Finance and Risk Analytics Part A

By Giridharan Velmurugan

Contents

List of Images

List of tables

Problem Statement

Businesses or companies can fall prey to default if they are not able to keep up their debt obligations. Defaults will lead to a lower credit rating for the company which in turn reduces its chances of getting credit in the future and may have to pay higher interest on existing debts as well as any new obligations.

From an investor's point of view, he would want to invest in a company if it is capable of handling its financial obligations, can grow quickly, and is able to manage the growth scale.

A balance sheet is a financial statement of a company that provides a snapshot of what a company owns, owe, and the amount invested by the shareholders. Thus, it is an important tool that helps evaluate the performance of a business.

Data Overview --

The dataset consists of primarily Int(64) and float(64) data types. "Co_Name" is the only feature that is of the object data type. The cell details of the data are given below. These values represent the raw data as it was given without any processing.

- **Number of Rows** → 2058
- **Number of Columns** → 58
- **Overall size** → 119364

We do have null values but only in some features, the majority of the features do not. We have to treat this before we start modeling.

We do not have duplicate values in the data-frame, this is very good given the volume and size of the data that we have here. This also means that we have unique company names("Co_Name") and company code ("Co_Code").

Another aspect of the data set is that a majority of the columns have outliers. The range of outliers present varies quite a bit.

Outlier treatment

Out of all the numerical features that we have, only 5 do not have outliers. This puts us in a tricky situation. The range of outliers present varies quite a bit, as low as 0.20% to all the way up to 21%.

A majority of the features have their outliers within the range of 5% to 15%. Processing them induces synthetic data(a lot in our case). This might hinder our model's performance and would also take us one step further from the actual scenario.

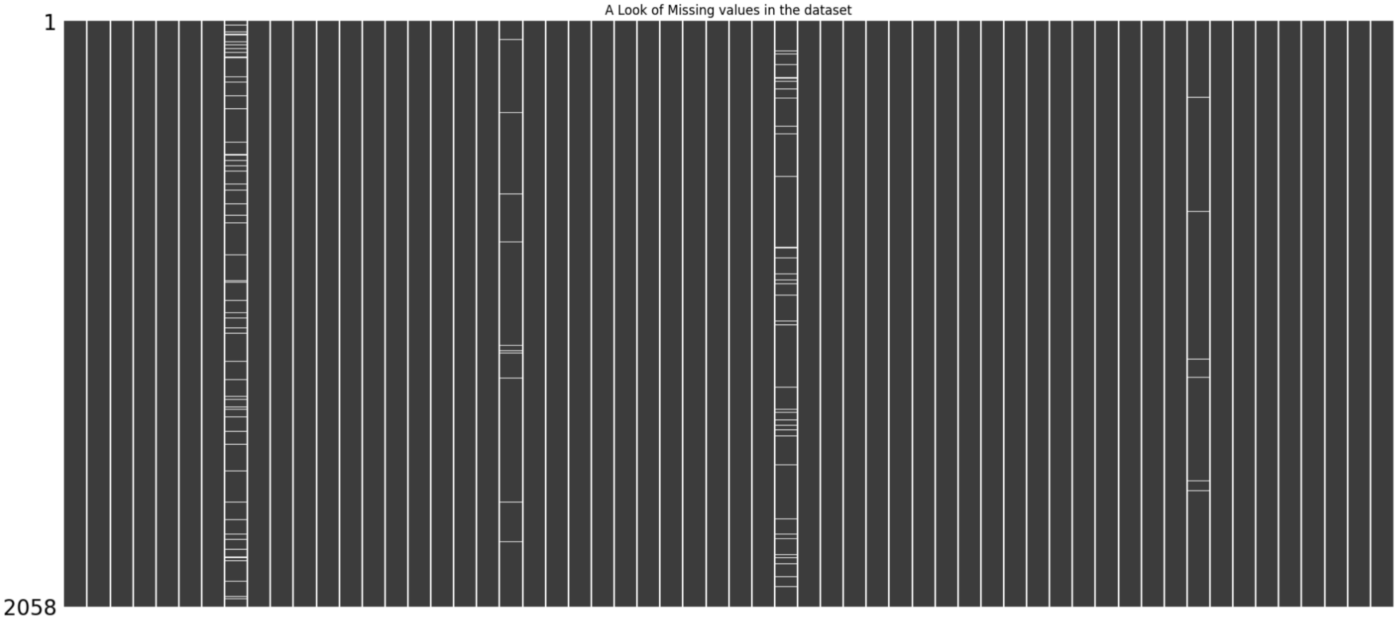
The below table shows the count of outliers to some of the features present in the dataset.

Feature name	Count of outliers
_Operating_Expense_Rate	0(for reference)
_Research_and_development_expense_rate	264
_Cash_flow_rate	206
_Interest_bearing_debt_interest_rate	94
_Tax_rate_A	42
_Cash_Flow_Per_Share	191
_Per_Share_Net_profit_before_tax_Yuan	186
_Realized_Sales_Gross_Profit_Growth_Rate	283
_Operating_Profit_Growth_Rate	317
_Continuous_Net_Profit_Growth_Rate	340
_Degree_of_Financial_Leverage_DFL	438(the max count)

Considering the sheer volume of the outliers, I have opted not to treat them and will be continuing the work with them as they are. The same will be upheld even for features with smaller counts.

Missing value treatment

The dataset does have some missing values but is not as broad as outlier count.



The above plot shows the null values present, the lines with some white breaks are the null values. We only have about 4 features that have missing values. Since all are numerical, we can impute them with appropriate data. Company Name and Company code do not have null values. From the low count we could assume that data collection has been done to a good standard.

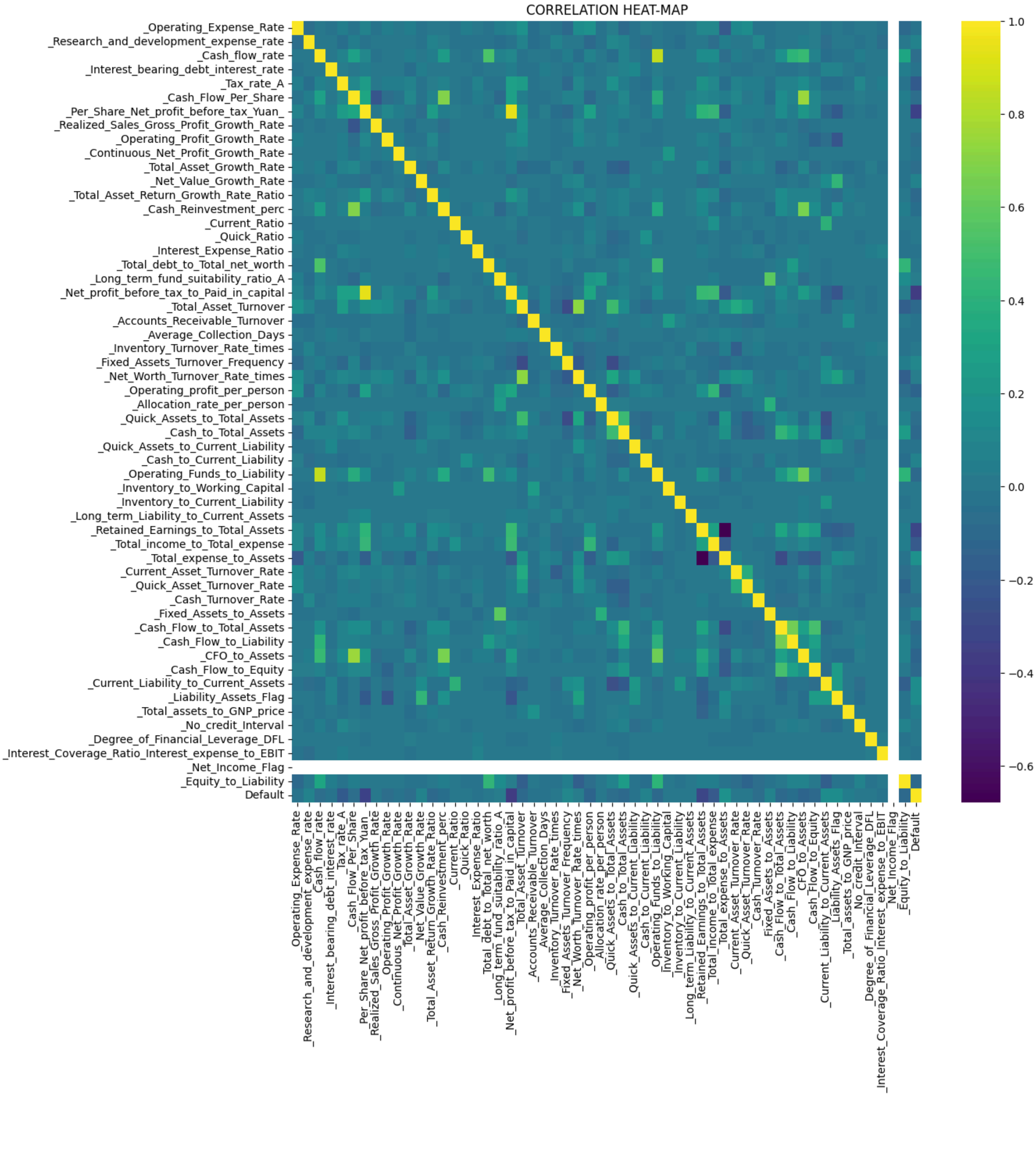
The following features which have missing values are handled as follows

- "_Cash_Flow_Per_Share" has 167 values → Mean
- "_Cash_to_Total_Assets" has 96 values → Mean
- "_Current_Liability_to_Current_Assets" has 14 values → Mean
- "_Total_debt_to_Total_net_worth" has 21 values → Median

The change of mean, standard deviation before and after imputation is minimal, the overall distribution has been retained.

Exploratory Data Analysis

Below is a heatmap of the entire dataset,



We should weed out the diagonal as it is the feature against itself. Apart from the yellow pixels in the diagonal we can see some more spots. These are feature pairs where correlation is considerable and has to be taken care of.

Please find the pairs below

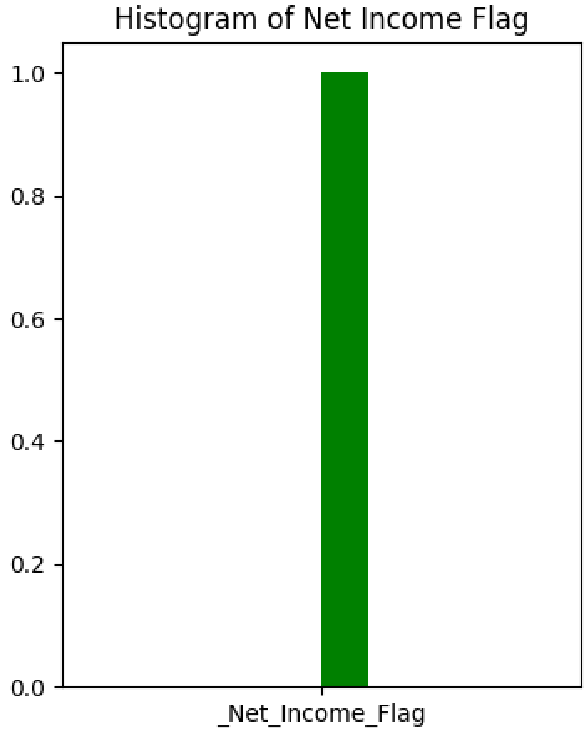
“_Per_Share_Net_profit_before_tax_Yuan_“ → “_Net_profit_before_tax_to_Paid_in_capital“

“_Cash_flow_rate“ → “_Operating_Funds_to_Liability“

“_Total_expense_to_Assets“ → “_Retained_Earnings_to_Total_Assets“

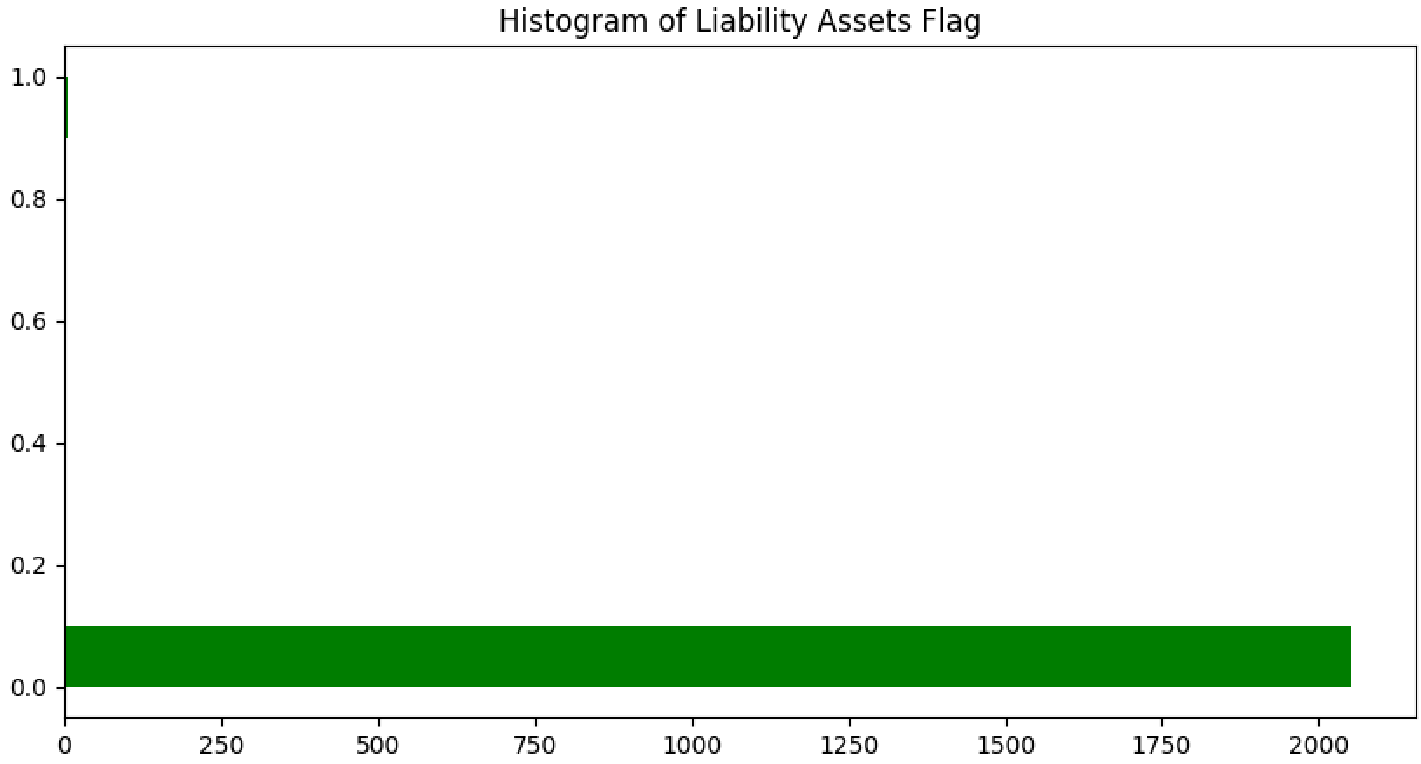
We will prioritize the ones that have a lower outlier count. The final pair is the only one that is negatively correlated.

“_Net_Income_Flag“ is the only feature that is fully white, this attributes to no variance. The feature against itself and also with other features does not show any trend. This is an interesting find, but we will remove this as it does not give the model any info to work with.



Digging deeper into Net Income Flag shows that all the values are zero, this explains why the heat map showed the white pixels.

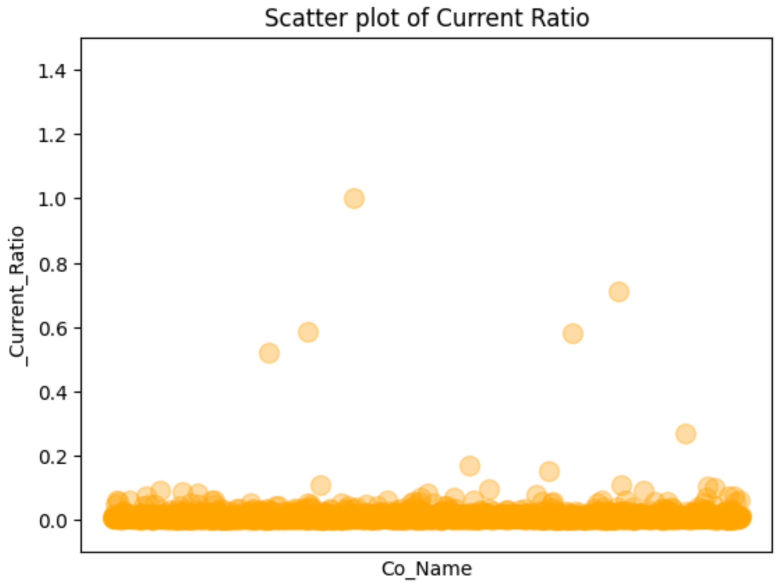
Next up we will have a plot that discusses the “Liability assets Flag”.



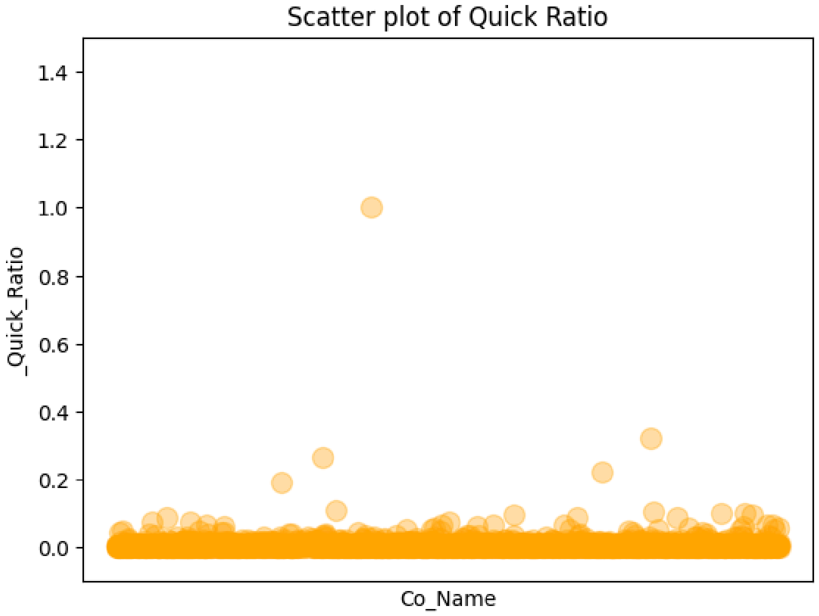
The Histogram of the said features show a very similar scenario to Net Income Flag, we see almost all of the values as zero, except if we zoom in at the bin that don’s the value one, we can see a very tiny hump.

This value having a lot of zeros and very less counter values would generally not help in modeling. It is heavily skewed towards one end, furthermore the weight would also be very negligible. There we will be removing this feature when we prepare classification models.

A peek at current ratio - A current ratio of greater than 1 indicates that the company has more assets than liabilities suggesting good stability, a ratio of less than 1 may indicate liquidity problems. Most of our data is under 0.20, this is not a good sign as most are on the verge of default or already are.

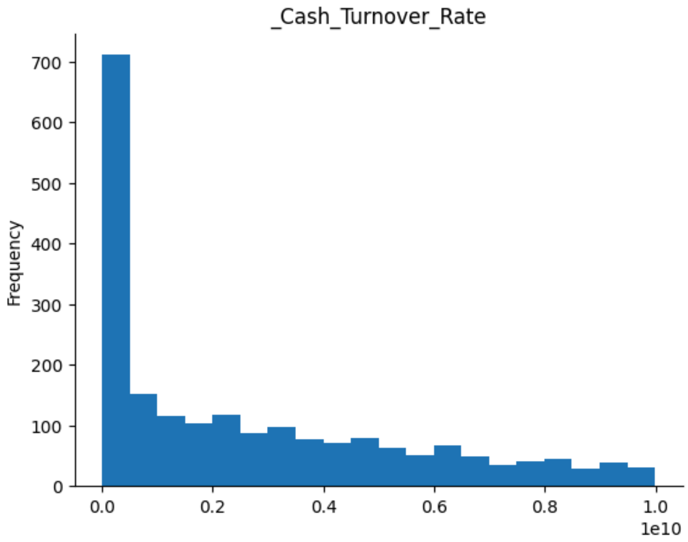


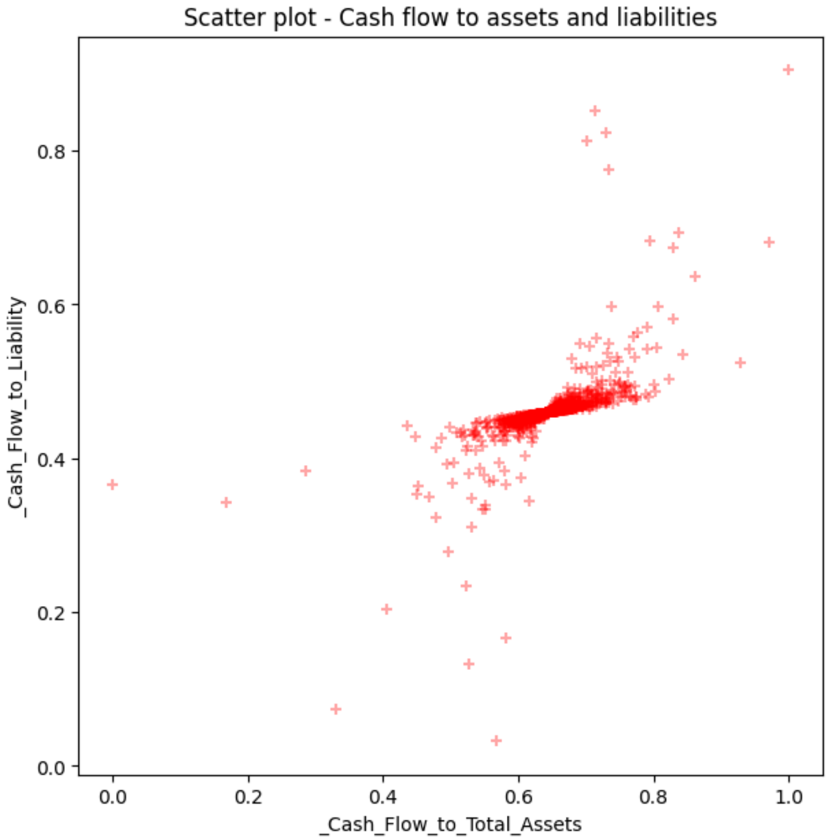
The plot of Quick ratio, describes a financial metric that measures a company's ability to pay off its current debts. A value of less than 1 states that assets are not available to pay off its short term obligations. Similar to the current ratio most of our companies are concentrated below 0.1, this is not a good sign.



Plot of Cash turn over rate

The bin of 0.0 has the highest count of greater than 700 cases, this is like almost 35% of the total volume. Higher cash turnover rate is important as it shows a company's ability to replenish it quickly.

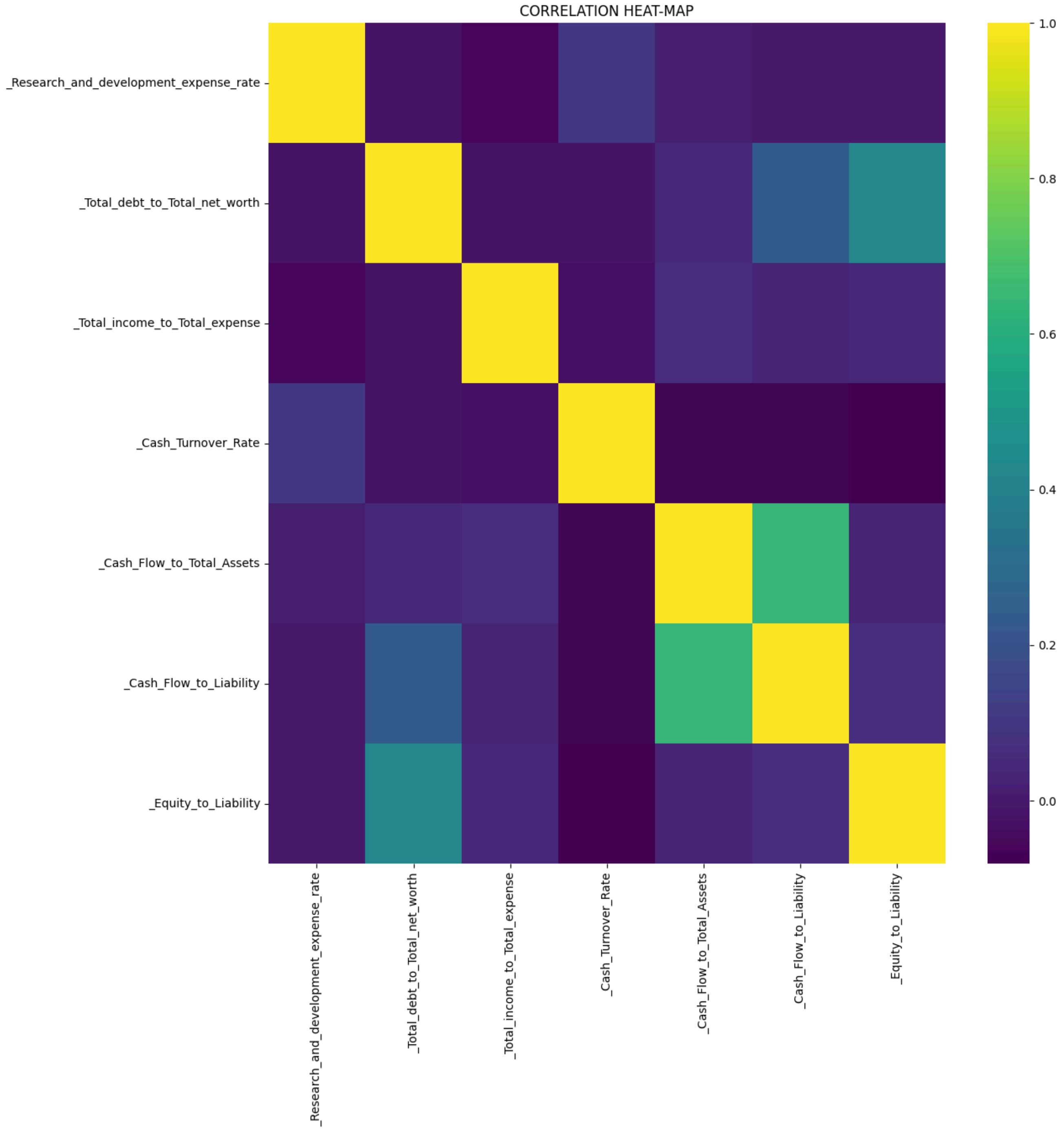




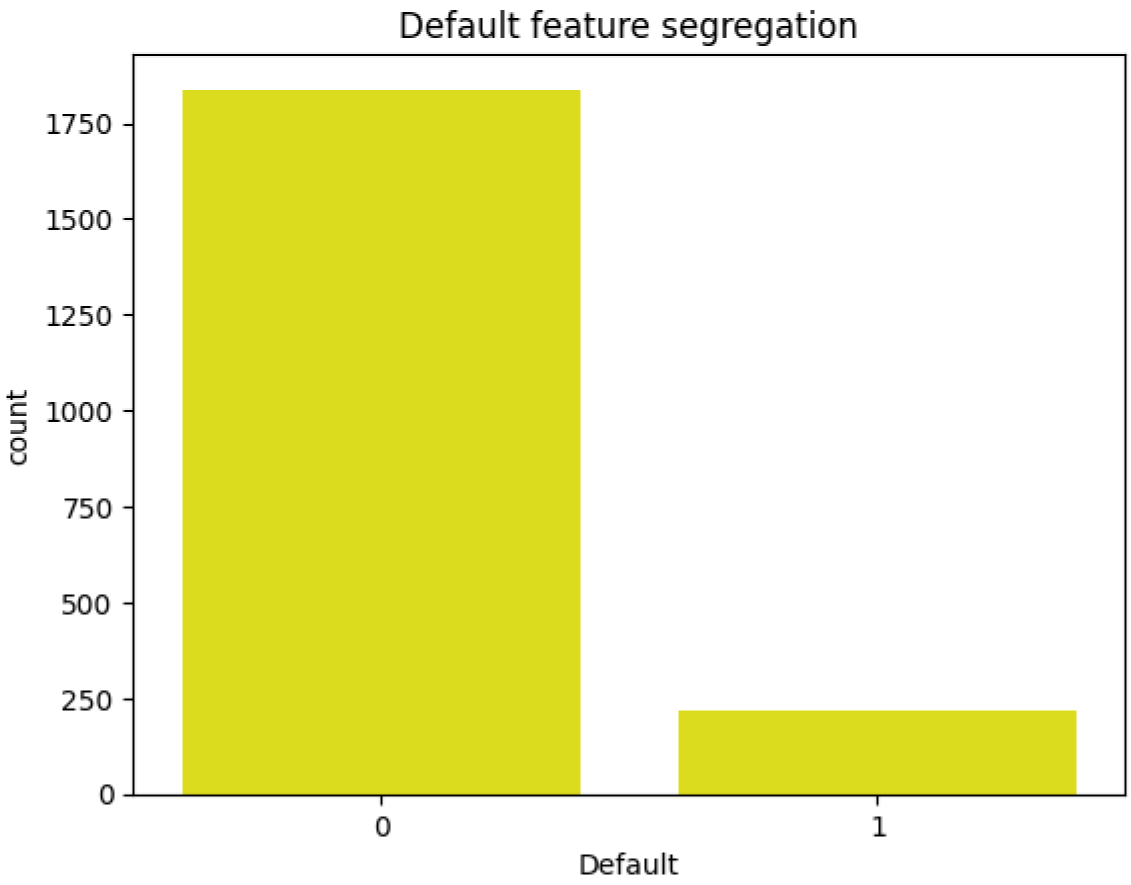
Scatter plot of Cash flow to assets and liabilities is shown on the left. There is a mild upwards trend, the denser spot between 0.4 and 0.6 would essentially have the trend line pass through it.

A higher value is desirable, generally as more assets means more credibility, but we should also not forget that the current suite of companies we have also garner more liabilities at probability slightly greater than a coin toss.

Heat map of the variables that were important from statsmodels logistic regression classifier model. -- The heat map shows that there is not much going on apart from the pair of features _cash_flow_to_assets and _cash_flow_to_liailites, this is what we saw previously as well. There is a correlation but manageable.



Finally, the “default“ feature, the one that is being predicted or classified.



The dataset seems to be skewed going in. This will definitely hinder the classification performance of “class 1”, where “class 0” will perform better.

Data Split up

Dataset was split up as per instructions in the ratio of 67:33 for train and test. We used a random state of 42 (*random_state=42*).

After splitting up the count is 1378 for the train dataset and 680 for the test dataset.

Building Models

What has been done till now

- We have treated the missing values with appropriate methods
- We have refrained from treating the outliers due to a considerable amount of volume present.
- We have removed features that provide us with very little information to make models work(the reasons discussed range from place-holder values, correlation and no meaning cell instances).

  A) Co_Code

  B) Co_Name

  C) _Per_Share_Net_profit_before_tax_Yuan_

  D) _Operating_Funds_to_Liability

  E) _Retained_Earnings_to_Total_Assets

  F) _Net_Income_Flag

  G) _Liability_Assets_Flag

- After removing from a list of 58 features we are reduced to 51 features.
- The feature 'Default' is our target variable.

Logistic Regression

Let's build a logistic regression classification model. We will use the train dataset to train and the test will be our performance validation set.

Since the scales of features are not consistent, we will be performing scaling to get them in order. This will help models like logistic regression to classify better over an unscaled model. Scaling technique used is Z-Score.

Applying Logistic regression, we are provided with the below report.

The “const” feature that must be added when we are working with Logistic regression. It represents the constant.

```
                           Logit Regression Results
==============================================================================
Dep. Variable:                Default   No. Observations:                 1378
Model:                          Logit   Df Residuals:                     1327
Method:                           MLE   Df Model:                           50
Date:                Sat, 24 Feb 2024   Pseudo R-squ.:                  0.4441
Time:                        10:06:52   Log-Likelihood:                -267.09
converged:                      False   LL-Null:                       -480.46
Covariance Type:            nonrobust   LLR p-value:                 3.122e-61
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
const         -6.6408   1780.489     -0.004      0.997   -3496.336    3483.054
x1             0.1295      0.126      1.026      0.305      -0.118       0.377
x2             0.3667      0.114      3.218      0.001       0.143       0.590
x3            -0.5213      1.210     -0.431      0.666      -2.892       1.849
x4            -0.1142      0.543     -0.210      0.833      -1.178       0.950
x5            -0.0591      0.154     -0.383      0.702      -0.362       0.243
x6            -0.0211      0.298     -0.071      0.944      -0.605       0.563
x7             0.0294      0.145      0.202      0.840      -0.255       0.314
x8            -0.1076      0.149     -0.720      0.472      -0.400       0.185
```

```
x9           -0.1041      0.098     -1.063      0.288     -0.296      0.088
x10          -0.1432      0.135     -1.058      0.290     -0.409      0.122
x11          -0.8643      2.172     -0.398      0.691     -5.122      3.393
x12           0.2241      0.473      0.473      0.636     -0.704      1.152
x13           0.1198      0.156      0.767      0.443     -0.186      0.426
x14          -4.0956      2.234     -1.834      0.067     -8.473      0.282
x15          -0.0173      0.078     -0.223      0.824     -0.169      0.135
x16           0.0132      0.079      0.166      0.868     -0.142      0.169
x17           4.8684      1.061      4.589      0.000      2.789      6.948
x18          -0.1507      0.511     -0.295      0.768     -1.153      0.852
x19          -0.5900      0.342     -1.723      0.085     -1.261      0.081
x20           0.0325      0.369      0.088      0.930     -0.690      0.755
x21          -1.0416      0.714     -1.459      0.145     -2.441      0.358
x22         -14.7846    1.6e+04     -0.001      0.999  -3.15e+04   3.14e+04
x23          -0.0378      0.120     -0.315      0.752     -0.273      0.197
x24           0.1944      0.108      1.808      0.071     -0.016      0.405
x25          -0.4859      0.299     -1.626      0.104     -1.072      0.100
x26           0.0832      0.179      0.464      0.643     -0.268      0.435
x27          -0.6663   5.42e+04  -1.23e-05      1.000  -1.06e+05   1.06e+05
x28          -0.0467      0.247     -0.189      0.850     -0.530      0.437
x29          -0.4174      0.227     -1.836      0.066     -0.863      0.028
x30           0.8159      2.629      0.310      0.756     -4.337      5.968
x31           0.0614      0.075      0.814      0.415     -0.086      0.209
x32          -0.1809      0.168     -1.075      0.282     -0.511      0.149
x33           0.1523      0.144      1.060      0.289     -0.129      0.434
x34          -0.0889      0.178     -0.500      0.617     -0.437      0.260
x35          -1.7299      0.550     -3.145      0.002     -2.808     -0.652
x36           0.2117      0.169      1.251      0.211     -0.120      0.543
x37          -0.0299      0.135     -0.222      0.824     -0.294      0.234
x38          -0.0046      0.131     -0.035      0.972     -0.261      0.252
x39          -0.3124      0.135     -2.314      0.021     -0.577     -0.048
```

```
x40             1.2170   2.39e+04    5.1e-05      1.000   -4.68e+04    4.68e+04
x41             1.9079      0.621      3.071      0.002       0.690       3.125
x42            -5.1243      1.181     -4.338      0.000      -7.439      -2.809
x43            -0.1145      0.344     -0.333      0.739      -0.789       0.560
x44            -0.5897      0.427     -1.380      0.167      -1.427       0.248
x45            -0.2659      0.175     -1.519      0.129      -0.609       0.077
x46            -0.0187      0.087     -0.214      0.830      -0.190       0.152
x47             0.0279      0.086      0.324      0.746      -0.141       0.197
x48            -0.0044      0.135     -0.032      0.974      -0.269       0.260
x49             0.0524      0.081      0.645      0.519      -0.107       0.212
x50            -7.0374      1.410     -4.991      0.000      -9.801      -4.274
==============================================================================
```

The above results produce a variety of information, from which we can cherry pick based on our needs. We also use this to tweak the model as required.

First off, the "converged: False" part is screaming for attention. This means the algorithm couldn't find the best-fit line for the data within the given iterations.

A Pseudo R-squared of “0.4441” tells us that your model explains a decent chunk of the variance in the dependent variable (Default), but there's still room for improvement.

Significant predictors based on P-value (less than 0.05, which is the chosen significance level). These variables significantly impact our dependent variable. We only have a handful of features out of 51 total. They are “x2“, “x17“, “x35“, “x39“, “x41“ and “x42“ who show significance and are important players in predicting Default.

The significant features are as follows

- “_Research_and_development_expense_rate“
- “_Total_debt_to_Total_net_worth“
- “_Total_income_to_Total_expense“
- “_Cash_Turnover_Rate“
- “_Cash_Flow_to_Total_Assets“
- “_Cash_Flow_to_Liability“
- “_Equity_to_Liability“

Accuracy and Precision-

Metric	Value
Accuracy	90.44%
True Positives (TP)	32
True Negatives (TN)	583
False Positives (FP)	30
False Negatives (FN)	35
Sensitivity (Recall)	≈ 47.76%
Specificity	≈ 95.11%
Precision	≈ 51.61%
F1 Score	≈ 49.54%

The model is highly accurate (90.44%), but this is largely due to its ability to identify the overwhelming number of non-defaults (0's).

The cut-off chosen was "0.4", This is a good balance where we are not sacrificing accuracy to a great extent. The standard cut-off is generally 0.5, if we want to take it to an extreme situation 0.3 or lower is generally opted.

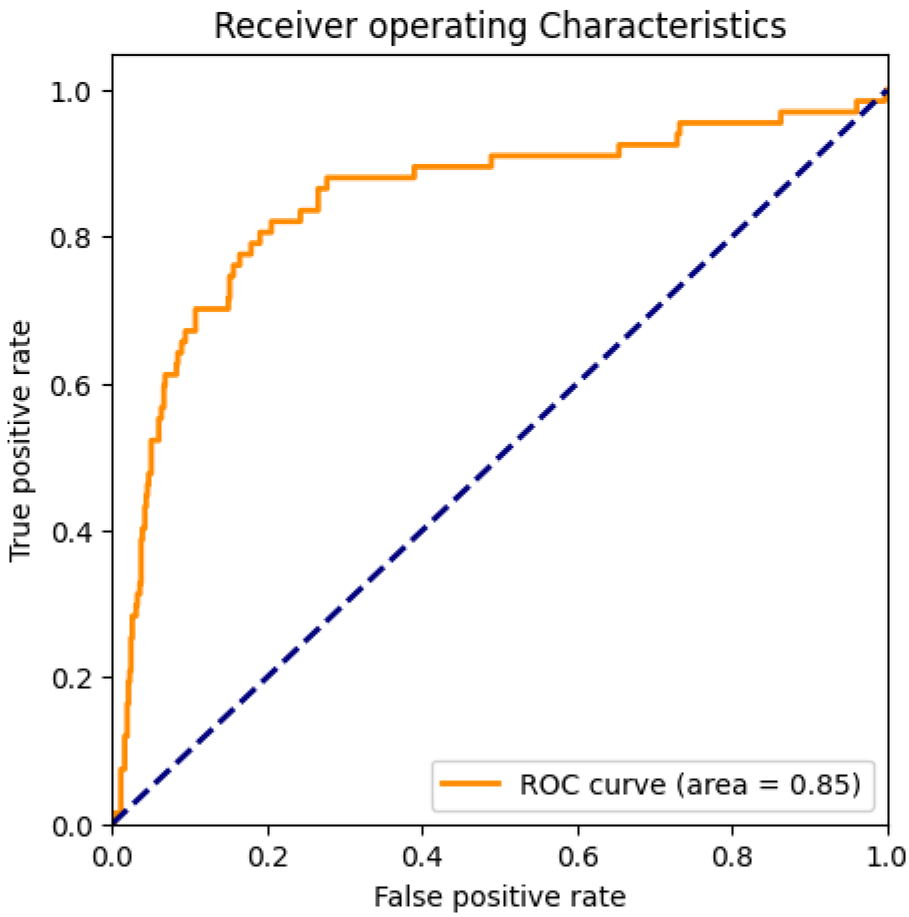
Sensitivity is low, meaning the model struggles to identify all actual defaults (1's) correctly, which is crucial in skewed datasets.

Specificity is very high, which is expected in a skewed dataset with many more non-defaults, the model is good at identifying these.

Precision is moderate, indicating that when the model predicts a default, it is correct half the time.

Our model is great at spotting who won't default but not as good at catching those who will.

ROC Curve –



The ROC curve area of 0.85 suggests that the model has a more than average performance in distinguishing between the classes, which is meaningful given the imbalanced dataset.

Our model has performed admirably within the situations that we have, it is capable of detecting those that won't default from those. One caveat is that the model ability to find those who do default is somewhat questionable as the available pool of defaulters is less.

Random Forest Classifier

Random forest classifier is a type that comes under the techniques called ensemble methods. The idea is that instead of having one complex model that does everything, we split multiple simpler models(even lower performance models if need be), combine them together as one single entity and then make predictive models.

We will use “grid search“ to help us get closer to the optimal parameters. Applying Random Forest Classifier we get the following performance metrics.

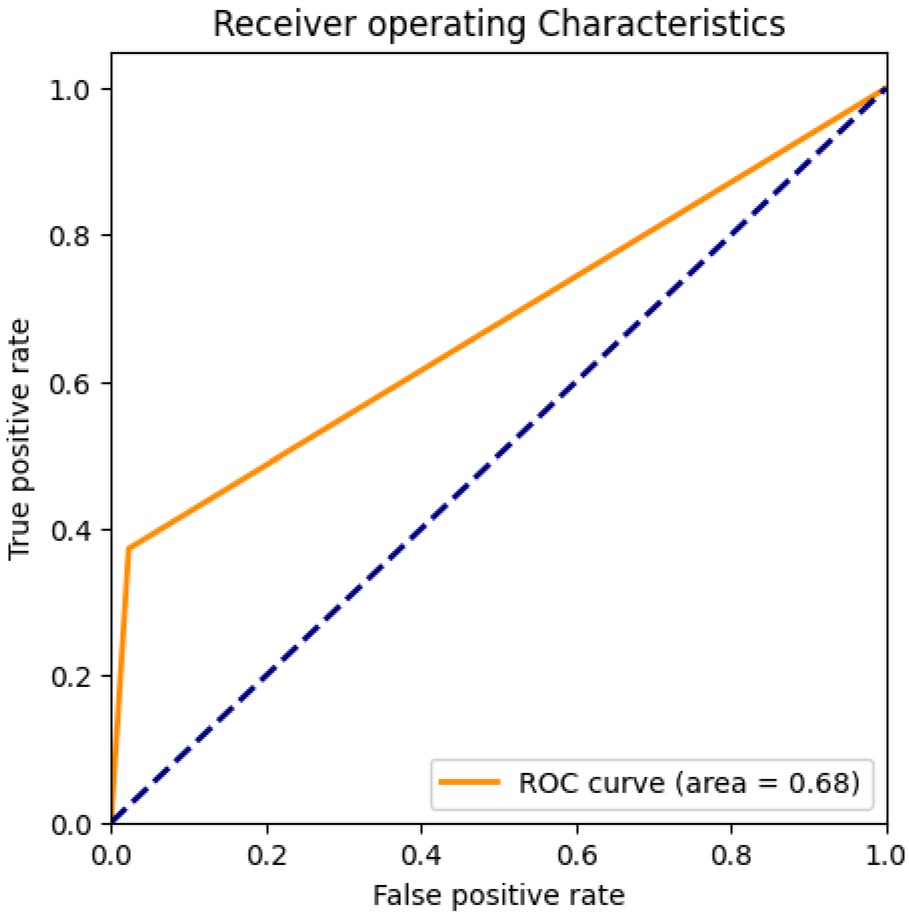
Metric	Value
Accuracy	91.76%
True Positives (TP)	25
True Negatives (TN)	599
False Positives (FP)	14
False Negatives (FN)	42
Sensitivity (Recall)	≈ 37.31%
Specificity	≈ 97.73%
Precision	≈ 64.10%
F1 Score	≈ 47.17%

Random forest has higher accuracy(91.76% vs. 90.44%), indicating it's better at making correct predictions overall. While logistic regression was better at identifying the positive cases, random forest is better at identifying the negative cases.

Precision has also improved with random forest by a considerable amount(64.10% vs. 51.61%), indicating a higher proportion of its positive predictions are correct.

Random forest provides a slight improvement in terms of accuracy, specificity and especially precision, it falls short in terms of recall and the number of true positives it captures compared to the logistic regression model with a cutoff of 0.4. This suggests that the logistic regression model at a 0.4 cutoff is more balanced in catching positive cases without greatly sacrificing precision.

ROC Curve –



The model has a fair ability to distinguish between the classes (with an AUC of 0.68).

There is definitely room for improvement, as models with AUC closer to 1 are more desirable.

The Random forest model has pro's to it, but it has sacrificed(which are the cons) quite a bit in other areas to get here. The model performance is average and would not sustain for long.

Linear Discriminant Analysis

Linear Discriminant Analysis is a method used for both classification and dimensionality reduction. In classification, LDA tries to find a decision boundary around each cluster of a class in such a way that it maximizes the distance between the means of different classes.

We will be using grid search to find the optimal parameter values for this model. Applying Linear Discriminant analysis post grid search we get the following performance parametrics.

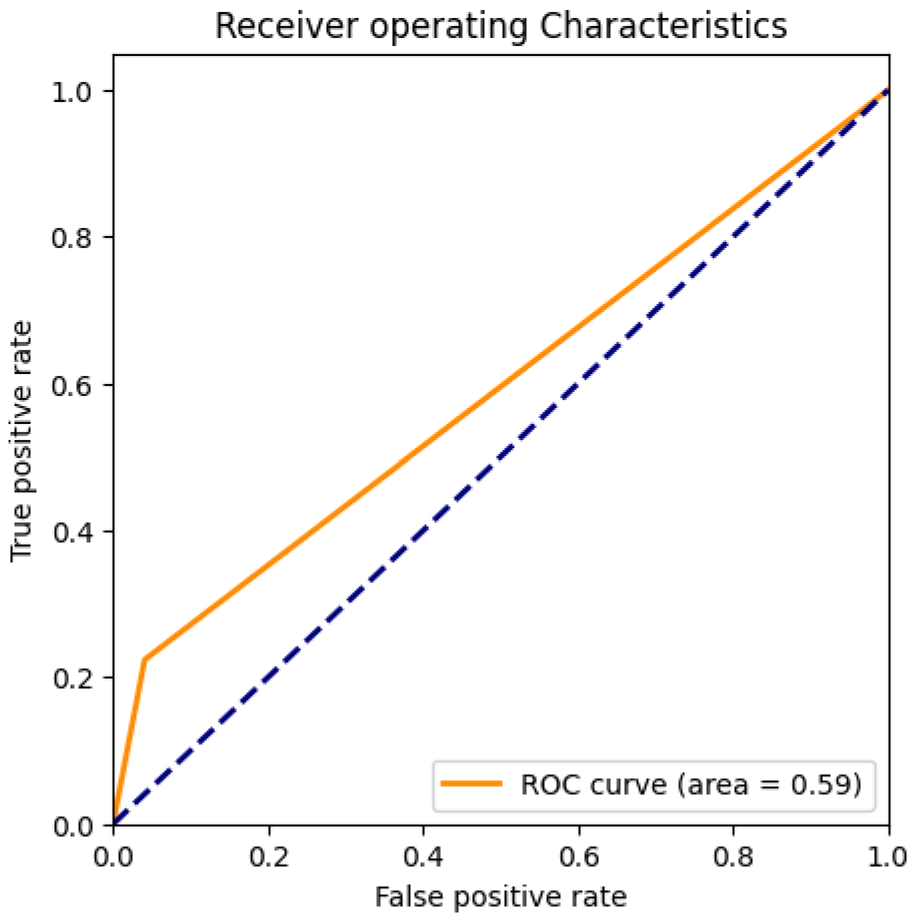
Metric	Value
Accuracy	89.85%
True Positives (TP)	23
True Negatives (TN)	588
False Positives (FP)	25
False Negatives (FN)	44
Sensitivity (Recall)	≈ 34.33%
Specificity	≈ 94.92%
Precision	≈ 47.92%
F1 Score	≈ 40.00%

The model has a recall of 34.33%, meaning it identified one-third of all actual positive cases. The specificity is 94.92%, showing that the model is quite good at identifying negative cases.

The precision is 47.92%, which means that when the model predicts an instance as positive, it is correct about half the time.

In summary, while the LDA model is fairly accurate and has high specificity, its recall, precision, and F1 score suggest that there is a significant number of positive cases it's not capturing. This could be due to the nature of the data.

ROC Curve –



An AUC of 0.59 suggests that the model is slightly better than a random guess (which would have an AUC of 0.5), There is room for improvement. The shape of the ROC curve shows that as the threshold for predicting a positive class decreases, the model identifies more true positives but also more false positives.

Comparison of performance

The Good:

Random Forest has the highest accuracy and specificity, making it reliable for predicting negatives and overall classification.

Logistic Regression has the best ROC AUC, which means it generally does a better job at distinguishing between classes.

The Bad:

LDA's ROC AUC is borderline random, indicating it's not a strong model for the task.

All models have low sensitivity, meaning they struggle to identify true positives (a critical flaw since false negatives are costly for us).

Logistic Regression and LDA have particularly low precision, indicating a higher rate of false positives within their positive predictions.

Logistic Regression might be the best model given our situation due to its highest ROC AUC and a better balance between sensitivity and precision. However, there are things that we have to improve.

Random Forest model did perform admirably, despite its precision, it did not catch enough positives to be reliable on its own for risk analysis.

Conclusion and Recommendations

- Logistic regression model is the best bet out of the three models.
- From a model training and fitting perspective it is preferable to have non-skewed data(default in this case) that will help the model in performing better. This inherently reduces bias and overfitting.
- We have some features like _Net_income_flag, which is full of zeros, we have to understand why it is the case.
- The scale of values present within the features given varies to a great extent, it is advisable to have categories. This not only helps in building the model, but also makes data collection more streamlined.
- Having segmented data also helps in handling outliers in a more coherent way.
- There were some missing values in the data-set, which is not desirable. Working on this could help us in the future.
-