FRA
Part A
Project Report

Preetam Sarmah

PGPDSBA.O.JAN23.A
January’ 23
Date: 19/11/2023

Table of Contents

List of Figures

References: *PreetamSarmah_FRA_PartA.ipynb*

Part A

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.

1.1) Outlier Treatment

As this is a Finance data, the extreme outlier might be valid or might have been a data collection error.

The count of outlier is first captured as all data points which are either less than Q1 – 1.5 IQR or data points greater than Q3 + 1.5 IQR.

Where Q1 = Quantile 1 or 25 percentile

Q3 = Quantile 3 or 75 percentile

and IQR = Q3 – Q1

There is a total of 11105 data points which we detect as outliers, based on the above approach.

To treat these outliers, we convert them to Nan or nulls, and would later impute them together with any other missing data points.

1.2) Missing Value Treatment.

The original data set has 298 missing data points.
As for the outlier treatment we have converted the outliers into missing data, the total count of missing data points in the data set is now 11403.
So, around 10 % of the data is now missing data.

We further notice that missing value percentage in the predictor variables is under 30%, so we won't be dropping any of the predictor variable columns.
The remaining missing values will be imputed using a KNN imputer set to 5 nearest neighbours.

1.3) Univariate and Bivariate Analysis.

Univariate Analysis

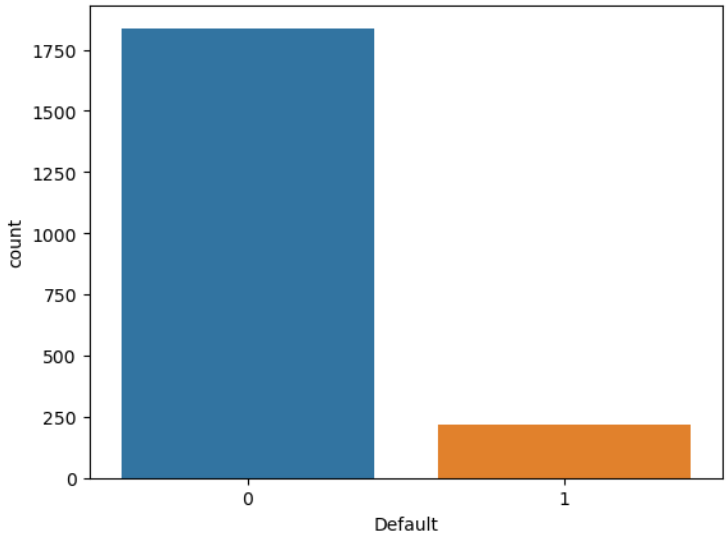


Fig 1 Count of defaults

It can be noted that defaulters class compromises a small fraction of the total customers around 11% are defaulters.

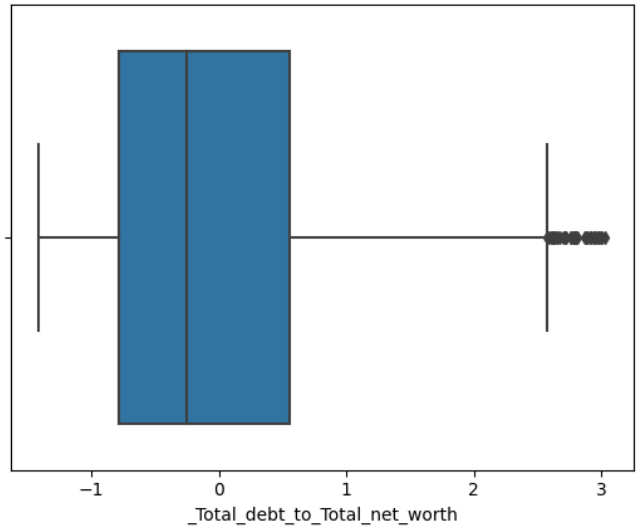


Fig 2 Boxplot of Total Debt to Total Net worth

The plot show a that most customers have a positive Total debt to total net worth

Bivariate Analysis

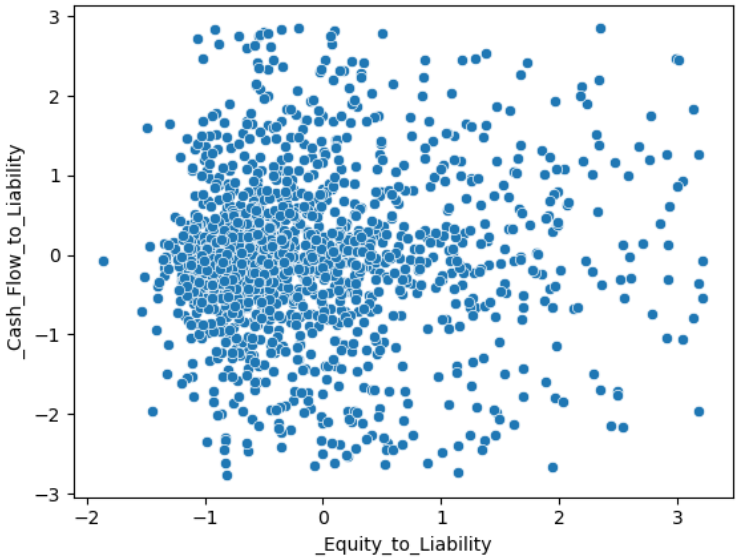


Fig 3 Cash Flow to Liability vs Equity to Liability

This plot shows not much relation between the two variables, though it can be noted that a most customers have a -1 to 0 equity to liability and -1 to 1 cash flow to liability.

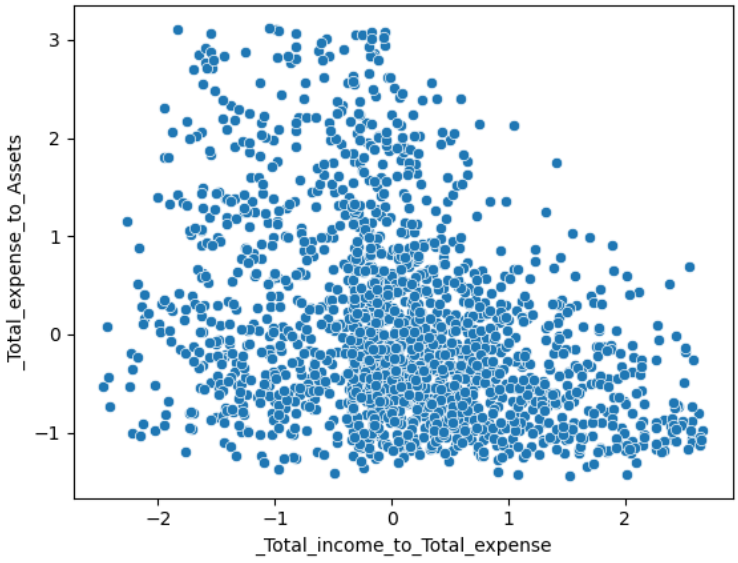


Fig 4 Total Expense to Assets vs Total income to Total expense

For the plot it can be noted that, individuals with high total income to total expense have a lower total expense to asset. Though not a strong relation, the two variables show a slight negative relation to one another.

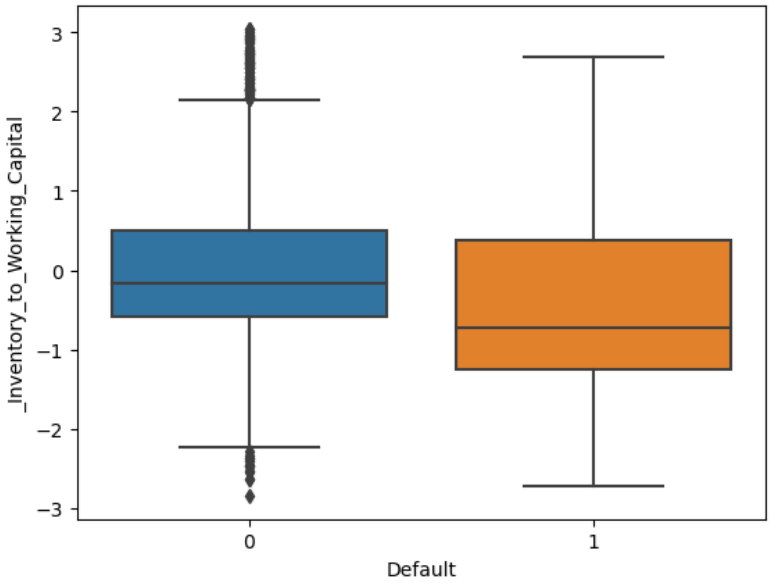


Fig 5 Inventory to working capital by Default

It can be noted that the defaulters on average have less inventory to working capital. Which can be expected as, it is the measurement of how much of a company's working capital is funded by its inventory, less this value high are the chances to default.

1.4) Train Test Split.

Before the dataset is split, we remove the following columns as they won't be help with model building process, 'Co_Code', 'Co_Name', '_Net_Income_Flag','_Liability_Assets_Flag'.

The Dataset is split into Train and Test data in the ratio 67:33.

As the number of defaults is very low around 11%, we call stratify on the Response variable, i.e., Default so that the ratio of defaults to non-defaults is preserved in both train and test data.

The random seed is set 42.

The train data has 1378 rows and 54 columns.

The test data has 680 rows and 54 columns.

1.5) Logistic Regression Model.

The Logistic Regression model is prone to outlier influence , hence we must drop the predictor variables one after the other based on the Variance Inflation Factor in descending order until all the predictor variables have a VIF of less than 5.

As the model building would be done on Train data, after VIF treatment we transform the train data into 1378 rows and 43 columns.

Using the feature selection, we end up with 14 features i.e., 1/3 of 43, for the logistic regression formula, but we notice that few of these predictor variables are insignificant i.e, P>|z| is above 0.05.

The final formula for the Logistic regression model is

f_1 = 'Default ~ _Research_and_development_expense_rate + _Total_debt_to_Total_net_worth + _Accounts_Receivable_Turnover + _Quick_Assets_to_Total_Assets + _Inventory_to_Working_Capital + _Total_income_to_Total_expense + _Total_expense_to_Assets + _Cash_Turnover_Rate + _Cash_Flow_to_Liability + _No_credit_Interval + _Equity_to_Liability'.

Logit Regression Results

Dep. Variable:	Default	**No. Observations:**	1378
Model:	Logit	**Df Residuals:**	1366
Method:	MLE	**Df Model:**	11
Date:	Sun, 19 Nov 2023	**Pseudo R-squ.:**	0.4071
Time:	18:32:14	**Log-Likelihood:**	-277.40
converged:	True	**LL-Null:**	-467.84
Covariance Type:	nonrobust	**LLR p-value:**	6.935e-75

	coef	std err	z	P>|z|	[0.025	0.975]
Intercept	-3.6119	0.207	-17.431	0.000	-4.018	-3.206
_Research_and_development_expense_rate	0.2103	0.109	1.938	0.053	-0.002	0.423
_Total_debt_to_Total_net_worth	0.6431	0.180	3.577	0.000	0.291	0.996
_Accounts_Receivable_Turnover	-0.6001	0.142	-4.218	0.000	-0.879	-0.321
_Quick_Assets_to_Total_Assets	-0.4323	0.144	-3.008	0.003	-0.714	-0.151
_Inventory_to_Working_Capital	-0.2632	0.101	-2.600	0.009	-0.462	-0.065
_Total_income_to_Total_expense	-1.1782	0.155	-7.616	0.000	-1.481	-0.875
_Total_expense_to_Assets	0.4208	0.119	3.537	0.000	0.188	0.654
_Cash_Turnover_Rate	-0.3161	0.129	-2.444	0.015	-0.570	-0.063
_Cash_Flow_to_Liability	-0.2742	0.135	-2.031	0.042	-0.539	-0.010
_No_credit_Interval	-0.3644	0.126	-2.896	0.004	-0.611	-0.118
_Equity_to_Liability	-0.5596	0.268	-2.087	0.037	-1.085	-0.034

Fig 6 Logistic Regression Model Summary

Based on the true positive rate and the false positive rate the optimum threshold, for the above model is 0.09.

The final model uses only 11 features to predict if a datapoint is going to be a defaulter or not.

1.6) Logistic Regression Model Evaluation on Test Data.

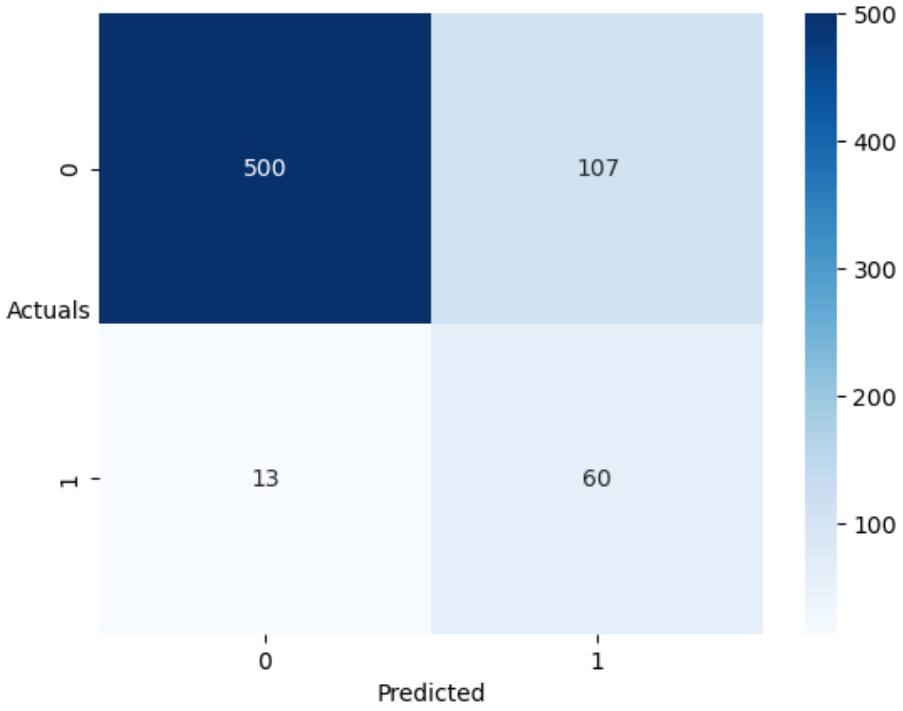


Fig 7 Logistic Regression Confusion Matrix

	precision	recall	f1-score	support
0.0	0.975	0.824	0.893	607
1.0	0.359	0.822	0.500	73
accuracy			0.824	680
macro avg	0.667	0.823	0.696	680
weighted avg	0.909	0.824	0.851	680

The Model has very good recall but poor precision for the default class.

Of the total defaults i.e., 73, the model is able to accurately predict 60 defaults, and for credit risk, this recall value is highly important as this gives us out of the actual defaulters how many the model is able to predict as default.

The model has an overall accuracy of 82.4%.

1.7) Random Forest Model

For Random Forests, checking VIF is not necessary as it is an ensemble technique where we combine multiple decision trees to make predictions.

Random Forests are robust o multi collinearity because they consider only a random subset of predictors at each split in decision tree.

Hence, we would be using our Train dataset, as it is.

The model is built with the following parameters, n_estimators = 100 and random_state = 42 on the train dataset.

1.8) Random Forest Model Evaluation on Test Data.

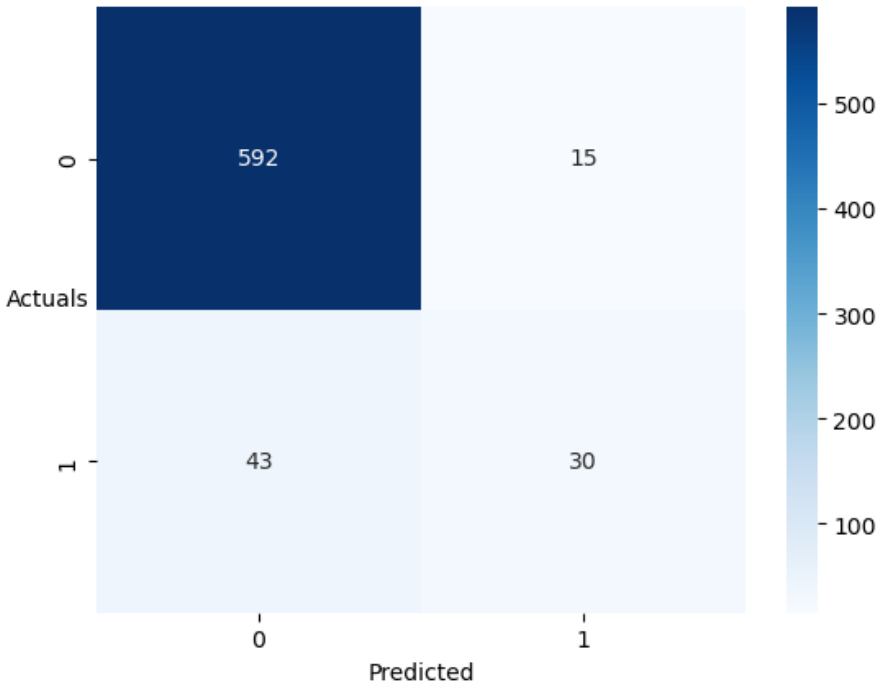


Fig 8 Random Forest Confusion Matrix

	precision	recall	f1-score	support
0.0	0.93	0.98	0.95	607
1.0	0.67	0.41	0.51	73
accuracy			0.91	680
macro avg	0.80	0.69	0.73	680
weighted avg	0.90	0.91	0.91	680

The Model has very poor recall but average precision for the default class.

Of the total defaults i.e., 73, the model is able to accurately predict 30 defaults, and for credit risk, this recall value is highly important as this gives us out of the actual defaulters how many the model is able to predict as default, hence this is not acceptable

The model has an overall accuracy of 91%.

1.9) LDA Model

In LDA Model, we don't need to check VIF, because LDA is not a regression-based technique, that assumes a relationship between predictor variables.

As unlike regression, it focuses on finding discriminant functions that maximizes class separation.

Hence, we would be using our Train dataset, as it is.

1.10) LDA Model Evaluation on Test Data.

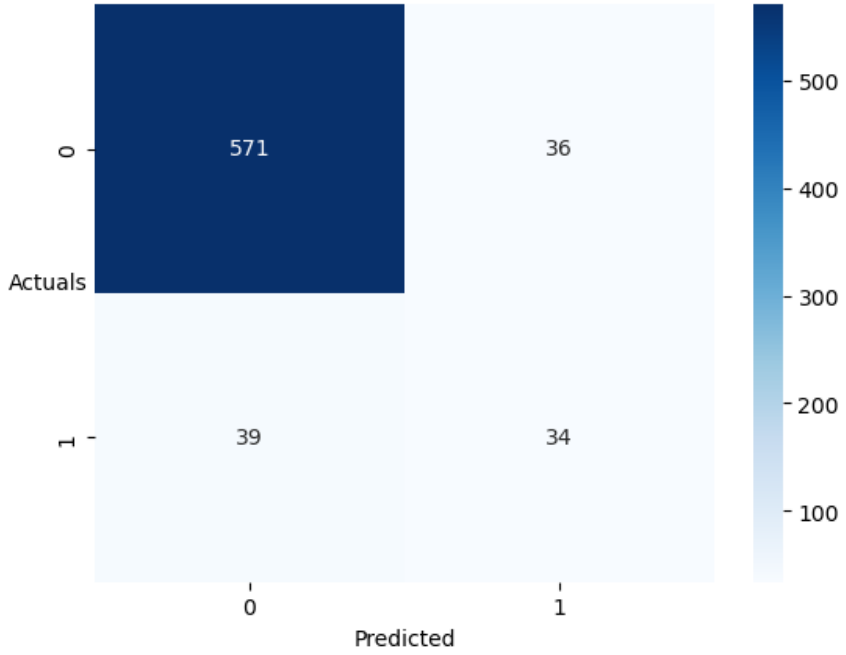


Fig 9 LDA Confusion Matrix

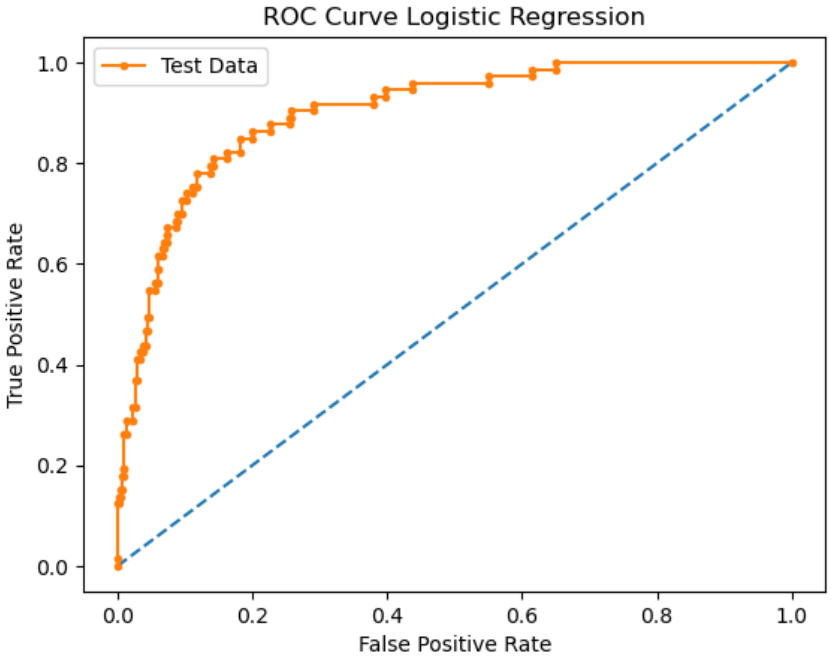
	precision	recall	f1-score	support
0.0	0.94	0.94	0.94	607
1.0	0.49	0.47	0.48	73
accuracy			0.89	680
macro avg	0.71	0.70	0.71	680
weighted avg	0.89	0.89	0.89	680

The Model has very average recall and average precision for the default class.

Of the total defaults i.e., 73, the model is able to accurately predict 34 defaults, and for credit risk, this recall value is highly important as this gives us out of the actual defaulters how many the model is able to predict as default, hence this is not acceptable

The model has an overall accuracy of 89%.

1.11) Logistic Regression vs Random Forest vs LDA



. Fig 10 ROC Curve Logistic Regression

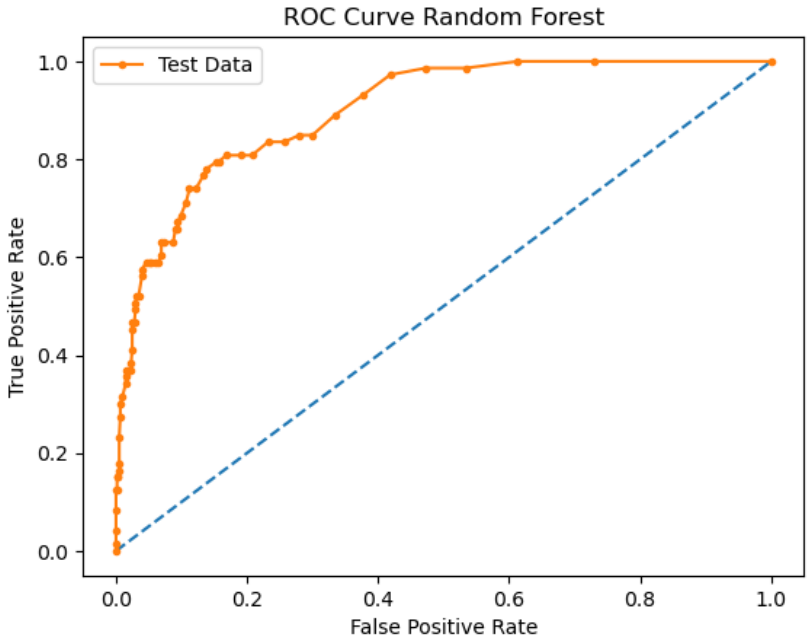


Fig 11 ROC Curve Random Forest

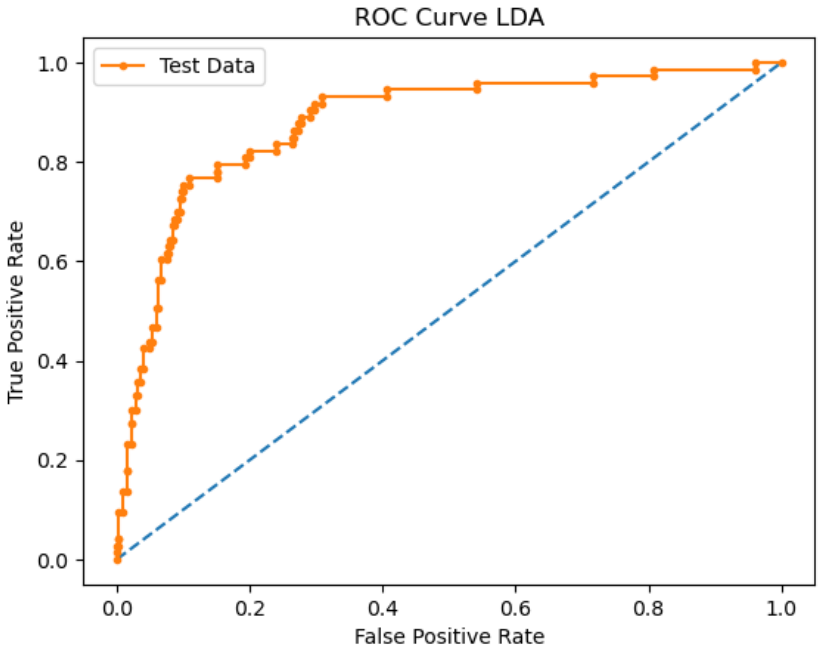


Fig 12 ROC Curve LDA

- All the 3 models have a good AUC score as exemplified by the ROC curves above
- But among the 3 models, the logistic regression preforms very well, as its able to accurately predict 82% of actual defaulters.
- But with regards to overall accuracy the Random Forest is better at 91% overall accuracy.

1.12) Conclusions and Recommendations.

- Defaults cost the bank a lot in terms of monetary, hence its always ideal to better predict the defaulters
- Among the models, the Logistic Regression performs really well when it comes to accurately predicting actual defaulters of the total defaulters.
- The model performance, can be improved if the data imbalances are treated, but as such in actuality the amount of defaulters in an organization will always be just a small percentage of the total customers.
- In terms of precision and overall accuracy, the Random Forest Model performs very well as out of the predicted defaulters, 67% of them are actually defaulters and has an overall accuracy of 91%.