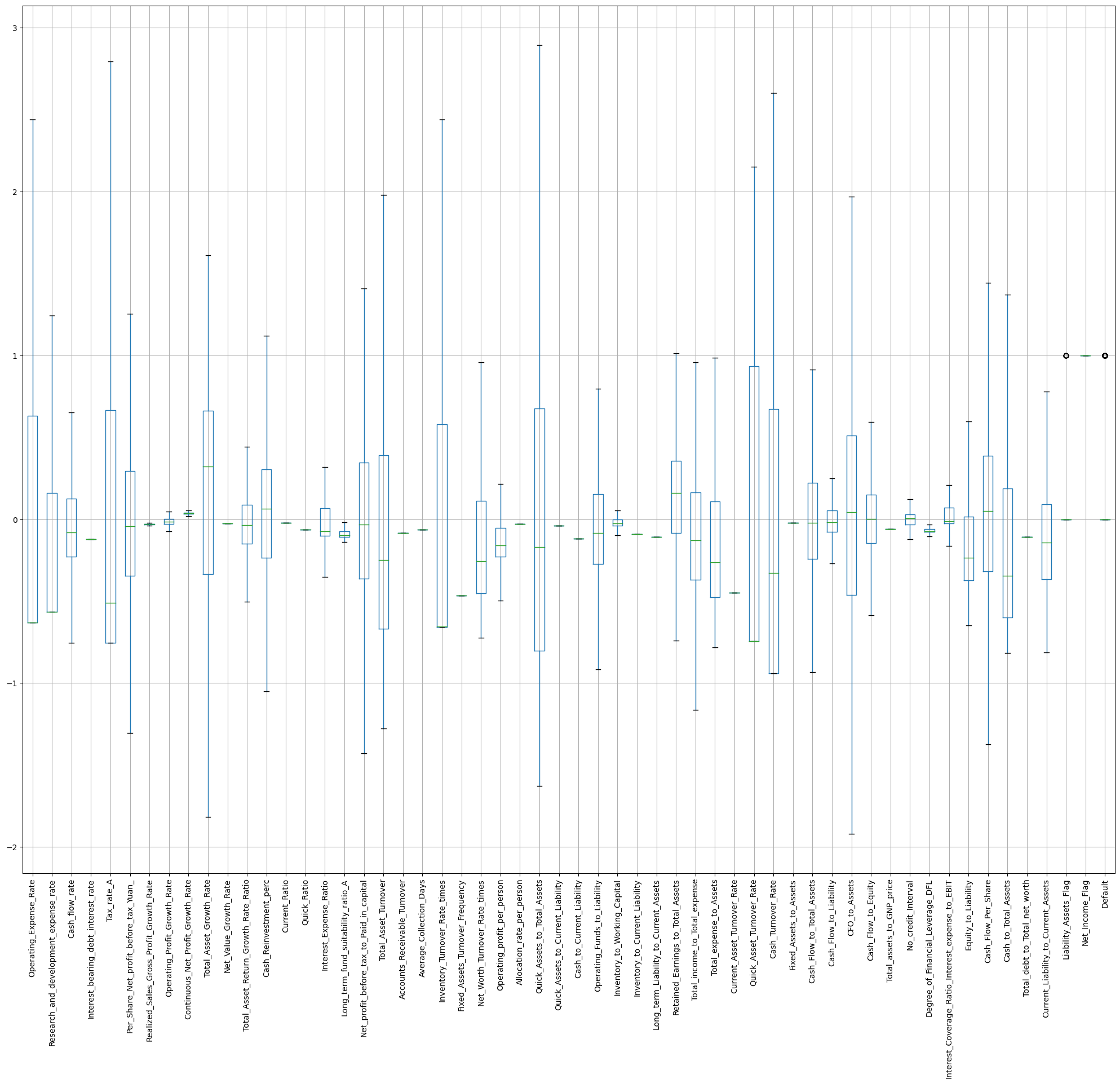


Fig.1.5. Boxplot of numerical variables after outlier treatment

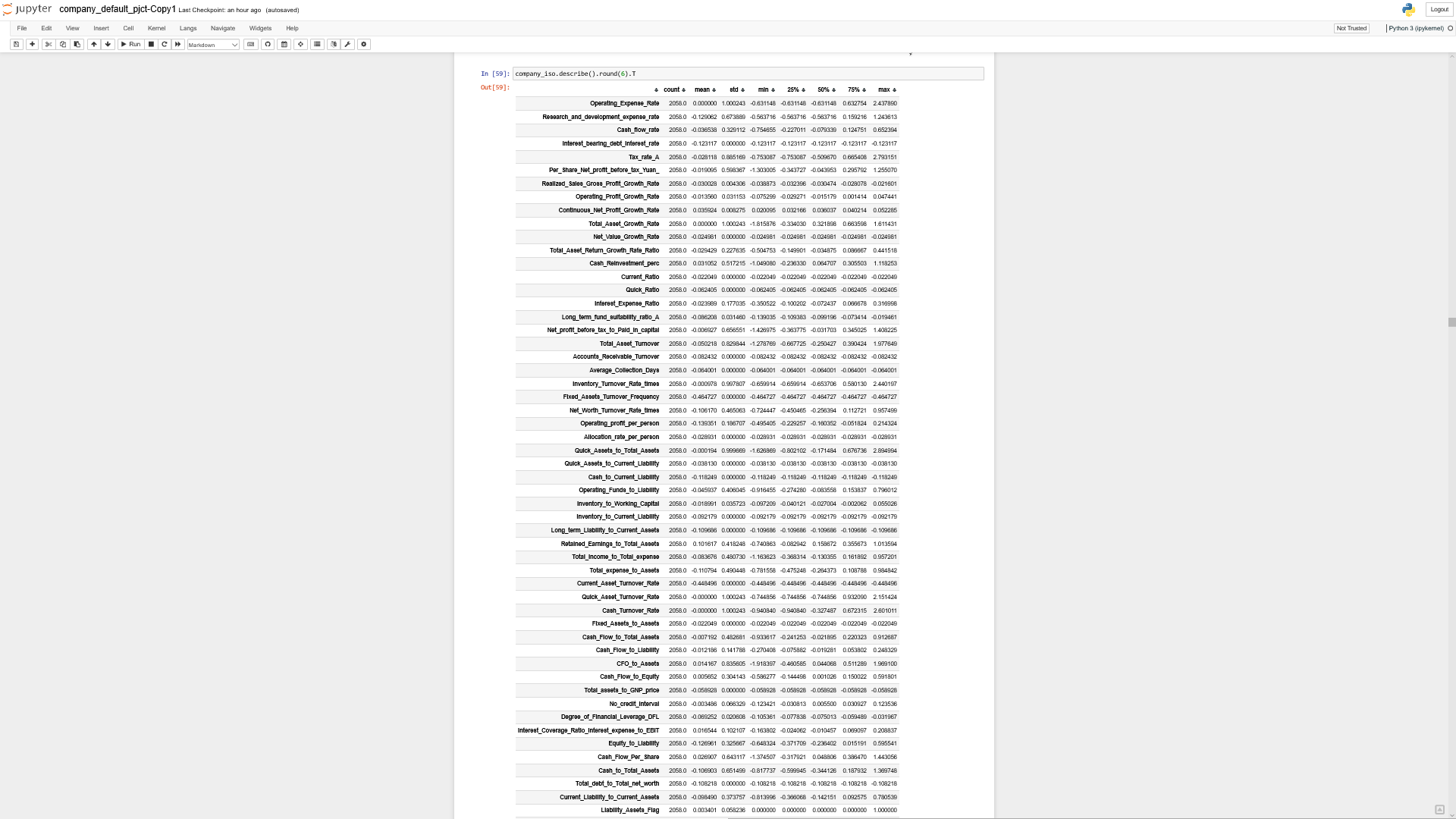


Fig.1.6. Description of dataset after outlier treatment

1. **Missing Value Treatment**

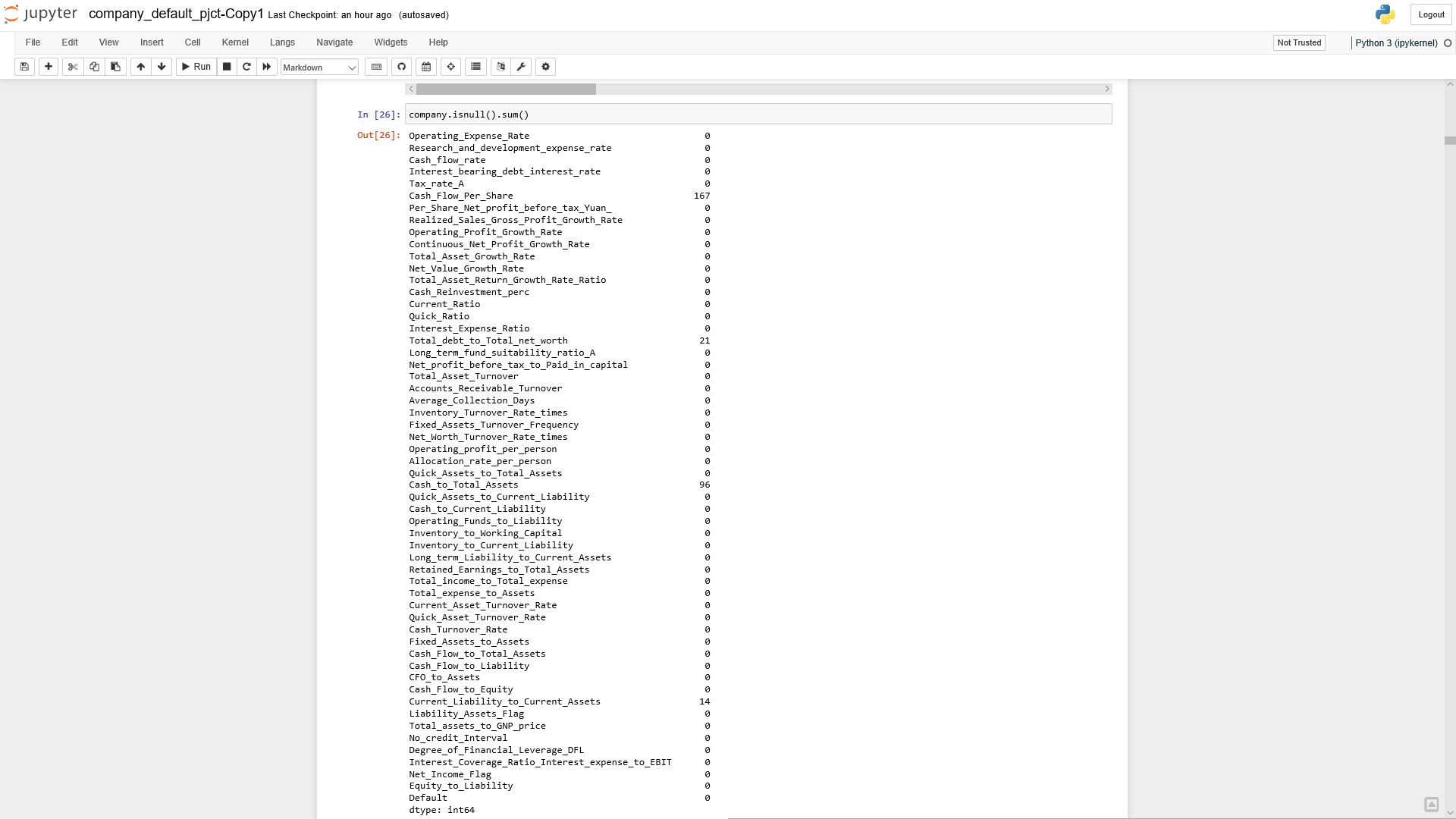


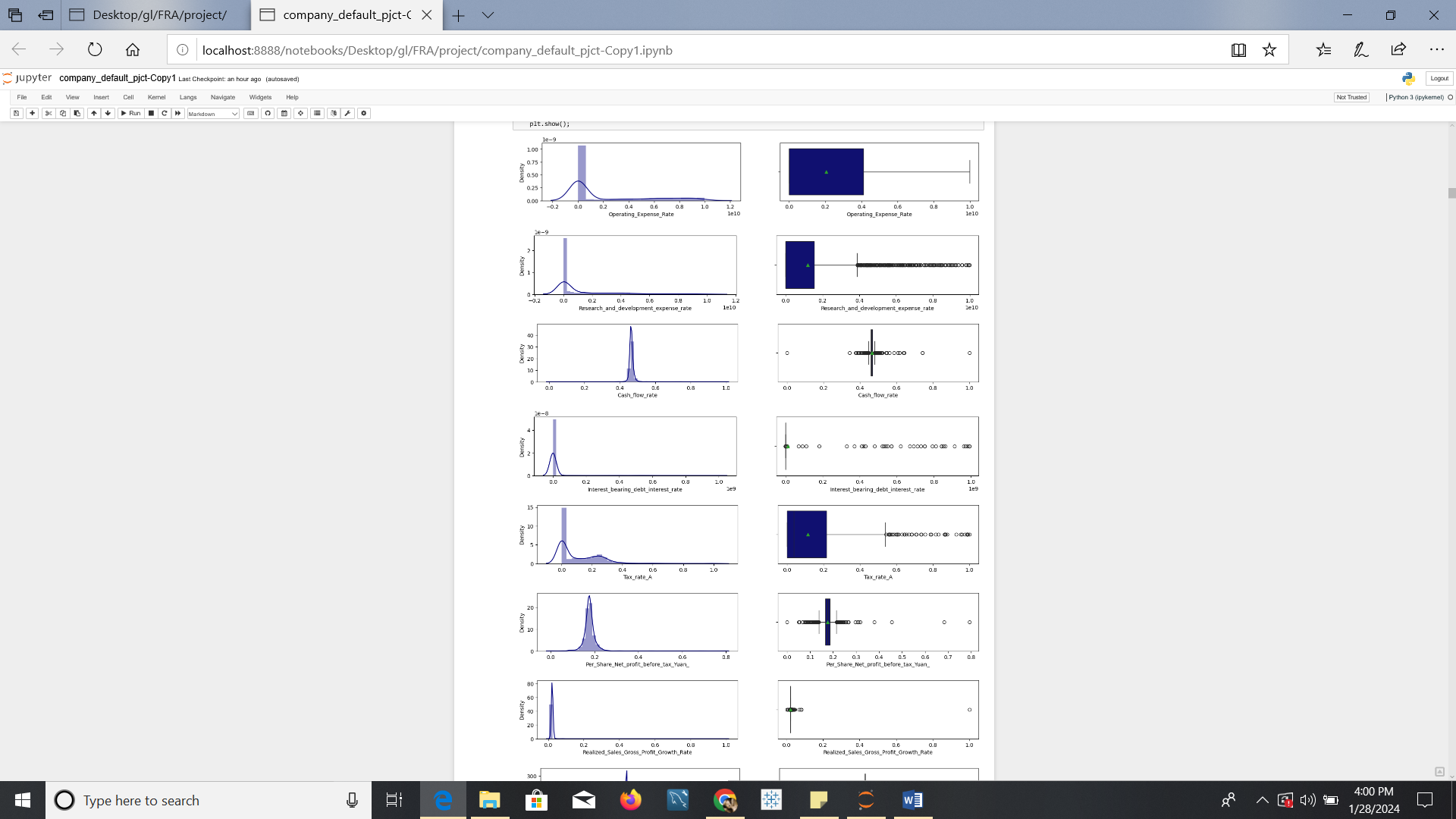
Fig.1.7. Number of missing values in each field in the dataset

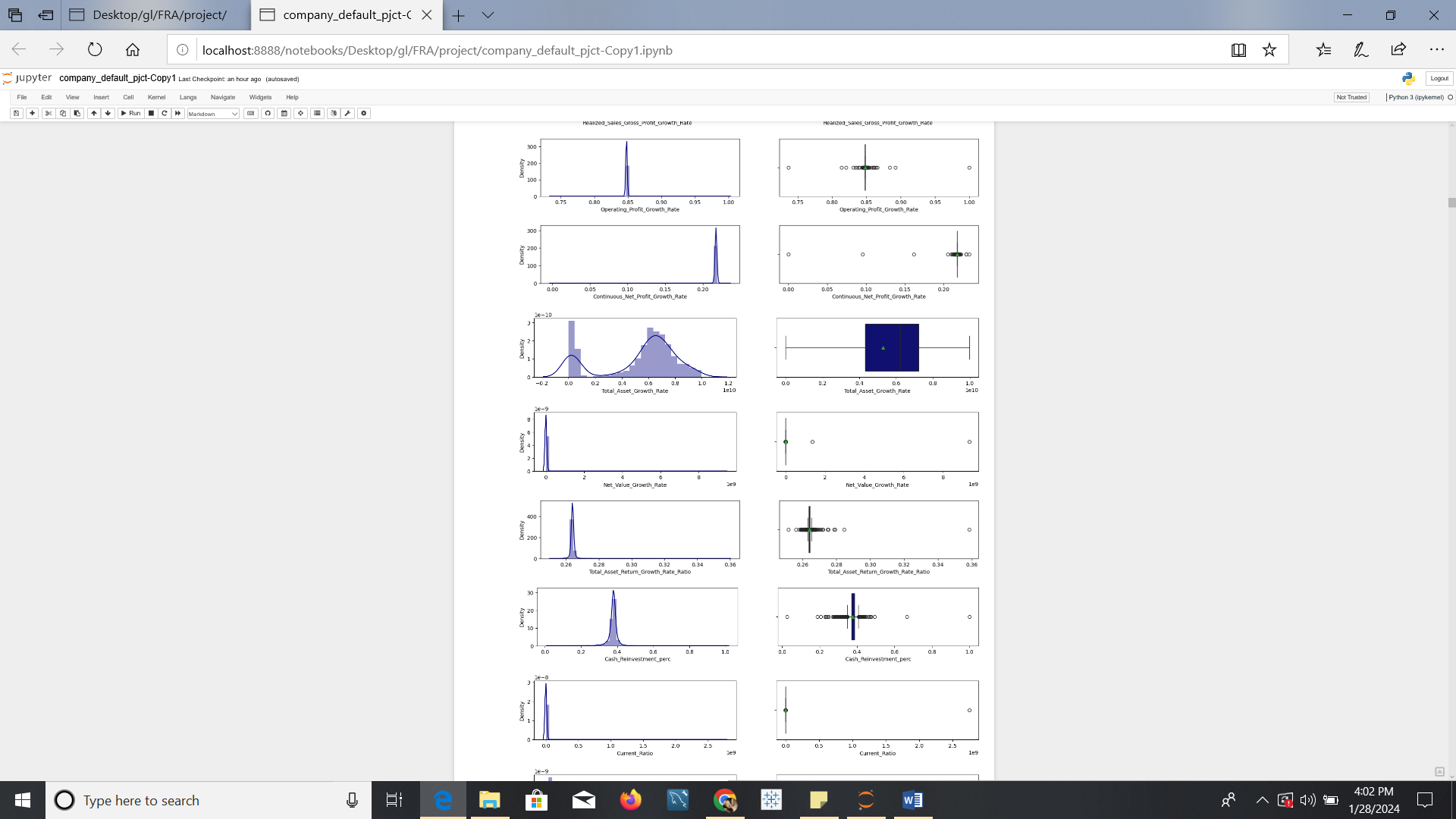
Observations:

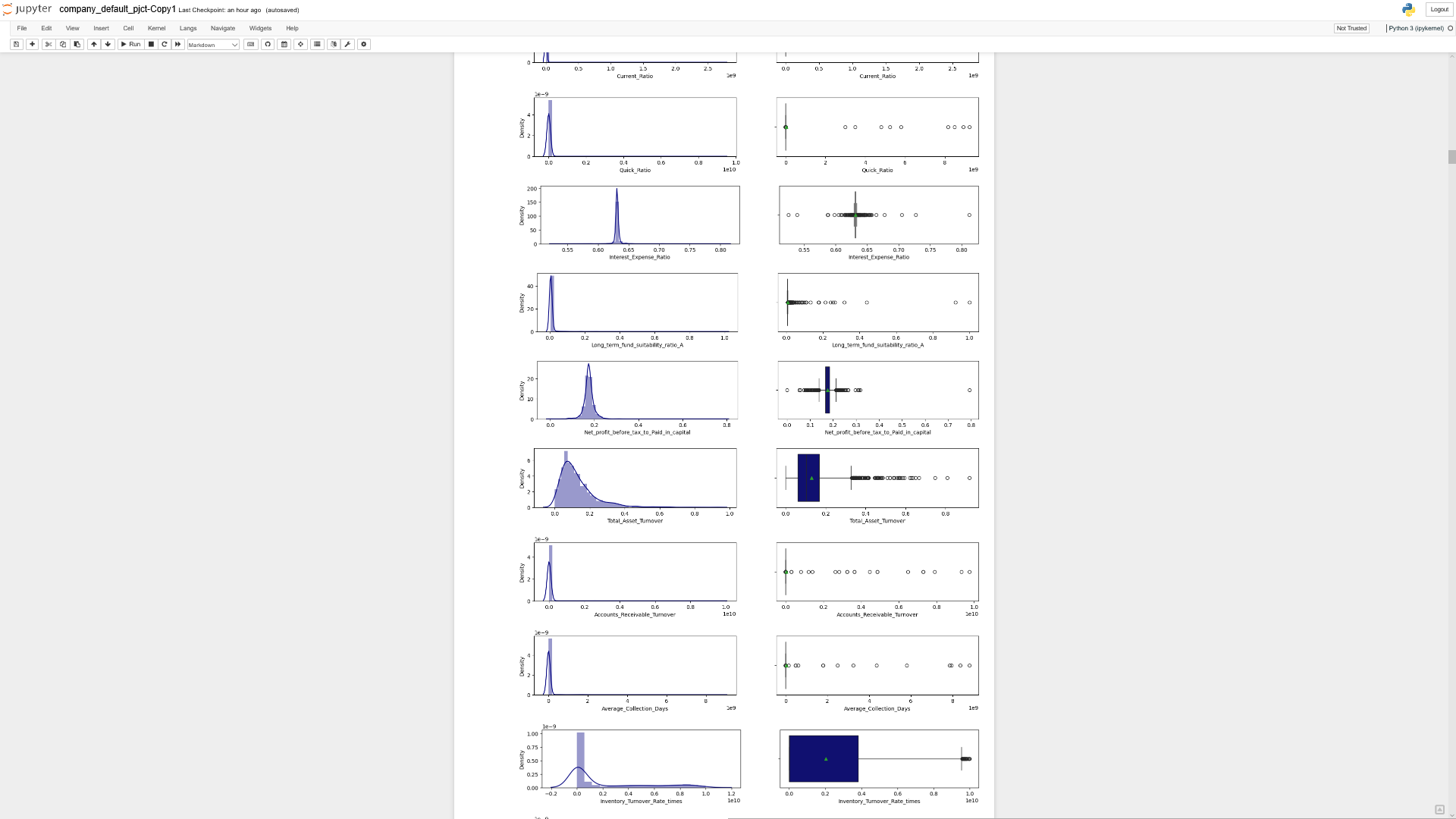
* There are about 5% of missing values in the Cash\_flow\_per\_share and 6.4% in Cash\_to\_total\_assets in rows where Default=1.
* Overall, about 13.18% of the rows having the desired class of the target variable have missing values
* Dropping these rows may impact the target variable and hence imputing might be a better option
* For the purpose of imputation, kNN imputer was used

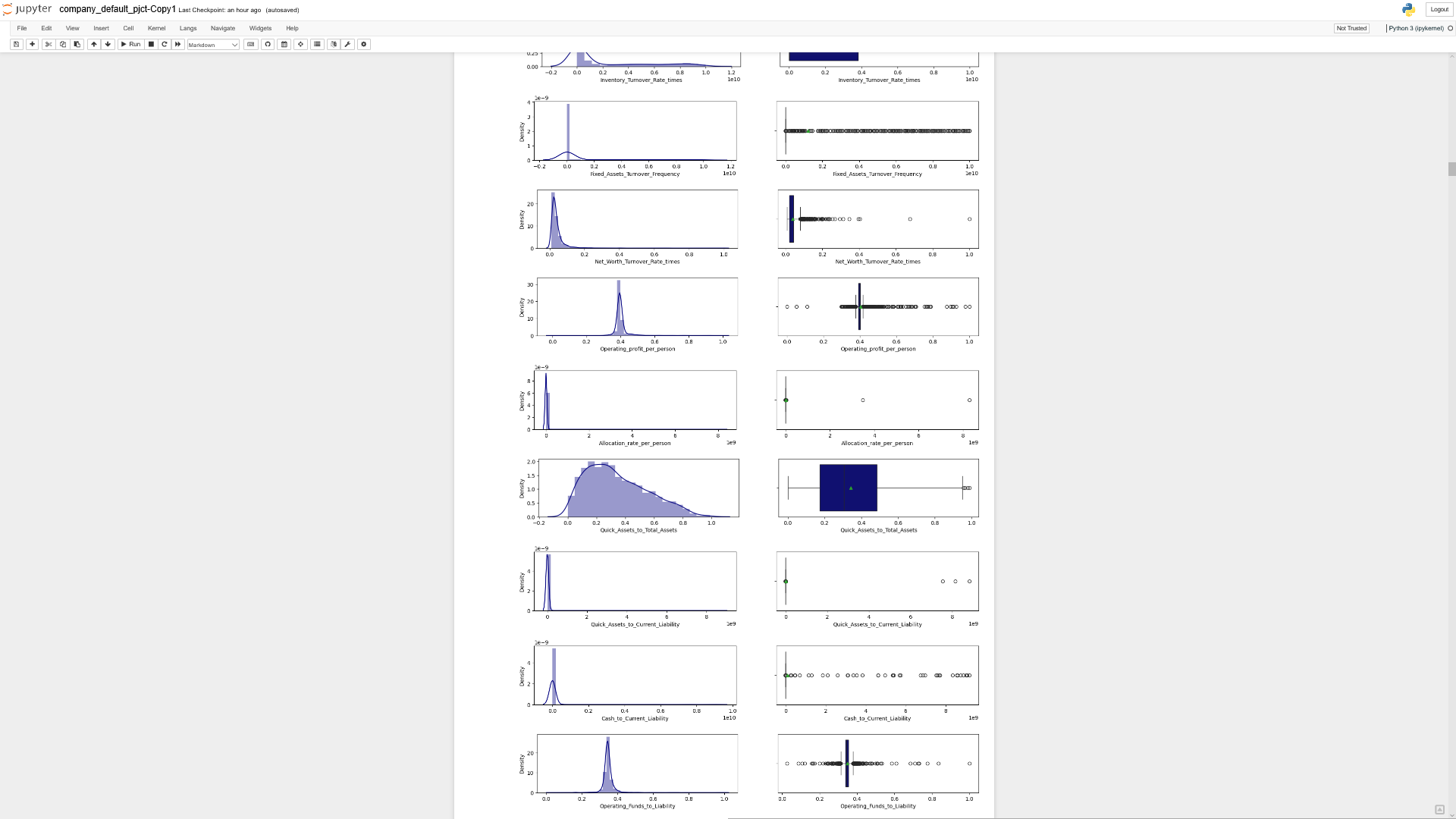
1. **Univariate (4 marks) & Bivariate (6 marks) analysis with proper interpretation. (You may choose to include only those variables which were significant in the model building)**

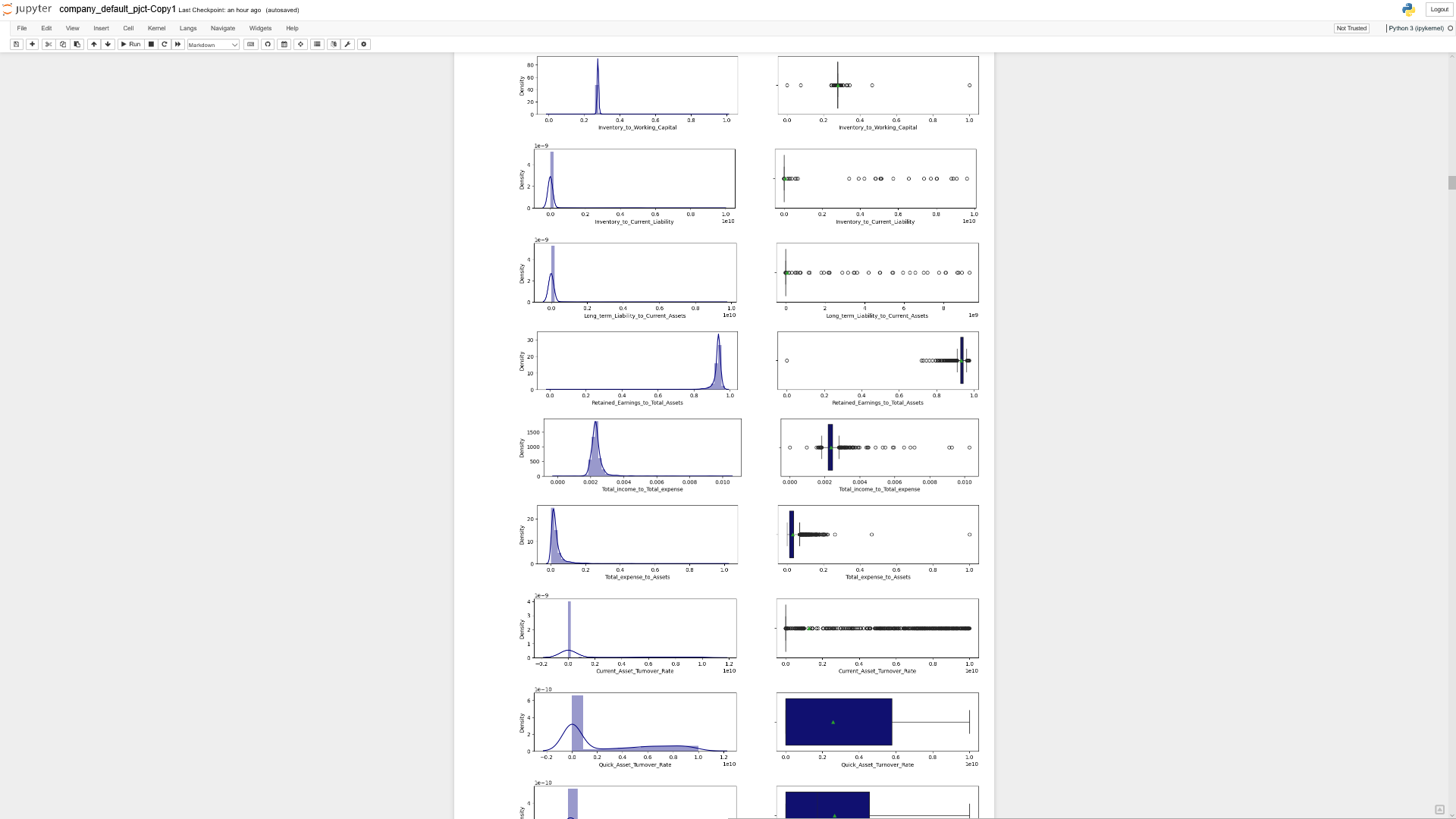
Univariate Analysis:

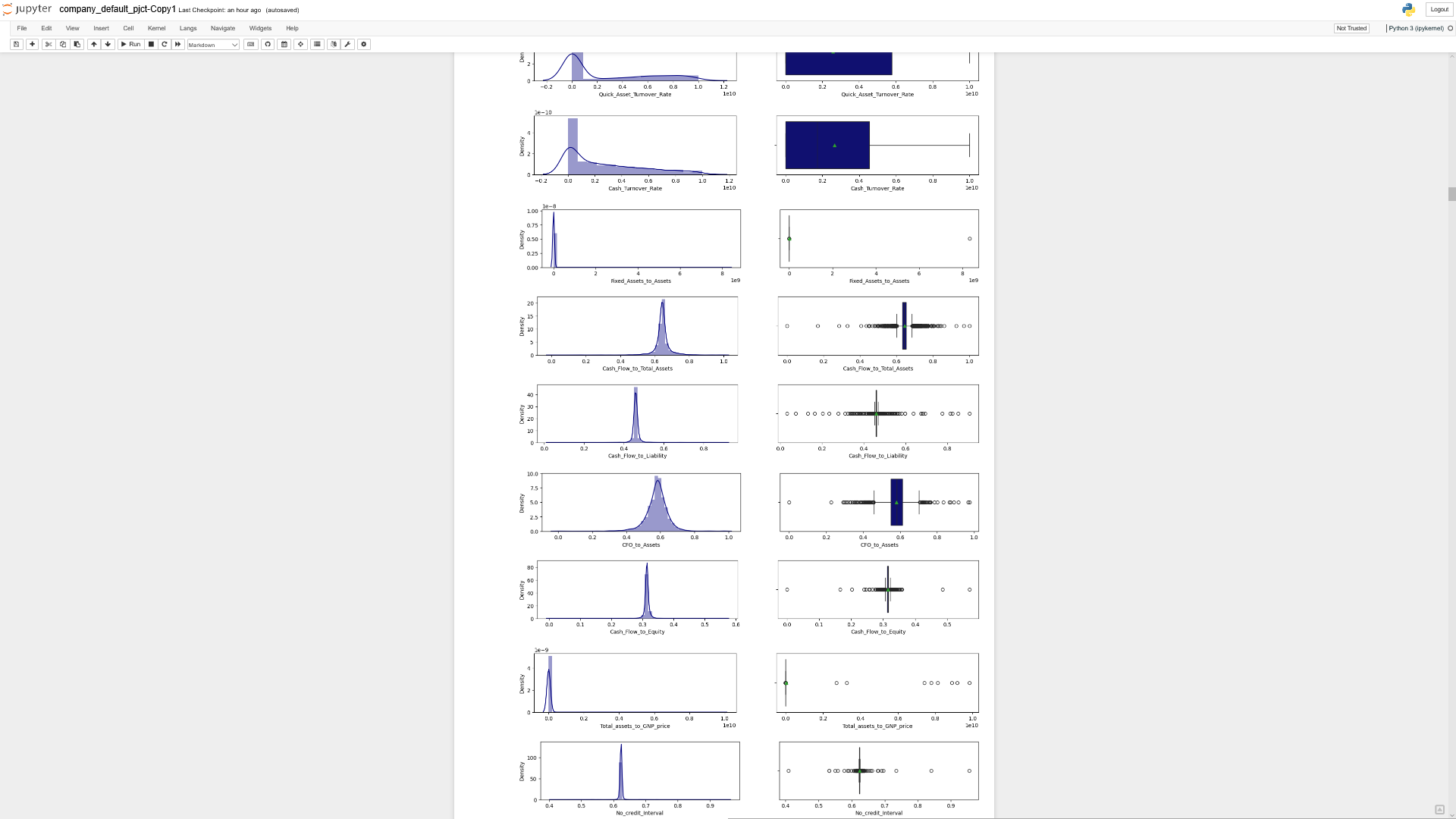












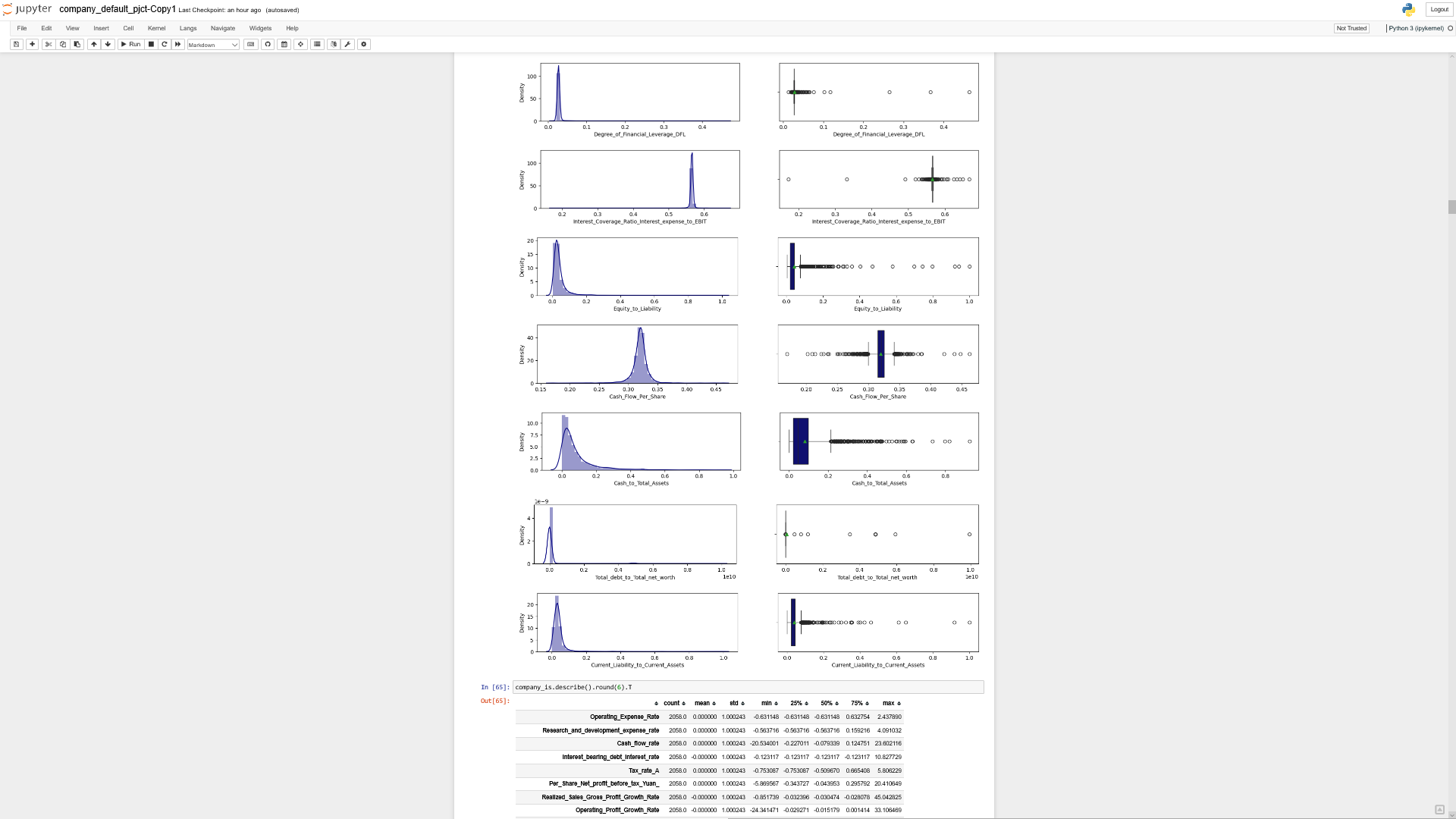


Fig.1.8. Univariate Analysis of Numerical Variables

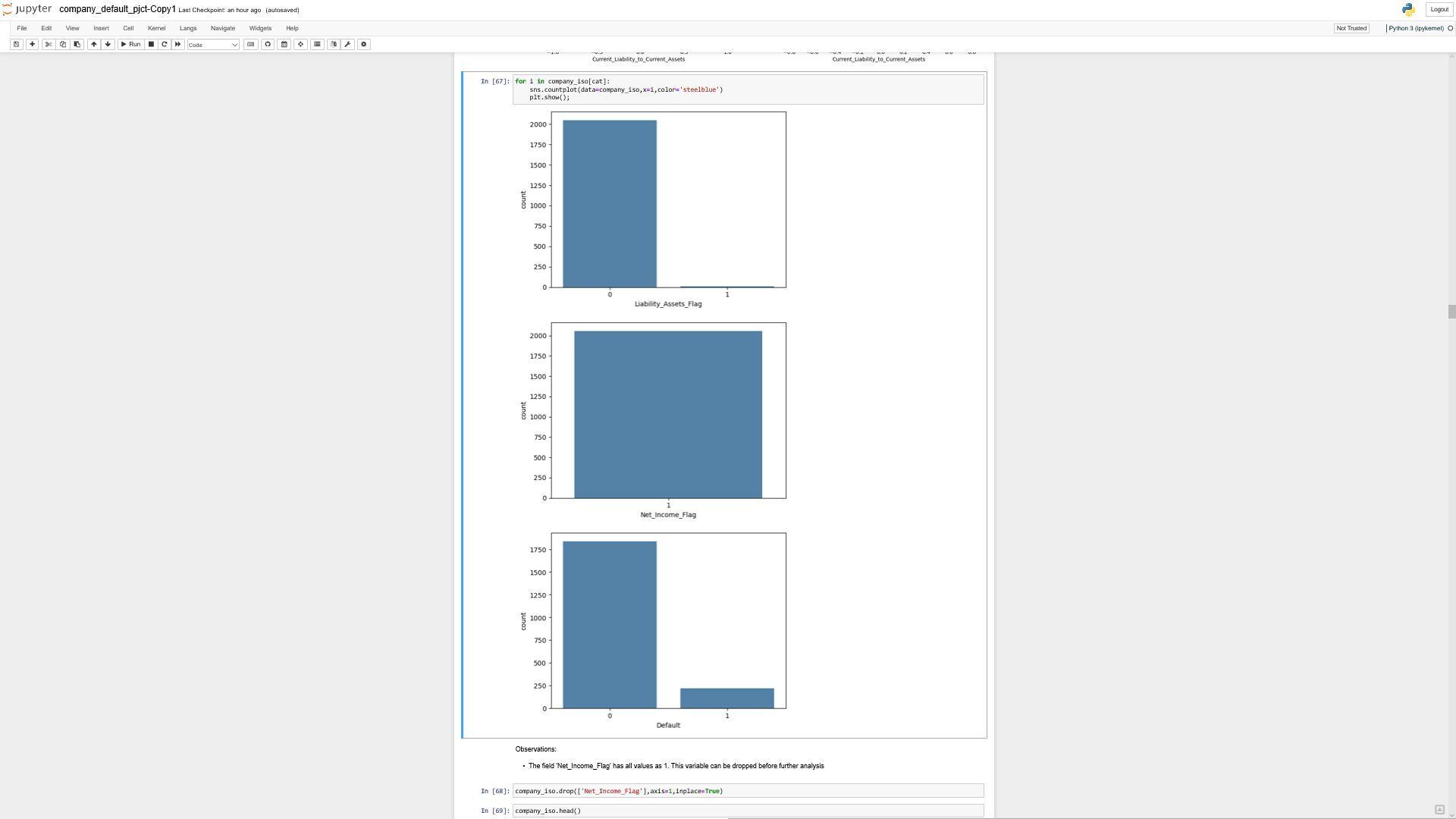


Fig.1.9. Univariate Analysis- Categorical Variables

Observations:

* The field 'Net\_Income\_Flag' has all values as 1. This variable can be dropped before further analysis

Bivariate Analysis:

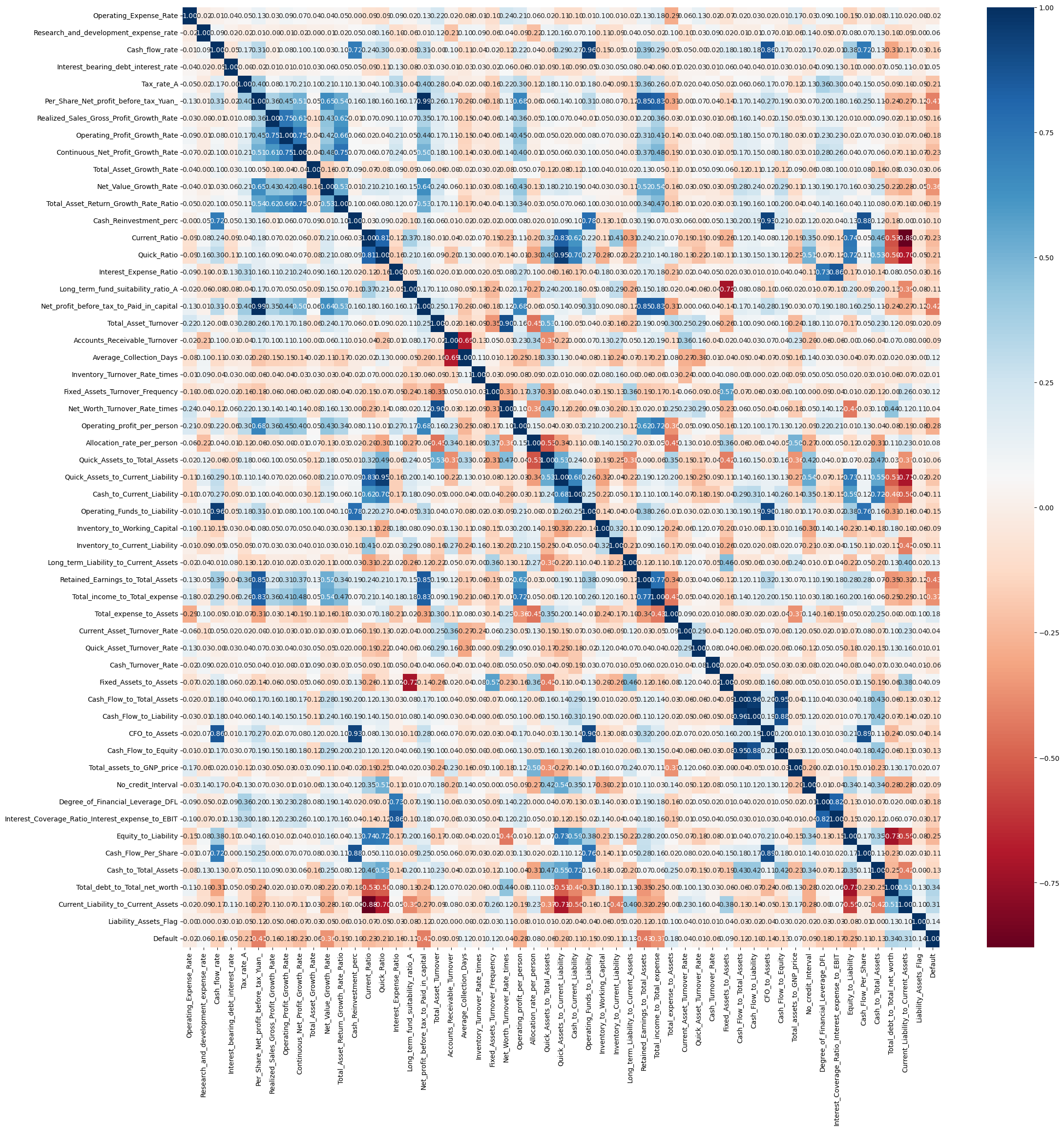
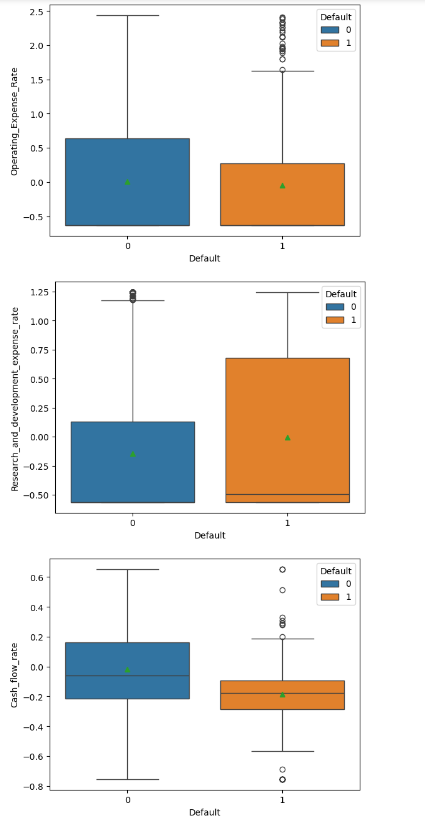
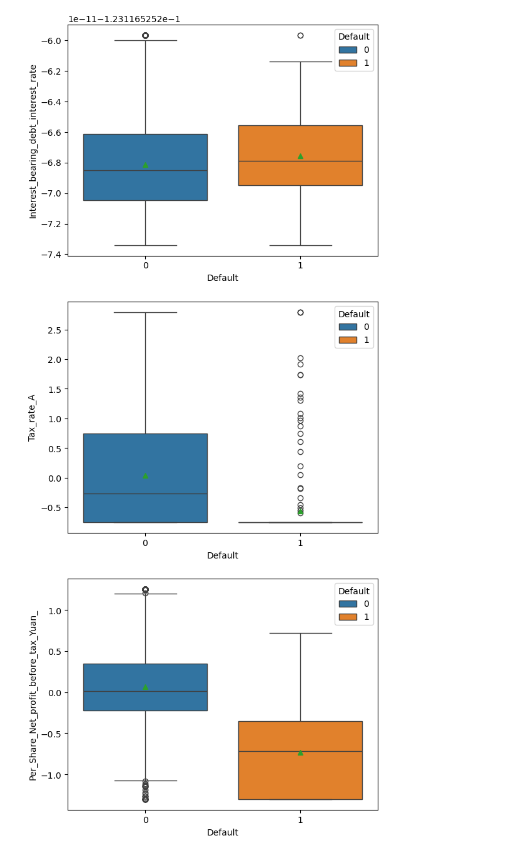


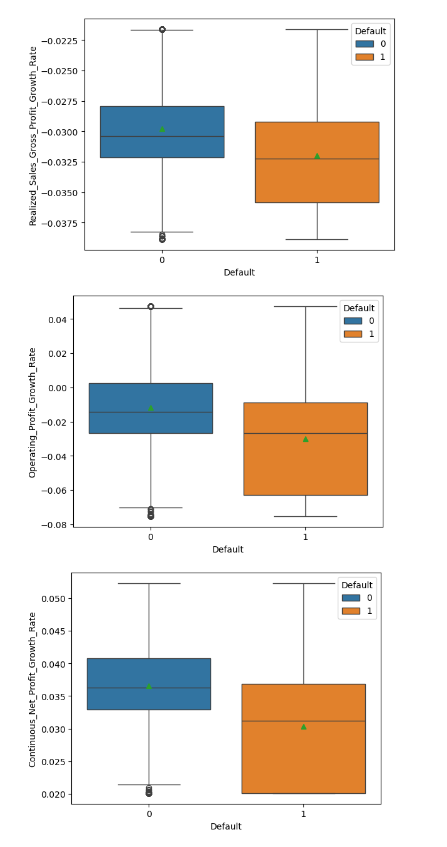
Fig.1.10 Bivariate analysis- Heatmap of numerical variables

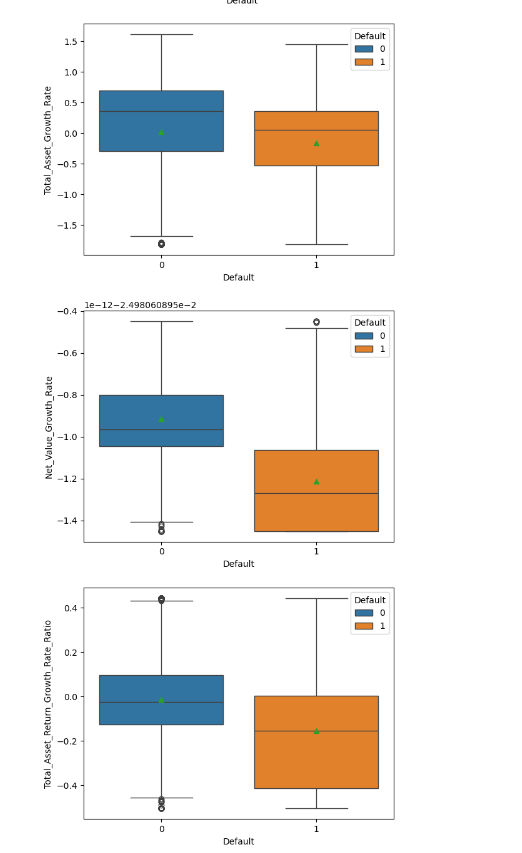
Observations:

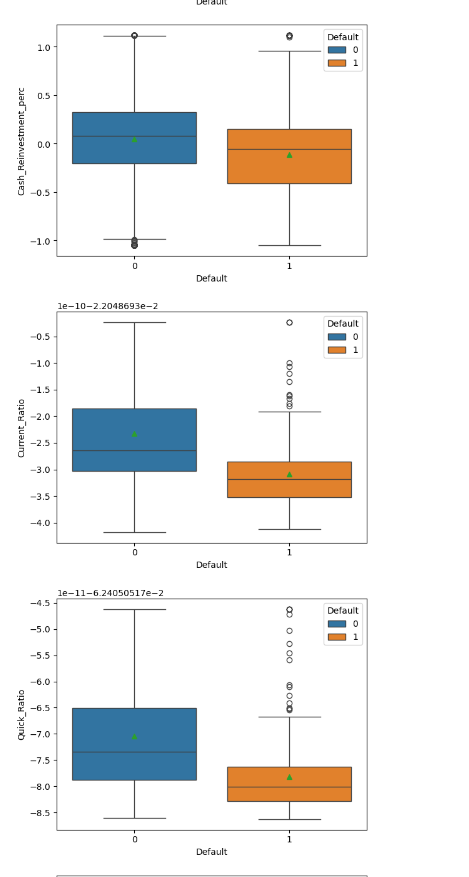
* Multicollinearity is present, which has to be addressed before further processing

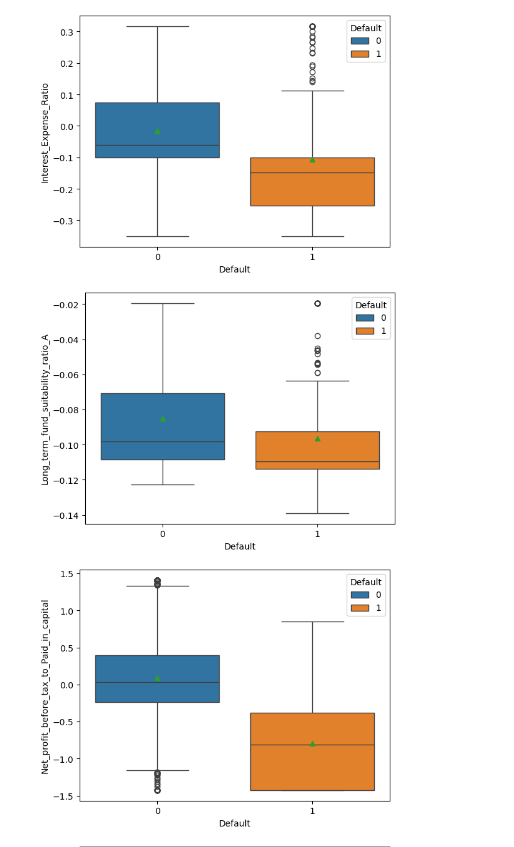


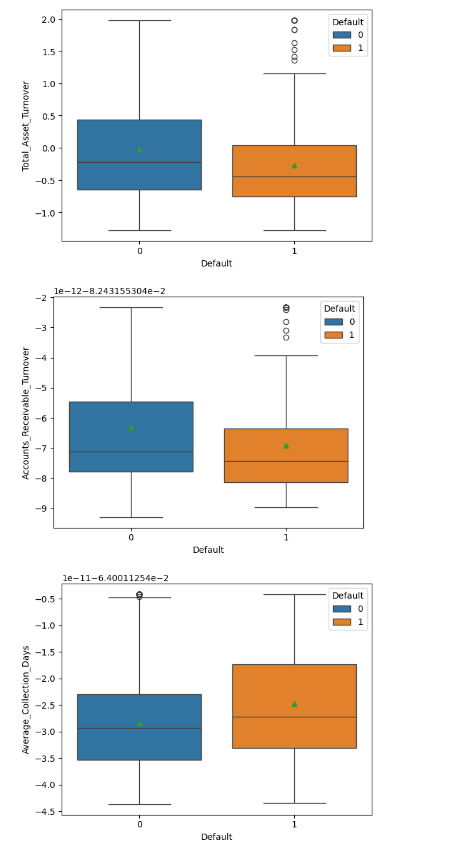


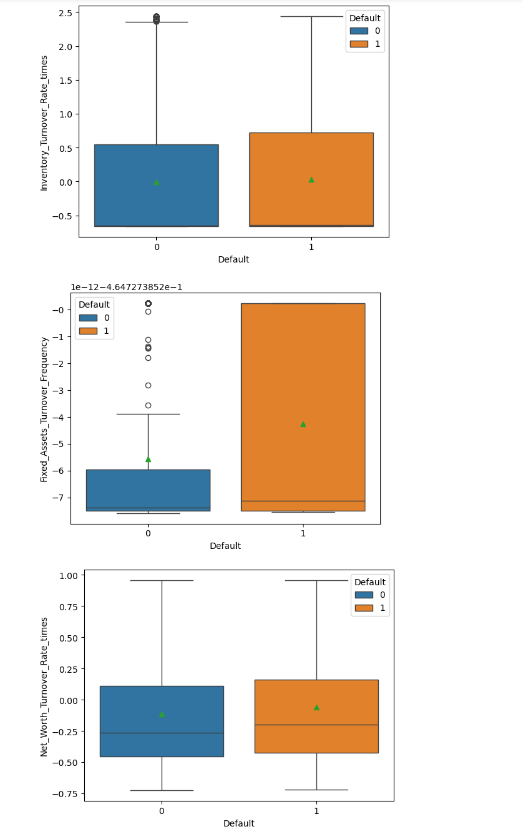


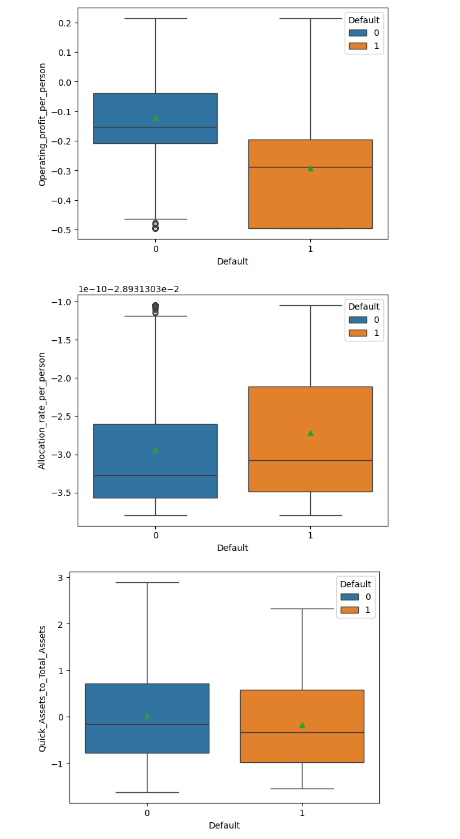


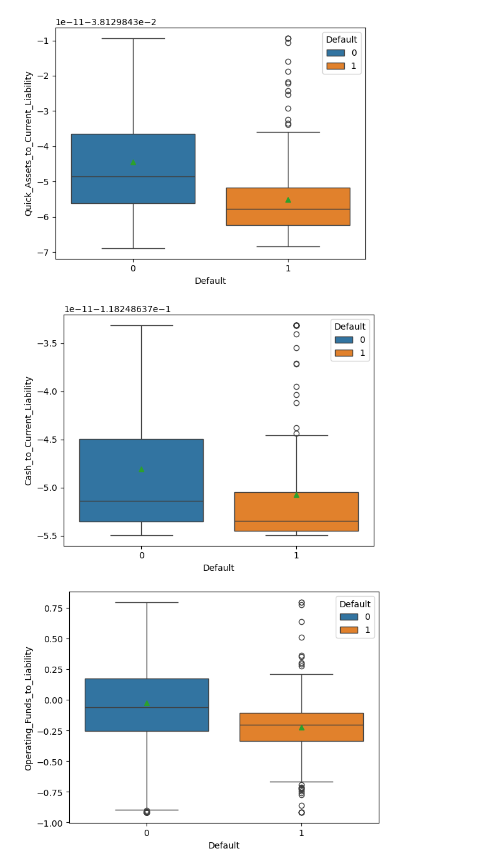


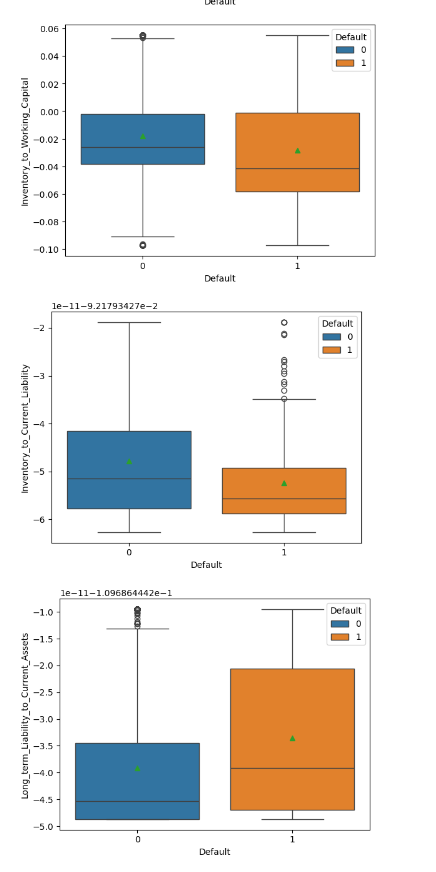


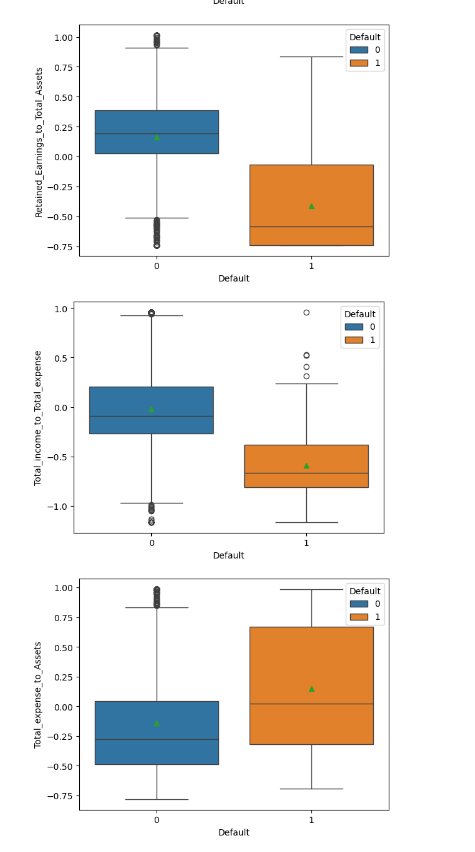


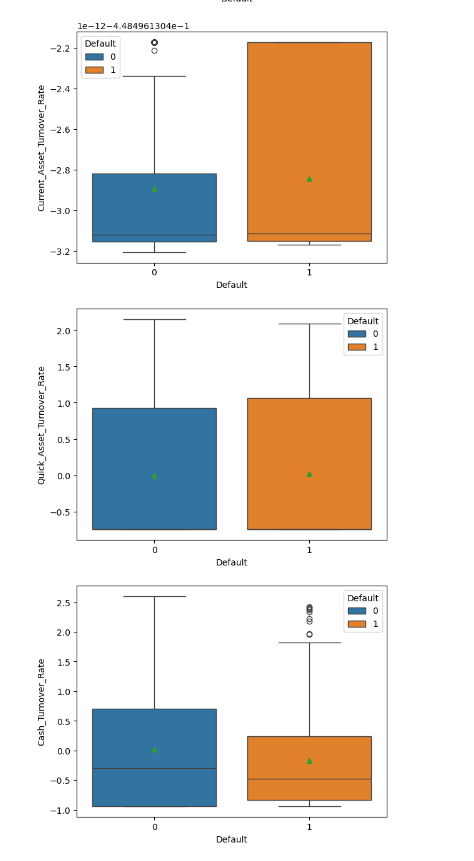


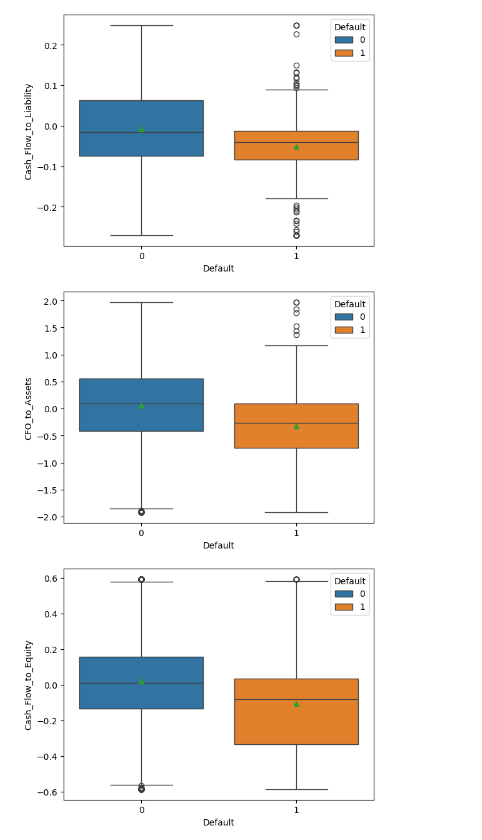


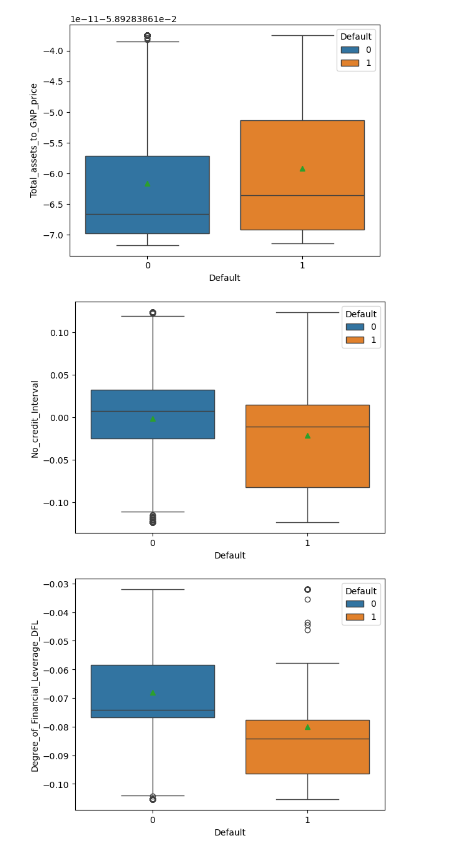


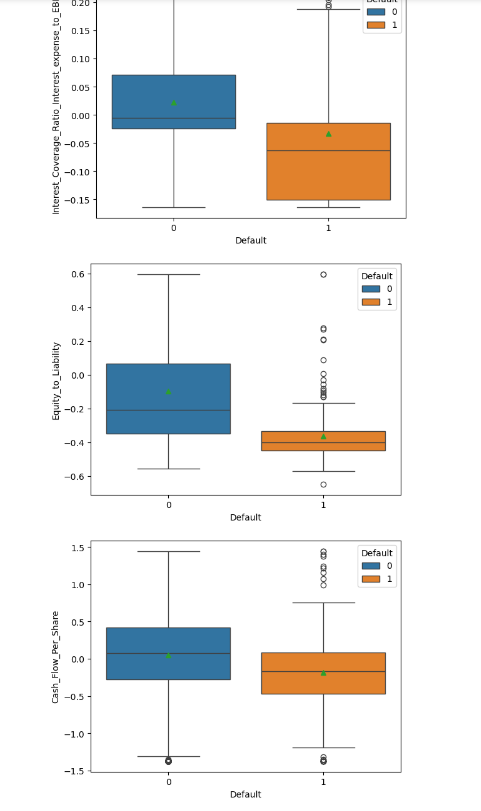












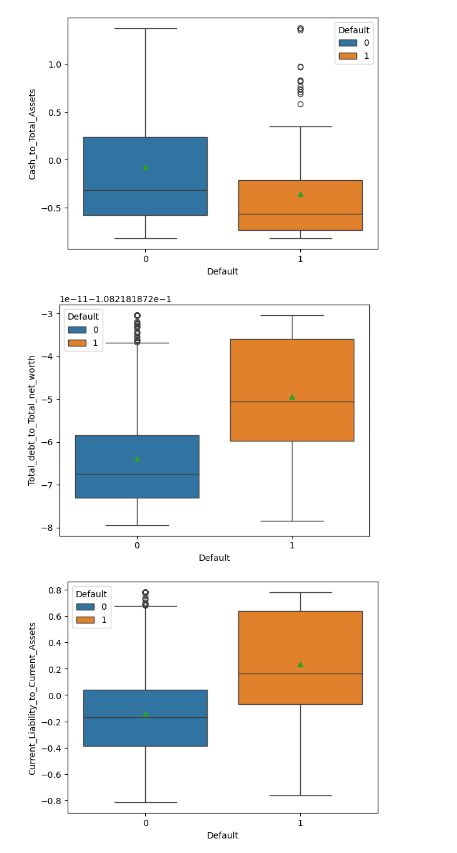


Fig.1.11. Bivariate Analysis- Numerical Variables vs Target Variables

Observations:

* For the defaulters, the mean and the median 'Research\_and\_development\_expense\_rate','Average\_Collection\_Days', 'Allocation\_rate\_per\_person','Long\_term\_Liability\_to\_Current\_Assets', 'Total\_expense\_to\_Assets', 'Fixed\_Assets\_to\_Assets', 'Total\_assets\_to\_GNP\_price', 'Total\_debt\_to\_Total\_net\_worth', 'Current\_Liability\_to\_Current\_Assets'are significantly higher compared to non-defaulters
* Also, the mean and median 'Cash\_flow\_rate', 'Tax\_rate\_A','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','Long\_term\_fund\_suitability\_ratio\_A', 'Net\_profit\_before\_tax\_to\_Paid\_in\_capital', 'Total\_Asset\_Turnover','Operating\_profit\_per\_person', 'Quick\_Assets\_to\_Current\_Liability', Cash\_to\_Current\_Liability','Operating\_Funds\_to\_Liability','Inventory\_to\_Working\_Capital', 'Inventory\_to\_Current\_Liability','Retained\_Earnings\_to\_Total\_Assets', 'Total\_income\_to\_Total\_expense', 'Cash\_Flow\_to\_Total\_Assets','Cash\_Flow\_to\_Liability', 'CFO\_to\_Assets', 'Cash\_Flow\_to\_Equity','No\_credit\_Interval', 'Degree\_of\_Financial\_Leverage\_DFL','Interest\_Coverage\_Ratio\_Interest\_expense\_to\_EBIT','Equity\_to\_Liability', 'Cash\_Flow\_Per\_Share', 'Cash\_to\_Total\_Assets'are significantly lower compared to non- defaulters
* The variables have high multicollinearity and there are 55 dimensions. So, in order to reduce dimensionality and reduce multicollinearity, VIF method was used to eliminate the variables that have VIF>5 one after the other

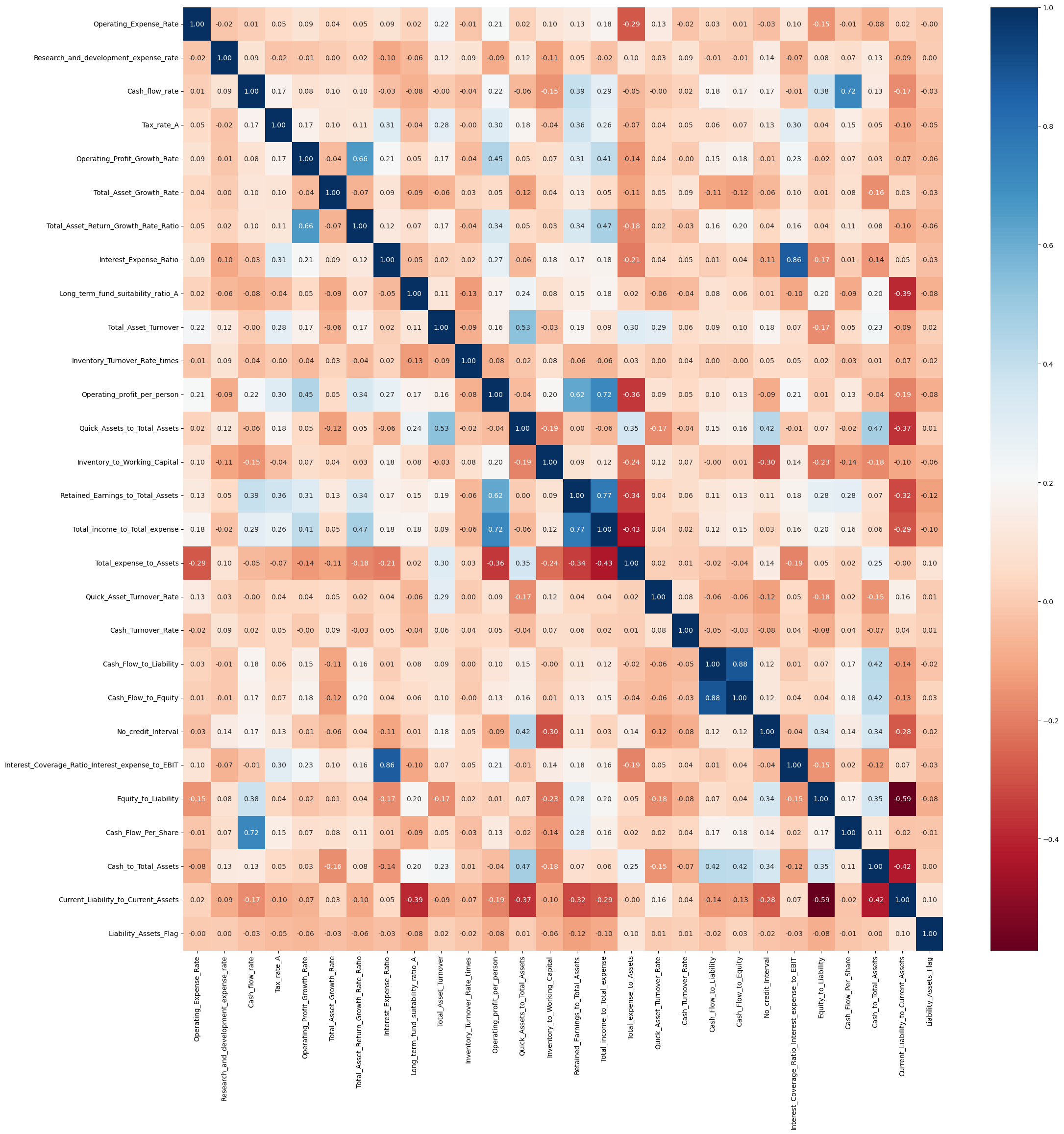
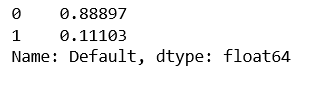
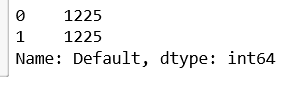


Fig.1.12. Heatmap of correlation between numerical variables after eliminating variables causing high multicollinearity using VIF

1. **Train Test Split**

Observations:

* The train-test split was done keeping a test size of 0.33, and random state 42
* After the split the shapes of the datasets are,
  + X\_train =(1378,28)
  + X\_test=(680,28)
  + Y\_train=(1378,)
  + Y\_test=(680,)
* On exploring the the y\_train dataset, the target variable proportions are:
* Thus, a SMOTE balancing was done, after which split the shapes of the datasets are,
  + X\_train =(2450,28)
  + X\_test=(680,28)
  + Y\_train=(2450,)
  + Y\_test=(680,)
* The split of proportion of the target variables for both the classes was 50%.
  + 

1. **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**

Approach:

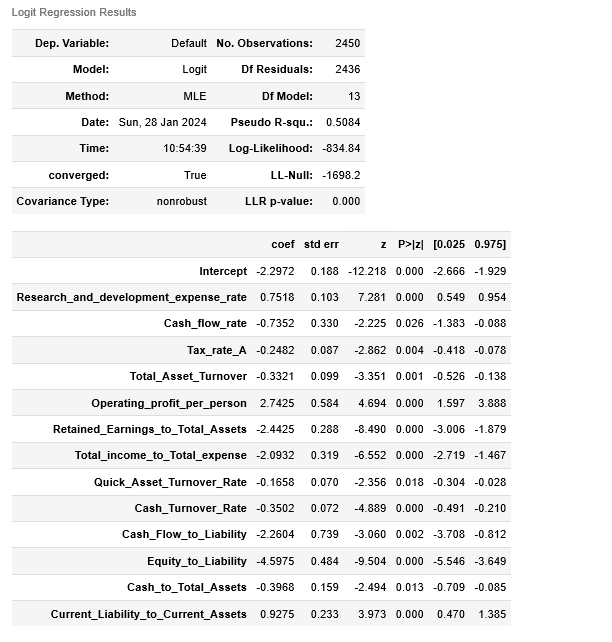
* The logistic regression from statsmodels library was used for model building.
* From the 28 predictor variables, the non-significant variables having p-value>0.05 were eliminated one by one.
* After 16 such iterations, the model was built using the formula:
  + logit\_model16= SM.logit(formula= 'Default ~ Research\_and\_development\_expense\_rate + Cash\_flow\_rate + Tax\_rate\_A + Total\_Asset\_Turnover + Operating\_profit\_per\_person + Retained\_Earnings\_to\_Total\_Assets + Total\_income\_to\_Total\_expense + Quick\_Asset\_Turnover\_Rate + Cash\_Turnover\_Rate + Cash\_Flow\_to\_Liability + Equity\_to\_Liability + Cash\_to\_Total\_Assets + Current\_Liability\_to\_Current\_Assets ' ,data=Default\_train).fit()
* Model Summary:
* 

Fig.1.13. Logistic Regression model summary

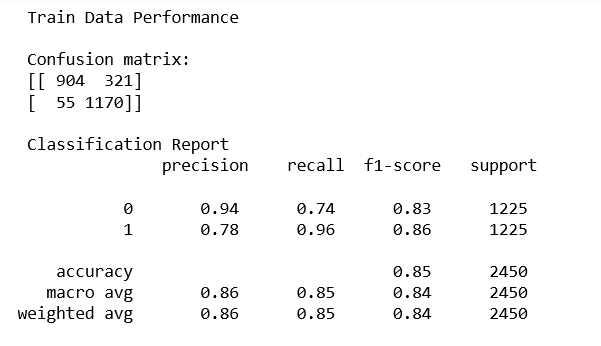


Fig.1.14. Logistic Regression- Train data performance

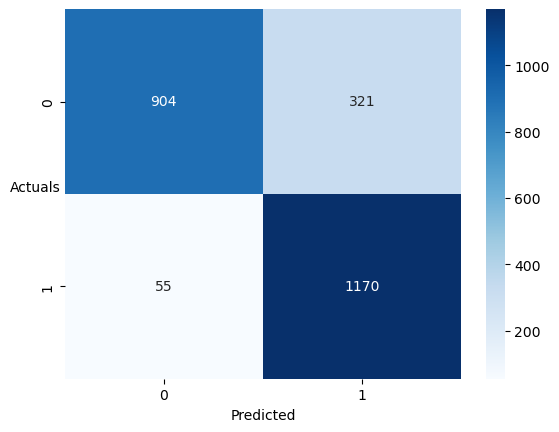


Fig.1.15. Logistic Regression- Confusion matrix- Train Data

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

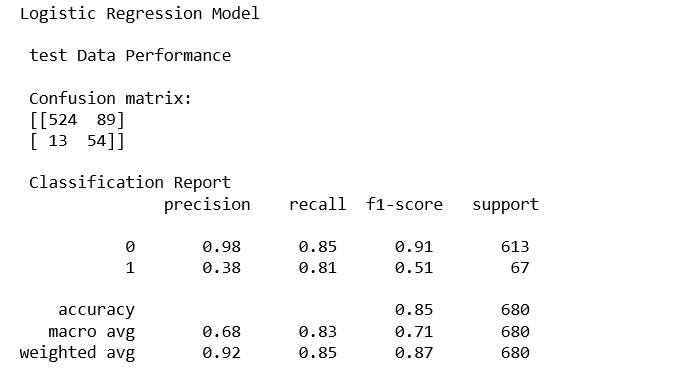


Fig.1.16. Logistic Regression Model- Test Data Performance

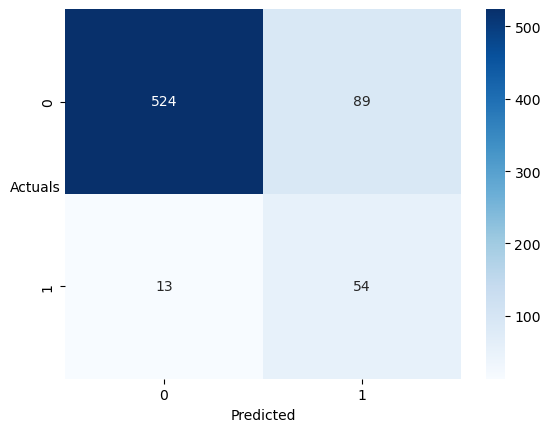


Fig.1.17. Logistic Regression- Confusion Matrix- Test Data

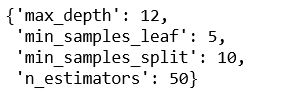
Interpretations:

* The model was overfit in the train set for precision metric.
* The performance of the model was consistent in both recall and accuracy across both train and test sets

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

Approach:

* The model was built by using grid search for choosing best parameters for optimized results
* Best parameters:



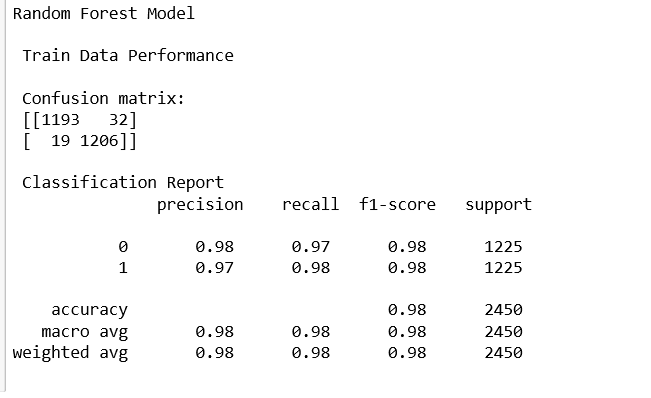


Fig.1.18. Random Forest model- Train Data performance

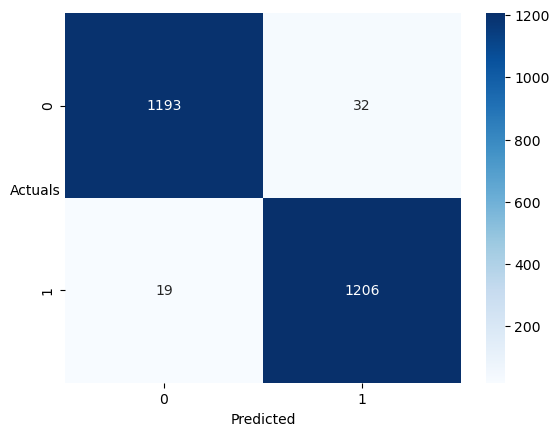


Fig.1.19. Random Forest model- Confusion Matrix- Train Data

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

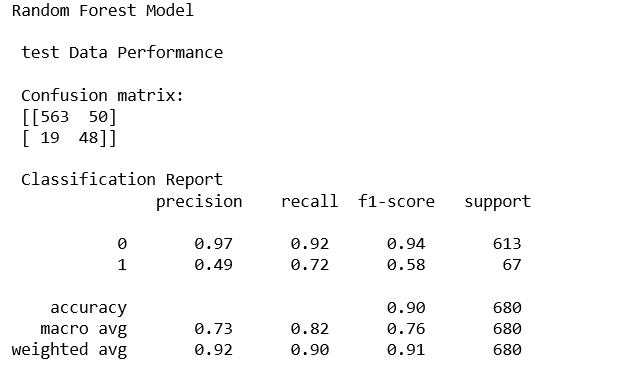


Fig.1.20. Random Forest model- Test data performance

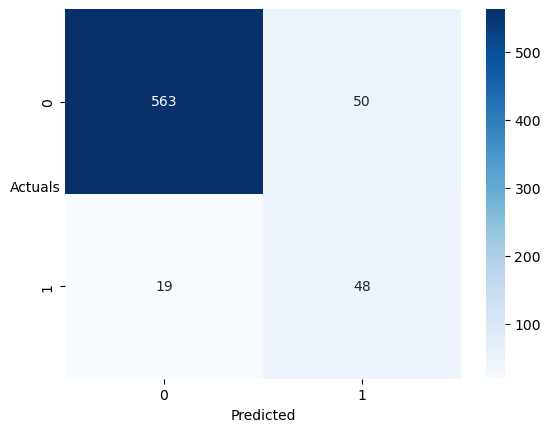


Fig.1.21. Random Forest model-Confusion matrix- test data

Interpretations:

* The model was extremely overfit, as all the scores- precision, recall, accuracy and f1 scores were excellent on train data
* However, on the test data, the model showed a good performance for accuracy, moderate performance for recall and a very poor performance for precision.

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

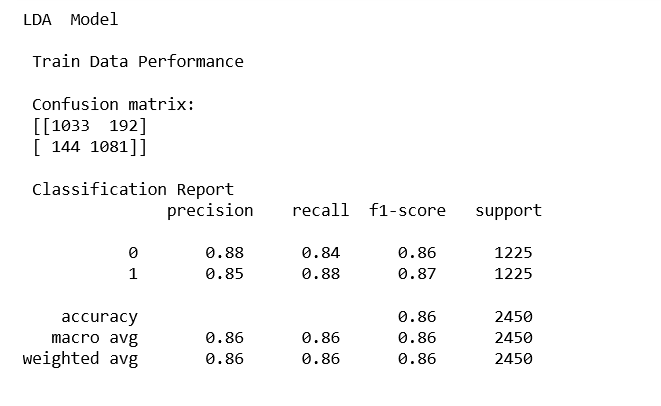


Fig.1.22. LDA model- Train data performance

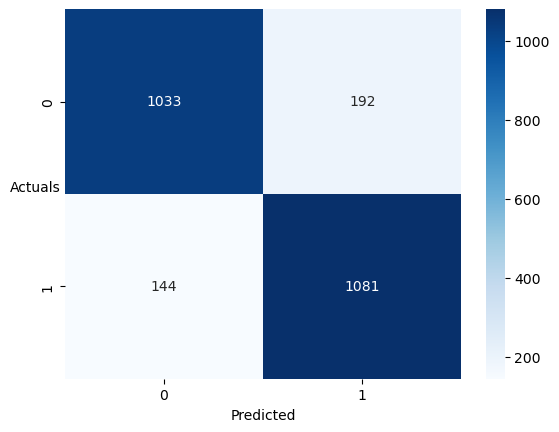


Fig.1.23. LDA model- Confusion Matrix- Train Data

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

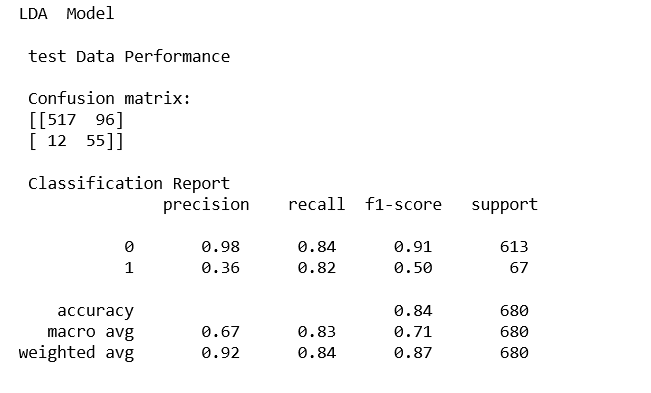


Fig.1.24. LDA model- Test Data Performance

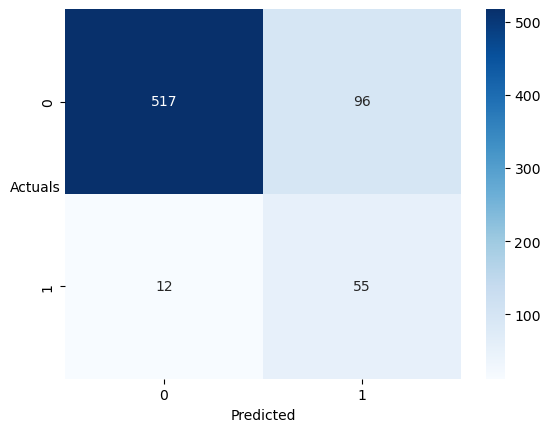


Fig.1.25. LDA model- Confusion Matrix- test data

Interpretations:

* The model had an overfit precision score on the train data
* The recall of the model for the default class was consistent on both the train and test data, as was the accuracy

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

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Model Name | Precision | Recall | F1 Score | Accuracy | ROC curve |
| Logistic Regression | 0.38 | 0.81 | 0.51 | 0.82 | C:\Users\India\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\392842B5.tmp |
| Random Forest | 0.49 | 0.72 | 0.58 | 0.90 | C:\Users\India\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\4FA270CF.tmp |
| LDA | 0.40 | 0.82 | 0.53 | 0.83 | C:\Users\India\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\A1BAB459.tmp |

Table 1.1. Test Data Preformance Comparison

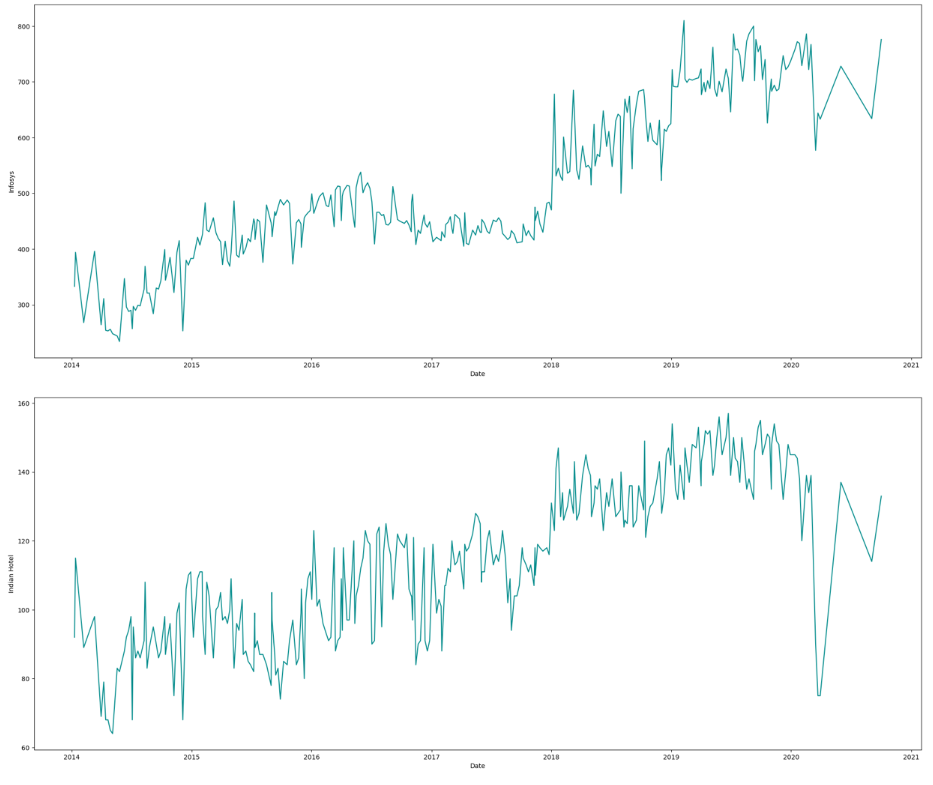
1. **Conclusions and Recommendations**

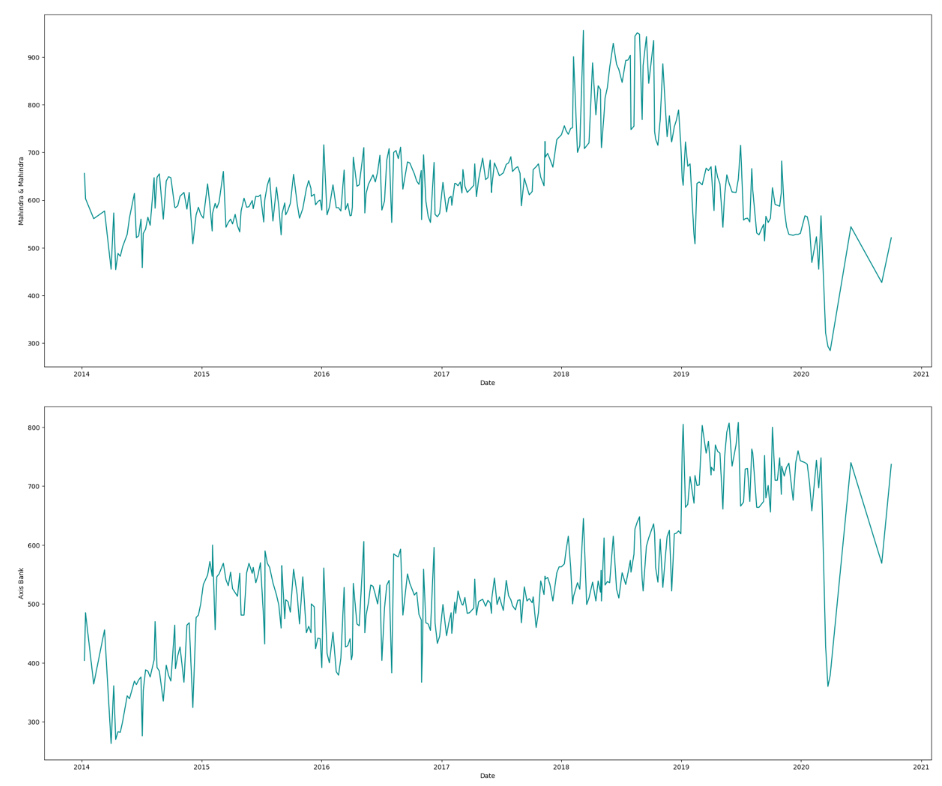
* This exercise was to identify a model for the classification of companies as Defaulter or Non- Defaulter based on certain financial parameters
* As the motive is credit analysis, this assessment will most likely be used to determine whether the companies are eligible for credit facility or not.
* Hence, considering the objective, the most important evaluation parameter would be recall, as it is better to make a misclassification as Defaulter than non-defaulter.
* Thus, based on recall Scores, the LDA model would be the most useful as evident from the table above
* However, the Random Forest model could be chosen for overall performance in all the metrics considered

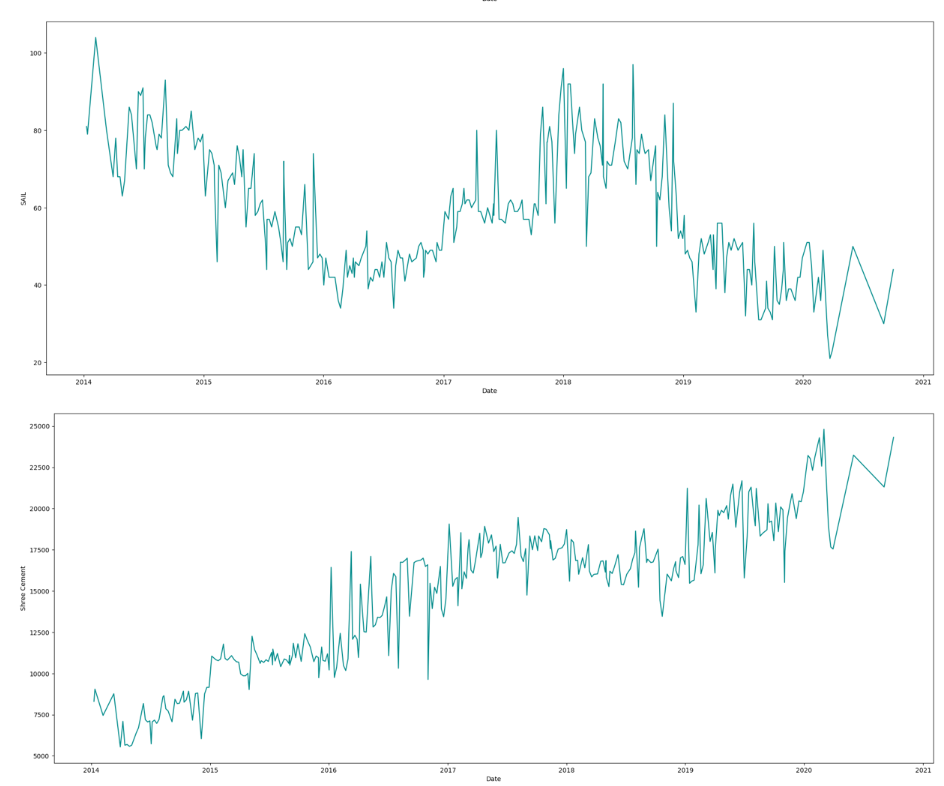
# Case 2: Market Risk Analysis

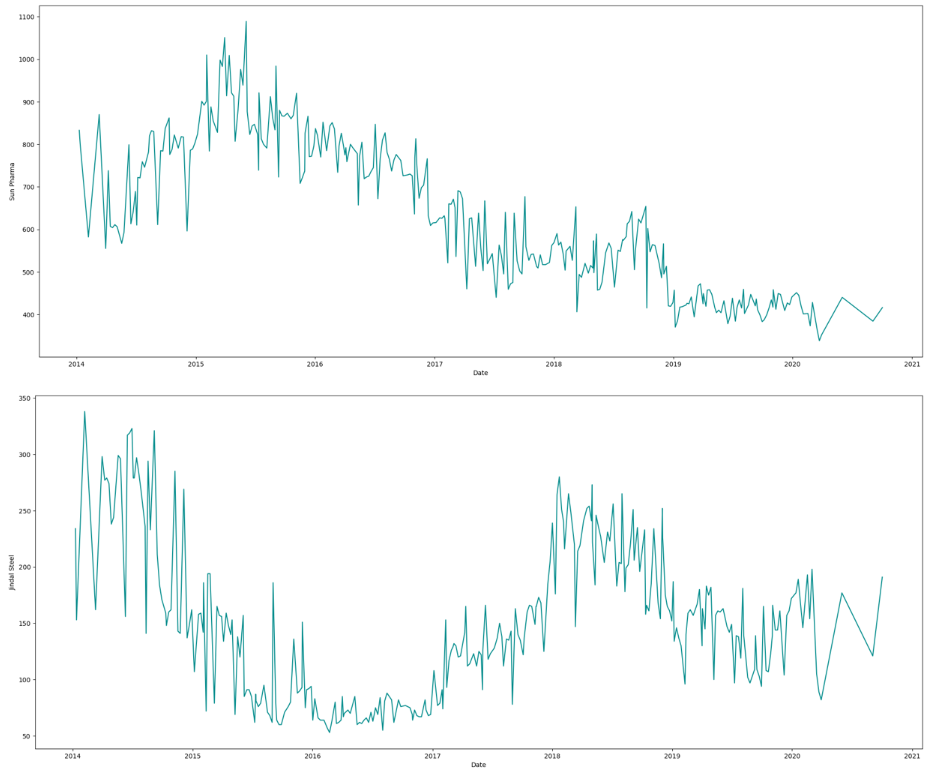
The dataset contains 6 years of information(weekly stock information) on the stock prices of 10 different Indian Stocks. Calculate the mean and standard deviation on the stock returns and share insights. You are expected to do the Market Risk Analysis using Python.

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









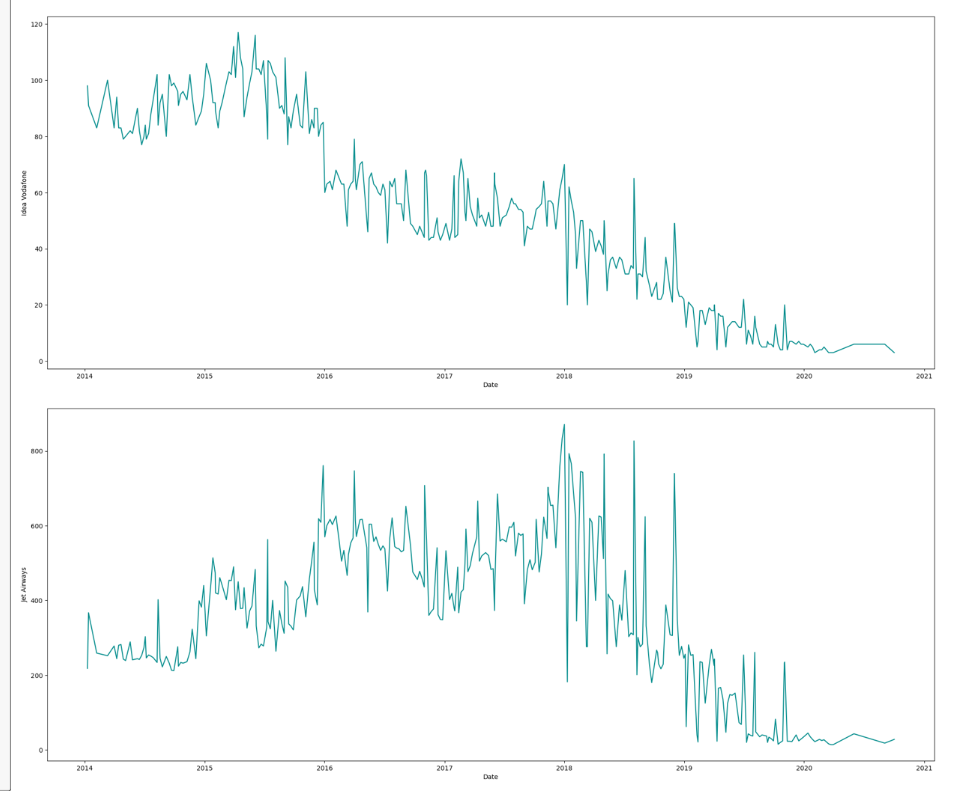


Fig.2.1. Price Vs time plots for stocks

Observations:

* Infosys, Indian Bank, Axis Bank and Shree Cement have a rising trend
* Mahindra & Mahindra , SAIL, Sun Pharma, Jindal Steel, Idea Vodafone , Jet Airways all show a declining trend
* Just within the scope of this data and from this preliminary analysis, bank and IT stocks seemed to have fared better

1. **Calculate Returns for all stocks with inference**

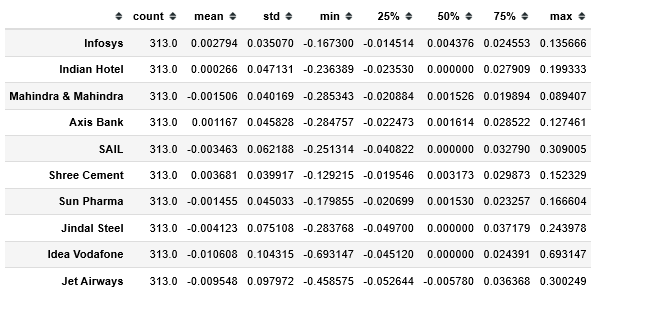


Fig.2.2. Market returns- Data Description

Observations:

* Infosys, Indian Hotel, Axis Bank, Shree Cement all have positive mean returns
* The rest have negative mean returns.
* These are consistent with the observations from the plots

1. **Calculate Stock Means and Standard Deviation for all stocks with inference**



Fig.2.3. Stocks- Returns and Volatility (Std deviation)

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

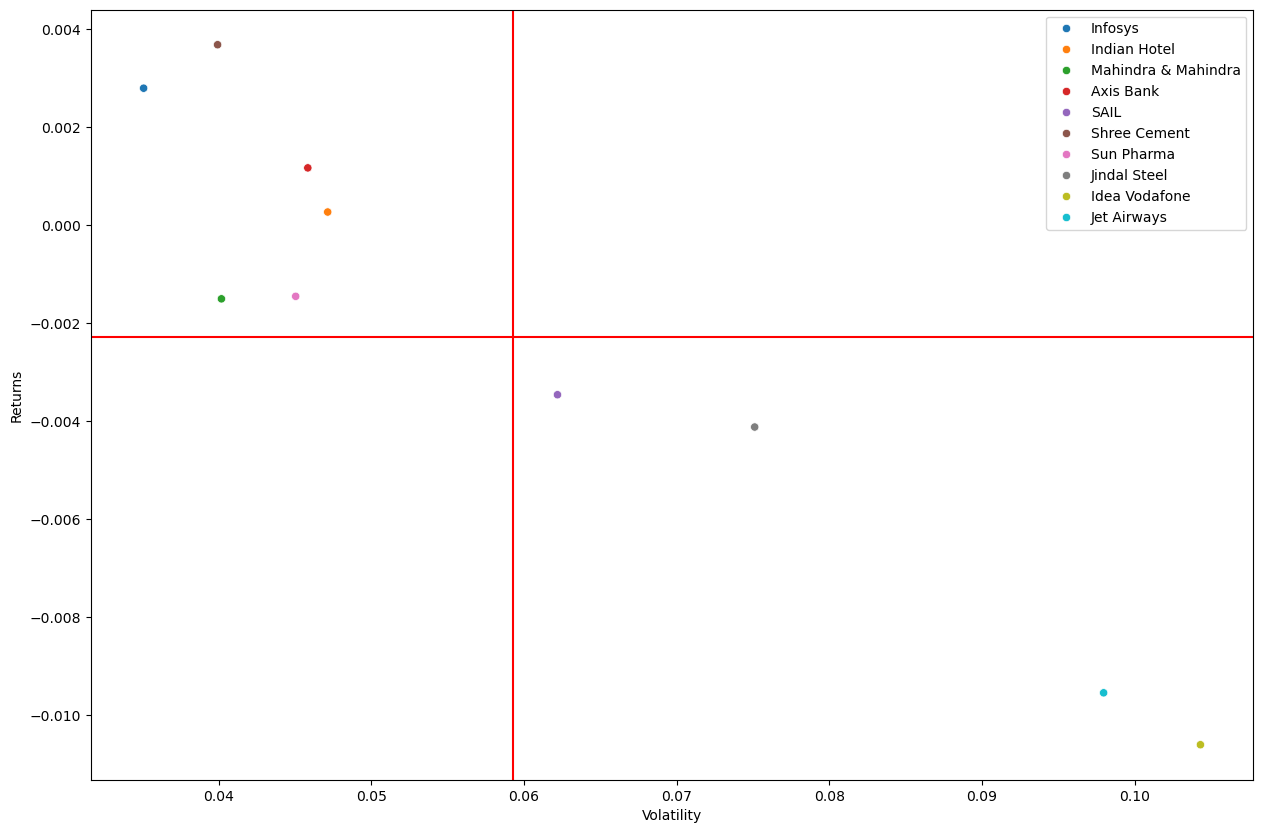


Fig.2.4. Returns vs Volatility

Observations:

* Infosys and Shree Cement are the best stocks, with high returns and low volatility
* Idea Vodafone and Jet Airways are the worst performing stocks, with low returns and high volatility
* Shree Cement has the highest returns and Idea Vodafone has the lowest returns
* Infosys has the lowest volatility and Idea Vodafone has the highest volatility

1. **Conclusions and Recommendations**

* The performance of Infosys and Shree Cement have been really good compared to other stocks, hence a buy or hold call is advised
* Idea Vodafone is the worst of all. Hence, a hold or sell call is advised
* Bank stock and IT stocks seem to be performing better overall.