INDIAN INSTITUTE OF TECHNOLOGY KANPUR

MTH416A: REGRESSION ANALYSIS

A PROJECT REPORT
ON

Bankruptcy Prediction Using Logistic Regression

Submitted by

Ananjay Kumar	**201267**
Manas Mishra	**201340**
Ritwik Vashishta	**201389**
Shivani Yadav	**201413**
Somesh Kr. Jha	**181147**

Under the Guidance of

Dr. Sharmishtha Mitra
Department Of Mathematics And Statistics,
IIT Kanpur

ABSTRACT

Estimating the risk of corporate bankruptcies is of large importance to creditors and investors. For this reason bankruptcy prediction constitutes an important area of finance and accounting research.The importance of the area is due in part to the relevance for creditors and investors in evaluating the likelihood that a firm may go bankrupt. In recent years artificial intelligence and machine learning methods have achieved promising results in corporate bankruptcy prediction settings. Therefore, in this study, we explore, build, and compare the different classification models. We have chosen the 'Polish Companies' bankruptcy data set and begin by carrying out data preprocessing and exploratory analysis where we impute the missing data values using some of the popular data imputation techniques like Mean, k-Nearest Neighbors and Multivariate Imputation by Chained Equations (MICE). To address the data imbalance issue, we apply Synthetic Minority Oversampling Technique (SMOTE) to oversample the minority class labels. Then, we build an appropriate model using Logistic Regression on our cleaned dataset and finally, analyze and evaluate the performance of the models on the validation datasets using several metrics such as accuracy, precision, recall, etc., and rank the models accordingly.

ACKNOWLEDGMENT

It is our esteemed pleasure to present a project on “Bankruptcy Prediction Using Logistic Regression”.

Any Achievement big or small should have a catalyst and constant encouragement and advice of valuable and noble minds for our efforts to bring out this project. The satisfaction that accompanies the successful completion of any task would be incomplete without mentioning those who made it possible because success is the epitome of hard work, determination and dedication.

We want to express our sincere gratitude to our instructor Dr. Sharmishtha Mitra for her constant help and support throughout the completion of the project. Without her valuable guidance and motivation, it was nearly impossible to work on this project as a team and understand the practical aspect of the course “MTH 416A: Regression Analysis”.

Last but not the least we are thankful to all faculty members and seniors without whose support at various stages, this project would not have materialized.

Ananjay Kumar
Manas Mishra
Ritwik Vashistha
Shivani Yadav
Somesh Kr. Jha

Contents

List of Tables

List of Figures

1 INTRODUCTION

1.1 Context

Bankruptcy prediction is a technique of forecasting and projecting on company financial distress of both public and firms. The purpose of predicting bankruptcy is fundamental in assessing the financial condition of a company and prospects in its operations. Corporate bankruptcy prediction is a very crucial phenomenon in economics. The financial soundness of a company is of great importance to the various actors and participants of the business cycle. The participants and interested parties include the policymakers, investors, banks, internal management, and the general public referred to as consumers. Accurate prediction of the financial performance of companies is of great importance to various stakeholders in making important and significant decisions concerning their relationship and engagement with companies. Financial distress is a global phenomenon that affects companies across all sectors of the economy.

Additionally, bankruptcy prediction is essential for investors as well as suppliers or retailers to the business. Credit lenders and investors need to evaluate the financial bankruptcy risk of a company before making an investment or credit-granting decisions to avoid a significant loss by banks and other credit lenders. A company's suppliers or retailers always conduct credit transactions with the company, and they also need to fully understand the company's financial status and make decisions on the credit transaction. To correctly predict a company's financial distress is of great concern to the various stakeholders of a company. Problems concerning bankruptcy have necessitated the need for studies to establish different stressors to companies to aid investors in making prudential investment decisions.

Corporate failures in significant economic companies have spurred research for better understanding to develop prediction capabilities that guide decision making in investments. Financial distress projections in companies are a product of available data from listed companies, public firms that have sunk. Available accounting ratios may be a vital indicator or signal to indicate danger. Typically, firms are quantified by many indicators that describe their business performance based on mathematical models constructed from past observations based on evidence from data.

Decisions of a corporate borrower on credit risk traditionally were exclusively based upon subjective judgments made by human experts, based on past experiences and some guiding principles. However, two significant problems associated with this approach include the difficulty to make consistent estimates and the fact that it tends to be reactive rather than predictive.

Bankruptcy prediction is of great importance to all participants in the insurance market, including insurance regulators, policyholders, agents, and insurance companies. As insurance products become more and more familiar to the public, they strengthen the consumers' willingness to buy products. However, as the service period of insurance products happens after the purchase of products, the consumer is very concerned when purchasing products of the insurance company about whether they will be able to pay in the future. Assessing the solvency of an insurance company in the future during the product service period is very important to the policyholder's purchase decision, and equivalently crucial to the operation of the insurance company.

The history of bankruptcy prediction includes application of numerous statistical tools which gradually became available, and involves deepening appreciation of various pitfalls in early analyses. Interestingly, research is still published that suffers pitfalls that have been understood for many years. Bankruptcy prediction has been a subject of formal analysis since at least 1932, when FitzPatrick published a study of 20 pairs of firms, one failed and one surviving, matched by date, size and industry, in The Certified Public Accountant. He did not perform statistical analysis as is now common, but he thoughtfully interpreted the ratios and trends in the ratios. His interpretation was effectively a complex, multiple variable analysis.

The purpose of the bankruptcy prediction is to assess the financial condition of a company and its future perspectives within the context of long-term operation on the market [4]. It is a vast area of finance and econometrics that combines expert knowledge about the phenomenon and historical data of prosperous and unsuccessful companies. Typically, enterprises are quantified by numerous indicators that describe their business condition that are further used to induce a mathematical model using past observations.

There are different issues that are associated with the bankruptcy prediction. Two main problems are the following: First, the econometric indicators describing the firm's condition are pro- posed by domain experts. However, it is rather unclear how to combine them into a successful model. Second, the historical observations used to train a model are usually influenced by imbalanced data phenomenon, because there are typically much more successful companies than the bankrupted ones. As a consequent, the trained model tends to predict companies as successful (majority class) even when some of them are distressed firms. Both these issues mostly influence the final predictive capability of the model.

To speak about the modern methods of approach for the field of bankruptcy prediction, it is worth noting that survival methods are now being applied. Option valuation approaches involving stock price variability have been developed. Under structural models, a default event is deemed to occur for a firm when its assets reach a sufficiently low level compared to its liabilities. Neural network models and other sophisticated models have been tested on bankruptcy prediction. Modern methods applied by business information companies surpass the annual accounts content and consider current events like age, judgements, bad press, payment incidents and payment experiences from creditors.

1.2 Objectives of the Study

In this project, we wish to predict Bankruptcy of Polish Companies using Logistic Regression. The key steps involved are as follows:

1. To deal with the problem of Missing Values using suitable Data Imputation Techniques.

2. To tackle the problem of Data Imbalance using suitable Resampling Methods.

3. To build the model using Logistic Regression.

4. To compare the accuracy of the model.

2 Methodology

In the previous section, we formally introduced the problem statement of bankruptcy prediction. In this section, we explain our step-by-step solution of how we achieved benchmark results for bankruptcy prediction. We started with introducing the Polish bankruptcy dataset and explaining the details of the dataset like features, instances, data organization, etc. Then, we delve into data preprocessing steps, where we state the problems present with the data like missing data and data imbalance, and explain how we dealt with them. Next, we introduce the classification models we have considered and explain how we train our data using these models. Later, we analyze and evaluate the performance of these models using certain metrics like accuracy, precision and recall.

2.1 Dataset Description

The dataset is about bankruptcy prediction of Polish companies. The data was collected from Emerging Markets Information Service (EMIS), which is a database containing information on emerging markets around the world. The bankrupt companies were analyzed in the period 2000-2012, while the still operating companies were evaluated from 2007 to 2013. The dataset is very apt for our research about bankruptcy prediction because it has highly useful econometric indicators as attributes (features) and comes with a huge number of samples of Polish companies that were analyzed in 5 different timeframes:
Based on the collected data five classification cases were distinguished, that depends on the forecasting period:

1. **1st Year:** The data contains financial rates from 1st year of the forecasting period and corresponding class label that indicates bankruptcy status after 5 years.

2. **2nd Year:** The data contains financial rates from 2nd year of the forecasting period and corresponding class label that indicates bankruptcy status after 4 years.
3. **3rd Year:** The data contains financial rates from 3rd year of the forecasting period and corresponding class label that indicates bankruptcy status after 3 years.
4. **4th Year:** The data contains financial rates from 4th year of the forecasting period and corresponding class label that indicates bankruptcy status after 2 years.
5. **5th Year:** The data contains financial rates from 5th year of the forecasting period and corresponding class label that indicates bankruptcy status after 1 years.

The dataset is summarized in Table 1 below:

Dataset Characteristic	Multivariate			
No. of features	64			
Number of Instances	**Data**	**Total Instances**	**Bankrupt Instances**	**Non-Bankrupt Instances**
	1st year	7027	271	6756
	2nd Year	10173	400	9773
	3rd Year	10503	495	10008
	4th Year	9792	515	9227
	5th Year	5910	410	5500
Feature Characteristics	Real Values			
Has missing data?	Yes			
Associated tasks	Classification			
Date donated	04-11-2016			

Table 1: Summary of Polish Bankruptcy Dataset

ID	Description	ID	Description
X1	net profit / total assets	**X33**	operating expenses / short-term liabilities
X2	total liabilities / total assets	**X34**	operating expenses / total liabilities
X3	working capital / total assets	**X35**	profit on sales / total assets
X4	current assests / short-term liabilities	**X36**	total sales / total assets
X5	[(cash + short-term securities + receivables - short-term liablities) / (operating expenses - depreciation)]*365	**X37**	(current assets - inventories) / long-term liabilities
X6	retained earnings / total assets	**X38**	constant capital / total assets
X7	EBIT / total assets	**X39**	profit on sales / sales
X8	book value of equity / total liabilities	**X40**	(current assets - inventory - receivables) / short-term liabilities
X9	sales / total assets	**X41**	total liabilities / (profit on operating activities + depreciation) * (12/365))
X10	equity / total assets	**X42**	profit on operating activities / sales
X11	(gross profit + extraordinary items + financial expenses) / total assets	**X43**	rotation receivables + inventory turnover in days
X12	gross profit / short-term liabilities	**X44**	(receivables * 365) / sales
X13	(gross profit + depreciation) / sales	**X45**	net profit / inventory
X14	(gross profit + interest) / total assets	**X46**	(current assets - inventory) / short-term liabilities
X15	(total liabilities * 365) / (gross profit + depreciation)	**X47**	(inventory * 365) / cost of products sold
X16	(gross profit + depreciation) / total liabilities	**X48**	EBITDA (profit on operating activities - depreciation) / total assets
X17	total assets / total liabilities	**X49**	EBITDA (profit on operating activities - depreciation) / sales
X18	gross profit / total assets	**X50**	current assets / total liabilities
X19	gross profit / sales	**X51**	short-term liabilities / total assets
X20	(inventory * 365) / sales	**X52**	(short-term liabilities * 365) / cost of products sold)
X21	sales (n) / sales (n-1)	**X53**	equity / fixed assets
X22	profit on operating activities / total assets	**X54**	constant capital / fixed assets
X23	net profit / sales	**X55**	working capital
X24	gross profit (in 3 years) / total assets	**X56**	(sales - cost of products sold) / sales
X25	(equity - share capital) / total assets	**X57**	(current assets - inventory - short-term liabilities) / (sales - gross profit - depreciation)
X26	(net profit + depreciation) / total liabilities	**X58**	total costs /total sales
X27	profit on operating activities / financial expenses	**X59**	long-term liabilities / equity
X28	working capital / fixed assets	**X60**	sales / inventory
X29	logarithm of total assets	**X61**	sales / receivables
X30	(total liabilities - cash) / sales	**X62**	(short-term liabilities *365) / sales
X31	(gross profit + interest) / sales	**X63**	sales / short-term liabilities
X32	(current liabilities * 365) / cost of products sold	**X64**	sales / fixed assets

Table 2: Summary of features in the Dataset

Table 1 shows the total number of features and instances in the dataset, and the number of samples in each class (bankrupt or not-bankrupt) of all the 5 datasets. The features are explained in Table 2 above. As shown in the table, there are 64 features labelled X1 through X64, and each feature is a synthetic feature. A synthetic feature is a combination of the econometric measures using arithmetic operations (addition, subtraction, multiplication, division). Each synthetic feature is as a single regression model that is developed in an evolutionary manner. The purpose of the synthetic features is to combine the econometric indicators proposed by the domain experts into complex features. The synthetic features can be seen as hidden features extracted by the neural networks but the fashion they are extracted is different. We have used only the 3rd Year dataset for our project.

2.2 Dataset Quality Assessment

Before moving on to assessing the quality of each of the two datasets, we first standardized our dataset as we can see that our dataset have large differences between their ranges which can possibly cause a lot of trouble to build a model. So, to prevent this problem, transforming features to comparable scales using standardization is the solution.
As we have mentioned earlier, the dataset suffers from missing values and data imbalance.

2.2.1 Missing Data

First, we look at some statistics of missing values. We plot the nullity matrix for the 3rd year dataset that explains the sparsity of 3rd Year data. The plot shown in Figure 1 was achieved using the library missingno. The nullity matrix gives us a data-dense display which lets us visually pick out the missing data patterns in the dataset. The white spaces indicate missing data values for the feature in the corresponding column. We notice that the feature X37 has the highest number of missing values.

We have visually seen the sparsity in the data. Now, let us see how much of data is actually missing. In Table 3 shown below, the second column shows the total number of instances in each dataset, and third column shows the number of instances or rows with missing values for at least one of the features. A naive approach of dealing with missing values would be to drop all such rows as in Listwise deletion. But dropping all such rows leads to a tremendous data loss. Column 4 shows the number of instances that would remain in each dataset if all rows with missing values were dropped. Column 5 shows the percent of data loss if all the rows with missing data values were indeed dropped. As the data loss in most of the datasets is over 50%, it is now clear that we cannot simply drop the rows with missing values, as it leads to severe loss in the representativeness of data.

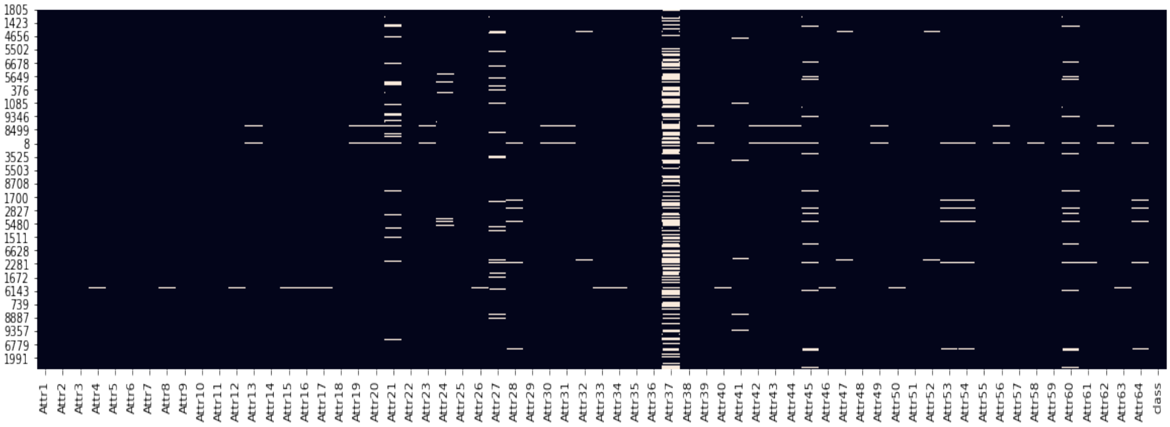


Figure 1: Sparcity Matrix for 3rd Year Dataset

Datal Set	**Total Instances**	**Instances With missing value**	**Instances that would remain if all rows with missing values were dropped**	**Data loss if rows with missing values were dropped**
Year 1	7027	3833	3194	54.54%
Year 2	10173	6085	4088	59.81%
Year 3	10503	5618	4885	53.48%
Year 4	9792	5023	4769	51.29%
Year 5	5910	2879	3031	48.71%

Table 3: Assessing the Missing Data for all the datasets.

2.2.2 Data Imbalance

We have covered the missing data aspect of the data quality assessment, let us now see the Data Imbalance aspect. Table 4 shown below summarizes the populations of class labels in each dataset. Column 2 shows the total instances, while Column 3 and Column 4 show the number of instances with class label as Bankrupt and Non-Bankrupt respectively. Looking at the numbers of Bankrupt class label, we can figure out that they are a minority when compare with the non-bankrupt class label. But Column 5 clearly shows the population percentage of the minority class, i.e., the Bankruptcy class label, among the total population of the dataset. These numbers in column 5 tell us that there is a huge data imbalance. If this imbalance is not cured, in the modeling stage that follows, the models will not have seen enough data from the minority class label and they train and hence perform poorly.

Datal Set	**Total Instances**	**Bankrupt instances in this forcasting period**	**Non-Bankrupt instances in this forcasting period**	**Percentage of minority class samples**
Year 1	7027	271	6756	3.85%
Year 2	10173	400	9773	3.93%
Year 3	10503	495	10008	4.71%
Year 4	9792	515	9277	5.25%
Year 5	5910	410	5500	6.93%

Table 4: Assessing the Data Imbalance for all the datasets.

2.3 Dealing with Missing Data

Missing data needs to treated since it causes mainly 3 problems:

1. It can introduce a substantial amount of bias in our model.
2. It makes the handling and analysis of the data more difficult.
3. Efficiency can be reduced because of this.

Dropping all the rows with missing values, introduces bias and affects representativeness of the results. So, we need to impute the missing data using suitable imputation techniques. Imputation is the process of replacing missing data with substituted values and it preserves all the cases by replacing missing data with an estimated value, based on other available information. In our project we explored 3 techniques of imputation:

1. Mean Imputation
2. k-Nearest Neighbors Imputation
3. Multivariate Imputation Using Chained Equations

2.3.1 Mean Imputation

Mean imputation technique is the process of replacing any missing value in the data with the mean of that variable in context. In our dataset, we replaced a missing value of a feature, with the mean of the other non-missing values of that feature. Mean imputation attenuates any correlations involving the variable(s) that are imputed. This is because, in cases with imputation, there is guaranteed to be no relationship between the imputed variable and any other measured variables. Thus, mean imputation has some attractive properties for univariate analysis but becomes problematic for multivariate analysis. Hence we opted Mean Imputation as a baseline method. We achieved mean imputation using scikit-learn's Imputer class.

2.3.2 k-Nearest Neighbors Imputation

The k-nearest neighbors algorithm or k-NN, is a non-parametric method used for classification and regression. In both cases, the input consists of the k closest training examples in the feature space. It can also be used as a data imputation technique k-NN imputation replaces NaNs in Data with the corresponding value from the nearest-neighbor row or column depending upon the requirement. The nearest-neighbor row or column is the closest row or column by Euclidean distance. If the corresponding value from the nearest-neighbor is also NaN, the next nearest neighbor is used. We used the fancyimpute library to perform k-NN data imputation, and we used 100 nearest neighbors for the process.

2.3.3 Multivariate Imputation using Chained Equations(MICE)

Multiple imputation using chained equations or MICE is an imputation technique that uses multiple imputations as opposed to single imputation. MICE is regarded as a fully conditional specification or sequential regression multiple imputation. It has become one of the principal methods of addressing missing data. Creating multiple imputations, as opposed to single imputations, accounts for the statistical uncertainty in the imputations. In addition, the chained equations approach is very flexible and can handle variables of varying types (for example., continuous or binary), as well as complexities such as bounds or survey skip patterns.

MICE is beneficial when the missing data is large. Because multiple imputation involves creating multiple predictions for each missing value, the analyses of multiply imputed data take into account the uncertainty in the imputations and yield accurate standard errors. In the MICE procedure a series of regression models are run whereby each variable with missing data is modeled conditional upon the other variables in the data. This means that each variable can be modeled according to its distribution, with, for example, binary variables modeled using logistic regression and continuous variables modeled using linear regression. We achieved MICE imputation using fancyimpute library.

2.4 Dealing with Data Imbalance

One of the shortcomings of the Polish Bankruptcy dataset is that it is highly imbalanced which we have already noticed in Table 4. Therefore, we need to deal with it.

Data Imbalance can be treated with Oversampling and/or Undersampling. In data analysis, Oversampling and Undersampling are opposite and roughly equivalent techniques of dealing with Data Imbalance, where they adjust the class distribution of a data set (i.e. the ratio between the different classes/categories represented). Oversampling is increasing the class distribution of the minority class label whereas Undersampling is decreasing the class distribution of the majority class label. In our project, we explored Synthetic Minority Oversampling Technique or SMOTE.

2.4.1 Synthetic Minority Oversampling Technique (SMOTE)

Synthetic Minority Oversampling Technique (SMOTE) is a widely used oversampling technique. To illustrate how this technique works consider some training data which has s samples, and f features in the feature space of the data. For simplicity, assume the features are continuous. As an example, let us consider a dataset of birds for clarity. The feature space for the minority class for which we want to oversample could be beak length, wingspan, and weight. To oversample, take a sample from the dataset, and consider its k nearest neighbors in the feature space. To create a synthetic data point, take the vector between one of those k neighbors, and the current data point. Multiply this vector by a random number x which lies between 0, and 1. Adding this to the current data point will create the new synthetic data point. SMOTE was implemented from the imbalanced-learn library.

Before moving further, we first divided the 3rd Year dataset into the training dataset and the testing dataset. We will build the model using the training dataset and will evaluate the performance of the model on the test dataset.

2.5 Multicollinearity

Next we move on to another important step of Model Adequacy Checking, that is, Checking Multicollinearity in our dataset. A basic assumption is multiple linear regression model is that the rank of the matrix of observations on explanatory variables is the same as the number of explanatory variables. In other words, such a matrix is of full column rank. This, in turn, implies that all the explanatory variables are independent, i.e., there is no linear relationship among the explanatory variables. It is termed that the explanatory variables are orthogonal.

In many situations in practice, the explanatory variables may not remain independent due to various reasons. The situation where the explanatory variables are highly intercorrelated is referred to as **multicollinearity**.

In order to check the multicollinearity in our dataset, we plot the ”Correlation HeatMap” for each of the imputed dataset. Figure 2, Figure 3 and Figure 4 show the Correlation HeatMap for the Mean Imputed Dataset, the kNN imputed Dataset and the MICE imputed Dataset respectively. The darker shades indicate high correlation (positive or negative) among the variables whereas the lighter shades indicate less correlation.

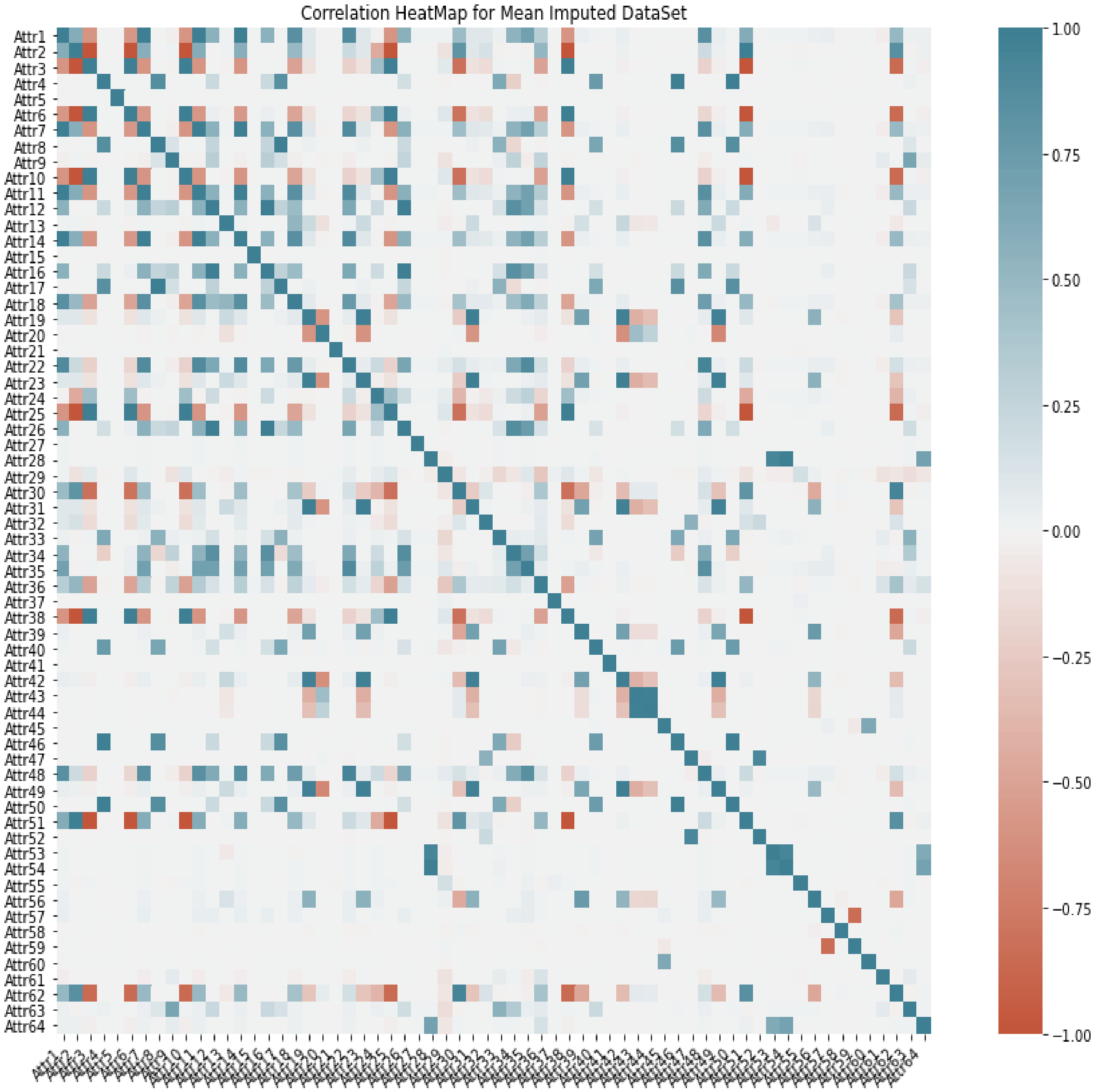


Figure 2: Correlation HeatMap for Mean Imputed DataSet

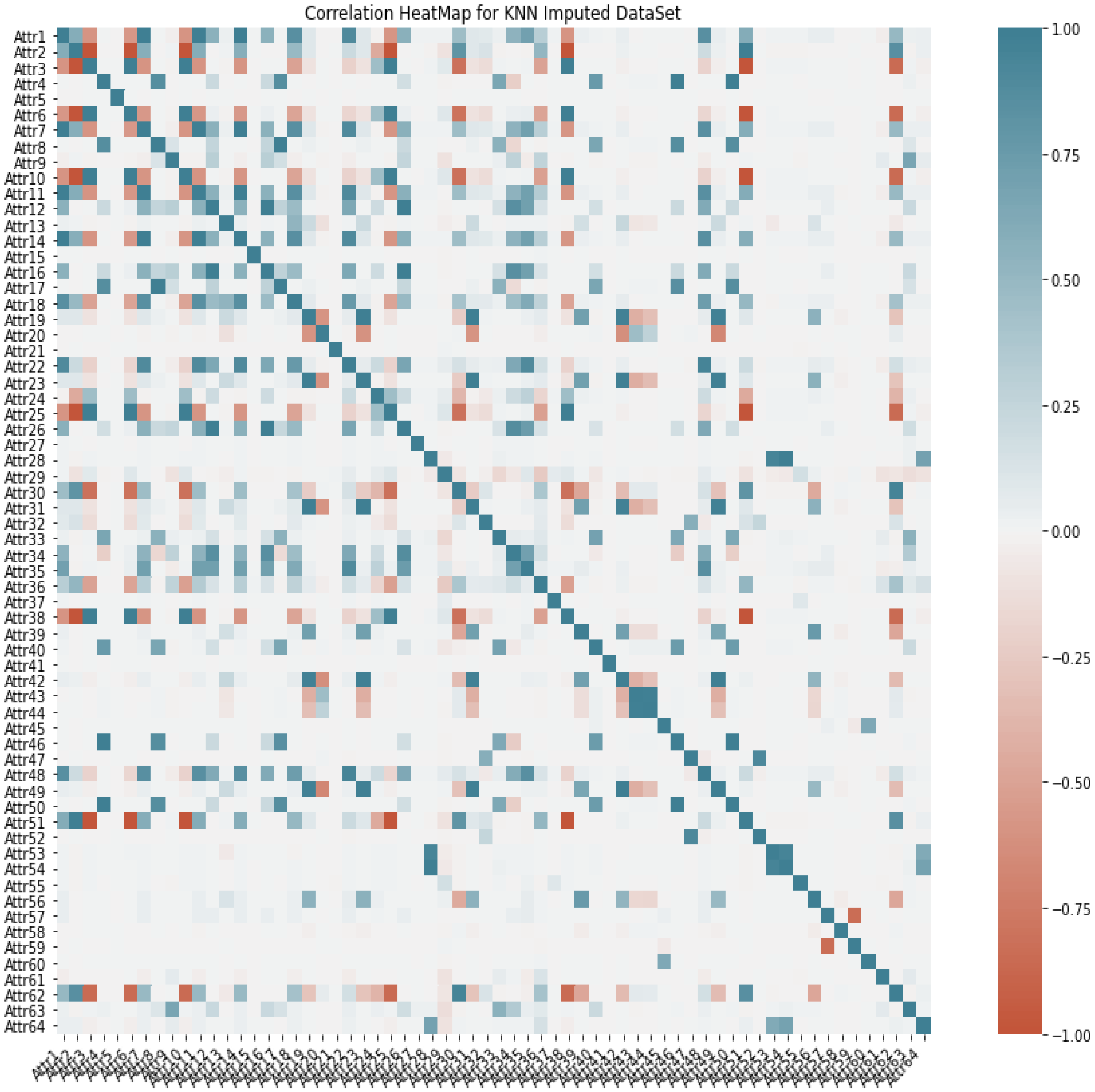


Figure 3: Correlation HeatMap for kNN Imputed DataSet

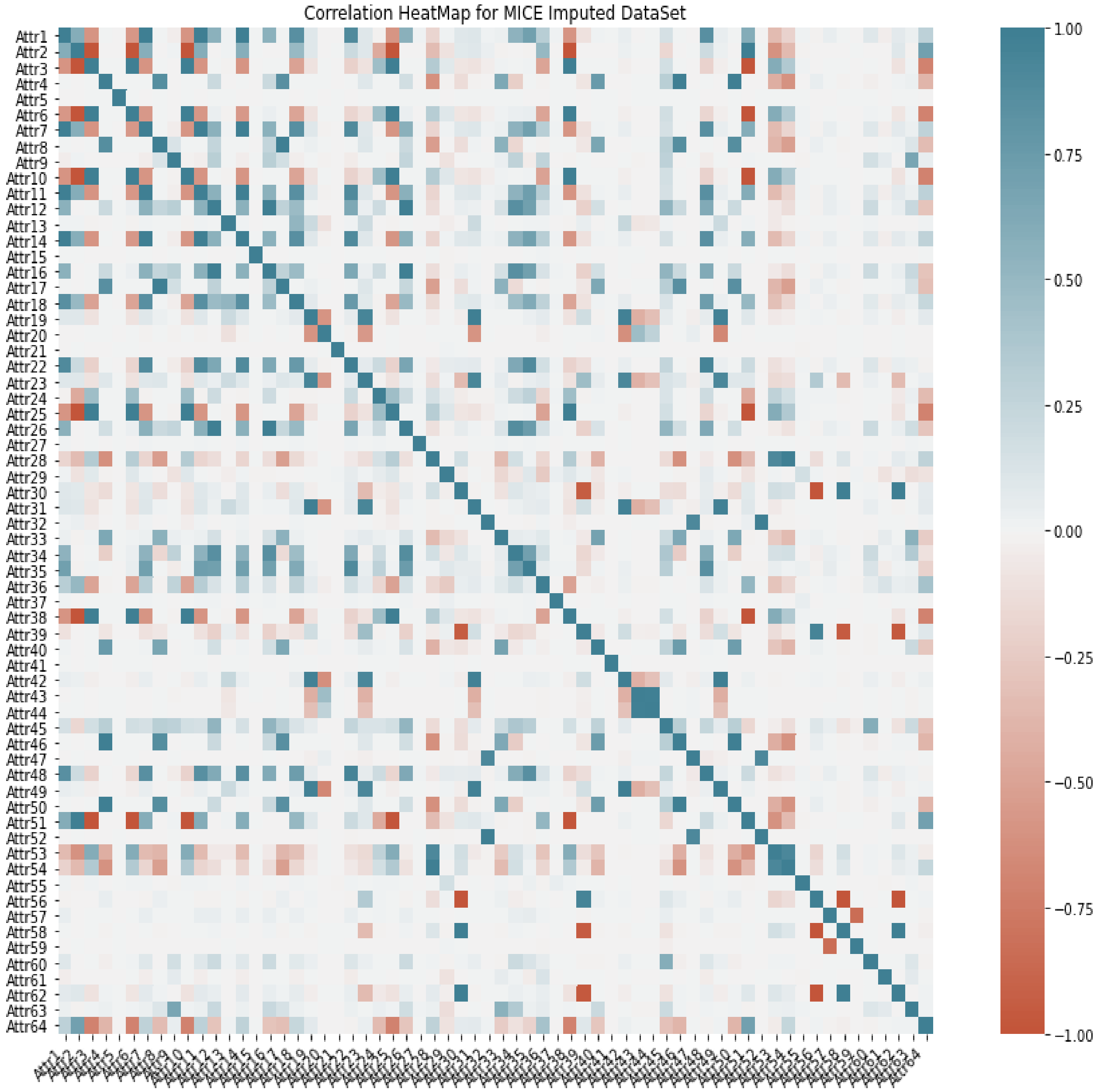


Figure 4: Correlation HeatMap for MICE Imputed DataSet

2.5.1 Dealing with Multicollinearity

Several multicollinearity diagnostic measures are available, and each of them is based on a particular approach. In our case, we have used the approach "Variance Inflation Factor" in order to detect the problem of multicollinearity. Based on this concept, the variance inflation factor for the j^{th} explanatory variable is defined as:

$$VIF_j = \frac{1}{(1 - {R_j}^2)}$$

where ${R_j}^2$ denotes the coefficient of determination obtained when X_j is regressed on the remaining $(k-1)$ variables excluding X_j.
In practice, usually, a $VIF > 5$ indicates that the associated regression coefficients are poorly estimated because of multicollinearity. Hence, in order to deal with it, we have used an iterative algorithm that drops variable with highest VIF and then checks VIF again and then drop until VIF of all variables is less than 5. After applying the Iterative VIF elimination, the remaining variables for each of the imputed datasets can be seen in Table 5.
We then verified the dropping of variables using the Variance Decomposition Analysis where we observed that the Condition Index for all the variables which are retained using the VIF method is less than 15. Hence, we can conclude that none of the remaining variables are involved in multicollinearity.
We have also used principal component analysis which is a common feature extraction technique that employs matrix factorization to reduce the dimensionality of data into lower spaces in order to deal with problem of multicollinearity.

S.No.	Mean Imputation	kNN Imputation	MICE Imputation
1	X5	X5	X5
2	X6	X6	X6
3	X8	X8	X8
4	X9	X9	X9
5	X13	X13	X13
6	X15	X15	X15
7	X20	X20	X20
8	X24	X24	X24
9	X27	X27	X27
10	X29	X29	X29
11	X32	X32	X31
12	X34	X34	X34
13	X36	X36	X36
14	X37	X37	X37
15	X39	X39	X40
16	X40	X40	X41
17	X41	X41	X44
18	X44	X44	X45
19	X45	X45	X48
20	X48	X48	X52
21	X52	X52	X55
22	X53	X53	X57
23	X55	X55	X58
24	X56	X56	X59
25	X57	X57	X60
26	X58	X58	X61
27	X59	X59	X63
28	X60	X60	X64
29	X61	X61	NaN
30	X63	X63	NaN
31	X64	X64	NaN

Table 5: Remaining Variables after applying Iterative VIF Elimination.

3 Variable Selection Method

The complete regression analysis depends on the explanatory variables present in the model. It is understood in the regression analysis that only correct and important explanatory variables appear in the model. In practice, after ensuring the correct functional form of the model, the analyst usually has a pool of explanatory variables which possibly influence the process or experiment. Generally, all such candidate variables are not used in the regression modelling, but a subset of explanatory variables is chosen from this pool.In our project, we have used two variable selection method, that is, Stepwise Regression and Lasso Regression.

3.1 Stepwise Regression

A combination of forward selection and backward elimination procedure is the stepwise regression. It is a modification of forward selection procedure and has the following steps.

- We start with the intercept model and compute the AIC for the model.
- We then compute AIC for all the possibilities of adding one more variable in our intercept only model. We select the variable with the smallest AIC if it has a lower AIC than intercept only model.
- We then again compute AIC for adding one more variable in the model along with the AIC for removing the already added variable. We sort the values by ascending order and variables are either added or the existing variable is subtracted depending on the value of AIC.
- We continue performing these steps until any further action - addition or subtraction, results in increase of AIC of the model.

Therefore, we have selected variables on the basis of Akaikie Information Criterion(AIC).
For **Mean Imputed dataset**, we have the following variables in our Model:
$X24, X41, X63, X34, X48, X36, X58, X61, X9, X55, X52, X32, X60, X29, X44, X64, X40, X5, X37, X57, X59$.
For **kNN Imputed dataset**, we have the following variables in our Model:
$X24, X63, X34, X52, X58, X55, X41, X48, X36, X9, X61, X60, X27, X37, X44, X40, X32, X64, X29, X5, X21, X8, X57, X59$
For **MICE Imputed dataset**, we have the following variables in our Model:
$X24, X41, X48, X64, X34, X36, X61, X58, X9, X55, X52, X29, X31, X60, X45, X13, X8, X5$

3.2 Lasso Regression

Lasso regression is a type of Regularization that uses shrinkage. Shrinkage is where data values are shrunk towards a central point, like the mean. The lasso

procedure encourages simple, sparse models (i.e. models with fewer parameters). This particular type of regression is well-suited for models showing high levels of multicollinearity or when you want to automate certain parts of model selection, like variable selection/parameter elimination. The acronym “LASSO” stands for Least Absolute Shrinkage and Selection Operator.
Lasso regression performs **L1 regularization** technique, which adds a penalty equal to the absolute value of the magnitude of coefficients. Larger penalties result in coefficient values closer to zero, which is the ideal for producing simpler models. On the other hand, L2 regularization (e.g. Ridge regression) doesn’t result in elimination of coefficients or sparse models. This makes the Lasso far easier to interpret than the Ridge.

4 Model: Logistic Regression

Since we have cleaned our dataset, now we are well set to define our model. As our response variable ’Y’ is a categorical variable with only TWO nominal categories, that is, 0 which indicates that there is no bankruptcy and 1 which indicates there is a bankruptcy. Logistic regression makes use of the canonical link function, $\ln \frac{p}{1-p}$.The logistic regression model is given as:

$$Y_i \sim Binomial(N_i, p_i)$$

$$\ln\left(\frac{p_i}{1-p_i}\right) = \beta_0 + \sum_{j=1}^{k} \beta_j x_{ij} = \boldsymbol{X\beta} \text{ for } i = 1, 2, ..., n$$

where x_{ij} is the element in the i^{th} row and j^{th} column of the model matrix $\boldsymbol{X}$.
To evaluate the accuracy of the classification or the logistic model, we use a confusion matrix.

4.1 Confusion matrix

Confusion Matrix as the name suggests gives us a matrix as output and describes the complete performance of the model.The confusion matrix shows the ways in which your classification model is confused when it makes predictions.
A confusion matrix is a technique for summarizing the performance of a classification algorithm. Classification accuracy alone can be misleading if you have an unequal number of observations in each class or if you have more than two classes in your dataset.
There are 4 important terms :

1. **True Positives :** The cases in which we predicted YES and the actual output was also YES.

2. **True Negatives :** The cases in which we predicted NO and the actual output was NO.

3. **False Positives:** The cases in which we predicted YES and the actual output was NO.
4. **False Negatives:** The cases in which we predicted NO and the actual output was YES.

4.2 Metrics of Confusion Matrix

4.2.1 Accuracy score

Overall, how often is our model correct?
As a rule of thumb, accuracy can tell us immediately whether a model is being trained correctly and how it may perform generally. However, it does not give detailed information regarding its application to the problem.In multilabel classification, this function computes subset accuracy: the set of labels predicted for a test dataset must exactly match the corresponding set of labels in y_pred.
The problem with using accuracy as your main performance metric is that it does not do well when you have a severe class imbalance.

$$\text{Acuuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of prediction made}}$$

4.2.2 Precision Score

It is the number of correct positive results divided by the number of positive results predicted by the classifier.
Precision talks about how precise/accurate your model is out of those predicted positive, how many of them are actual positive.Precision is a good measure to determine, when the costs of False Positive is high.

$$\text{Precison} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}}$$

4.2.3 Recall Score

It is the number of correct positive results divided by the number of all relevant samples (all samples that should have been identified as positive).It is also known as the sensitivity. Recall actually calculates how many of the Actual Positives our model capture through labelling it as Positive (True Positive).

$$\text{Recall} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Negetive}}$$

4.2.4 Specificity

Specificity, also known as the **true negative rate(TNR)**, measures the proportion of actual negatives that are correctly identified as such. It is the opposite of the recall.

$$\text{Specificity} = \frac{\text{True Negastive}}{\text{True Negative + False Positive}}$$

4.2.5 F1 Score

The F1 Score is a measure of a test's accuracy, that is, it is the harmonic mean of precision and recall. It can have a maximum score of 1 and a minimum of 0. Overall, it is a measure of the preciseness and robustness of the model.

$$\text{F1 Score} = \frac{2(\text{precision} \times \text{recall})}{\text{precision} + \text{recall}}$$

$$= \frac{\text{2True Positive}}{\text{2True Positive + False Positive + False Negative}}$$

4.2.6 Precison recall curve

The precision-recall curve is constructed by calculating and plotting the precision against the recall for a single classifier at a variety of thresholds. For example, if we use logistic regression, the threshold would be the predicted probability of an observation belonging to the positive class.

4.2.7 ROC curve

An ROC curve (receiver operating characteristic curve) is a graph showing the performance of a classification model at all classification thresholds. This curve plots two parameters: True Positive Rate. False Positive Rate.

The ROC curve is produced by calculating and plotting the true positive rate against the false positive rate for a single classifier at a variety of thresholds. For example, in logistic regression, the threshold would be the predicted probability of an observation belonging to the positive class.

4.3 Model Building and Comparison

In our project, we have built four models using each of the imputed datasets, which are as follows:

1. Logistic Regression using Principal Component Analysis.
2. Logistic Regression after Stepwise Regression without Regularization.
3. Logistic Regression after Stepwise Regression with Regularization
4. Logistic Regression using Lasso Regression.

We then compared the F1-score and recall score of each of the model with the help of confusion matrix and classification report. Figure 5 and Figure 6 show the model comparison of each of the model without and with regularization in stepwise regression respectively.

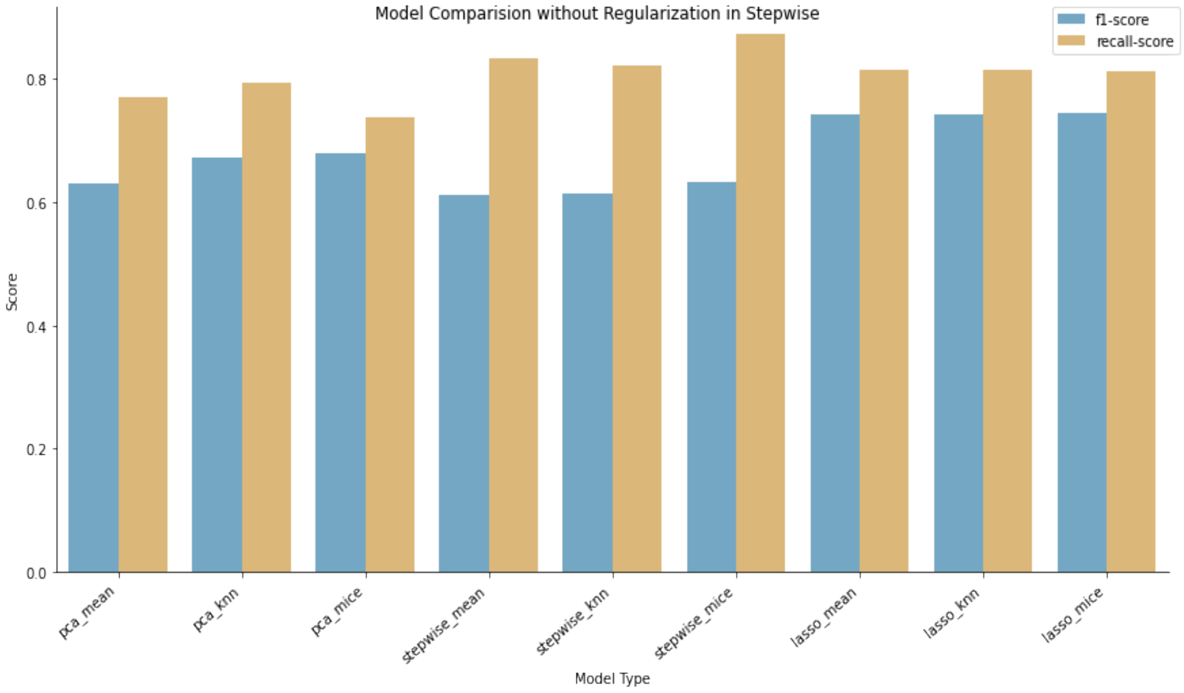


Figure 5: Model Comparison Without Regularization in Stepwise.

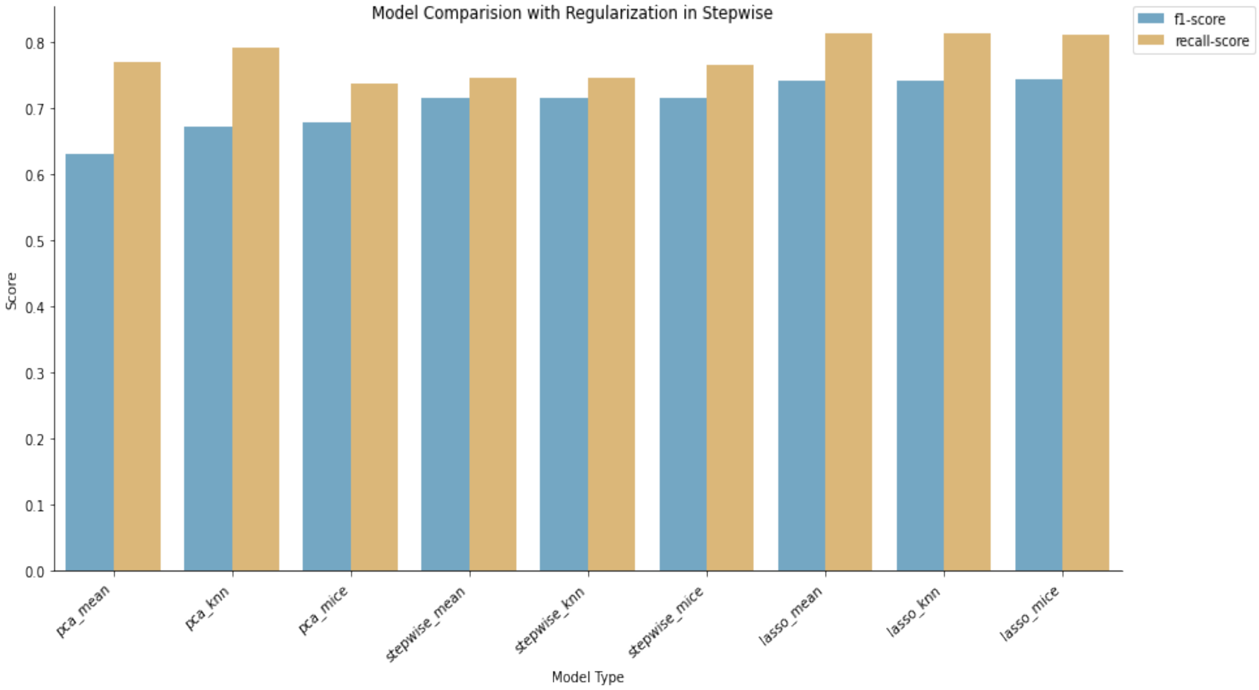


Figure 6: Model Comparison With Regularization in Stepwise.

We can see that the lasso regression model has higher accuracy than other models, that is, Model Accuracy is approximately 70%. Therefore, we consider the Lasso Regression Model for Mean Imputed dataset. Let us explore the confusion matrix and classification report of this model which are shown in Figure 7 and Figure 8 respectively.

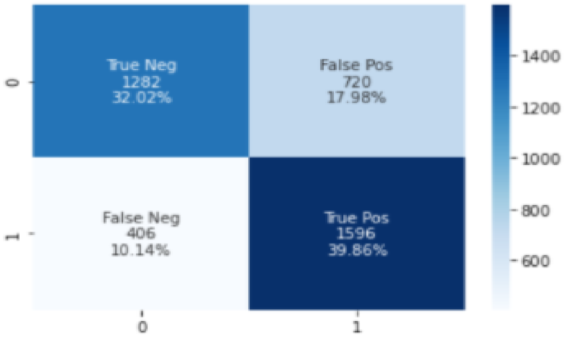
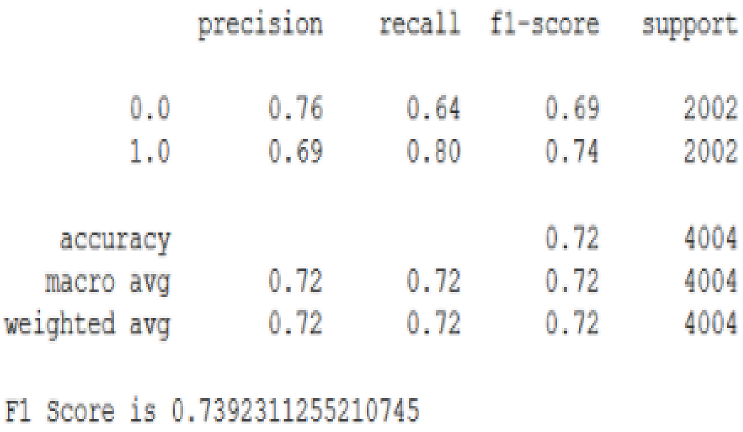


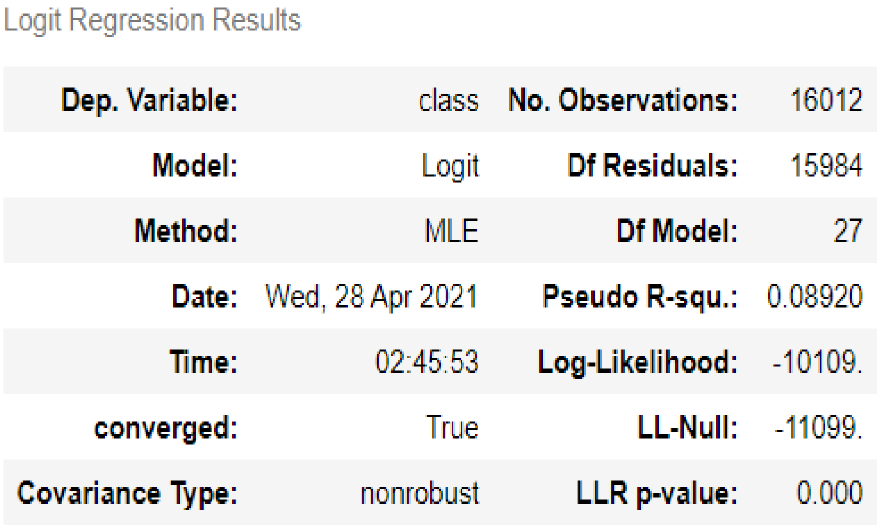
Figure 7: Confusion Matrix for Lasso regression for Mean Imputed dataset.



```
              precision    recall  f1-score   support

         0.0       0.76      0.64      0.69      2002
         1.0       0.69      0.80      0.74      2002

    accuracy                           0.72      4004
   macro avg       0.72      0.72      0.72      4004
weighted avg       0.72      0.72      0.72      4004

F1 Score is 0.7392311255210745
```

Figure 8: Classification Report for Lasso regression for Mean Imputed dataset.

Now the Table 6 (shown below) is the table of estimates of the coefficients, standard errors, Z-scores, Probability $Z > |Z|$ and the confidence intervals for each of the variables which are included in the model. The summary of our model can be seen in Figure 9.



Logit Regression Results

Dep. Variable:	class	**No. Observations:**	16012
Model:	Logit	**Df Residuals:**	15984
Method:	MLE	**Df Model:**	27
Date:	Wed, 28 Apr 2021	**Pseudo R-squ.:**	0.08920
Time:	02:45:53	**Log-Likelihood:**	-10109.
converged:	True	**LL-Null:**	-11099.
Covariance Type:	nonrobust	**LLR p-value:**	0.000

Figure 9: Summary of the Lasso regression model for Mean Imputed dataset.

	Coef.	**Std. Err.**	**z**	**P>|z|**	**[0.025**	**0.975]**
const	-0.252	0.021	-12.228	0	-0.292	-0.212
X5	0.0149	0.026	0.582	0.561	-0.035	0.065
X11	-0.1964	0.154	-1.279	0.201	-0.497	0.105
X18	-0.0366	0.043	-0.861	0.39	-0.12	0.047
X21	-0.0129	0.024	-0.54	0.589	-0.06	0.034
X22	-1.24	0.269	-4.609	0	-1.767	-0.713
X24	-1.797	0.157	-11.471	0	-2.104	-1.49
X28	2.96e-17	0.042	7.12e-16	1	-0.082	0.082
X29	0.025	0.02	1.267	0.205	-0.014	0.064
X31	0.0807	0.073	1.109	0.267	-0.062	0.223
X33	0.1599	0.194	0.826	0.409	-0.22	0.539
X34	1.0075	0.094	10.765	0	0.824	1.191
X35	-1.1495	0.12	-9.549	0	-1.385	-0.914
X36	0.2386	0.034	7.031	0	0.172	0.305
X37	-0.0074	0.037	-0.201	0.84	-0.079	0.064
X38	-0.7933	0.263	-3.013	0.003	-1.309	-0.277
X41	-0.0097	0.024	-0.404	0.686	-0.057	0.037
X48	2.0274	0.255	9	0	1.586	2.469
X52	-0.042	0.024	-1.75	0.08	-0.089	0.005
X55	-0.1327	0.037	-3.569	0	-0.206	-0.06
X56	0.0501	0.044	1.126	0.26	-0.037	0.137
X57	-0.0683	0.044	-1.568	0.117	-0.154	0.017
X58	-0.0187	0.025	-0.738	0.46	-0.068	0.031
X59	-0.0734	0.046	-1.602	0.109	-0.163	0.016
X60	-0.0157	0.027	-0.576	0.565	-0.069	0.038
X61	-0.1172	0.059	-1.985	0.047	-0.233	-0.001
X63	-0.7677	0.218	-3.521	0	-1.195	-0.34
X64	-0.0704	0.048	-1.461	0.144	-0.165	0.024

Table 6: Estimates of Coefficients etc.

4.4 Model Diagnostics

Now, to check the overall performance of our fitted model, we considered the Pearson's Chi-Squared Goodness of Fit, ϕ Coefficient and Contingency Coefficient.

1. **Pearson's Chi-Squared test** (χ^2) is a statistical test applied to sets of categorical data and enables us to find if the deviation of the experiment from theory is just by chance or is it really due to the inadequacy of the theory to fit the observed data. In other words, it is a test of independence between the predicted and actual response variable. It has the following hypotheses:

   $$H_0 : Y \text{ and } \hat{Y} \text{ are independent } \textbf{against } H_1 : Y \text{ and } \hat{Y} \text{ are dependent.}$$

   From the confusion matrix, we can define each term of the model as f_{jk}. Also, we can define the term $f_{j.} = \sum_{j=1}^{J} f_{jk}$ for all $j = 1, 2, .., J$ and $f_{.k} = \sum_{k=1}^{K} f_{jk}$ for all $k = 1, 2, .., K$. Here in our model, $J = 2$ and $K = 2$.
   The Pearson Statistic is given as:

   $$\chi^2{}_p = \sum_{j=1}^{J} \frac{\sum_{k=1}^{K} \left(f_{jk} - \frac{f_{j.}f_{.k}}{n} \right)^2}{\frac{f_{j.}f_{.k}}{n}} \sim \chi^2{}_{1-\alpha,(J-1)(K-1)}$$

   Here in our case $\alpha = 0.05$ and $J = 2$ and $K = 2$.
   So,

   $$\chi^2{}_{1-\alpha,(J-1)(K-1)} = \chi^2{}_{0.95,1} = 3.84146$$

   If the value of our pearson statistic comes greater than $\chi^2{}_{0.95,1}$, then the predicted $\hat{Y}$ and the actual response Y are not independent. Otherwise, they are dependent on each other.
   From our confusion matrix, by calculating $\chi^2{}_p$, we get the following result:

   $$\chi^2{}_p = 791.95 > \chi^2{}_{0.95,1}$$

   This tells us that our model is a good fit and the predicted response by our classifier is highly dependent or correlated on the actual response, and hence our classifier will give us better results of prediction.

2. ϕ **Coefficient** is a measure of association for two binary variables. Higher the value of ϕ, stronger is the association. It is given by:

   $$\phi = \frac{f_{11}f_{00} - f_{10}f_{01}}{\sqrt{f_{1.}f_{0.}f_{.0}f_{.1}}}$$

   In our case, we get $\phi = 0.4447$.

3. **Contingency Coefficient** can also be used to estimate the extent of the relationship between two variables, or to show the strength of a relationship. The contingency coefficient is computed as the square root of chi-square divided by chi-square plus n, the sample size. The larger the contingency coefficient the stronger the coefficient is.

$$\text{Contingency Coefficient} = \sqrt{\frac{{\chi^2}_p}{{\chi^2}_p + n}}$$

In our case, it is coming out to be 0.40636.

5 Results

In the beginning, we first dealt with the problem of the missing values and data imbalance in our dataset. Then we moved on to deal with the problem of multicollinearity and to tackle this, we used VIF iterative algorithm for eliminating variables from our model. We also used Principal Component Regression and using AIC criterion, we selected the suitable variables in our model. Then, we built different classification models, and on comparison, we got that for Lasso Regression Model with the mean imputed dataset, the results are comparatively satisfactory. We get an overall accuracy of approximately 70%. Finally, we used **Pearson Chi-squared test for goodness of fit**, ϕ **coefficient** and **Contingency Coefficient** to check how good our model is and what is it's measure of association. The results are quite decent but the measures of association is not high because of extra false positives in our model.

6 Discussion

We saw from the model comparison graphs that the Model on Mean Imputed Data Set with Lasso Regularisation had arguably the highest F-1 Score and Recall Score. In fact, there was not much difference in the results of all the 3 models in which Lasso Regularisation was applied. Our reasoning for selecting the Mean Model was that since it doesn't matter which imputation Method is being used, the model with least complex imputation should be preferred. Therefore, we decided to select Model on Mean Imputed Data Set with Lasso Regularisation as our final model for classification.

We further observed during modelling that using L1 regularisation on Stepwise Regression Model boosted the scores. However, it doesn't make practical sense to use L1 regularisation after eliminating Multicollinearity and doing variable selection.

In case of models on Principle Components we observed that they performed poorly in comparison to other 2 class of models in our study. Despite explaining 95% of variablity in the data and removing multicollinearity, Principal Complements were not upto mark as regressors.

7 Conclusion

In our study, we have successfully modelled the problem of Bankruptcy Prediction in Polish Data using different variants of Logistic Regression.

Initially, we started with the data pre processing steps such as dealing with Missing Data and Imbalanced Data. We found during these steps that the Polish Dataset suffered from various serious problems and necessary actions were needed before modelling the data. We therefore used multiple data imputation techniques and created synthetic samples to deal with issues of missing and imbalance data.

After obtaining a complete dataset, we found that there is severe multicollinearity in our dataset. This is due to the nature of the Variables in dataset which are synthetic combinations of common financial Variables. To tackle the problem of multicollinearity, we primarily used VIF iterative algorithm for eliminating variables from our model. We also used Principal Component Regreression as an alternative to variable elimination.

Post elimination of Multicollinearity,we switched to the problem of Variable Selection and implemented Stepwise Regression to reduce dimensionality of our model. We also considered Lasso Regression to deal with the combined problem of Multicollinearity and Variable Selection.

At last we compared all the models and found that Lasso Regression Models outperformed other class of models and in that class, all the models performed on the same level. As Mean Imputation was the easiest among all the imputation techniques, we considered Lasso Regression Model using Mean Imputation Method to be our final model.

The results for our final model were comparable with models considered by other researchers. In class of Logistic Regression models and perhaps Linear Classifiers , our final model achieved one of the highest possible scores. However, it's performance is slightly poor in comparison to other non linear Machine Learning Classifiers. This shows that data considered for our project is essentially Non Linear in nature and could be better modelled by other Non Linear Classifiers.

But despite it's relatively low performance, Logistic Regression Model provides the advantage of explainability which is not present in case of most Non Linear Classifiers. This advantage boosts the trust of regulators , users in the model and is therefore considered very often in modelling this type of problems.

References

[1] Zieba, M., Tomczak, S.K., & Tomczak, J.M.(2016) *Ensemble Boosted Trees with Synthetic Features Generation in Application to Bankruptcy Prediction. Expert Systems with Applications.*

[2] `https://github.com/smaddikonda/Bankruptcy-Prediction/blob/a0391f131c24fc8f8c32ea47542d53dbb50cdf68/Bankruptcy%20Prediction%20Report.pdf`

[3] `https://home.iitk.ac.in/~shalab/regression/Chapter9-Regression-Multicollinearity.pdf`

[4] `https://home.iitk.ac.in/~shalab/regression/Chapter13-Regression-VariableSelectionAndModelBuilding.pdf`

[5] `https://home.iitk.ac.in/~shalab/regression/Chapter14-Regression-LogisticRegressionModels.pdf`