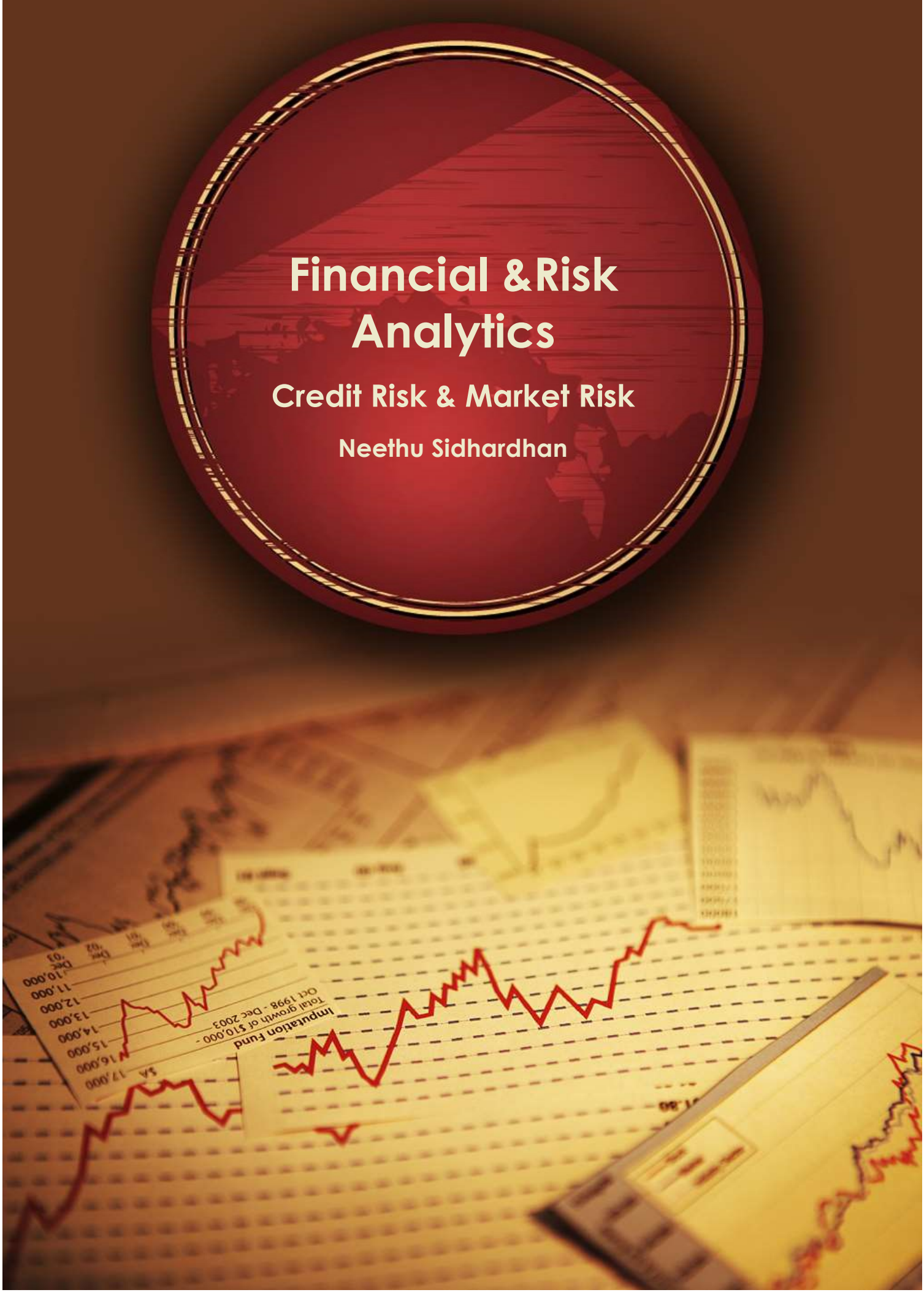
Financial &Risk Analytics

Credit Risk & Market Risk

Neethu Sidhardhan



1

Credit Risk

Credit risk is the risk arising due to the borrower's failure to strictly comply with the terms of the credit contract. This might happen when the customer is late in debt repayment, not fully pays the debt amount or fails to pay debt when principal and interest amounts are due, causing financial losses and difficulties in the business activities of commercial banks..

A credit risk is risk of default on a debt that may arise from a borrower failing to make required payments.[1] In the first resort, the risk is that of the lender and includes lost principal and interest, disruption to cash flows, and increased collection costs. The loss may be complete or partial. In an efficient market, higher levels of credit risk will be associated with higher borrowing costs

Case Study 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 interests 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 (2015). Also, information about the Networth of the company in the following year (2016) is provided which can be used to drive the labeled field.

Explanation of data fields available in Data Dictionary, 'Credit Default Data Dictionary.xlsx'

We need the below libraries for processing the case study. Import all the libraries.

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import seaborn as sns # for making plots with seaborn
color = sns.color_palette()
import sklearn.metrics as metrics
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn import metrics
from sklearn.metrics import roc_auc_score,roc_curve,classification_report,confusion_matrix
import warnings
warnings.filterwarnings("ignore")
```

Import the dataset and check the heading to confirm its imported perfectly.

```
Default = pd.read_csv('Company data.csv')
```

	Num	Networth Next Year	Total assets	Net worth	Total income	Change in stock	Total expenses	Profit after tax	PBDITA	PBT	...	Debtors turnover	Finished goods turnover	WIP turnover	Raw material turnover	Shares outstanding	Equity face value	EPS
0	1	8890.6	17512.3	7093.2	24965.2	235.8	23657.8	1543.2	2860.2	2417.2	...	3.85	200.55	21.78	7.71	42381675.0	10.0	35.52
1	2	394.3	941.0	351.5	1527.4	42.7	1454.9	115.2	283.0	188.4	...	5.70	14.21	7.49	11.46	11550000.0	10.0	9.97
2	3	92.2	232.8	100.6	477.3	-5.2	478.7	-6.6	5.8	-6.6	...	5.07	9.24	0.23	NaN	8149090.0	10.0	-0.5(
3	4	2.7	2.7	2.7	NaN	NaN	NaN	NaN	NaN	NaN	...	0.00	NaN	NaN	0.00	52404.0	10.0	0.0(
4	5	109.0	478.5	107.6	1580.5	-17.0	1558.0	5.5	31.0	6.3	...	9.46	12.68	7.90	17.03	619635.0	10.0	7.9

The dataset consists of 52 columns and 3541 rows.

The columns of the dataset are having spaces and symbols in the column header.

Hence before proceeding to further analysis it required to correct the column header.

```
Index(['Num', 'Networth Next Year', 'Total assets', 'Net worth',
       'Total income', 'Change in stock', 'Total expenses', 'Profit after tax',
       'PBDITA', 'PBT', 'Cash profit', 'PBDITA to total income',
       'PBT to total income', 'PAT to total income',
       'Cash profit to total income', 'PAT to net worth', 'Sales',
       'Income from financial services', 'Other income', 'Total capital',
       'Reserves and funds', 'Deposits (accepted by commercial banks)',
       'Borrowings', 'Current liabilities & provisions',
       'Deferred tax liability', 'Shareholders funds',
       'Cumulative retained profits', 'Capital employed', 'TOL/TNW',
       'Total term liabilities / tangible net worth',
       'Contingent liabilities / Net worth (%)', 'Contingent liabilities',
       'Net fixed assets', 'Investments', 'Current assets',
       'Net working capital', 'Quick ratio (times)', 'Current ratio (times)',
       'Debt to equity ratio (times)', 'Cash to current liabilities (times)',
       'Cash to average cost of sales per day', 'Creditors turnover',
       'Debtors turnover', 'Finished goods turnover', 'WIP turnover',
       'Raw material turnover', 'Shares outstanding', 'Equity face value',
       'EPS', 'Adjusted EPS', 'Total liabilities', 'PE on BSE'],
      dtype='object')
```

Replace all the symbols and extra spaces in the column header by using the below string replace function.

```
Default.columns = Default.columns.str.strip().str.replace(' ', '_').str.replace('(', '').str.replace(')', '').str.replace('%',
```

After the string replacement the column headers will be standardized.

```
Index(['Num', 'Networth_Next_Year', 'Total_assets', 'Net_worth',
       'Total_income', 'Change_in_stock', 'Total_expenses', 'Profit_after_tax',
       'PBDITA', 'PBT', 'Cash_profit', 'PBDITA_to_total_income',
       'PBT_to_total_income', 'PAT_to_total_income',
       'Cash_profit_to_total_income', 'PAT_to_net_worth', 'Sales',
       'Income_from_financial_services', 'Other_income', 'Total_capital',
       'Reserves_and_funds', 'Deposits_accepted_by_commercial_banks',
       'Borrowings', 'Current_liabilities_&_provisions',
       'Deferred_tax_liability', 'Shareholders_funds',
       'Cumulative_retained_profits', 'Capital_employed', 'TOL/TNW',
       'Total_term_liabilities_/_tangible_net_worth',
       'Contingent_liabilities_/_Net_worth_perc', 'Contingent_liabilities',
       'Net_fixed_assets', 'Investments', 'Current_assets',
       'Net_working_capital', 'Quick_ratio_times', 'Current_ratio_times',
       'Debt_to_equity_ratio_times', 'Cash_to_current_liabilities_times',
       'Cash_to_average_cost_of_sales_per_day', 'Creditors_turnover',
       'Debtors_turnover', 'Finished_goods_turnover', 'WIP_turnover',
       'Raw_material_turnover', 'Shares_outstanding', 'Equity_face_value',
       'EPS', 'Adjusted_EPS', 'Total_liabilities', 'PE_on_BSE'],
      dtype='object')
```

Exploratory Data Analysis (EDA):

Exploratory data analysis is an approach to analyzing data sets to summarize their main characteristics, often with visual methods. A statistical model can be used or not, but primarily EDA is for seeing what the data can tell us beyond the formal modeling or hypothesis testing task.

Shape of the dataset:

The dataset contains of 3541 rows and 52 columns

```
The number of rows (observations) is 3541
The number of columns (variables) is 52
```

The dataset is related to profit and loss account; hence the values of the dataset are all in numeric and the type of the data is in float64.

Statistical analysis of the dataset:

	Num	Networth_Next_Year	Total_assets	Net_worth	Total_income	Change_in_stock	Total_expenses	Profit_after_tax	PBDITA
count	3541.000000	3541.000000	3.541000e+03	3541.000000	3.343000e+03	3083.000000	3.402000e+03	3410.00000	3410.000000
mean	1772.451567	1616.256199	3.443382e+03	1295.862214	4.582823e+03	41.489264	4.262931e+03	277.36044	578.056305
std	1023.731393	17459.639673	3.097089e+04	13387.891867	5.568124e+04	440.573761	5.302869e+04	3064.40606	5653.770427
min	1.000000	-74265.600000	1.000000e-01	0.000000	0.000000e+00	-3029.400000	-1.000000e-01	-3908.30000	-440.700000
25%	886.000000	31.700000	9.130000e+01	31.300000	1.064500e+02	-1.800000	9.582500e+01	0.50000	6.900000
50%	1773.000000	116.300000	3.097000e+02	102.300000	4.449000e+02	1.600000	4.077000e+02	8.80000	35.400000
75%	2658.000000	456.100000	1.098700e+03	377.300000	1.440900e+03	18.050000	1.359775e+03	52.27500	150.250000
max	3545.000000	805773.400000	1.176509e+06	613151.600000	2.442828e+06	14185.500000	2.366035e+06	119439.10000	208576.500000

There are no duplicates row and columns in the dataset.

Missing Value:

There are missing values in the dataset that need to be treated.

Total_income	198
Change_in_stock	458
Total_expenses	139
Profit_after_tax	131
PBDITA	131
PBT	131
Cash_profit	131
PBDITA_to_total_income	68
PBT_to_total_income	68
PAT_to_total_income	68
Cash_profit_to_total_income	68
PAT_to_net_worth	0
Sales	259
Income_from_financial_services	935
Other_income	1295
Total_capital	4
Reserves_and_funds	85
Deposits_accepted_by_commercial_banks	3541
Borrowings	366
Current_liabilities_&_provisions	96
Deferred_tax_liability	1140
Shareholders_funds	0
Cumulative_retained_profits	38

```
Contingent_liabilities                      1188
Net_fixed_assets                             118
Investments                                 1435
Current_assets                                66
Net_working_capital                           32
Quick_ratio_times                             93
Current_ratio_times                           93
Debt_to_equity_ratio_times                     0
Cash_to_current_liabilities_times             93
Cash_to_average_cost_of_sales_per_day         85
Creditors_turnover                           333
Debtors_turnover                             328
Finished_goods_turnover                      740
WIP_turnover                                 640
Raw_material_turnover                        361
Shares_outstanding                           692
Equity_face_value                            692
EPS                                            0
Adjusted_EPS                                   0
Total_liabilities                              0
PE_on_BSE                                   2194
```

1.2 Missing Value Treatment:

In this data, the column header "Deposits accepted by commercial bank" is having zero values. Hence, we can drop the column here as the null values is 3541. In this whole dataset we have 18,533 missing values.

The treatment of the missing values are done by using the median of each column values.

To do the correction we need to import the library from SKLearn.

```
from sklearn.impute import SimpleImputer
imputer = SimpleImputer(missing_values=np.nan, strategy='median')

Default = pd.DataFrame(imputer.fit_transform(Default))
Default.columns=col
Default.head()
```

	Num	Networth_Next_Year	Total_assets	Net_worth	Total_income	Change_in_stock	Total_expenses	Profit_after_tax	PBDITA	PBT	.
0	1.0	8890.6	17512.3	7093.2	24965.2	235.8	23657.8	1543.2	2860.2	2417.2	.
1	2.0	394.3	941.0	351.5	1527.4	42.7	1454.9	115.2	283.0	188.4	.
2	3.0	92.2	232.8	100.6	477.3	-5.2	478.7	-6.6	5.8	-6.6	.
3	4.0	2.7	2.7	2.7	444.9	1.6	407.7	8.8	35.4	12.4	.
4	5.0	109.0	478.5	107.6	1580.5	-17.0	1558.0	5.5	31.0	6.3	.

5 rows × 51 columns

1.3Transform target variable into 0 and 1:

Here the target variable is considered as "Networth_next _year" as 0 and 1as our dataset has not specified any defaulter. As we are retrieving the data from a financial statement. If the credit worthiness of the company is positive, then the income generated from sales after deducting all the expenses then the credit risk of the company is 0. When the value is negative the credit risk is 1. Those net worth greater than 0 will be considered as non-defaulters and those less than zero are considered as defaulters.

```
Default['default'] = np.where((Default['Networth_Next_Year'] > 0), 0, 1)
```

We have got 3298 non defaulters and 243 defaulters

We could identify that 6.8% chances are there for the defaulters to be in risk.

The statistical description of the default variable is as below:

```
count    3541.000000
mean        0.068625
std         0.252851
min         0.000000
25%         0.000000
50%         0.000000
75%         0.000000
max         1.000000
```

The mean of the new variable is 0.068 where the standard deviation is 0.025.

Profit before tax on total income is the main factor that will derive if the company has the capability to make their corporate tax. This is also considered as the company earnings.

```
count     3541.000000
mean       -16.883635
std        421.337332
min     -21340.000000
25%          0.600000
50%          3.310000
75%          8.630000
max        100.000000
```

Before considering the earnings of defaulter and non-defaulters will check the statistical analysis of the PBT values in the financial statement. The mean of the PBT is showing as -16.88. we can consider this as we have paid more interest to the loans that we gained out of the investment.

Checking on the summary of non-defaulters: -

```
count    3298.000000
mean       -3.855588
std       196.861553
min     -9700.000000
25%         0.982500
50%         3.580000
75%         9.172500
max       100.000000
Name: PBT_to_total_income, dtype: float64
```

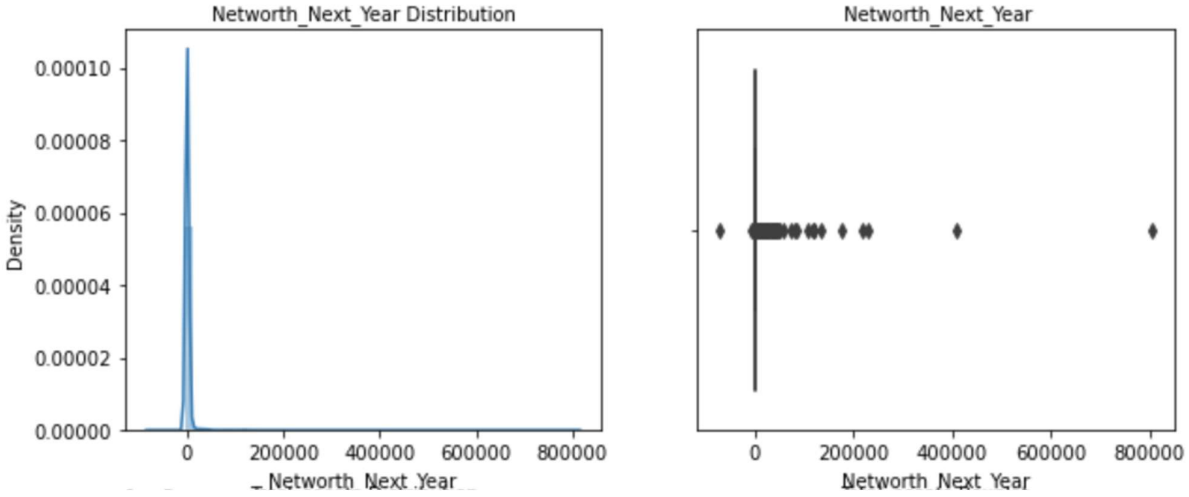
Checking the summary of defaulters: -

```
count      243.000000
mean      -193.700494
std       1426.590111
min     -21340.000000
25%        -31.705000
50%         -5.090000
75%          0.000000
max         92.380000
Name: PBT_to_total_income, dtype: float64
```

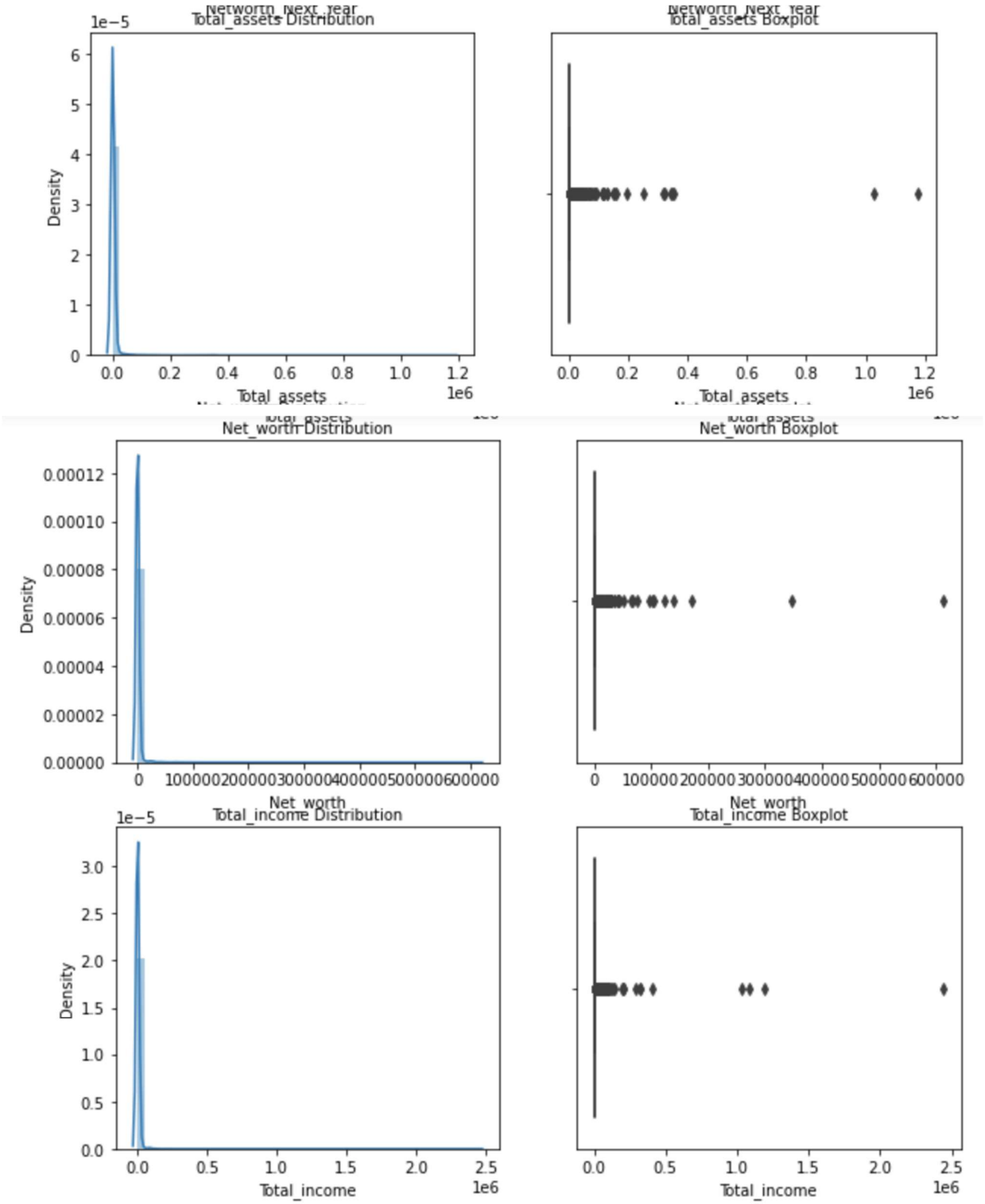
1.4 Univariate & bivariate analysis (including Heatmap)

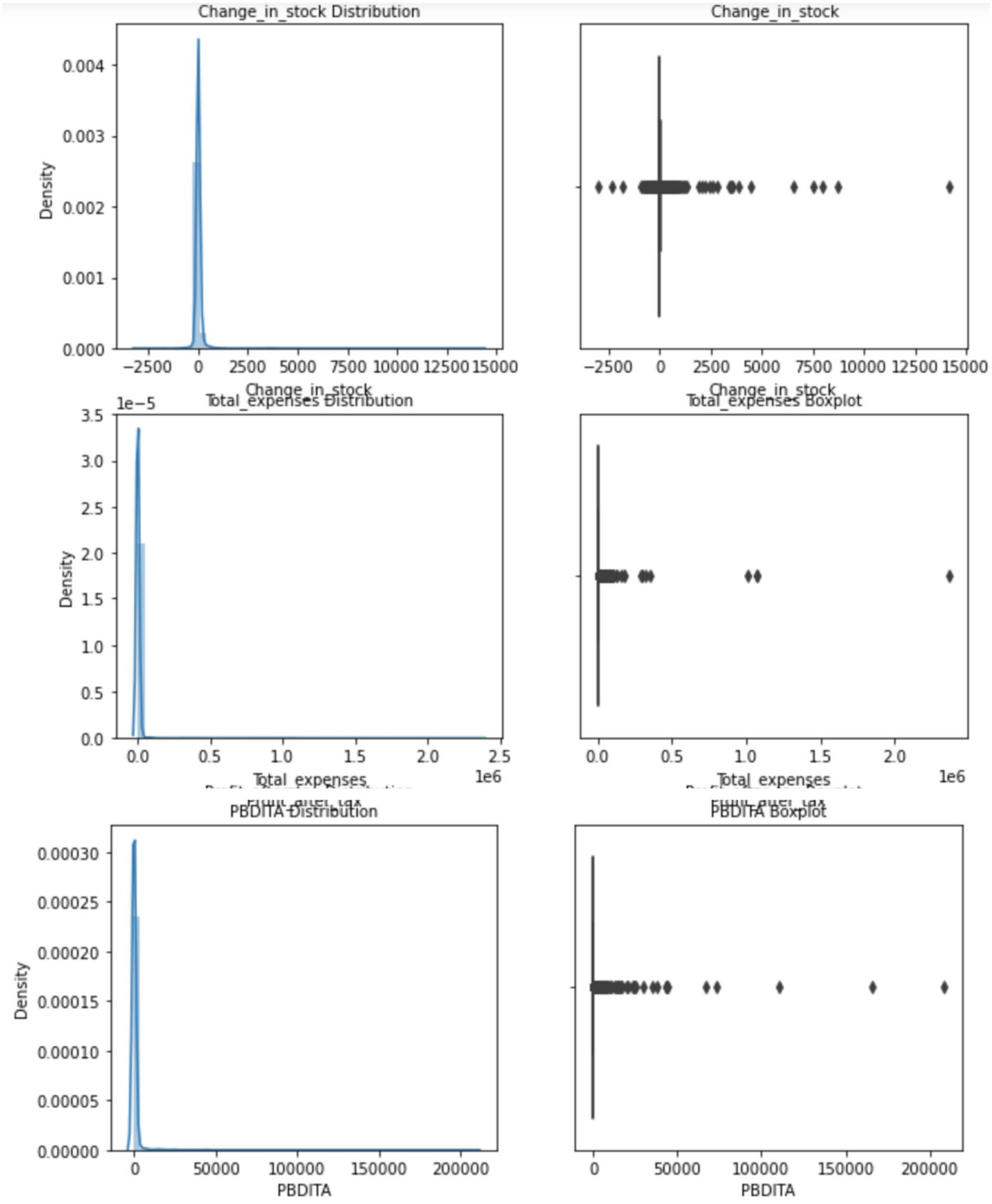
Outliers:

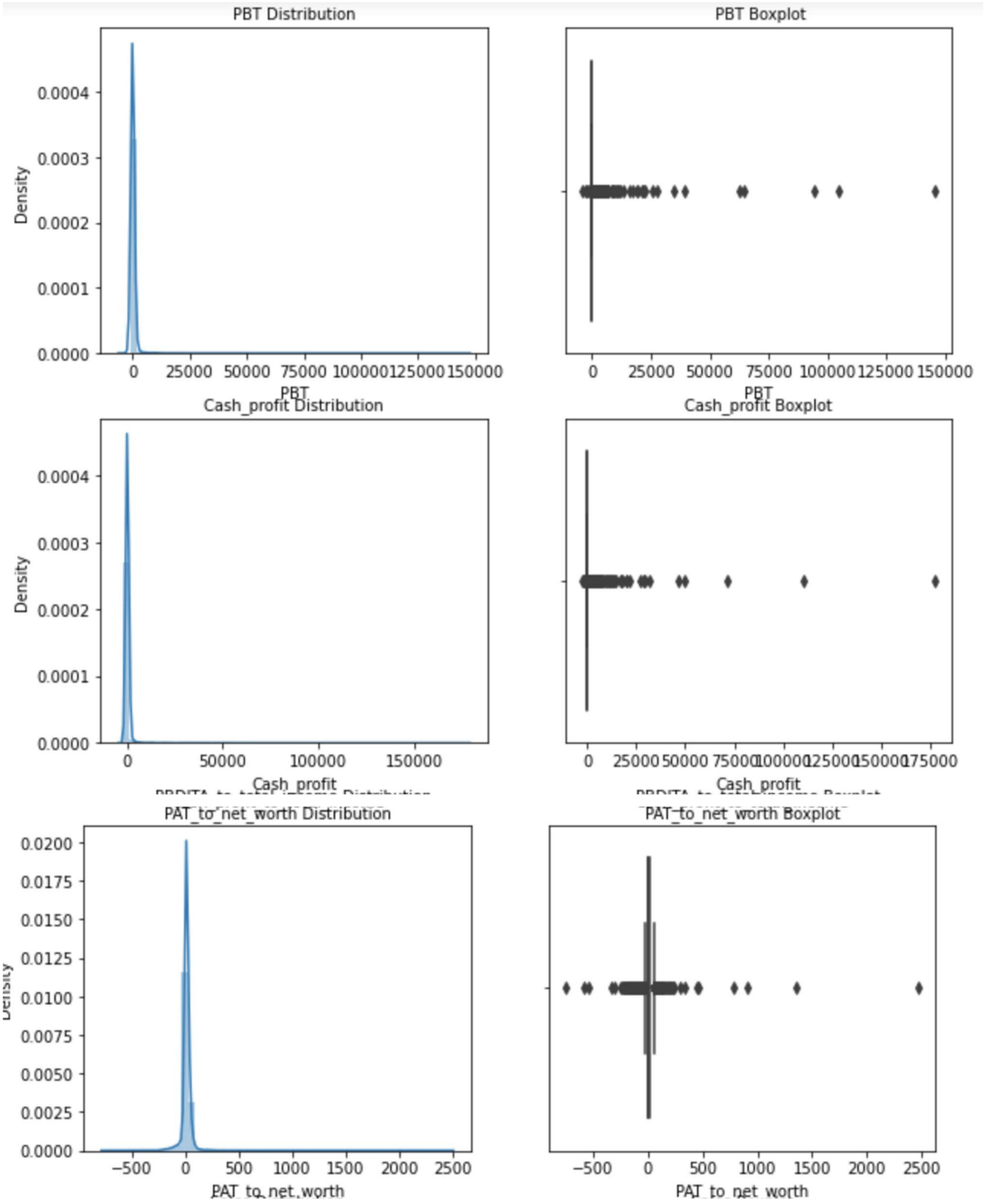
We could find there are visible outliers in the dataset. The analysis of the outliers is as below:

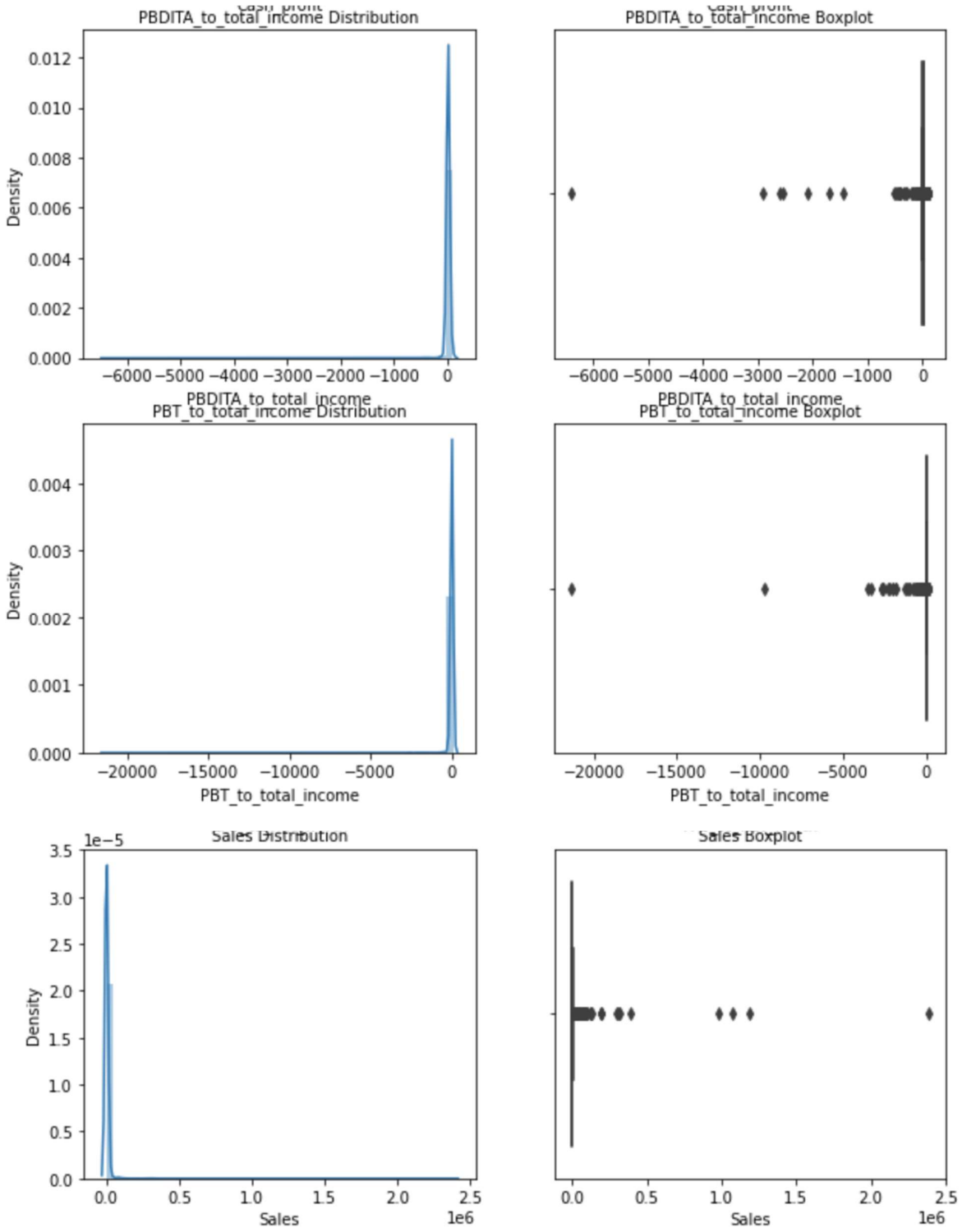


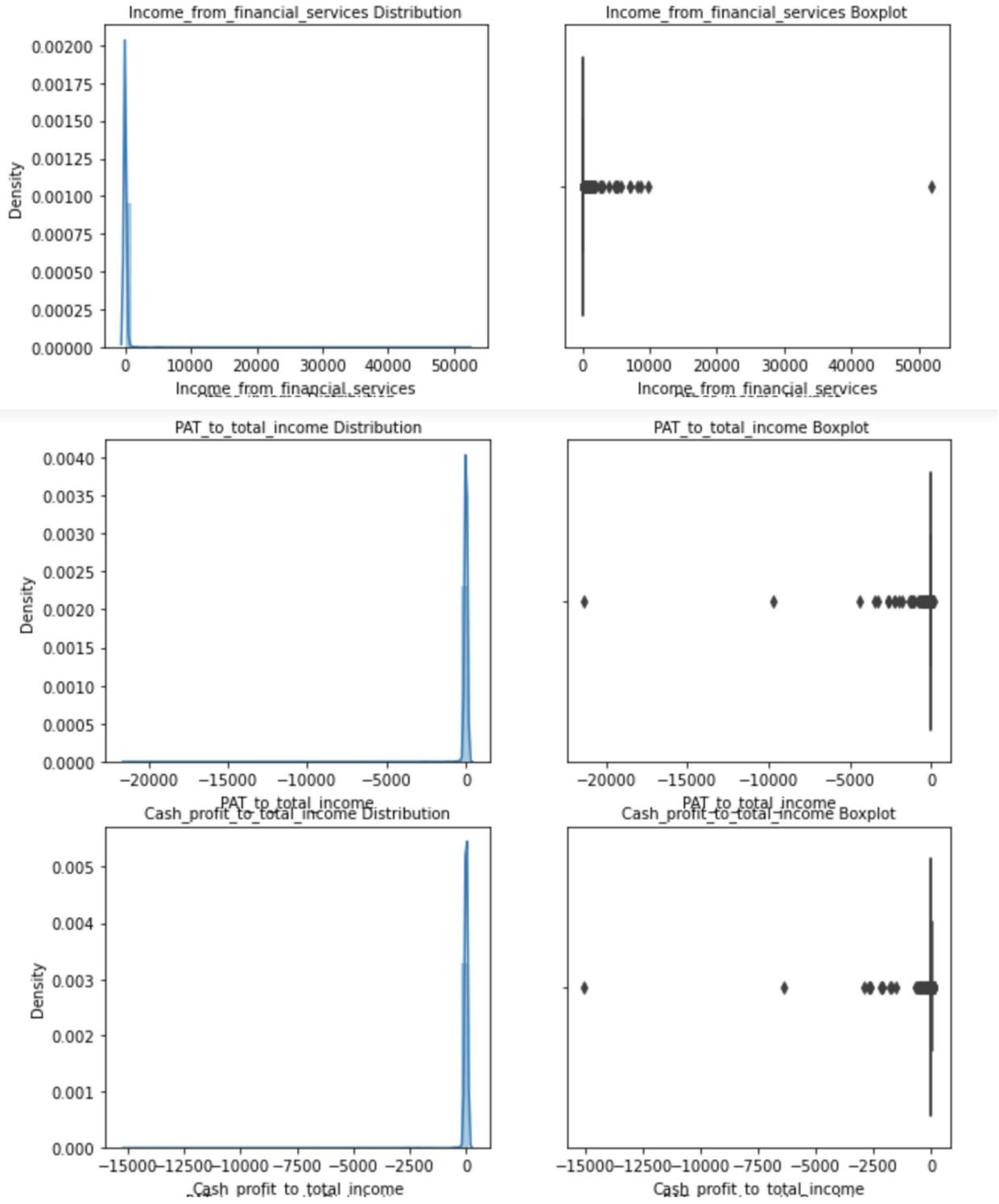
The credit worthiness of the company is having outliers

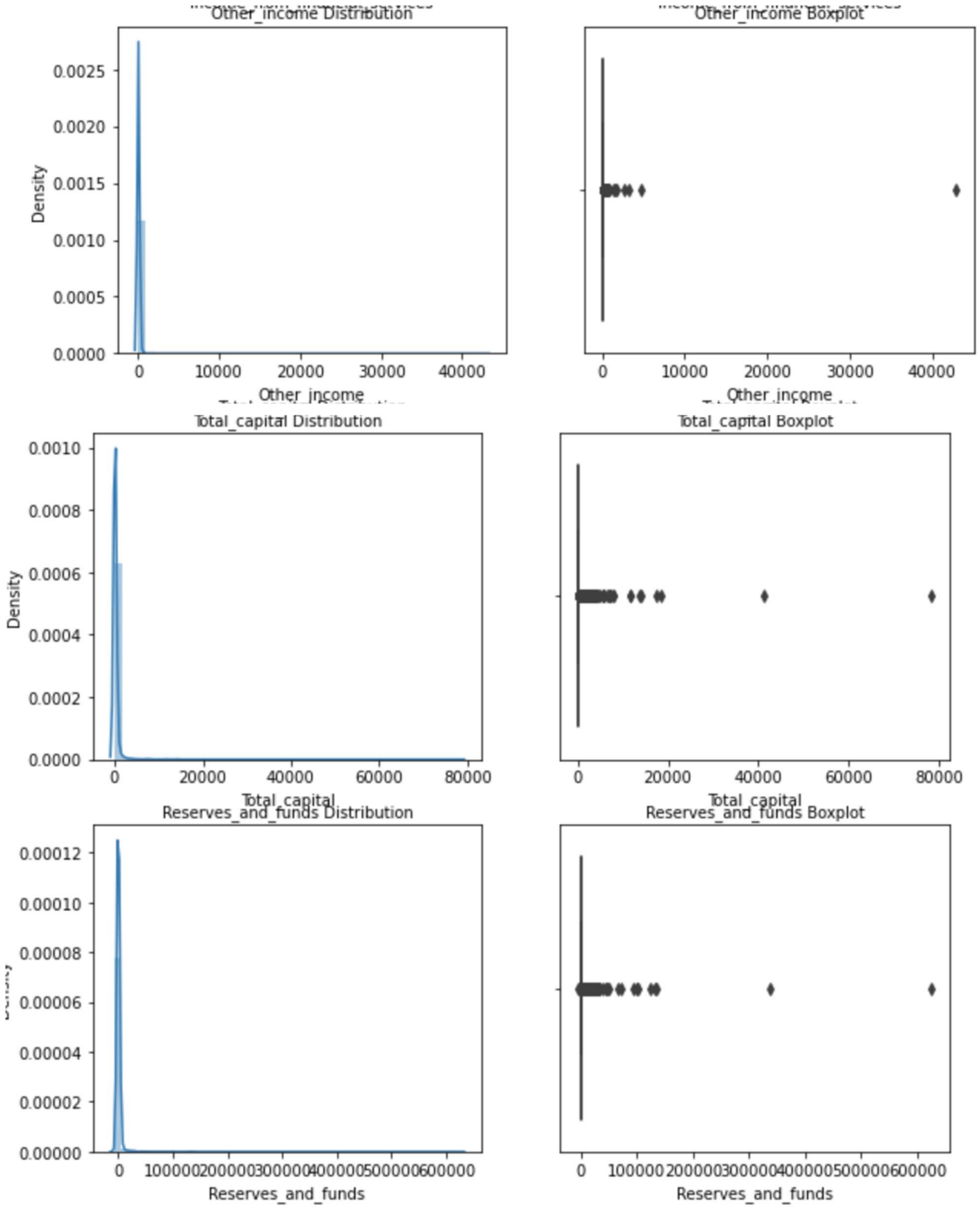


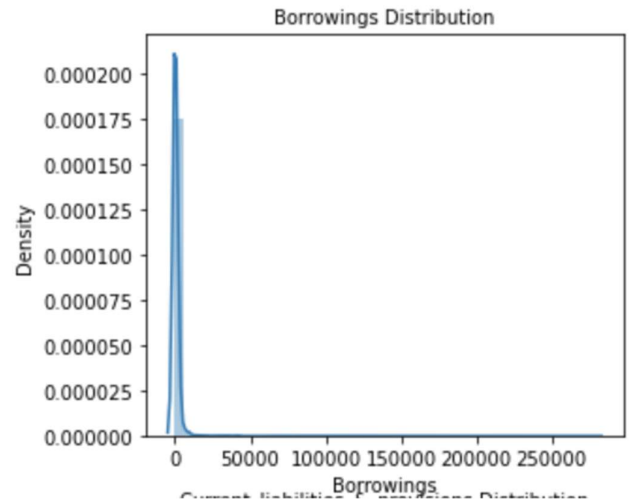


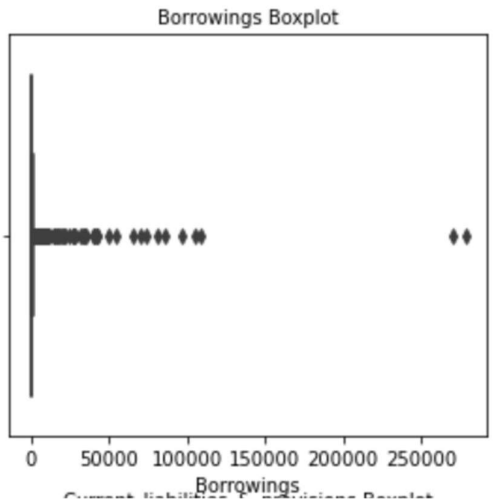


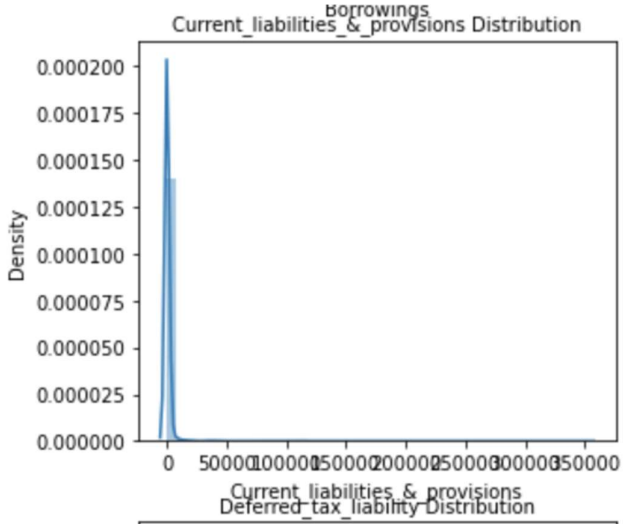


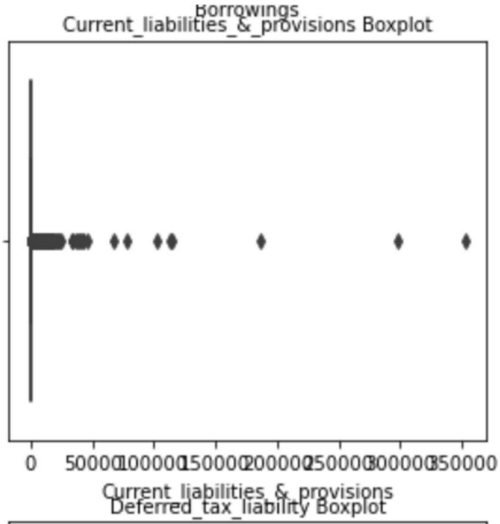


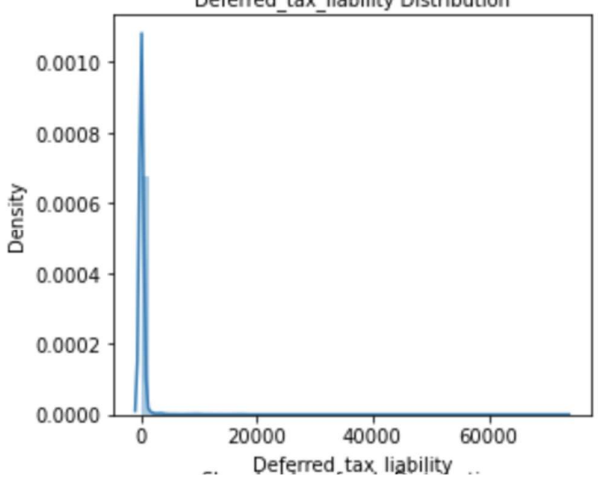


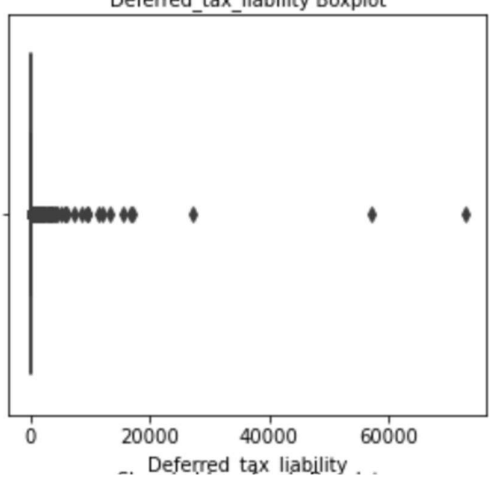


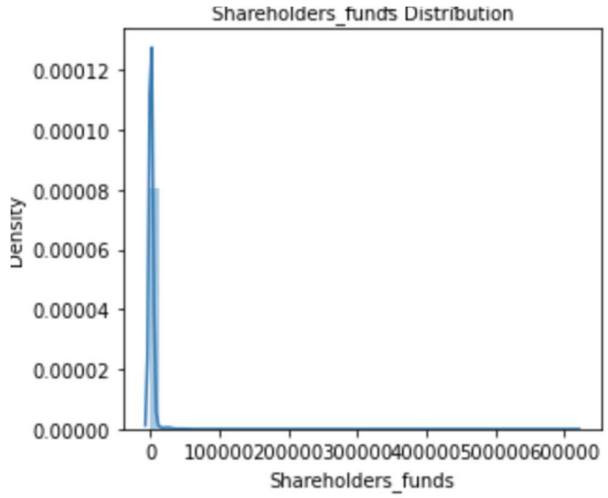


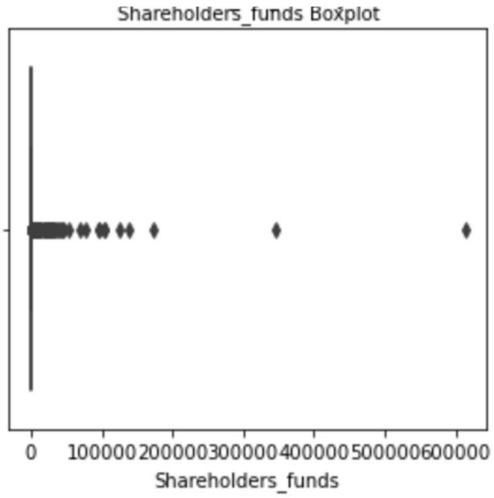


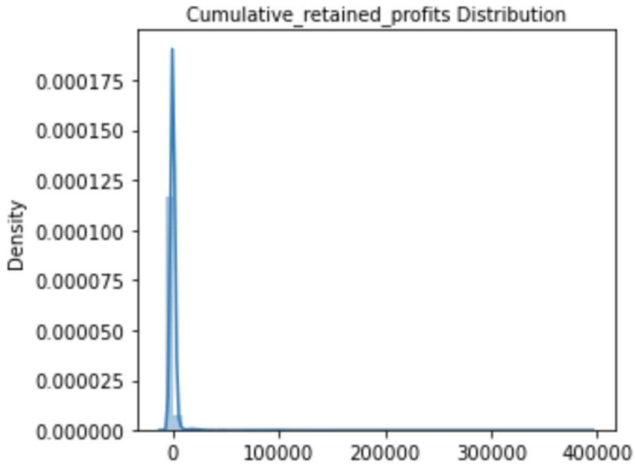


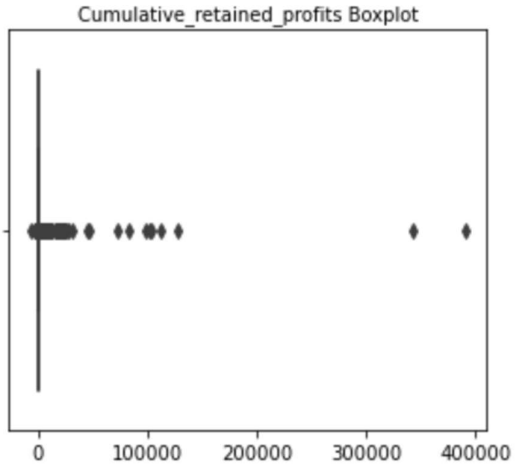


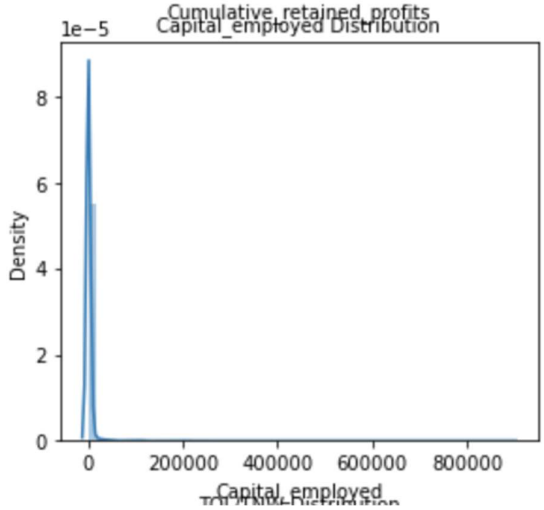


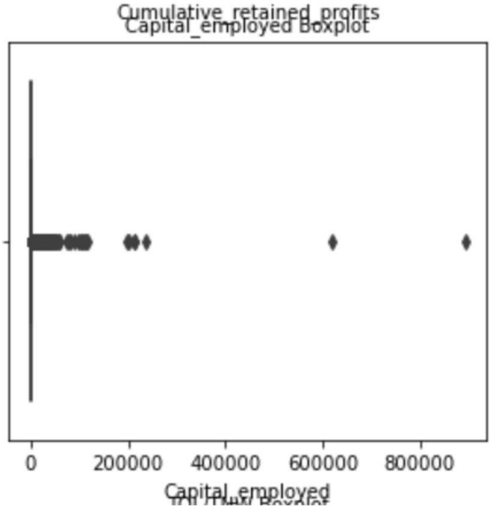


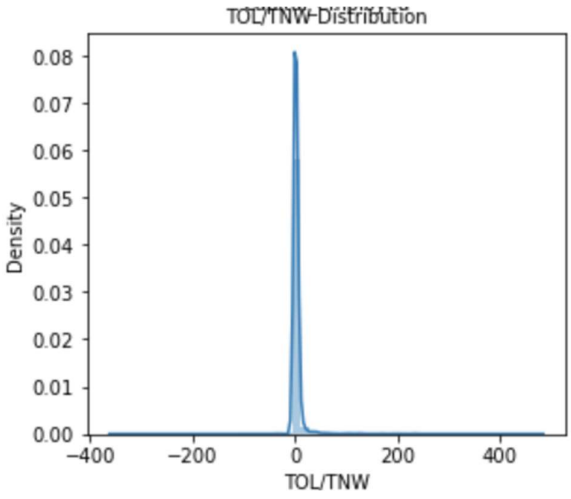


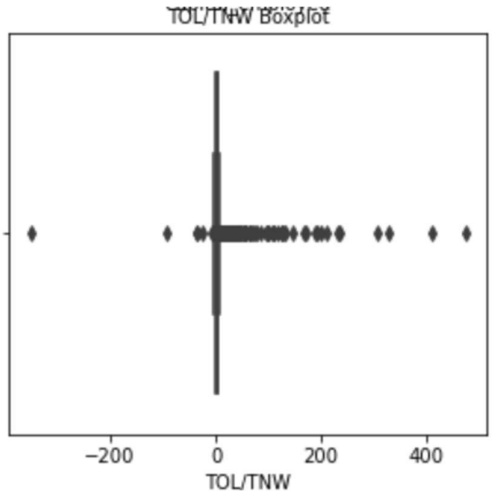


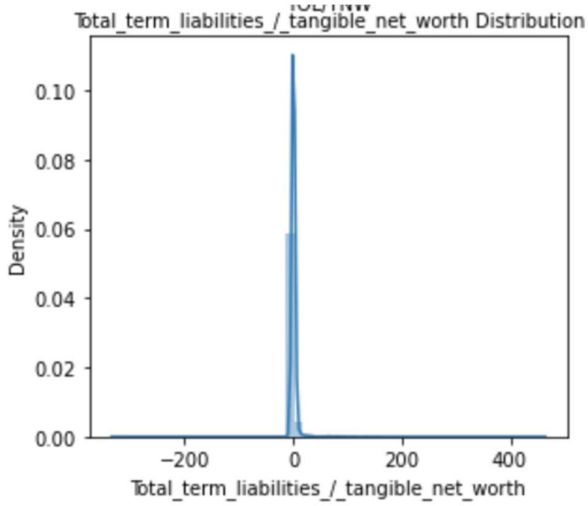


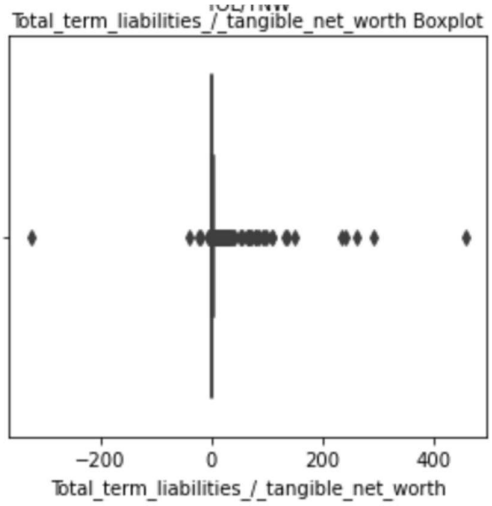


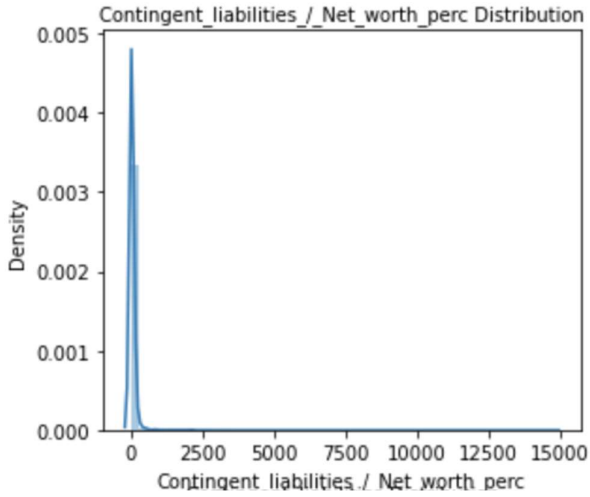


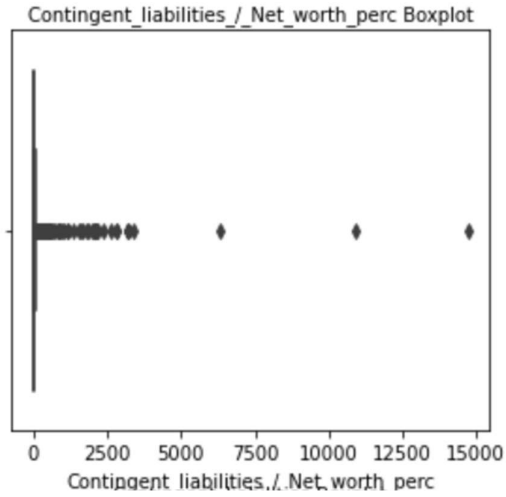


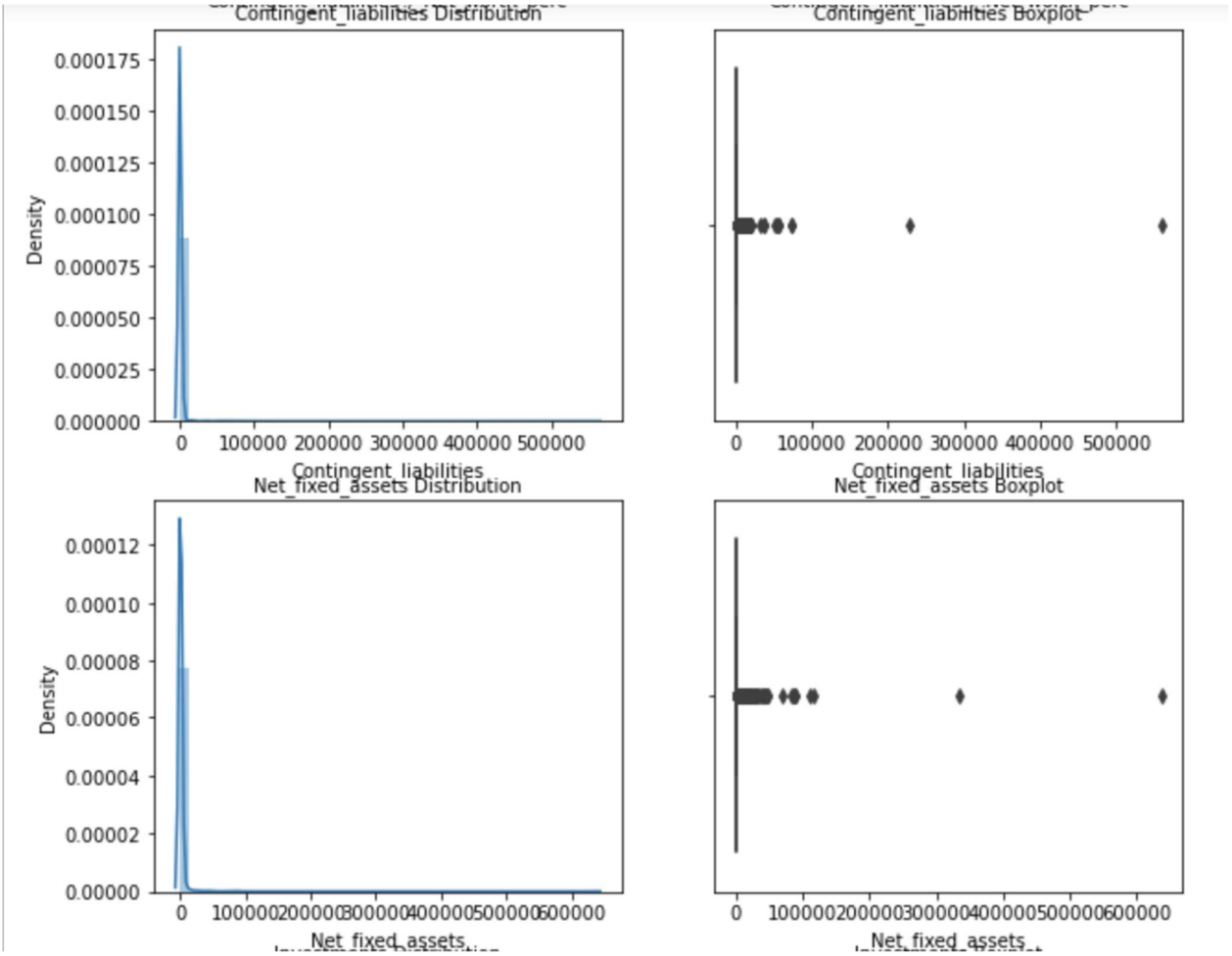


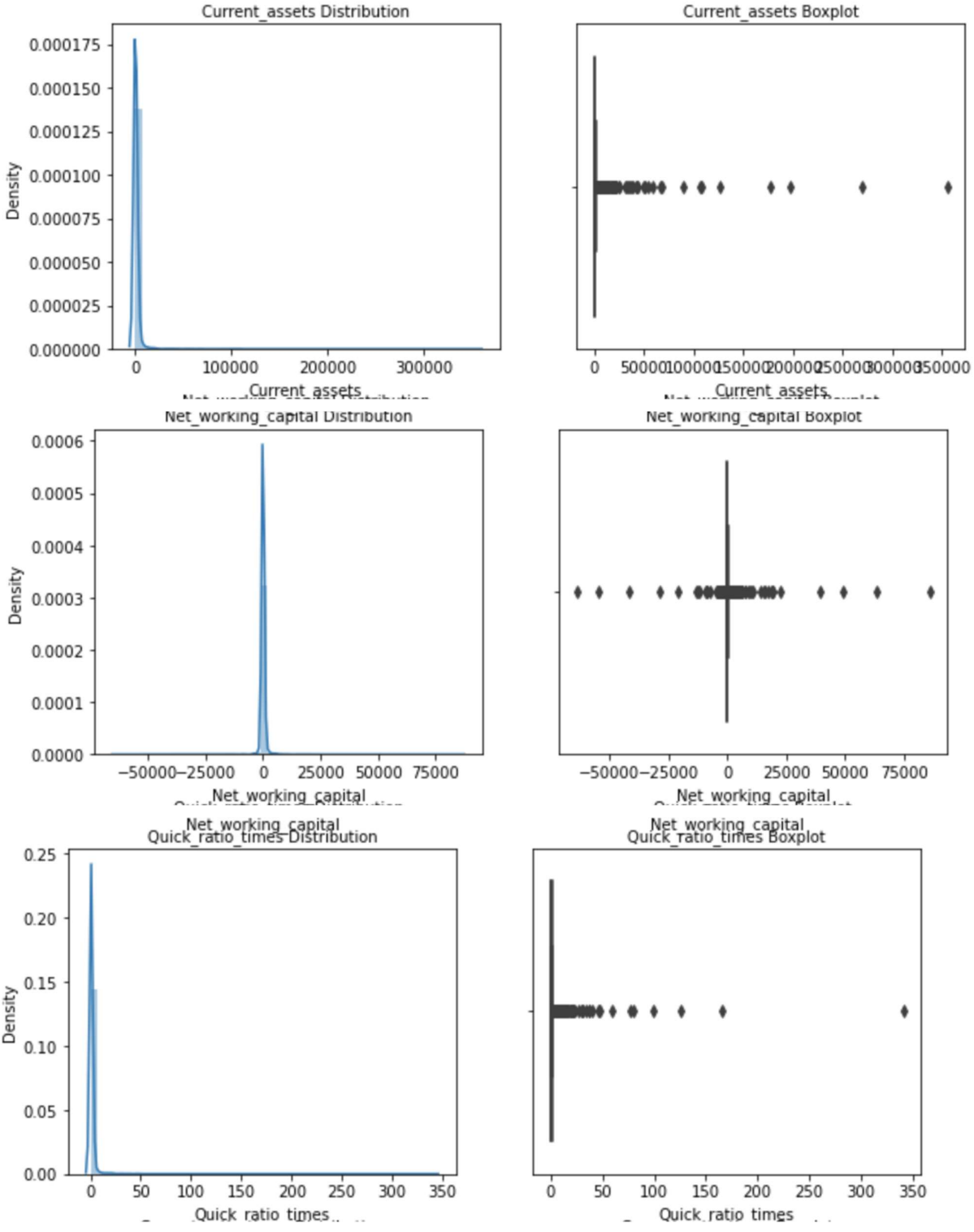


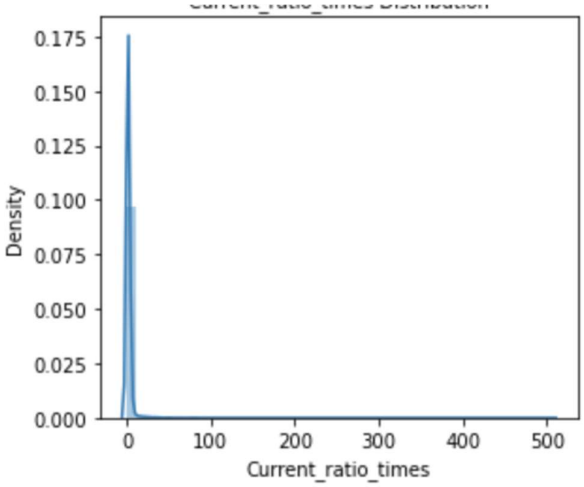


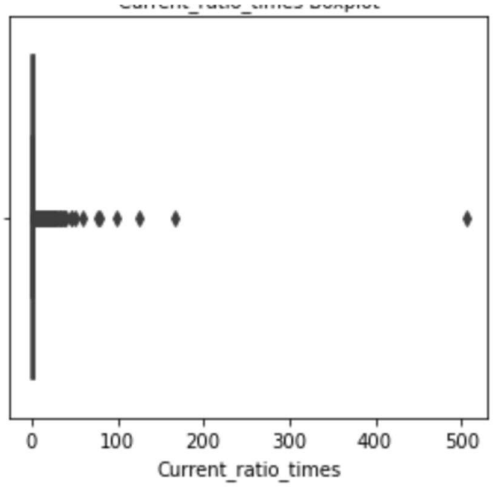


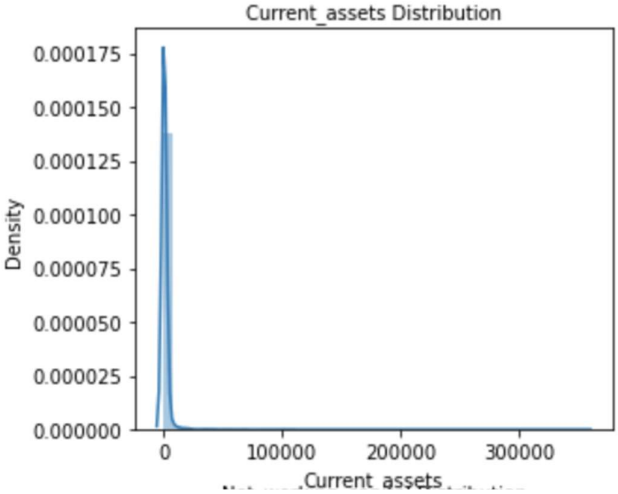


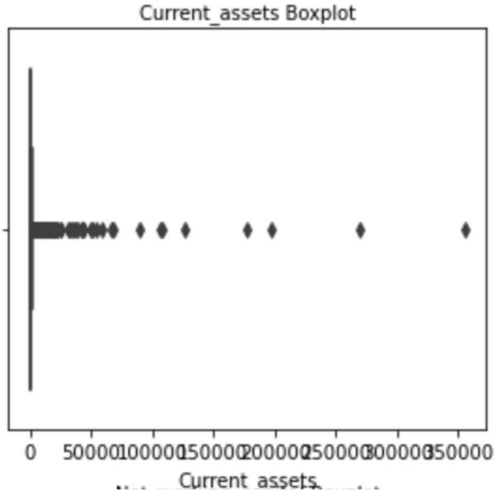


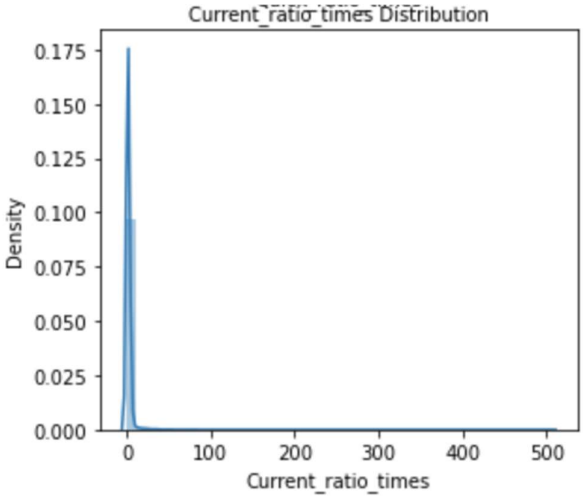


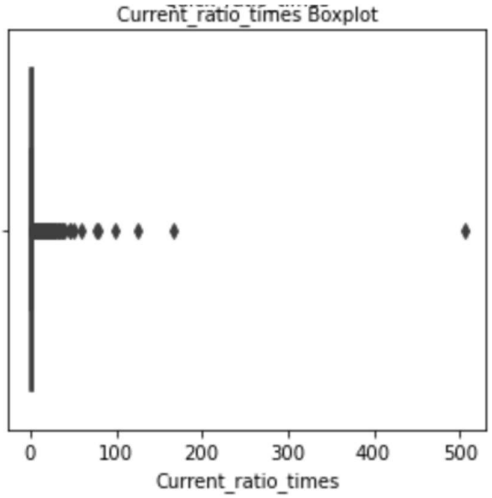


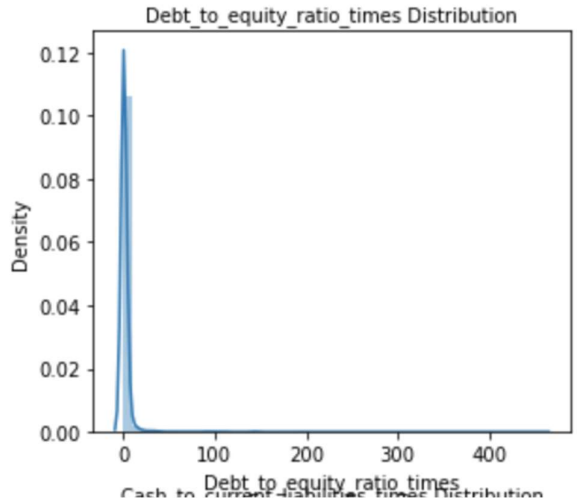


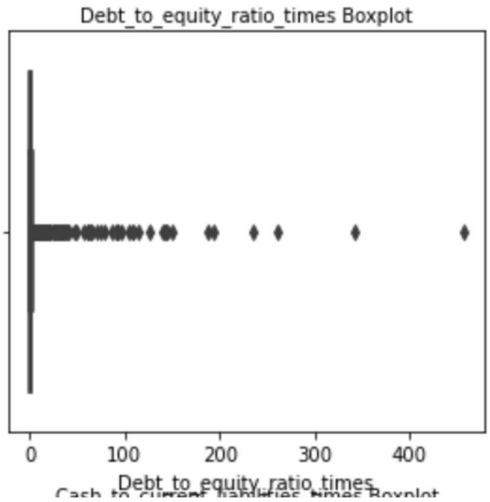


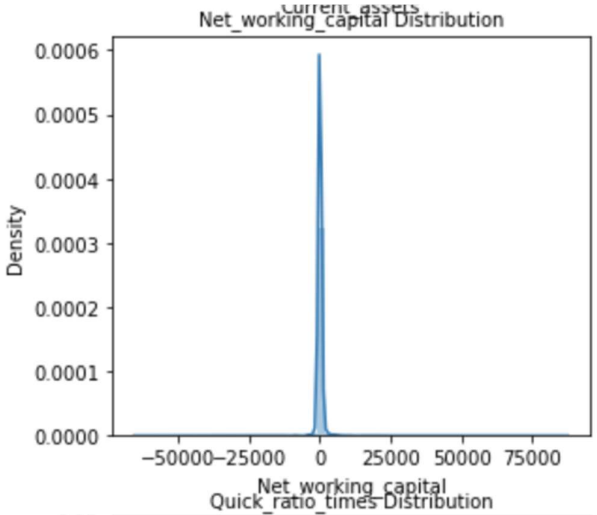


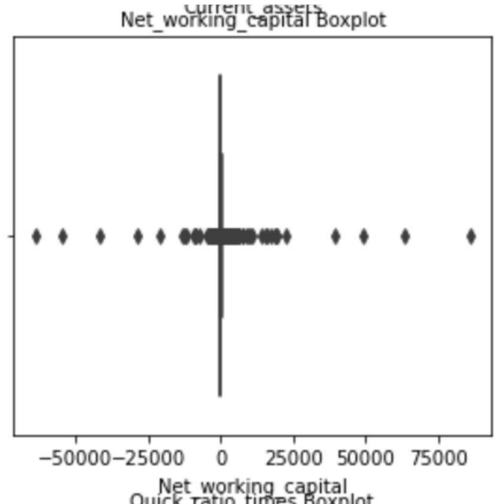


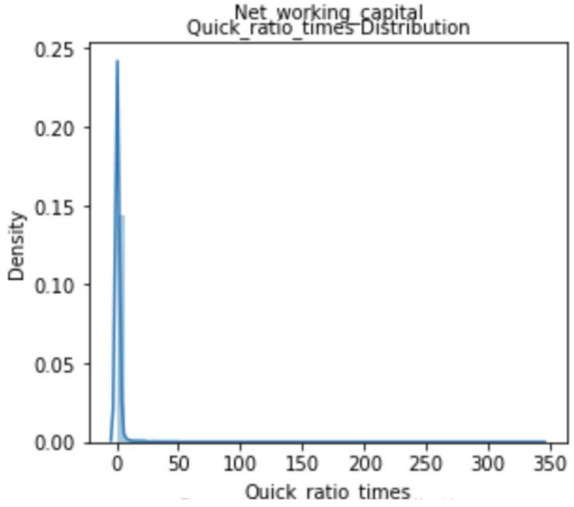


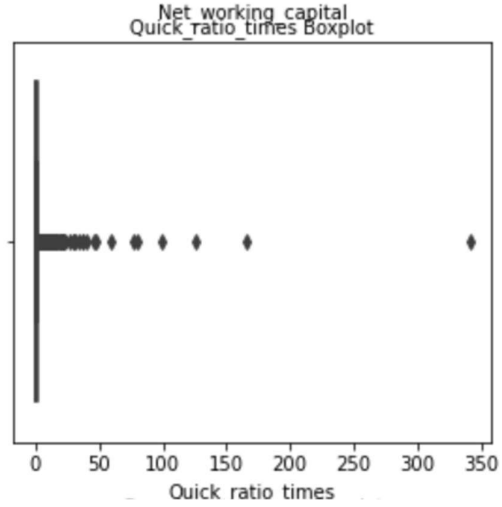


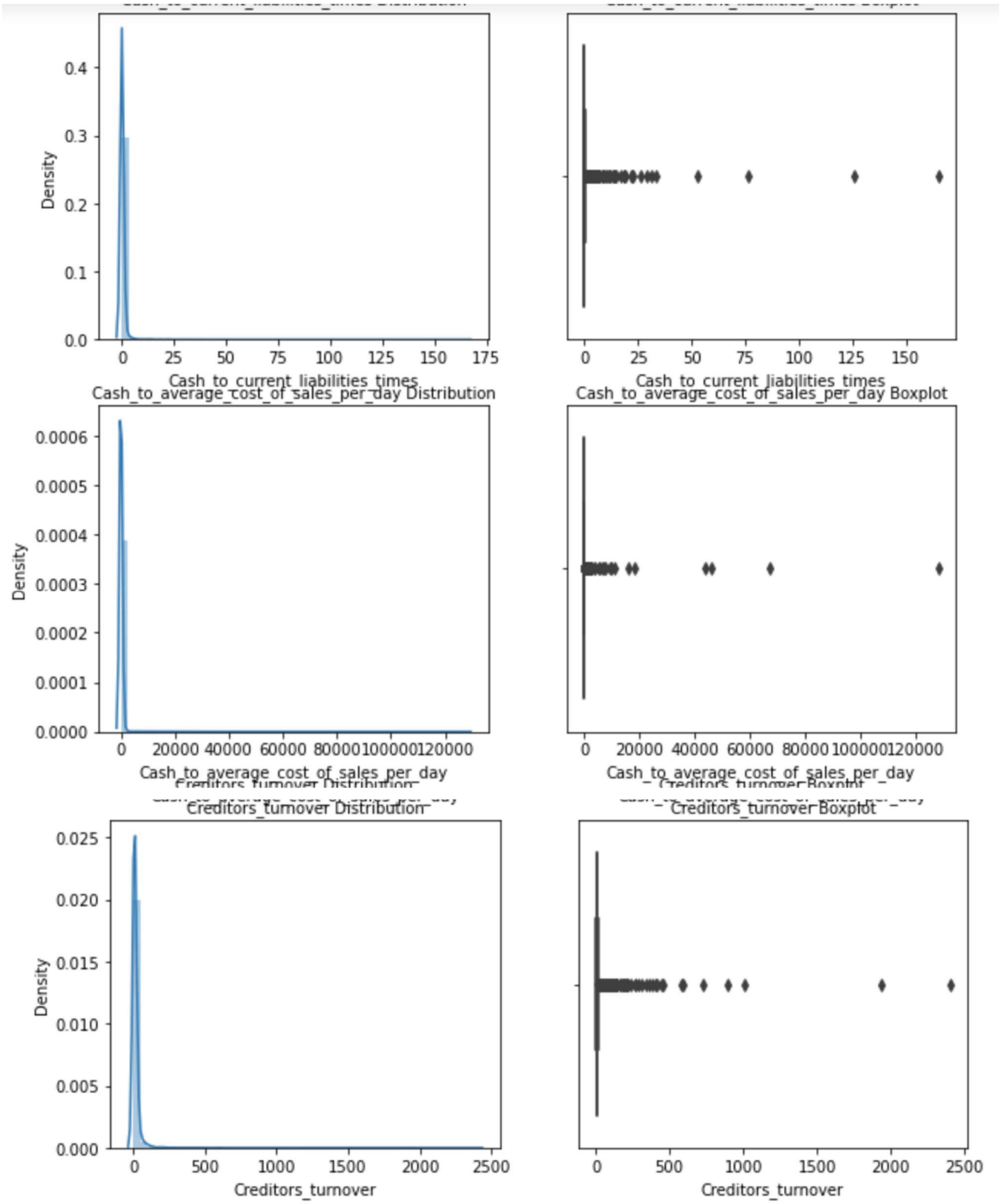


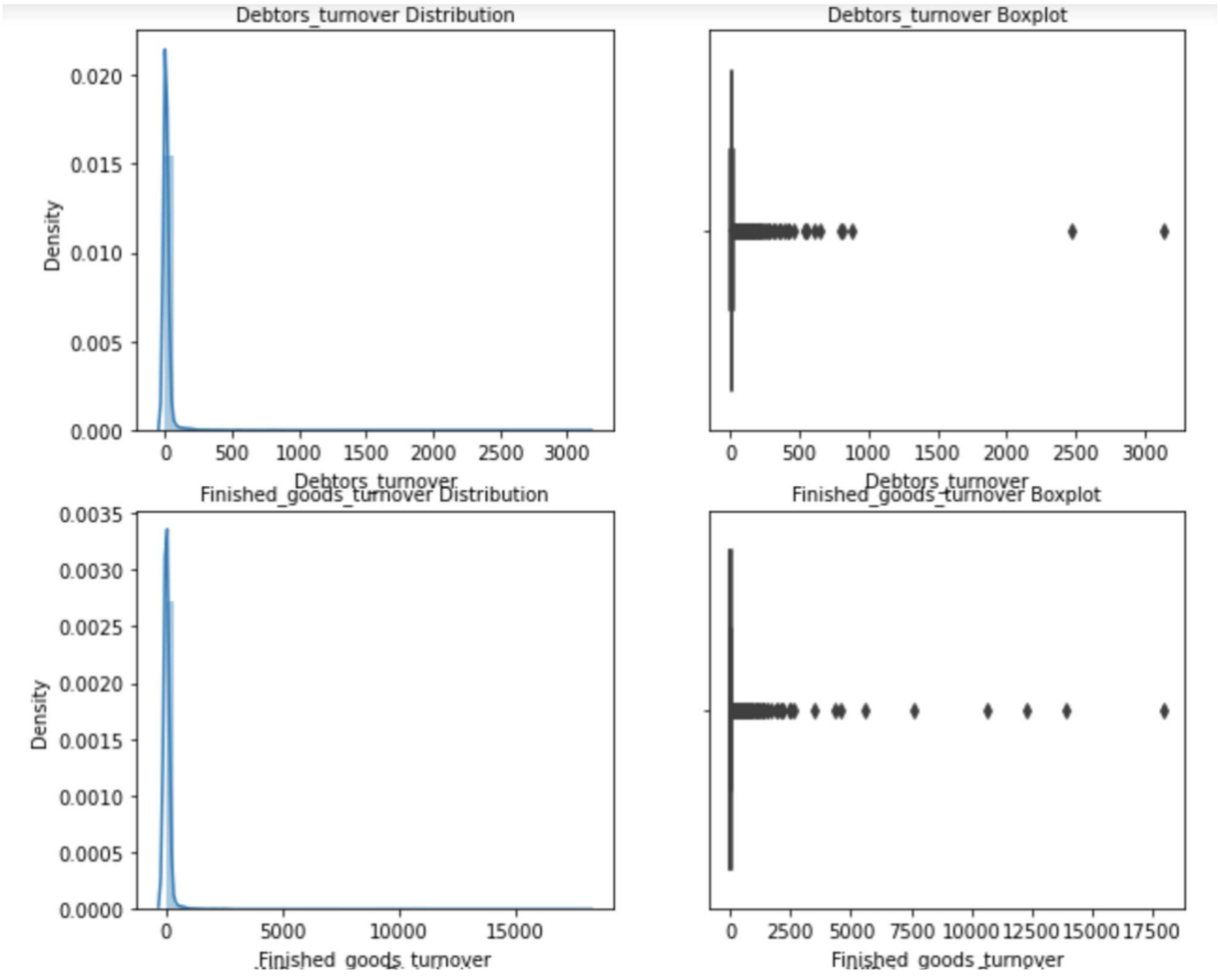


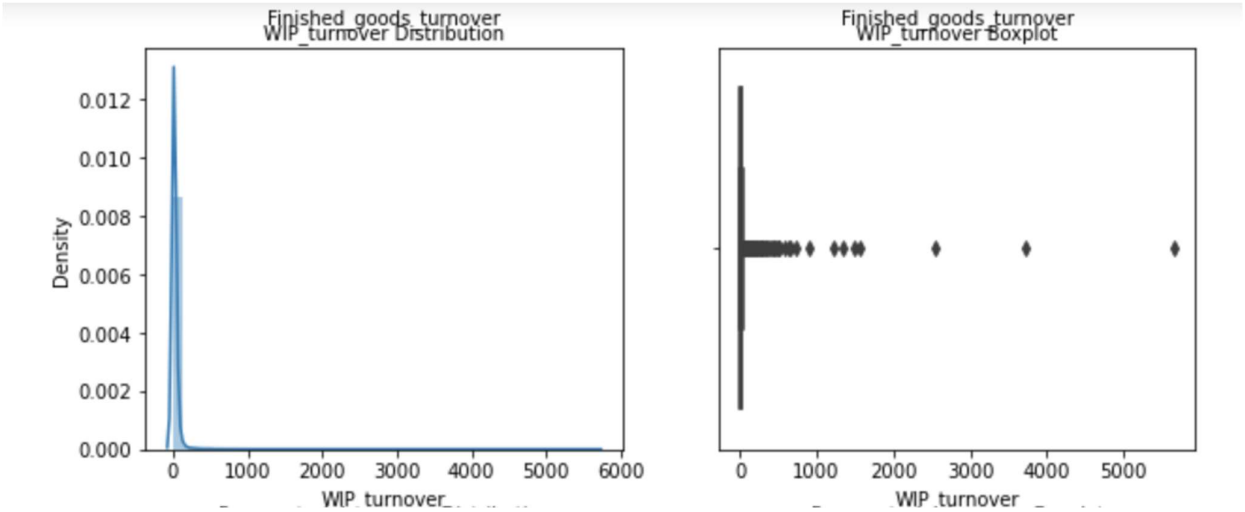


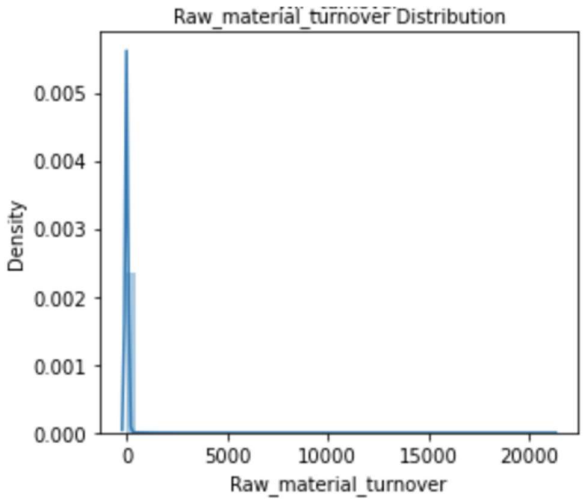


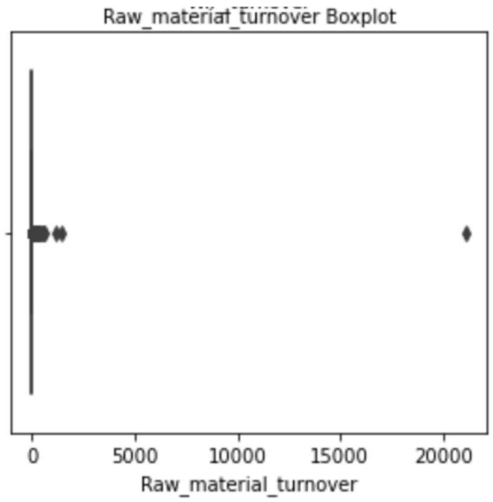


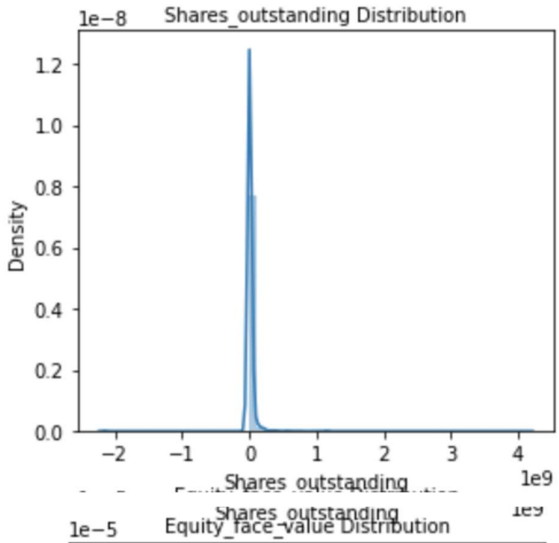


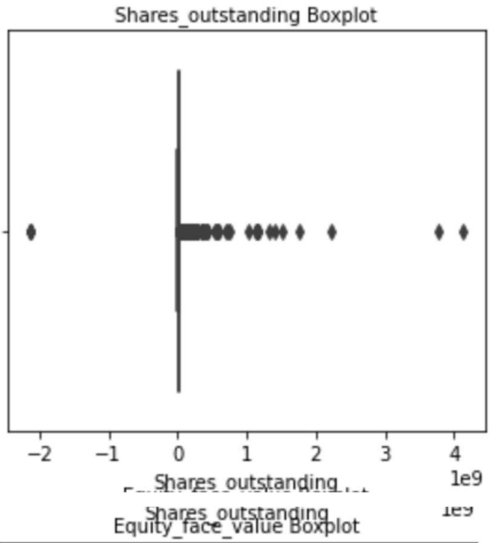


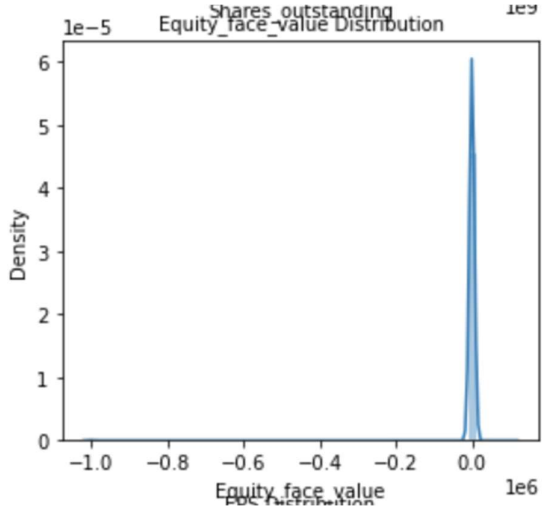


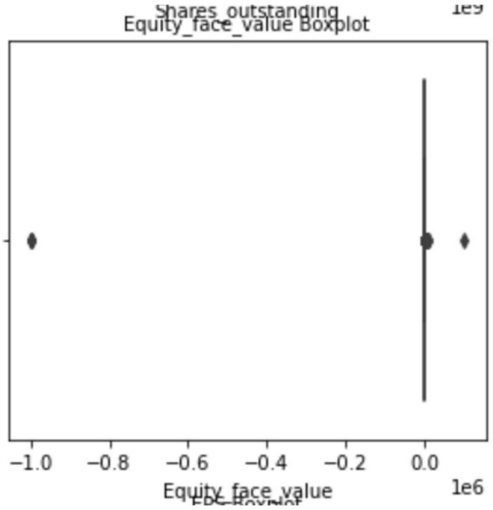


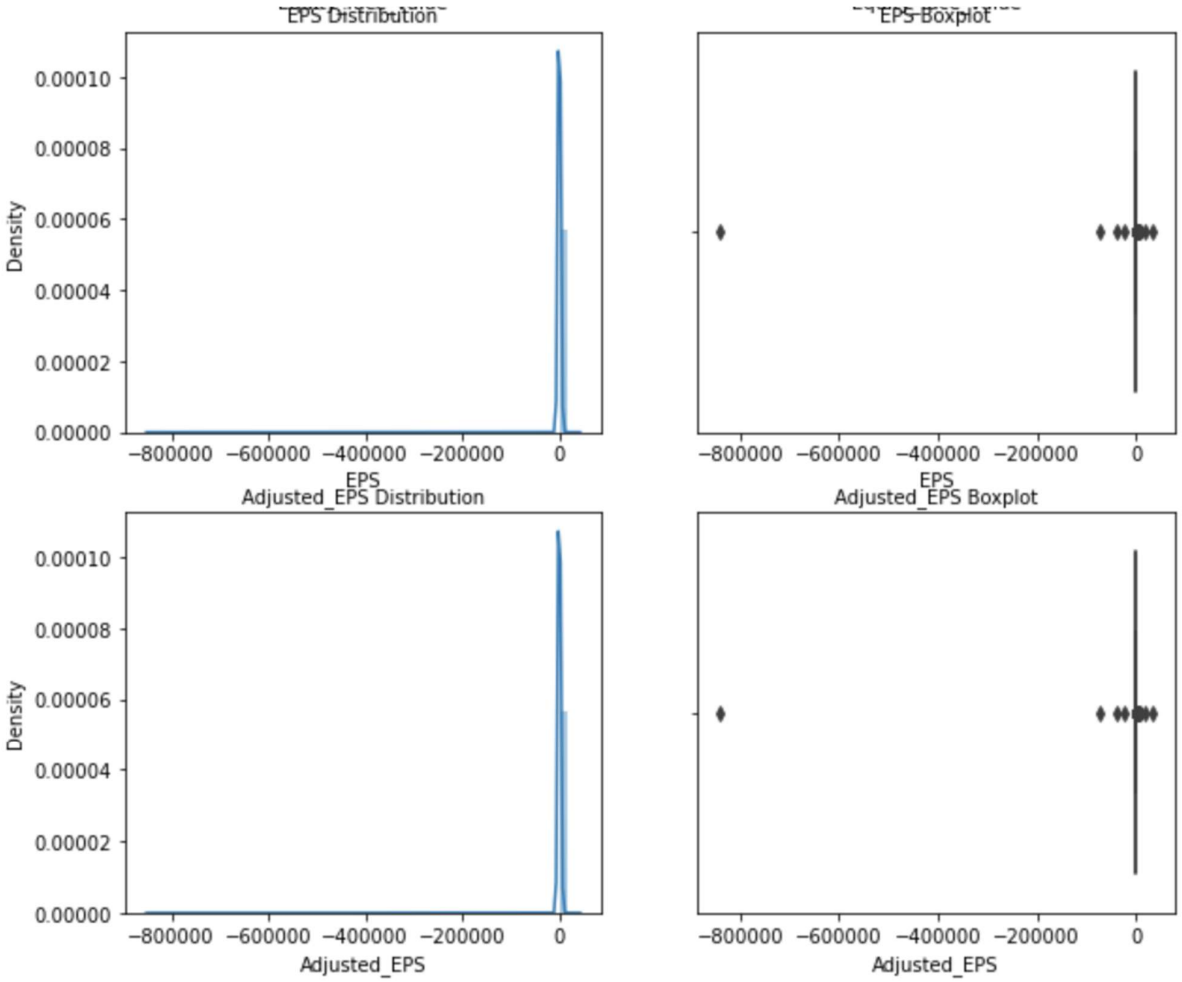


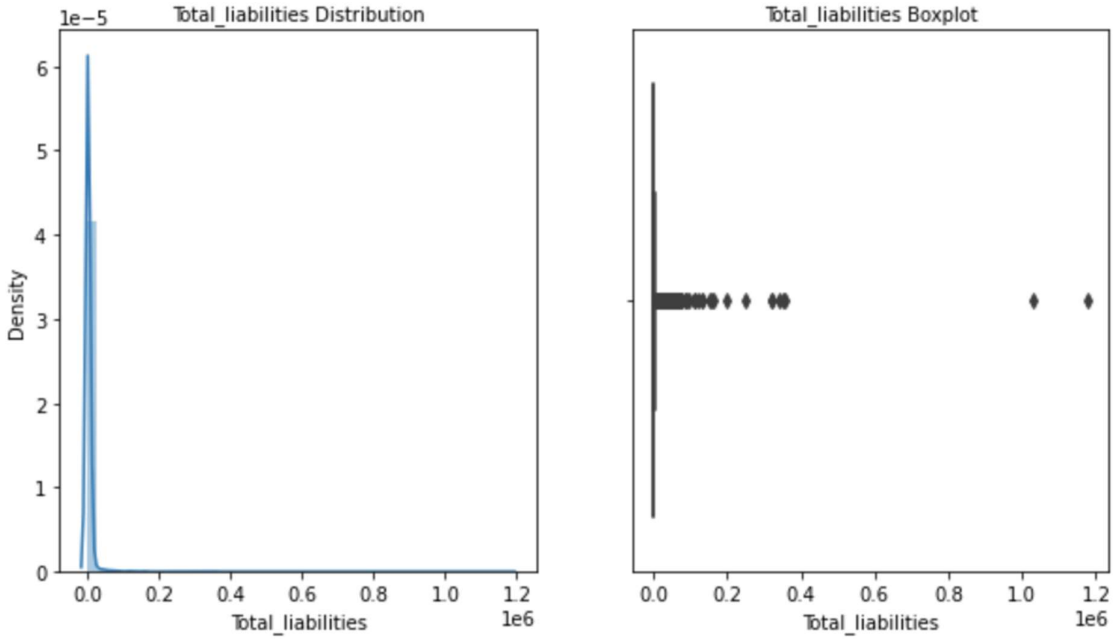


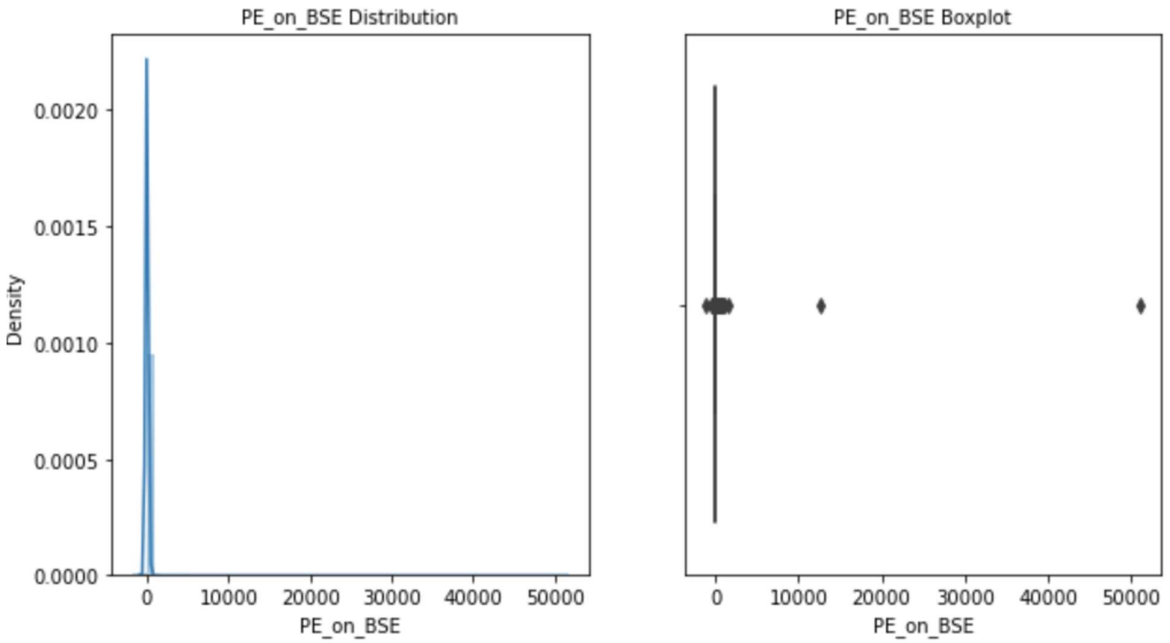












We could find that all the variables are having outliers in the data that need to be treated.

Correlation: The heatmap helps in identifying the correlation of the each variable with the other variables. Darker the color in heat map the higher the correlation and lighter the color of the heat map shows the lower the correlation. In our correlation map we can identify that there are certain variables like debt equity ratio has negatively correlated with total income, PBTIA total income, PBT to total income, PBT to net worth and adjusted EPS.

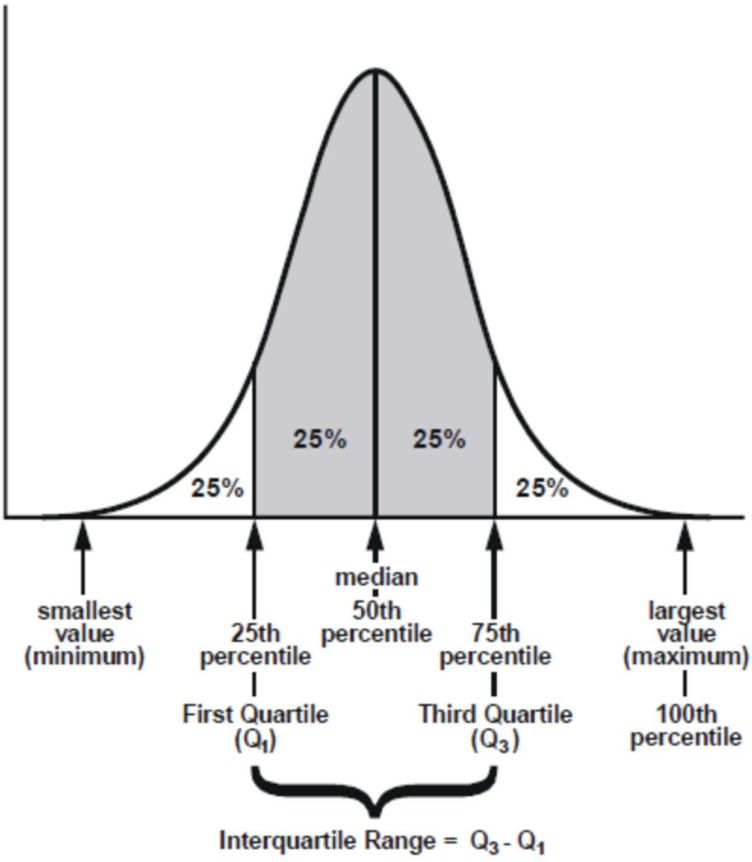
1.1 Outlier Treatment:

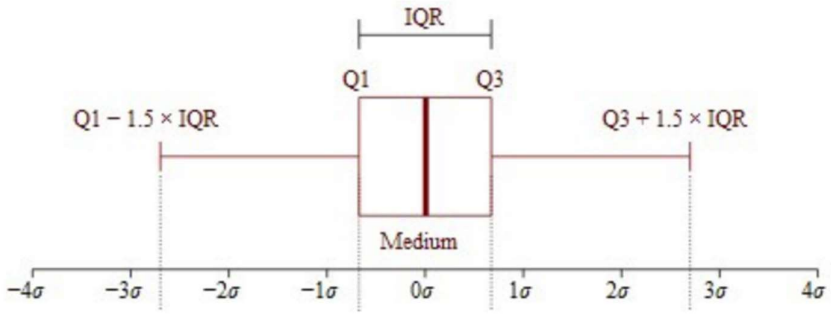
```
def remove_outlier(col):
    sorted(col)
    Q1,Q3=np.percentile(col,[25,75])
    IQR=Q3-Q1
    lower_range= Q1-(1.5 * IQR)
    upper_range= Q3+(1.5 * IQR)
    return lower_range, upper_range
```

```
for column in Default.iloc[:, 2:4].columns:
    lr,ur=remove_outlier(Default[column])
    Default[column]=np.where(Default[column]>ur,ur,Default[column])
    Default[column]=np.where(Default[column]<lr,lr,Default[column])
```

The treatment of the outlier treatment are done on extreme value analysis.

In case of Gaussian distribution, the outliers will lie outside the mean plus or minus 3 times the standard deviation of the variable. Box plot generally helps us in identifying the outliers present in the dataset.

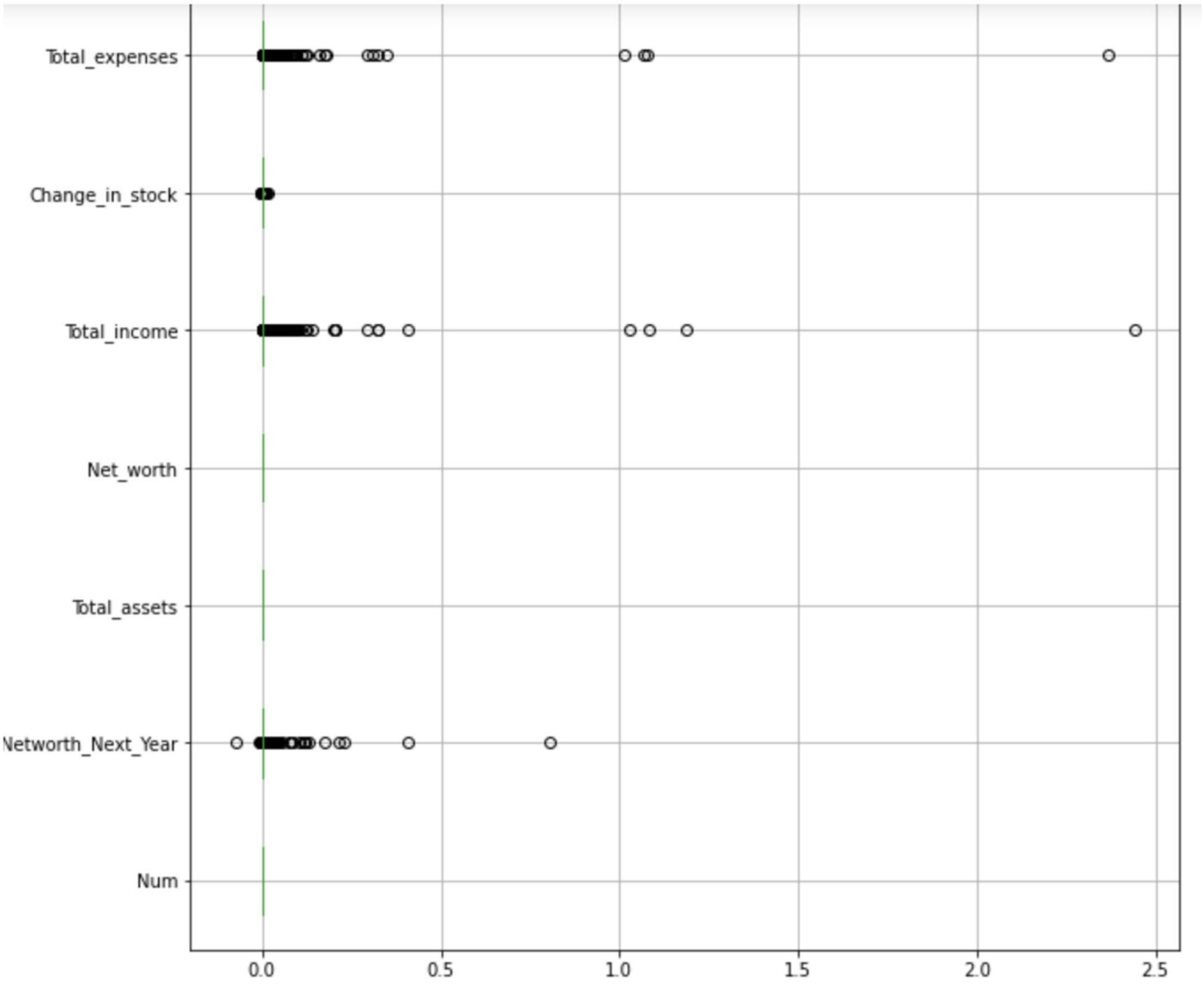




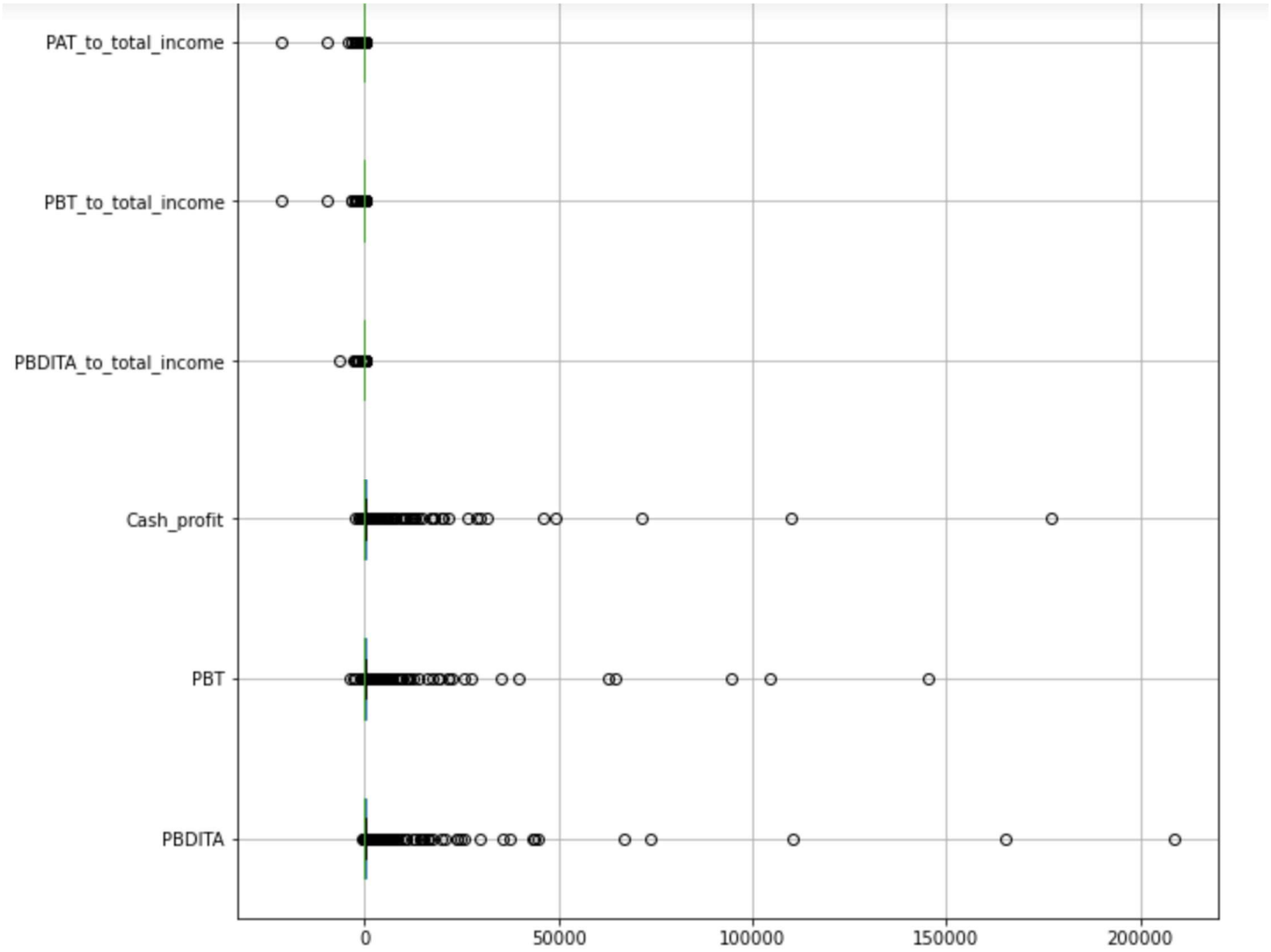
IQR (Inter quantiles range) = 75th quantile — 25th quantile

Upper Boundary = 75th quantile +(IQR * 3)

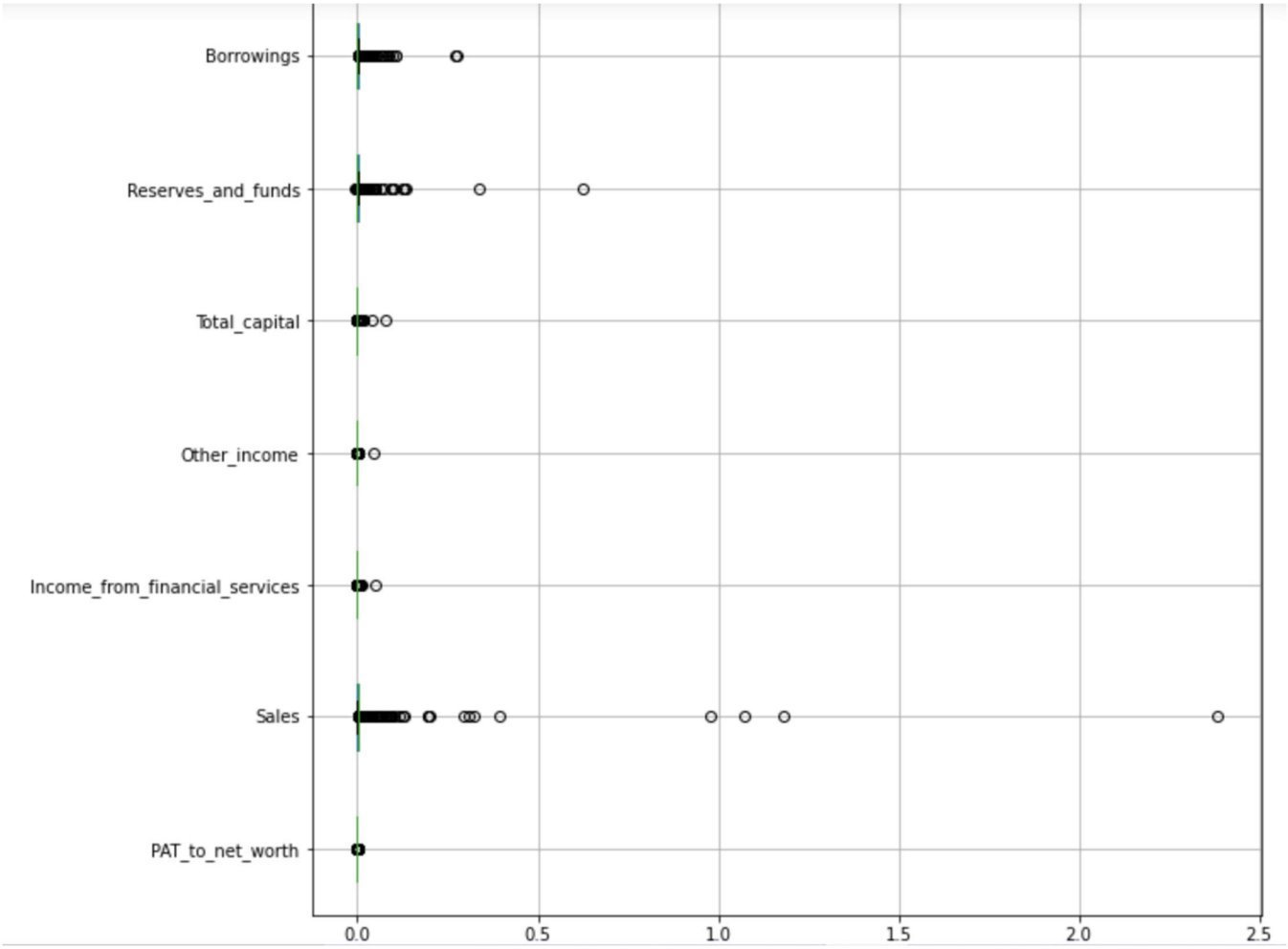
Lower Boundary = 25th quantile — (IQR * 3)



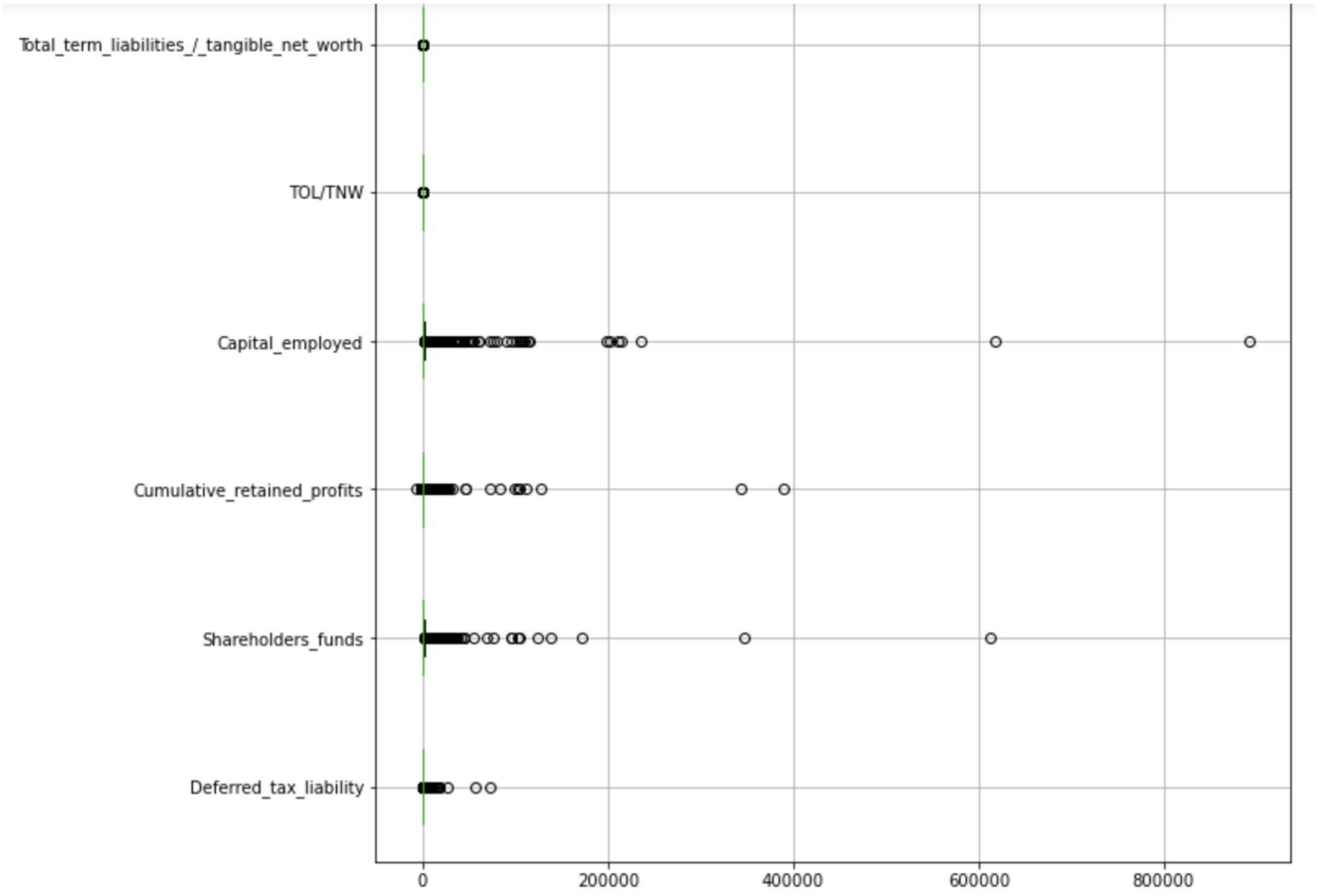
Even after the treatment of the outlier for Net worth next year, total income, total expenses are still showing the outliers present in the dataset.



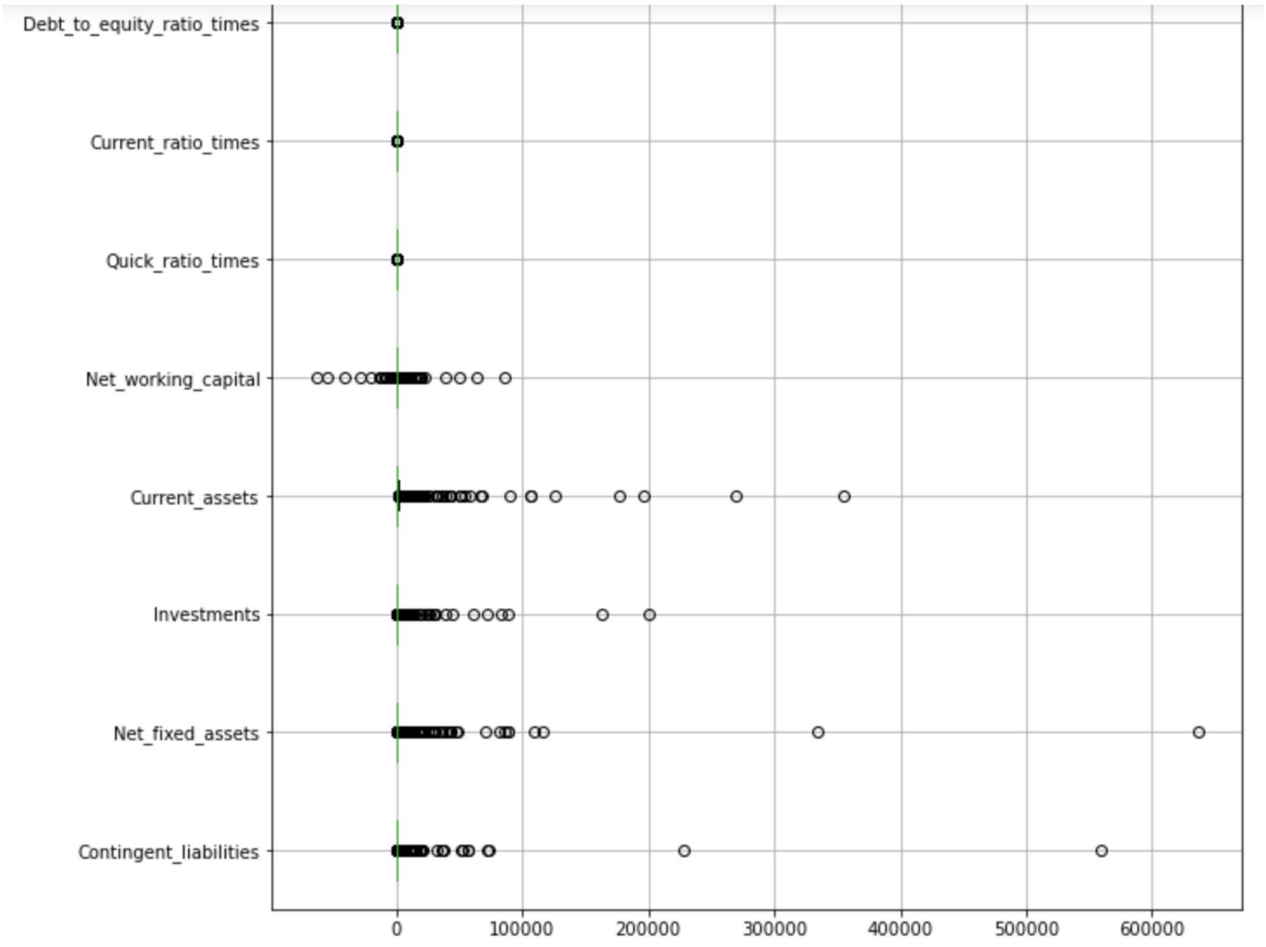
Even after the treatment of the outlier for PBDITA, PBT, Cash Profit are still showing the outliers present in the dataset.



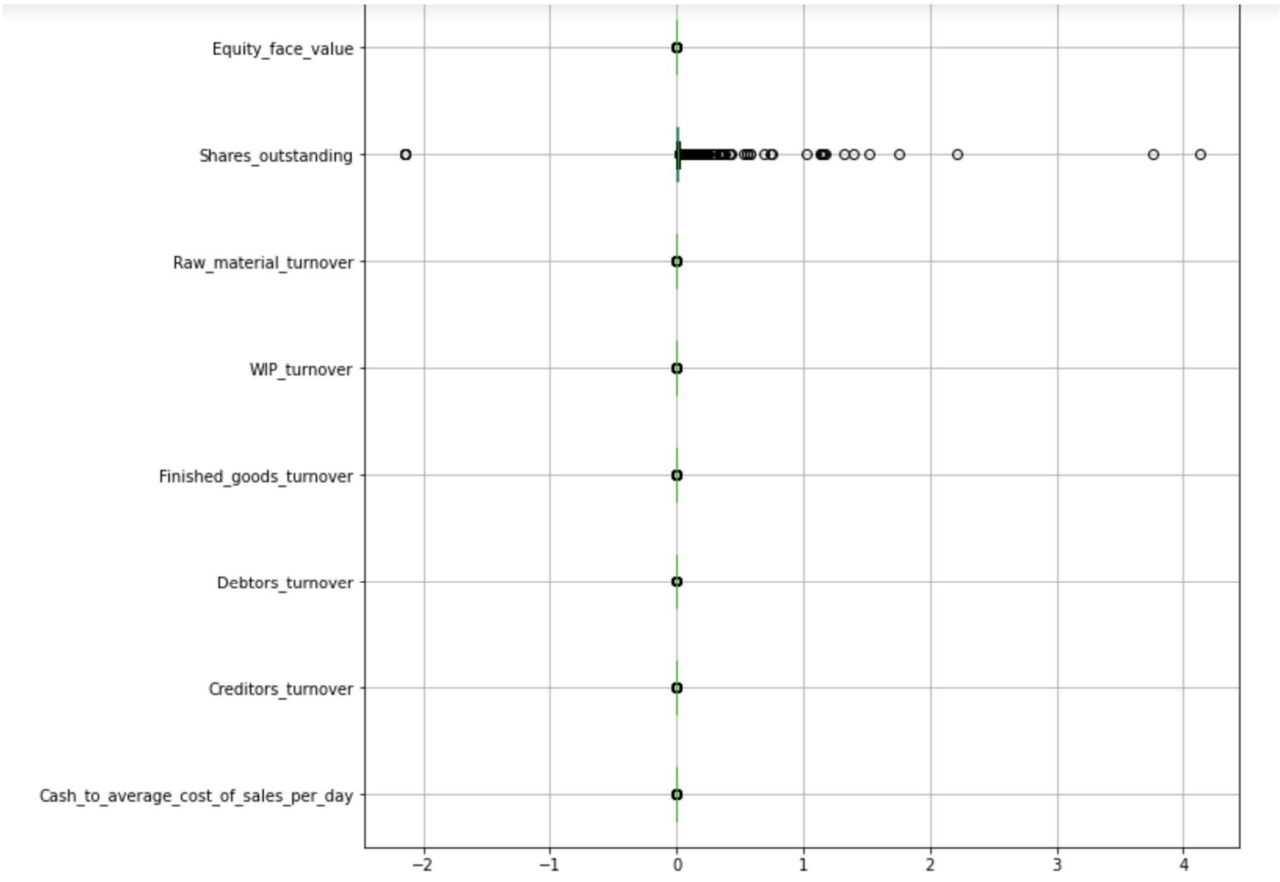
Even after the treatment of the outlier for PAT to net worth, sales, reserves and funds and borrowings are still showing the outliers present in the dataset.



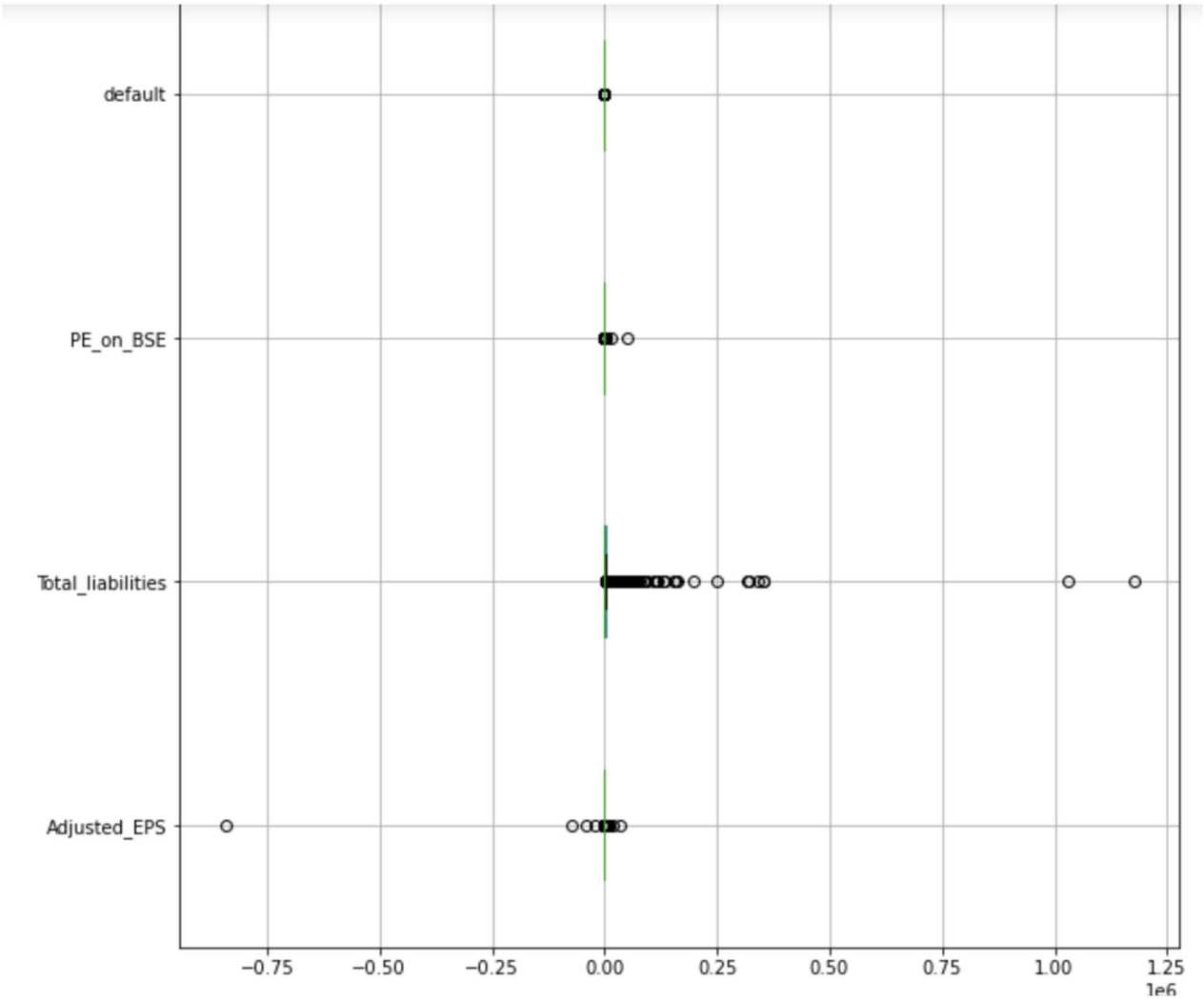
Even after the treatment of the outlier for Deferred tax liability, shareholders fund, cumulative retained profits and capital employed are still showing the outliers present in the dataset.



Even after the treatment of the outlier for contingent liabilities, net fixed assets, investments and current assets, net working capital are still showing the outliers present in the dataset.



Even after the treatment of the outlier for shares outstanding is still showing the outliers present in the dataset.



Even after the treatment of the outlier for total liabilities is still showing the outliers present in the dataset.

1.5 Train-Test split

```
X = Company.drop('default',axis=1)

# Copy target into the y dataframe.
y = Company['default']
```

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

Default variable is considered as the target variable and the train and test split are done on the basis of the Target variable. The test and train proportion is as 67:33 and the random state is 42

Modeling

1.6 Build a Logistic Regression Model on most important variables on Train Dataset

1. *Model building using Logistic Regression for "Probability at default" by using the variable profit before tax on total income.*

Probability of default is a formal quantification framework that enables the calculation of probability of default risk measure on the basis of quantitative and qualitative information.

Formula for calculating the probability at default.

- EL = PD × LGD × EAD = PD × (1 − RR) × EAD, where: PD = probability of default LGD = loss given default EAD = exposure at default RR = recovery rate (RR = 1 − LGD)
- model = SM.logit(formula='Dependent Variable ~ Σ$Independent\ Variables\ (k)$' data = 'Data Frame containing the required values').fit()

Profit before tax shows the net worthiness of the company. This will show the actual profit of the company and the company has the worthiness to make the payment of corporate tax. Hence, we will be considering the first variable to build the model in logistic regression as "PBT_to_total_income".

Model Summary:

Dep. Variable:	default	**No. Observations:**	3541
Model:	Logit	**Df Residuals:**	3539
Method:	MLE	**Df Model:**	1
Date:	Tue, 15 Dec 2020	**Pseudo R-squ.:**	0.01658
Time:	23:02:36	**Log-Likelihood:**	-870.81
converged:	True	**LL-Null:**	-885.49
Covariance Type:	nonrobust	**LLR p-value:**	6.025e-08

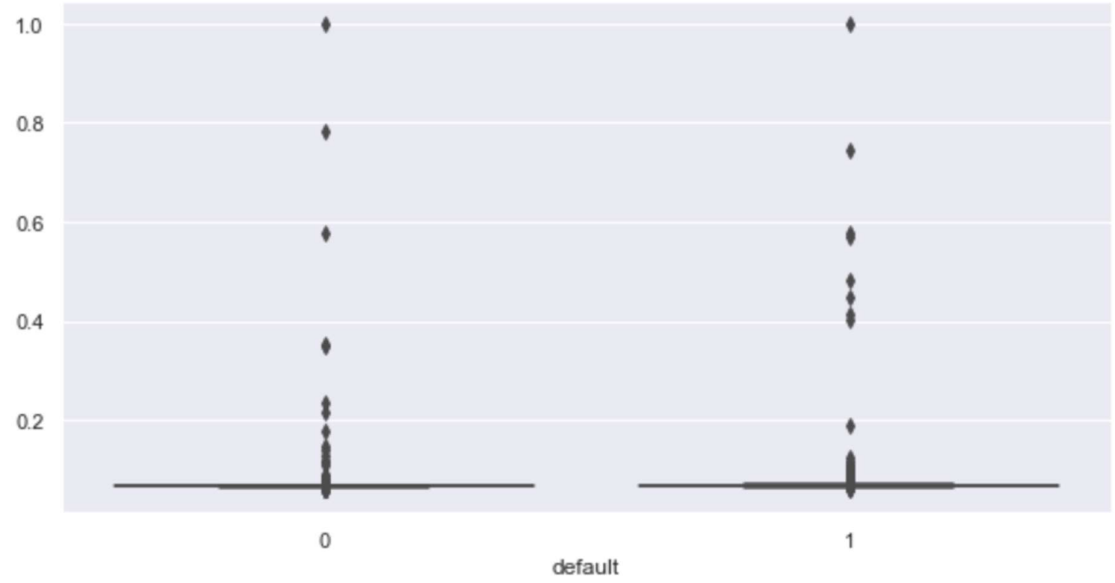
	coef	std err	z	P>|z|	[0.025	0.975]
Intercept	-2.6414	0.068	-39.088	0.000	-2.774	-2.509
PBT_to_total_income	-0.0011	0.000	-4.541	0.000	-0.002	-0.001

The p value if considered under 0.05, here the value is less than p value, hence we will accept the hypothesis. The coefficient of the variable is -0.0011 and the standard deviation is 0.

The coefficient intervals on the basis of train and test data is as below

```
                          0         1
Intercept           0.062425  0.081357
PBT_to_total_income 0.998396  0.999363
```

Plotting of the actual default vs Predicted default



Here we can identify the defaulter and the non-defaulter are below 0.2. Compared to the non-defaulter , defaulter are below the variable 0.2.

2. *Model building using Logistic Regression for "Probability at default" by using the variable debt to equity ratio.*

The debt-to-equity ratio is calculated by dividing the company's total liability by its shareholder equity. This ratio will show us the financial leverage of the company.

Model Summary:

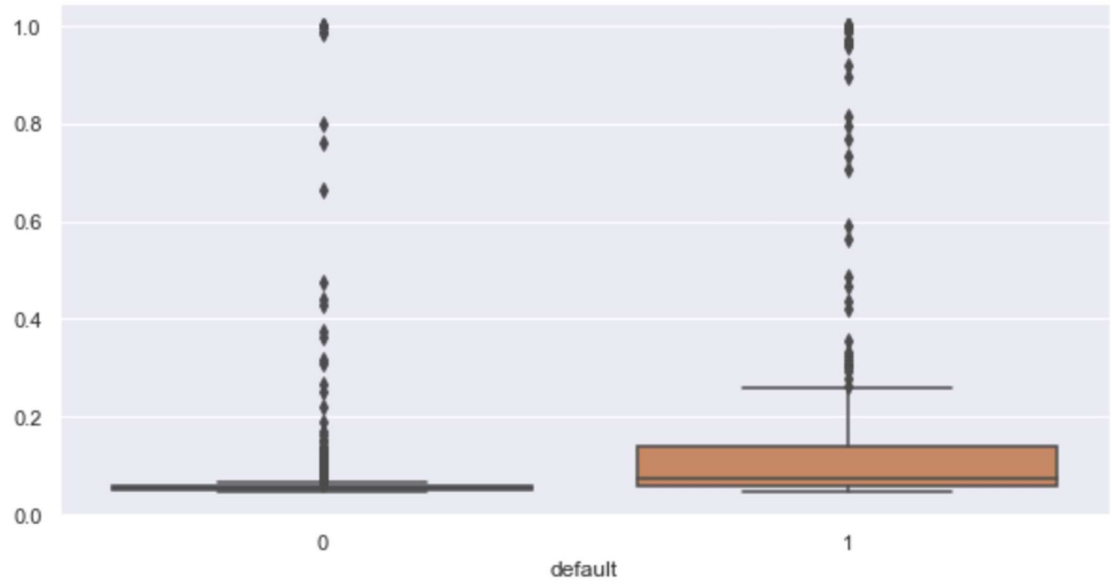
Logit Regression Results

Dep. Variable:	default	**No. Observations:**	3541
Model:	Logit	**Df Residuals:**	3538
Method:	MLE	**Df Model:**	2
Date:	Tue, 15 Dec 2020	**Pseudo R-squ.:**	0.1063
Time:	23:02:36	**Log-Likelihood:**	-791.38
converged:	True	**LL-Null:**	-885.49
Covariance Type:	nonrobust	**LLR p-value:**	1.355e-41

	coef	std err	z	P>|z|	[0.025	0.975]
Intercept	-2.8997	0.076	-37.905	0.000	-3.050	-2.750
PBT_to_total_income	-0.0010	0.000	-3.886	0.000	-0.002	-0.000
Debt_to_equity_ratio_times	0.0667	0.009	7.765	0.000	0.050	0.084

This model also shows us that the p value of the variables are less than 0.05 hence we accept the null hypothesis. The coefficient of the model is showing 0.06 which is less than 1.5 where the model is considered as good.

Plotting of the actual default vs Predicted default

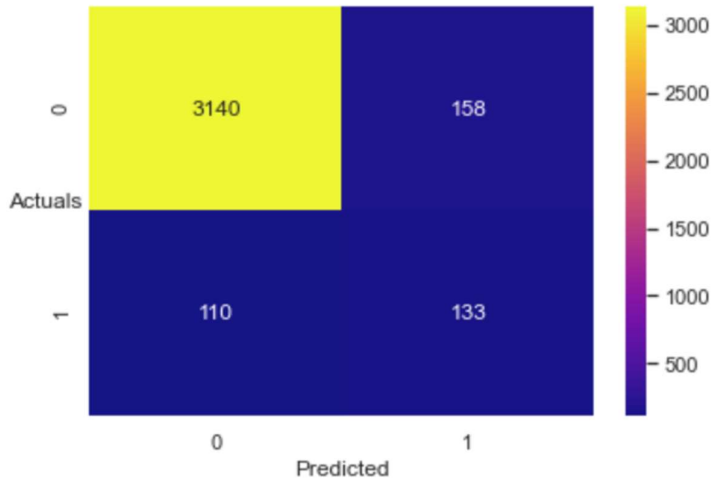


The defaulters are showing the chances of being default is falling between 40 to 60%

1.7 Validate the Model on Test Dataset and state the performance matrices

The mean average of the proportion of defaulters is derived as 0.0686, hence we will proceed up with our prediction by using the range between 0.05 to 0.07 in order to identify the best model.

@0.07



The confusion matrix:

Sensitivity aka Recall (true positives / all actual positives) = TP / TP + FN

Specificity (true negatives / all actual negatives) =TN / TN + FP

Accuracy (all correct / all) = TP + TN / TP + TN + FP + FN

Misclassification (all incorrect / all) = FP + FN / TP + TN + FP + FN

Confusion Matrix for default cutoff at 0.07

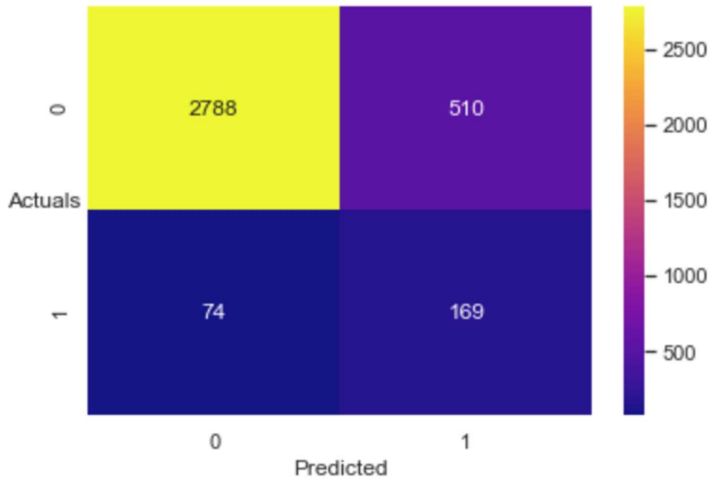
Sensitivity of the model is 54%

Specificity of the model is 95%

Accuracy of the model is 92%

Misclassification rate is 0.07%

@0.06



Confusion Matrix for default cutoff at 0.06

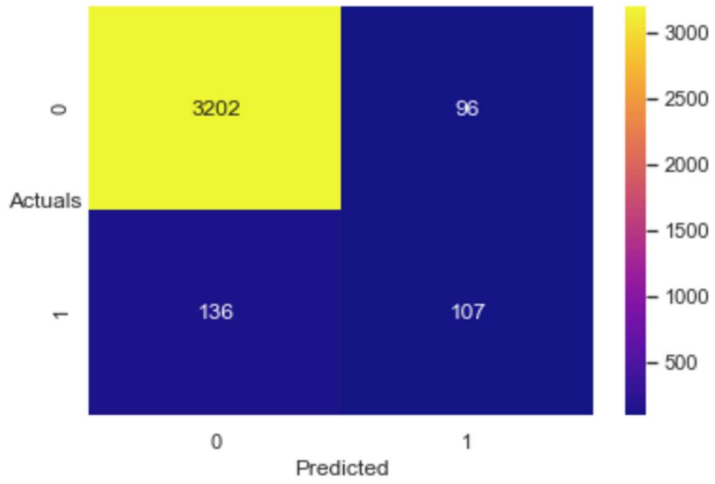
Sensitivity of the model is 70%

Specificity of the model is 85%

Accuracy of the model is 84%

Misclassification rate is 0.16%

@0.08



Confusion Matrix for default cutoff at 0.08

Sensitivity of the model is 44%

Specificity of the model is 97%

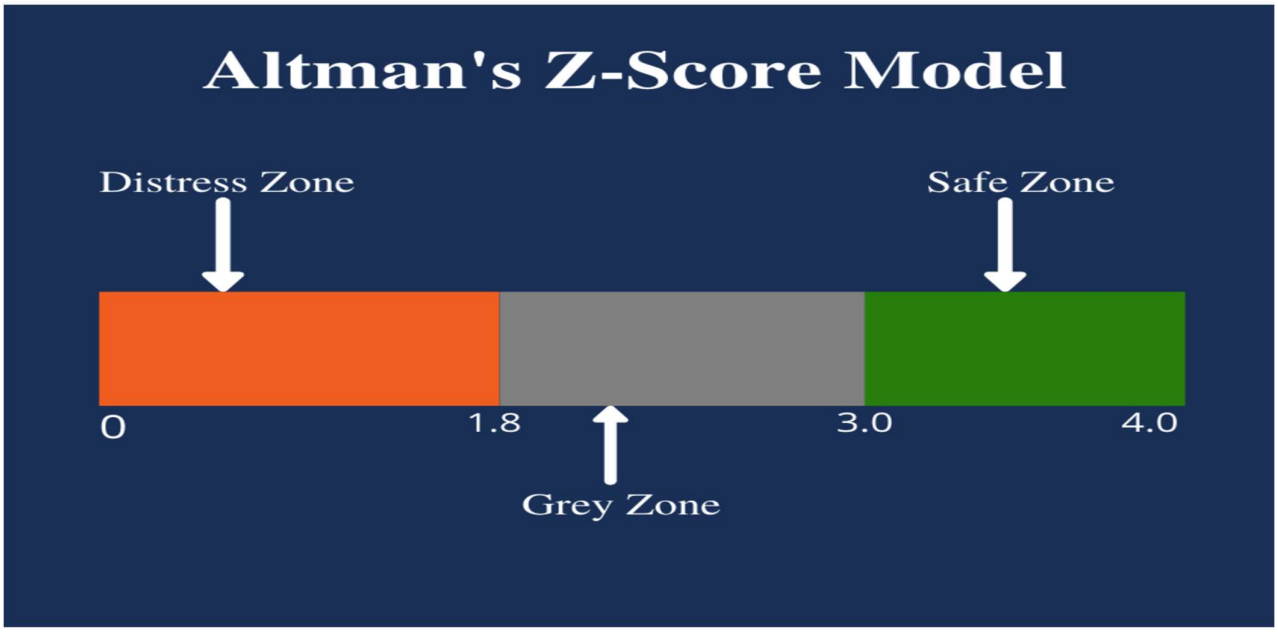
Accuracy of the model is 93%

Misclassification rate is 0.07%

By comparing all the models, we can identify that higher the rate of cutoff is higher the accuracy of the model with very low error rate. Hence the best model is considered to be as 0.07.

Altman's Z Score:

Altman's Z-Score model is a numerical measurement that is used to predict the chances of a business going bankrupt in the next two years. The model was developed by American finance professor Edward Altman in 1968 as a measure of the financial stability of companies.



Altman's Z-score model is considered an effective method of predicting the state of financial distress of any organization by using multiple balance sheet values and corporate income. Altman's idea of developing a formula for predicting bankruptcy started at the time of the Great Depression when businesses experienced a sharp rise in incidences of default

Formula:

$\zeta = 1.2A + 1.4B + 3.3C + 0.6D + 1.0E$

Zeta (ζ) is the Altman's Z-score

A is the Working Capital/Total Assets ratio

B is the Retained Earnings/Total Assets ratio

C is the Earnings Before Interest and Tax/Total Assets ratio

D is the Market Value of Equity/Total Liabilities ratio

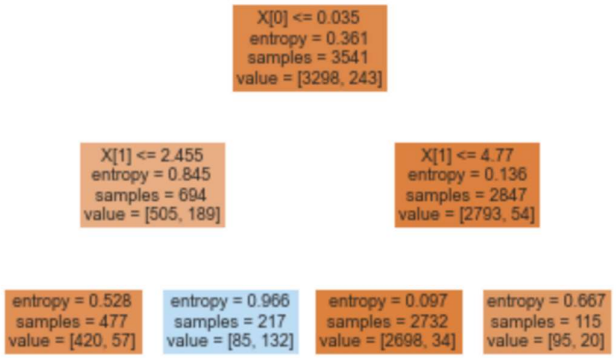
E is the Total Sales/Total Assets ratio

Interpretation:

The lower the Z-score, the higher the odds that a company is heading to be defaulters. A Z-score that is lower than 1.8 means that the company is in financial distress and with a high probability of going defaulters. On the other hand, a score of 3 and above means that the company is in a safe zone and is unlikely to file for defaulter. A score of between 1.8 and 3 means that the company is in a grey area and with a moderate chance of filing for being defaulter.

```
count     3541.000000
mean         9.186016
std        667.317560
min     -28706.178110
25%          1.573883
50%          2.344135
75%          3.563892
max       6275.269000
```

Decision Tree



Confusion Matrix for default

Sensitivity of the model is 54%

Specificity of the model is 97%

Accuracy of the model is 94%

Misclassification rate is 0.05%

Under the criterion "entropy" we have received the best results as the accuracy of this model best with 94%.

1.8 Build a Random Forest Model on Train Dataset

Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes or mean/average prediction of the individual trees

Confusion Matrix on train data:

```
1.0
[[2203    0]
 [   0  169]]
              precision    recall  f1-score   support

           0       1.00      1.00      1.00      2203
           1       1.00      1.00      1.00       169

    accuracy                           1.00      2372
   macro avg       1.00      1.00      1.00      2372
weighted avg       1.00      1.00      1.00      2372
```

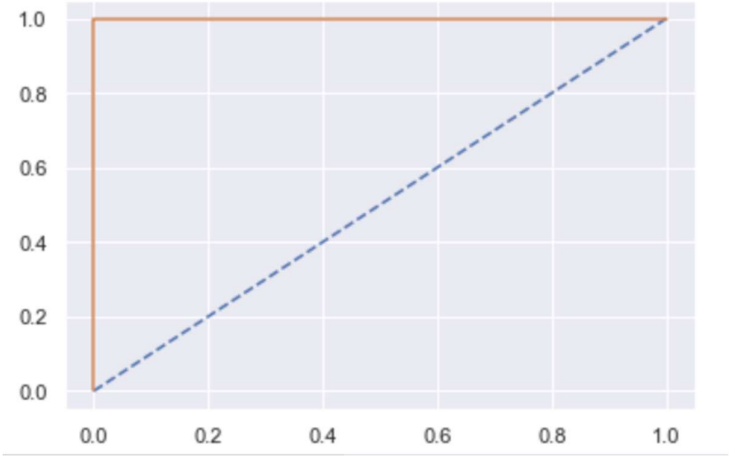
The model shows accuracy as 100%

The train precision is 100%

The train recall is 100%

The train F1 score is also 100%

ROC Curve:



The AUC curve is showing as 100%

1.9 Validate the Random Forest Model on test Dataset and state the performance matrices

Confusion Matrix on test data:

```
0.998289136013687
[[1093    2]
 [   0   74]]
              precision    recall  f1-score   support

           0       1.00      1.00      1.00      1095
           1       0.97      1.00      0.99        74

    accuracy                           1.00      1169
   macro avg       0.99      1.00      0.99      1169
weighted avg       1.00      1.00      1.00      1169
```

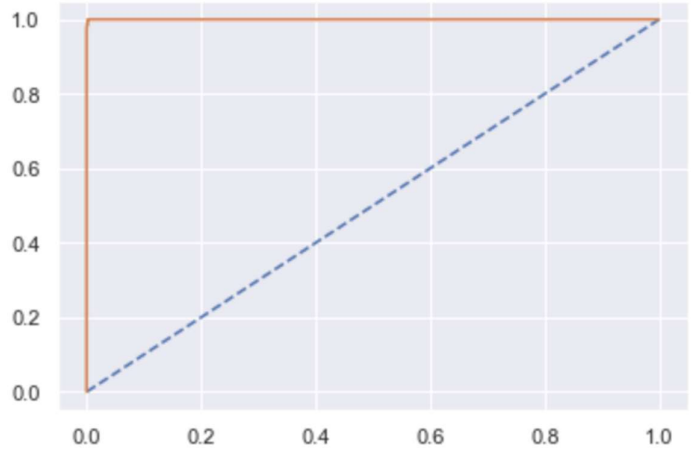
The accuracy of the test model is 99.8%

The precision of the test model is 100%

The recall of the test model is 99%

The f1 score of the model is 0.97%

ROC Curve:



The AUC curve is showing as 100%

1.10 Build an LDA Model on Train Dataset

Linear Discriminant Analysis means a dimensionality reduction technique. Dimensionality reduction techniques reduce the number of dimensions (i.e. variables) in a dataset while retaining as much information as possible.

Confusion Matrix on train data:

```
0.9397133220910624
[[2187   16]
 [ 127   42]]
              precision    recall  f1-score   support

           0       0.95      0.99      0.97      2203
           1       0.72      0.25      0.37       169

    accuracy                           0.94      2372
   macro avg       0.83      0.62      0.67      2372
weighted avg       0.93      0.94      0.93      2372
```

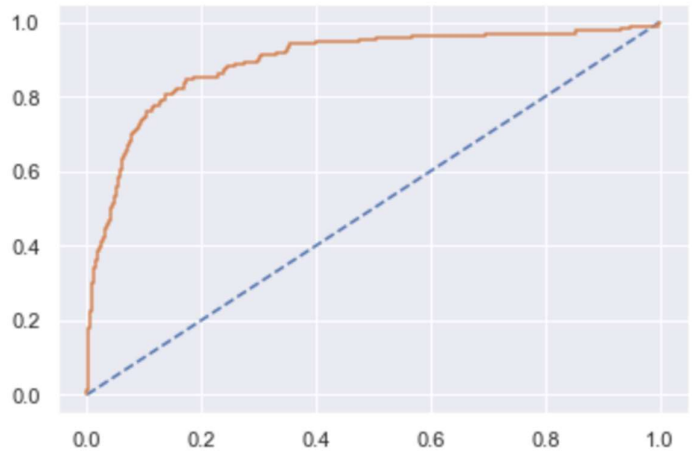
The accuracy of the model is showing as 94%.

The precision of the model is showing as 25%

The recall of the model is showing as 37%

The F1 score is 72%

ROC Curve:



The AUC curve shows 89% on train data.

1.11 Validate the LDA Model on test Dataset and state the performance matrices

Confusion Matrix on Test Data:

```
0.9443969204448246
[[1083   12]
 [  53   21]]
              precision    recall  f1-score   support

           0       0.95      0.99      0.97      1095
           1       0.64      0.28      0.39        74

    accuracy                           0.94      1169
   macro avg       0.79      0.64      0.68      1169
weighted avg       0.93      0.94      0.93      1169
```

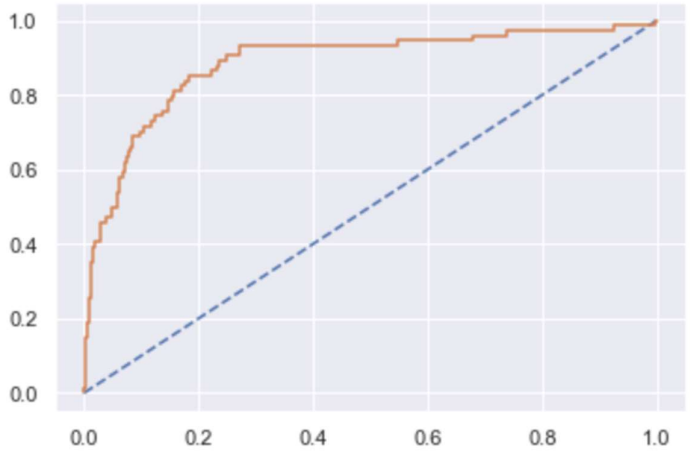
The accuracy of the model shows 94%

The precision of the model shows 28%

The recall on test model is 39%

The F1 score is 64%

ROC Curve:



The AUC curve is 88%

1.12 Compare the performances of all the three models (include ROC Curve)

	Random Forest Train	Random Forest Test	Logistic Regression Train	Logistic Regression Test	LDA Train	LDA Test
Recall	1.0	0.99	0.09	0.03	0.37	0.39
Precision	1.0	1.00	0.05	0.01	0.25	0.28
F1 Score	1.0	0.97	0.73	0.50	0.72	0.64

Comparing all the three models, we can go with random forest as the data is showing 100% consistency on both test data and train dataset. When compared with the logistic regression model under stats model we can say that the best model that gives us the correct decision to make on is through a cut off proportion of 0.07 where the accuracy of the data is more than 94%. The test results on logistic regression says that there are only 20% below chances are there for the financial institution to be under defaulter.

1.13 Recommendation

The analysis shows us that there is very less chance for the firm to go under credit risk. Hence this financial institution to be more a non-defaulter rather than being defaulter. Its recommended that the firms can be approached for loans as they have the credit worthiness to make their payments.

2

Market Risk

Market risk is the possibility of an investor experiencing losses due to factors that affect the overall performance of the financial markets in which he or she is involved. Market risk, also called "systematic risk," cannot be eliminated through diversification, though it can be hedged against in other ways

Market risk is the risk of losses on financial investments caused by adverse price movements

Market risk is one of the three core risks all banks are required to report and hold capital against, alongside credit risk and operational risk. The standard method for evaluating market risk is value-at-risk

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.

Import all libraries and dataset:

We have to import all relevant packages for the analysis.

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import seaborn as sns # for making plots with seaborn
color = sns.color_palette()
import sklearn.metrics as metrics

import warnings
warnings.filterwarnings("ignore")
```

Once the packages are imported, we will be importing the dataset to the work base.

```
stock_prices = pd.read_csv('Market+Risk+Dataset.csv')
```

To confirm whether the data is successfully uploaded, check the header of the dataset.

	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

We could find the dataset is successfully uploaded with 11 columns and 314 rows.

We need to amend the heading of the column heading as there are special characters included in the heading.

```
Index(['Date', 'Infosys', 'Indian Hotel', 'Mahindra & Mahindra', 'Axis Bank',
       'SAIL', 'Shree Cement', 'Sun Pharma', 'Jindal Steel', 'Idea Vodafone',
       'Jet Airways'],
      dtype='object')
```

Once the column headings are corrected, to the specific format then we can process further to the analysis.

	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

```
The number of rows (observations) is 314
The number of columns (variables) is 11
```

Type of dataset:

```
 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)
```

The dataset does not have any missing values or any null values. The data is related to stock pricing hence the type of the data is numeric and are integers.

Description of the dataset:

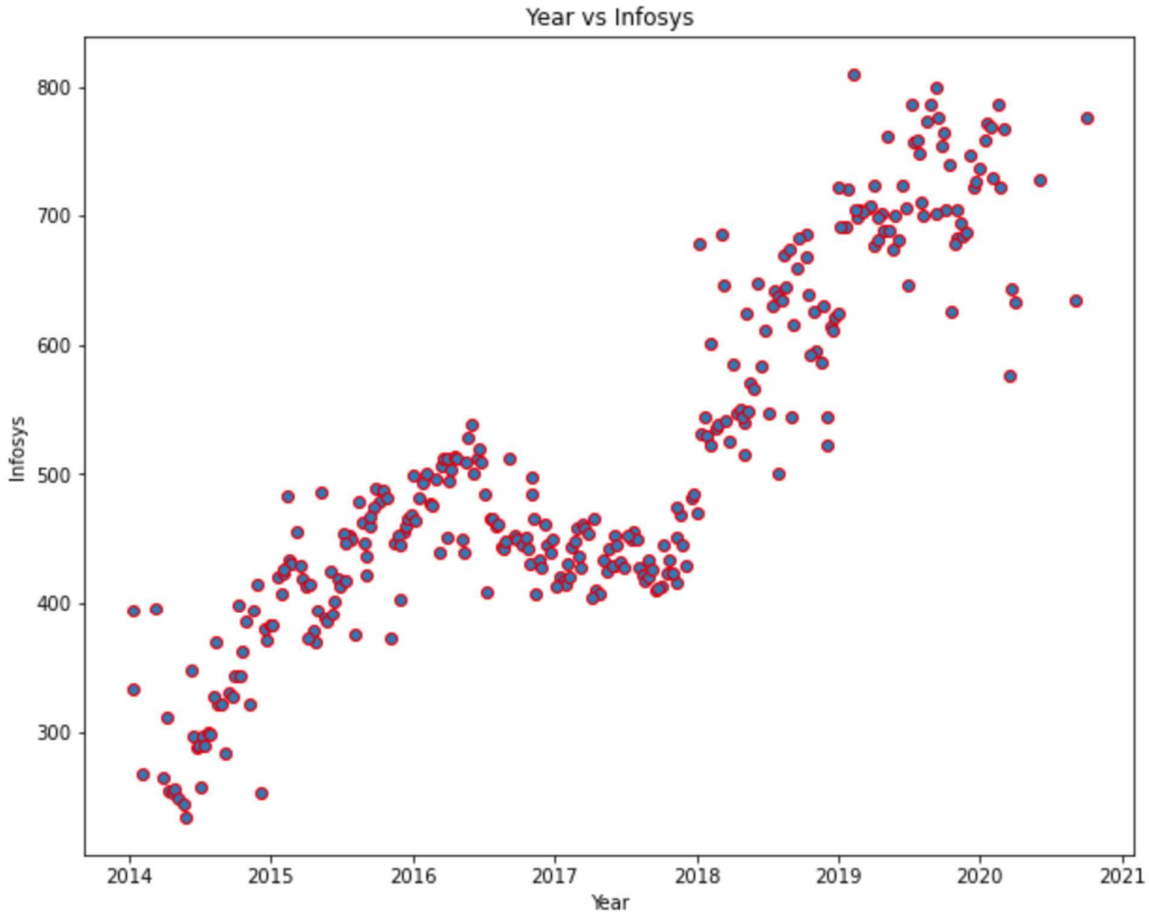
	Infosys	Indian_Hotel	Mahindra__Mahindra	Axis_Bank	SAIL	Shree_Cement	Sun_Pharma	Jindal_Steel	Idea_Vodafone	Jet_Airways
count	314.000000	314.000000	314.000000	314.000000	314.000000	314.000000	314.000000	314.000000	314.000000	314.000000
mean	511.340764	114.560510	636.678344	540.742038	59.095541	14806.410828	633.468153	147.627389	53.713376	372.659236
std	135.952051	22.509732	102.879975	115.835569	15.810493	4288.275085	171.855893	65.879195	31.248985	202.262668
min	234.000000	64.000000	284.000000	263.000000	21.000000	5543.000000	338.000000	53.000000	3.000000	14.000000
25%	424.000000	96.000000	572.000000	470.500000	47.000000	10952.250000	478.500000	88.250000	25.250000	243.250000
50%	466.500000	115.000000	625.000000	528.000000	57.000000	16018.500000	614.000000	142.500000	53.000000	376.000000
75%	630.750000	134.000000	678.000000	605.250000	71.750000	17773.250000	785.000000	182.750000	82.000000	534.000000
max	810.000000	157.000000	956.000000	808.000000	104.000000	24806.000000	1089.000000	338.000000	117.000000	871.000000

We have the mean, median, standard deviation, minimum and maximum of the data set.

2.1 Draw Stock Price Chart for any 2 variables

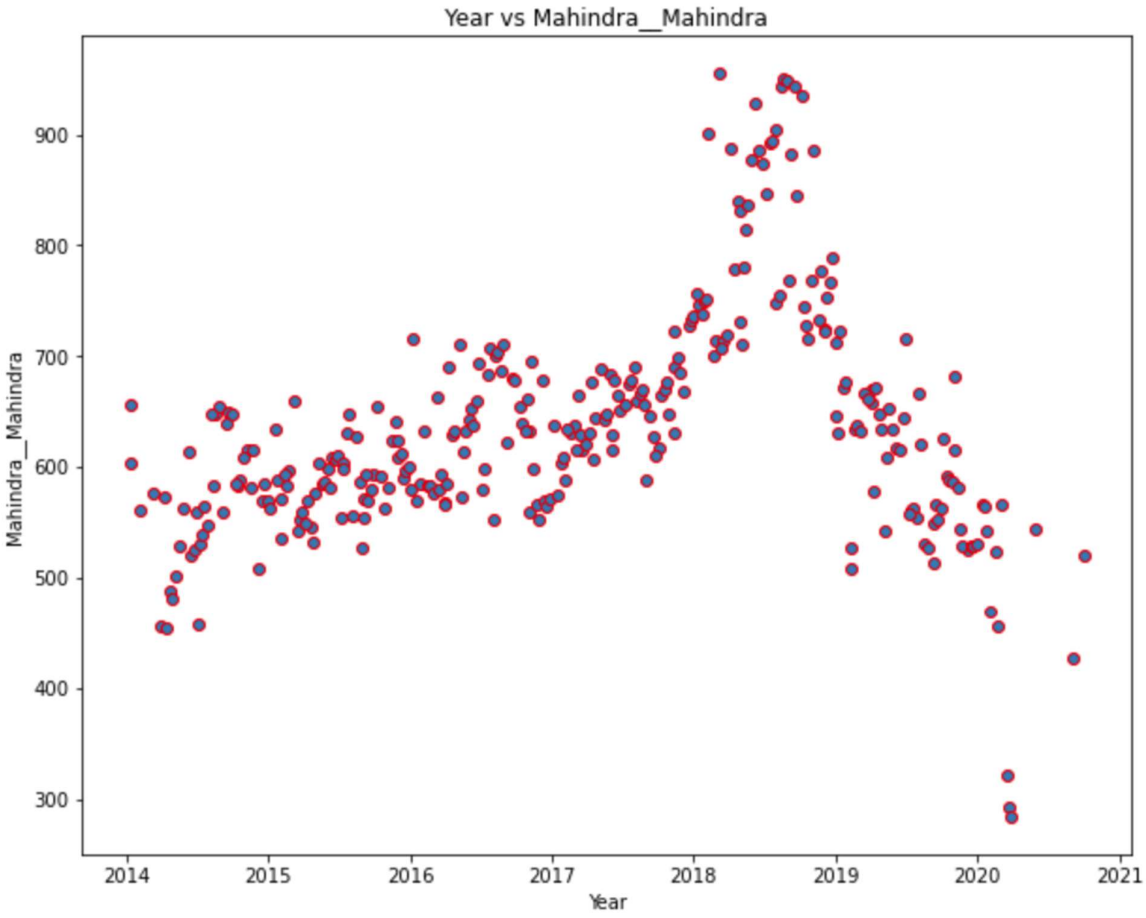
The dataset has the values of the shares related to Infosys, Indian Hotel, Mahindra & Mahindra, Axis Bank, SAIL, Shree Cements, Sun Pharma, Jindal Steel, Idea Vodafone and Jet airways. The maximum priced shares are for Shree Cements, Sun Pharma, Infosys, Axis bank and Jet airways. To analyze the movement of the stock for various years, will take a chart by taking year and shares as variables.

1. Year vs Infosys:



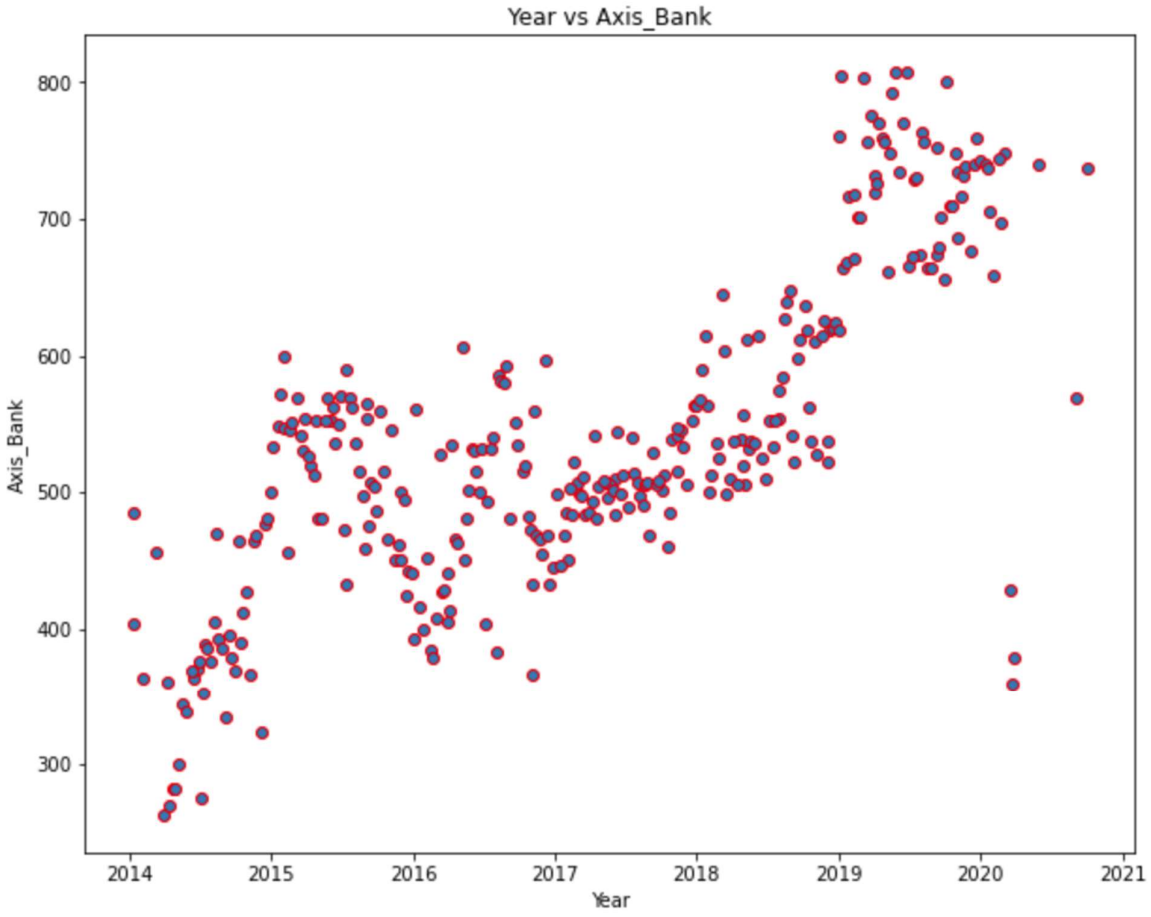
The value of the stock for Infosys was below 300 in between 2014-15, where the stock value of the shares has increased from 2015 to 2017 where the price has gone above 550 in mid of 2016. The shares have again dropped down for the next two years and again from 2018 the value of the shares has gone high to 800 in mid of 2019-20.

2. Year vs Mahindra & Mahindra



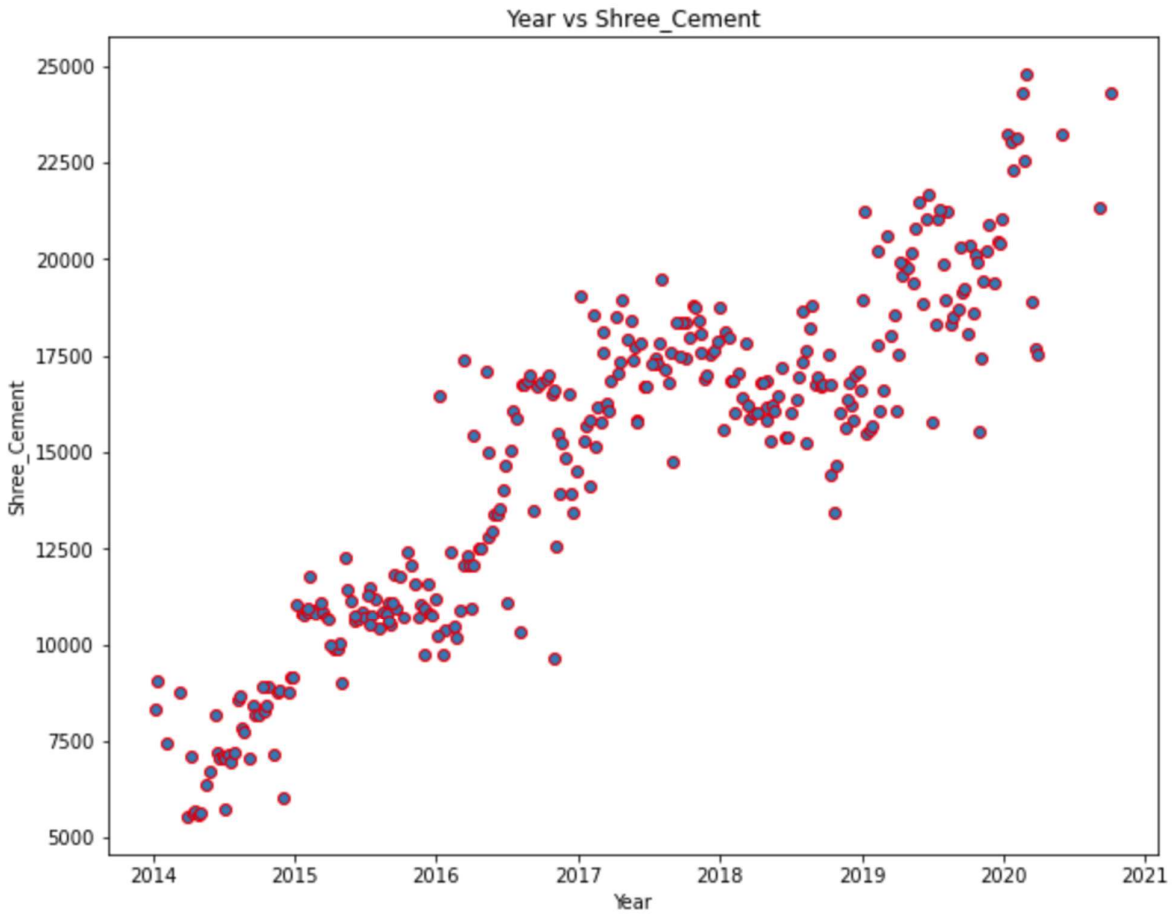
Currently the stock value of Mahindra & Mahindra is showing a bear trend. The turnover period for the shares of Mahindra & Mahindra was during 2018-19 where the shares have reached above 900. The value of the shares has started to decline from 2019 till date. The value of the shares has gone below 300 in 2020. This decline is due to the reason of pandemic and the operations of all manufacturing units were suspended and this is one of the reasons for the decline of the shares of Mahindra & Mahindra.

3. Year vs Axis Bank



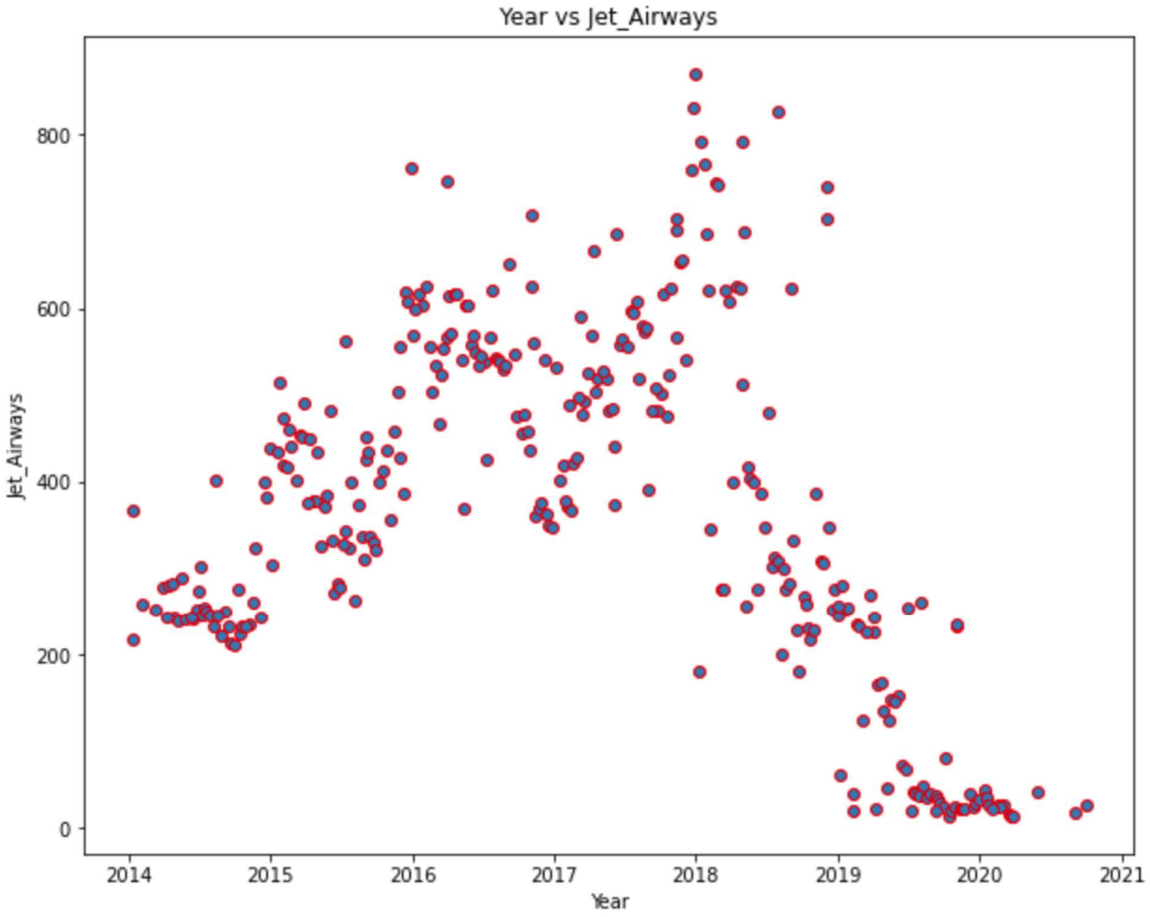
Axis bank shares are having a bull trend in the stock market. The value of the shares is showing a hike from 2014 till 2020. Where the shares have not gone down below 300 even in pandemic stages. The best year that gained the highest stock gains was during the period 2019 to 2020. The shares have reached more than 800 points in the period.

4. Year vs Shree Cements:



The trend of Shree Cements is showing bull trend. The shares of Shree cements where at 5000 in 2014 and the value has gone high to more than 25000. The trend of Shree cements shows that every year beginning the shares will come down and then the shares increases.

5. Year vs Jet Airways



The tend of Jet airways is duck market now. The value of the shares is tremendously down from 800 to 100. The shares where above 200 in year 2014 and there was an increasing trend from 2014 till 2018. The value of the shares started to decline from 2018 and the shares has gone down to 100. The reasons why the fall of the shares is due to the suspension of the air services as they ran out of money and financial institutions where not ready to support them due to the credit risk.

2.2 Calculate Returns

Return on investment (ROI) is a financial ratio used to calculate the benefit an investor will receive in relation to their investment cost. It is a ratio that compares the gain or loss from an investment relative to its cost. It is as useful in evaluating the potential return from a stand-alone investment as it is in comparing returns from several investments.

Formula for calculating the ROI:

First method:

$$ROI = \frac{\text{Net Return on Investment}}{\text{Cost of Investment}} \times 100\%$$

Second method:

$$ROI = \frac{\text{Final Value of Investment} - \text{Initial Value of Investment}}{\text{Cost of Investment}} \times 100\%$$

Here we use the below formula in calculating the returns

```
stock_returns = np.log(stock_prices.drop(['Date','dates'],axis=1)).diff(axis = 0, periods = 1)
```

Results of the return calculation:

	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	-0.026873	-0.014599	0.006572	0.048247	0.028988	0.032831	0.094491	-0.065882	0.011976	0.086112
2	-0.011742	0.000000	-0.008772	-0.021979	-0.028988	-0.013888	-0.004930	0.000000	-0.011976	-0.078943
3	-0.003945	0.000000	0.072218	0.047025	0.000000	0.007583	-0.004955	-0.018084	0.000000	0.007117
4	0.011788	-0.045120	-0.012371	-0.003540	-0.076373	-0.019515	0.011523	-0.140857	-0.049393	-0.148846

We have 314 rows and 10 rows in the results derived from the returns calculation.

2.3 Calculate Stock Means and Standard Deviation

Mean of returns:

```
Infosys               0.002794
Indian_Hotel          0.000266
Mahindra__Mahindra   -0.001506
Axis_Bank             0.001167
SAIL                 -0.003463
Shree_Cement          0.003681
Sun_Pharma           -0.001455
Jindal_Steel         -0.004123
Idea_Vodafone        -0.010608
Jet_Airways          -0.009548
dtype: float64
```

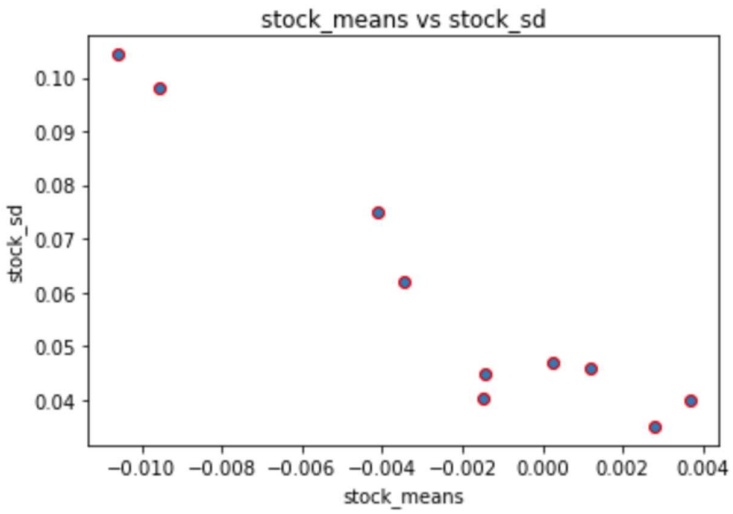
We have four shares that are having a positive means and the rest all are showing a negative trend in the returns. Infosys, Indian Hotels, Axis bank and Shree cements are the top four shares that are giving a average return to the investors.

Standard Deviation of the returns:

```
Infosys               0.035070
Indian_Hotel          0.047131
Mahindra__Mahindra    0.040169
Axis_Bank             0.045828
SAIL                  0.062188
Shree_Cement          0.039917
Sun_Pharma            0.045033
Jindal_Steel          0.075108
Idea_Vodafone         0.104315
Jet_Airways           0.097972
dtype: float64
```

Standard deviation of returns measures the volatility of risk. Larger the return standard deviation larger the variation on expected returns.

2.4 Draw a plot of Stock Means vs Standard Deviation and share insights



The chart helps us in identifying the market risk of investment of various shares. Higher the standard deviation means higher the risk on returns. We will get 0.004 means where the risk is below 0.05.