**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, 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.

**EXPLORATORY DATA ANALYSIS (EDA)**

Data has 58 variables and 2058 rows. There are 4 integer type,53 float and 1object type data point. Here is a snapshot of the data.

A screenshot of a computer

Description automatically generated

There is no duplicate data in the dataset.

A screenshot of a computer

Description automatically generated

Here is a snapshot of data description. AS we can see from the snapshot data has few missing values and data also have outliers.

A black text on a white background

Description automatically generatedA screenshot of a computer

Description automatically generated

A white text with black text

Description automatically generated with medium confidence

We check for missing values and find that some columns have missing values, but most don’t. Here is a snippet of the column info.

We also check for weights of the target variable in the data. We find that data is imbalanced.

**Outlier Treatment**

A graph with lines and dots

Description automatically generated

As we can see from the snippet that all the columns have outliers. We now define the outliers and check for outliers in each column.

A screenshot of a computer screen

Description automatically generated We can see from the snippet the number of outliers per column. We will not impute these with upper bound or lower bound. We will convert them to NaN values and then use KNN imputer to impute previous and new missing values.

**Missing value treatment**

A screenshot of a computer

Description automatically generated

We now scale the data and prepare it for KNN- imputer.

We then split the data in with test size of 33% and random state =42. We also stratify the date to capture the imbalances in the data in our test and train datasets.

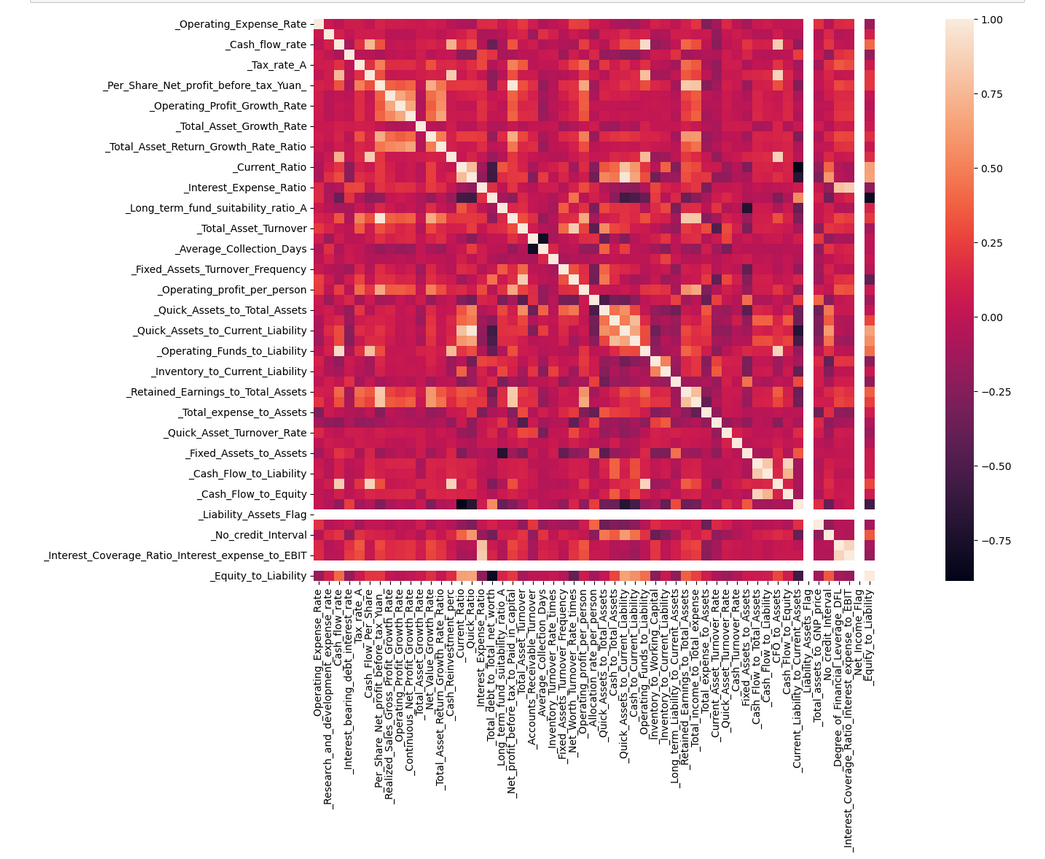
We then use KNN-imputer with n\_neighbors=5 and impute missing values, here is a snapshot of the data after the treatment.

A screenshot of a graph

Description automatically generated

After this there is no missing value present in the dataset.

Multivariate Analysis



As we can see from the heatmap that we have a few columns which are correlated. WE will remove those in further workflow.

**Train Test Split**

As we have previously split the data into train and test during application of KNN-imputer, we will now just split the data into predictor and target variables.

**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.**

A screenshot of a computer

Description automatically generated

We have to work on feature selection and remove values which are correlated. To ensure there is no multicollinearity in the data we use VIF and we eliminate any variable which have VIF of more than 5.

We drop each column with VIF >=5 one by one and measure VIF at the end of removal.

A screenshot of a computer

Description automatically generated

Here is the snapshot of the VIF chart after removing ‘\_Net\_profit\_before\_tax\_to\_Paid\_in\_capital’ which had the highest VIF. We do this process until we find that our VIF chart doesn’t have values more than 5.

A screenshot of a computer screen

Description automatically generated

Here is the data description after removing 12 columns which had VIF of more than 5.

We then join the target and predictor variables as statsmodel requires that.

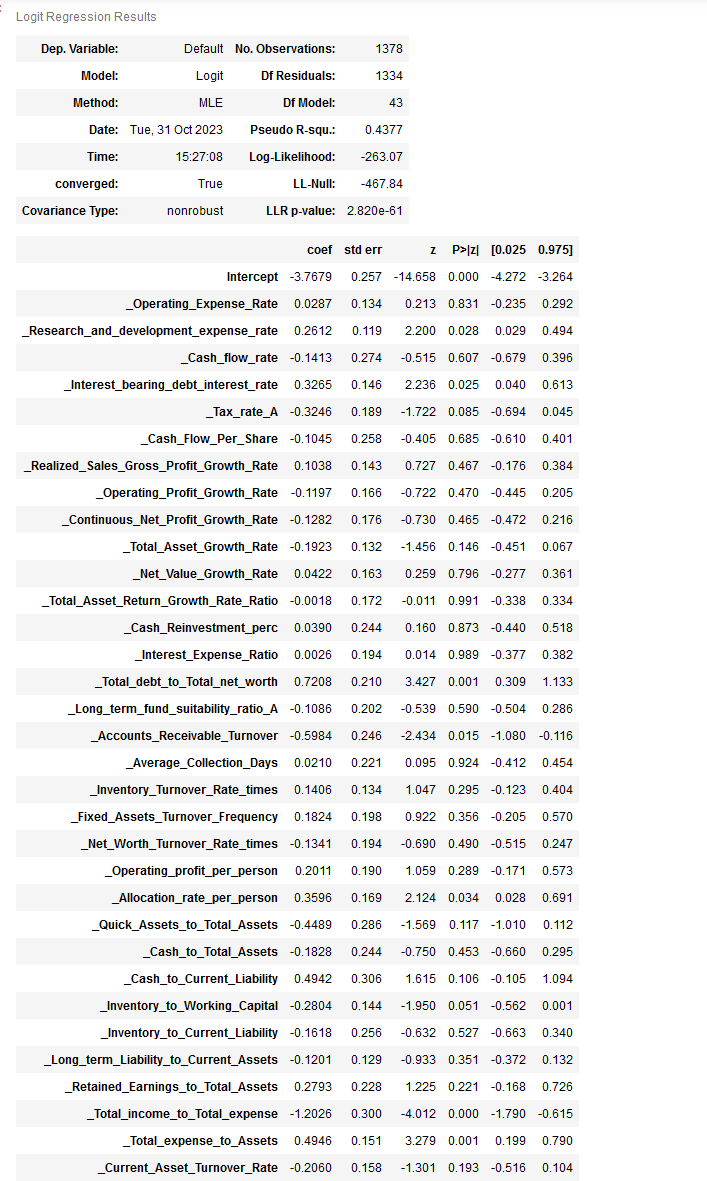
We now create model with these variables.

Model 1:

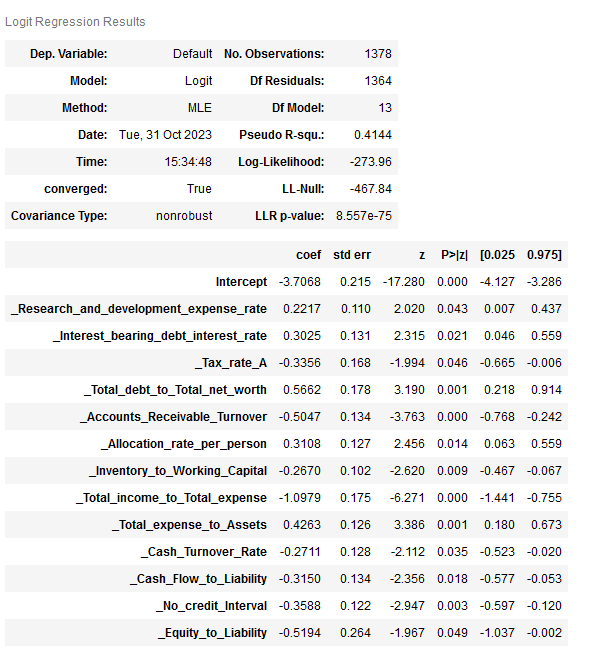
Here is the equation we used for model 1

f\_1= 'Default ~ \_Operating\_Expense\_Rate + \_Research\_and\_development\_expense\_rate + \_Cash\_flow\_rate + \_Interest\_bearing\_debt\_interest\_rate + \_Tax\_rate\_A + \_Cash\_Flow\_Per\_Share + \_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 + \_Interest\_Expense\_Ratio + \_Total\_debt\_to\_Total\_net\_worth + \_Long\_term\_fund\_suitability\_ratio\_A + \_Accounts\_Receivable\_Turnover + \_Average\_Collection\_Days + \_Inventory\_Turnover\_Rate\_times + \_Fixed\_Assets\_Turnover\_Frequency + \_Net\_Worth\_Turnover\_Rate\_times + \_Operating\_profit\_per\_person + \_Allocation\_rate\_per\_person + \_Quick\_Assets\_to\_Total\_Assets + \_Cash\_to\_Total\_Assets + \_Cash\_to\_Current\_Liability + \_Inventory\_to\_Working\_Capital + \_Inventory\_to\_Current\_Liability + \_Long\_term\_Liability\_to\_Current\_Assets + \_Retained\_Earnings\_to\_Total\_Assets + \_Total\_income\_to\_Total\_expense + \_Total\_expense\_to\_Assets + \_Current\_Asset\_Turnover\_Rate + \_Quick\_Asset\_Turnover\_Rate + \_Cash\_Turnover\_Rate + \_Fixed\_Assets\_to\_Assets + \_Cash\_Flow\_to\_Liability + \_Cash\_Flow\_to\_Equity + \_Current\_Liability\_to\_Current\_Assets + \_Total\_assets\_to\_GNP\_price + \_No\_credit\_Interval + \_Degree\_of\_Financial\_Leverage\_DFL + \_Equity\_to\_Liability'

Snapshot of model summary



We can see from the snapshot that model 1 still has some columns in which p value is not significant. We remove those at once to build a better model with less predictors.



Here is our model with all collinearity removed.

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

Here is our classification report and confusion matrix of the train data with 0.5 threesold.

A screenshot of a computer screen

Description automatically generated

We checked that our threshold is 0.07669278006648413. We change our threshold and check our model performance on train data again.

A blue squares with numbers and a white background

Description automatically generated

As we can see that even though precision has decreased, recall has improved greatly.

We will now check our model performance on test data with changed threshold.

A screenshot of a graph

Description automatically generated

As we can see from the confusion matrix and classification report that the model performance was consistent across train and test data.

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

We now build a random forest model with hyper parameter turning. We use gridsearchcv to search for best parameter.

A screenshot of a computer program

Description automatically generated

{'max\_depth': 3,

'min\_samples\_leaf': 15,

'min\_samples\_split': 45,

'n\_estimators': 25}

This is the best parameter, and we build our model with these hyperparameters.

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

A screenshot of a computer screen

Description automatically generated

As we can see our precision is good, but recall has dropped. We check test data performance.

A screenshot of a graph

Description automatically generated

We can see precision and recall is almost consistent with train data. Recall is very low. We now check if changing threshold improves the model performance.

A screenshot of a graph

Description automatically generatedA screenshot of a graph

Description automatically generated  
We can see that our model performance has not changed.

A blue and orange line graph

Description automatically generated  
We can see that the model performance is not that good, recall is very low. So, we will discard this model.

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

We now build a LDA model.

A screenshot of a computer code

Description automatically generated

We fit the train data and do the prediction on test data.

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

We check the model performance on train and test data.

A screenshot of a graph

Description automatically generated

We can see from the confusion matrix and classification report that our precision and recall has dropped. We now check test performance.

A screenshot of a graph

Description automatically generated