CONTENTS

PREFACE

Financial Institutions lend out credit (money) to borrowers (obligators) who are required to return the money with added interest. This interest is a financial institution's cost of operation or even profit. Nevertheless, this process of crediting out money and making a return on the investment (so to say) is the functional premise and fundamental approach of a financial institution in a broad sense. The risk associated with the ability of the borrower/obligator to pay back the loaned amount in maybe under stipulated timelines, agreed upon interest rates or contracted installment patterns is called Credit Risk.

Financial institutions rely of credit risk models to evaluate the credit risk of a potential obligator. These evaluations help them determine whether to approve a particular loan application or not. The credit risk model may also be used to estimate the interest rate of the lending amount to be charged based on the person's likeliness to payback the amount or credit worthiness.

OBJECTIVE

The objective of this report to walk you through the process of Credit Risk Modelling, using Logistic Regression. The training dataset which we will use to train and validate our model has 3541 rows and 51 columns or variables. These variables are a combination financial institution's key data features. The data variables largely fall under these categories: profitability, leverage, company's size and liquidity.

We shall start with quality control of our data by replacing NA values using most relevant imputation techniques, exploratory data analysis using density graphs and relationship trends, checking for multicollinearity, creation of new financial ratios and variables, outlier treatment and finally, the logistic framework significance, model performance and validation exercises.

INTRODUCTION

ENVIRONMENT SETUP:

```
setwd("C:/Users/Hp/Desktop/R Programming")
getwd()

TrainData <- read_excel("raw-data.xlsx" )
TestData <- read_excel("validation_data.xlsx")
```

We start by setting up our working directory and reading the training dataset excel file as TrainData and the validation data as TestData.

LIBRARIES USED

Libraries	Description
ggplot2	For visualizations. Building plots step by step from multiple sources.
readxl	Reading excel formal files
corrplot	Multicollinearity Plots
dplyr	Data Manipulation
Hmisc	Useful for data analysis,high level graphics and utility operations
data.table	Fast aggregation of large datasets
rms	Multicollinearity check within a model (VIF)
pscl	McFadden Psuedo R2 Test
lmtest	Likelihood Ratio test
ROCR	Calculating K.S and AUC.
pROC	Plotting ROC

VARIABLE EXPLANATION:

Variable Name	Description
Networth Next Year	Net worth of the customer in next year
Total assets	Total assets of customer
Net worth	Net worth of the customer of present year
Total income	Total income of the customer
Change in stock	Value of current stock - Value of stock in last trading day
Total expenses	Total expense done by customer
Profit after tax	Profit after tax deduction
PBDITA	Profit before depreciation, income tax and amortization
PBT	Profit before tax deduction
Cash profit	Total Cash profit
PBDITA as % of total income	PBDITA / Total income
PBT as % of total income	PBT / Total income
PAT as % of total income	PAT / Total income
Cash profit as % of total income	Cash Profit / Total income
PAT as % of net worth	PAT / Net worth
Sales	Sales done by customer
Income from financial services	Income from financial services
Other income	Income from other sources
Total capital	Total capital of the customer
Reserves and funds	Total reserves and funds of the customer
Deposits	All blank values
Borrowings	Total amount borrowed by customer
Current liabilities & provisions	current liabilities of the customer
Deferred tax liability	Future income tax customer will pay
Shareholders funds	Amount of equity in a company
Cumulative retained profits	Total cumulative profit retained by customer

Capital employed	Current asset minus current liabilities
TOL/TNW	Total liabilities/Total net worth
TTL/ TNW	Short + long term liabilities/Tangible net worth
CL/ Net worth (%)	Contingent liabilities / Net worth
CL	Liabilities because of uncertain events
Net fixed assets	purchase price of all fixed assets
Investments	Total invested amount
Current assets	Assets expected to convert to cash within a year
Net working capital	Difference of current liabilities and current assets
Quick ratio (times)	Total cash divided by current liabilities
Current ratio (times)	Current assets divided by current liabilities
Debt to equity ratio (times)	Total liabilities divided by its shareholder equity
Cash to current liabilities (times)	Total liquid cash divided by current liabilities
Cash to average cost of sales per day	Total cash divided by average cost of the sales
Creditors turnover	Net credit purchase/average trade creditors
Debtors turnover	Net credit sales /average accounts receivable
Finished goods turnover	Annual sales divided by average inventory
WIP turnover	Cost of goods sold for a period/Average inventory period
Raw material turnover	Cost of goods sold/Average inventory for the same period
Shares outstanding	Number of issued - number of share held in the company
Equity face value	cost of the equity at the time of issuing
EPS	Net income divided by total number of outstanding share
Adjusted EPS	Adjusted net earning /weighted average number of CSO
Total liabilities	Sum of all type of liabilities
PE on BSE	Company current stock price/Earning per share

EXPLORATORY DATA ANALYSIS

MISSING VALUE IDENTIFICATION:

On investigating for missing values in the dataset, we straight away notice that a significant amount of variables contain NA's. For narrowing down to top significant variables to be used to build our model, we first need to under stand the correlations between these assumed independent predictors. That requires all them to be justly converted to continuous variables and free of missing values.

For this exercise, I have imputed mostly all with replacing the NA's with the median for that continuous distribution. To avoid over generalization, I have imputed the values using both Default and Not Default as filter to estimate a more realistic value.

As shown below, most other variable's na's have been treated similarly. Imputation of NA's in total income:

```
summary(TrainData$`Total income`[TrainData$Default == "Yes"])
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.     NA's
##     0.10    13.65    98.20   880.77   419.80 33935.20       44
```

```
summary(TrainData$`Total income`[TrainData$Default == "No"])
```

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.      NA's
##       0.0     122.8     482.6    4817.1    1513.4 2442828.2       154
```

```
TrainData$`Total income`[is.na(TrainData$`Total income`) &
TrainData$Default == "Yes"] =
  median(TrainData$`Total income`[TrainData$Default == "Yes"], na.rm = T)

TrainData$`Total income`[is.na(TrainData$`Total income`) &
TrainData$Default == "No"] =
  median(TrainData$`Total income`[TrainData$Default == "No"], na.rm = T)
```

If a NA value occurs in the total income and the person is defaulting, then the median of the income when a person defaults would be appropriated into the NA columns which in the above case is 98.20. In the same way,

If a NA value occurs in the total income and the person is not defaulting, then the median of the income when a person is not defaulting will be assigned into that NA entry which is 482.6. This has been repeated for other variables as well, until we get:

```
sum(is.na(TrainData))
```

```
## [1] 0
```

FEATURE ENGINEERING (NEW RATIO VARIABLES):

```
TrainData$Default <- factor(ifelse(`Networth Next Year` > 0, "No" , "Yes s
summary(TrainData$Default)
```

```
##   No  Yes
## 3298  243
```

Using the continuous variable net worth next year, we create two classes or bucket of Default and Non Defaulters. If the net worth next year falls into the negative zone or below 0, these class of potential borrowers would be labeled as Default set to Yes and for who have net worth next year in positive will be set as Default to No.

```
TrainData$`Deposits (accepted by commercial banks)` <- NULL
TrainData$Num <- NULL
TrainData$`Income from financial services` <-  NULL
TrainData$`Other income` <- NULL
```

```
TrainData$`Deferred tax liability` <- NULL
TrainData$`Contingent liabilities` <- NULL
TrainData$Investments <- NULL
TrainData$`PE on BSE` <- NULL
TrainData$`Equity face value` <- NULL
TrainData$`Shares outstanding` <- NULL
TrainData$`WIP turnover` <- NULL
TrainData$`Finished goods turnover` <- NULL
```

The missing data in these variables, on an average, accounts for 40% of the total values present respectively for each. Hence I consider dropping these variables as opposed to imputing them with values.

```
TrainData$Size.Ratio.CapitalAsPerc.Assets <- (TrainData$`Total capital`/Tr
ainData$`Total assets`) *100
summary(TrainData$Size.Ratio.CapitalAsPerc.Assets)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.03    4.90   11.88   29.02   28.53 4716.67
```

The first financial ratio is the total capital as a percentage of total assets. This will fall under the category of Company's Size.

```
TrainData$PBDITA.PERC.TotalIncome <- (TrainData$PBDITA/TrainData$`Total in
come`)*100
summary(TrainData$PBDITA.PERC.TotalIncome)
```

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.      NA's
## -6400.000     5.236     9.414     4.731    16.164   100.000         1
```

Second financial ratio falls under profitability, this variable is profit before depreciation, income tax and amortization as a percentage of total income.

```
TrainData$Leverage.Ratio <- (TrainData$`Total liabilities` - TrainData$Bor
rowings)/`Total liabilities`
summary(TrainData$Leverage.Ratio)
```

```
##       Min.    1st Qu.     Median       Mean    3rd Qu.       Max.
## -1016.0000     0.5052     0.6670    -2.1783     0.8238     1.0000
```

Third financial ratio falls under the purview of leverage and is basically the total borrowing subtracted from total liabilities and the divided again with total liabilities.

```
TrainData$Liquidity.Ratio.CapitalByLiabilities <- (TrainData$`Net working
capital`/TrainData$`Total liabilities`)
summary(TrainData$Liquidity.Ratio.CapitalByLiabilities)
```

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
## -21.00000  -0.01169   0.08982   0.14023   0.21042  48.50000
```

Fourth and final financial ratio is the liquidity ratio captured by dividing total capital by total liabilities.

CHECK FOR MULTICOLLINEARITY:

```
Matrix1 <- cor(TrainData[,2:15])
corrplot(Matrix1, method = "pie", type = "upper")

Matrix2 <- cor(TrainData[,16:30])
corrplot(Matrix2,method = "pie", type = "upper")

Matrix3 <- cor(TrainData[ ,c(31:40,42,43,44,45)])
corrplot(Matrix3, method = "pie", type = "upper")
```

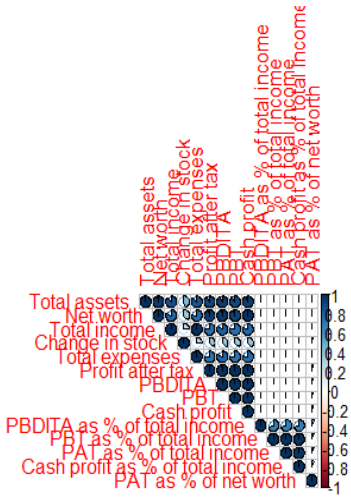


Fig 1.1

Fig 1.2

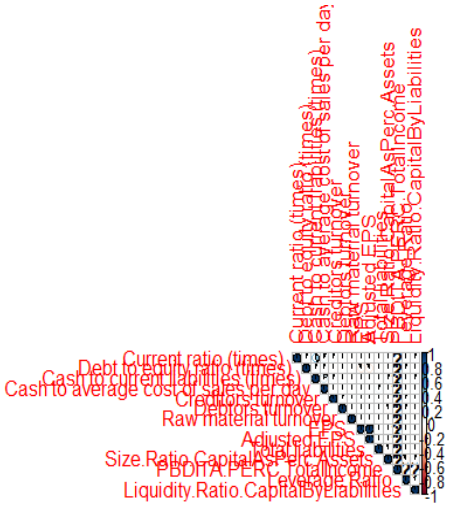


Fig 1.3

In Fig 1.1, from this plot we can understand that there is high positive correlation between some of the company's size parameters such as total assets, total income, net worth e and also some of the profitability para meters have some high correlations such as profit after tax, PBDITA, PBT etc. The different profits as a ratio to total income are correlated to each

other except for profit after tax as a percentage of net worth which is a good indicator that it would be significant while building our model.

In Fig 1.2, here just like the previous fig, we can see high correlation between variables from the company's size parameter such as sales, total capital to have high correlation to variables such as current liabilities, EPS and Adjusted EPS. For our model building perspective choosing a significant variable from liquidity and leverage is also imperative.

Fig 1.3 Although we don't see high correlations between the liquidity ratios such as debt to equity, cash to capital ratio and turnovers such as creditors turnover, debtors turnover etc, it is safe to assume that due to similarity in the information being captured by multiple variable, it would be best to iteratively build our logistic model and try various predictors to obtain an optimal model.

OUTLIER TREATMENT:

```
summary(TrainData$`Total income`)
```

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
##       0.0     108.3     482.6    4348.8    1340.3 2442828.2
```

```
IncomeOut = subset(TrainData, `Total income` < LLIncome | `Total income` >
ULIncome)
dim(IncomeOut)
```

```
## [1] 455  45
```

```
TrainData$`Total income`[TrainData$`Total income` > 3171.7] = 3171.7
```

The total number of outliers present in total income are 455. Here we treat the outliers by the method of capping or flooring i.e the values of outliers is equated to the max value in the third quantile which is 3171.7. The boxplot below shows the distribution of total income post treatment.

```
summary(TrainData$`Profit after tax`)
```

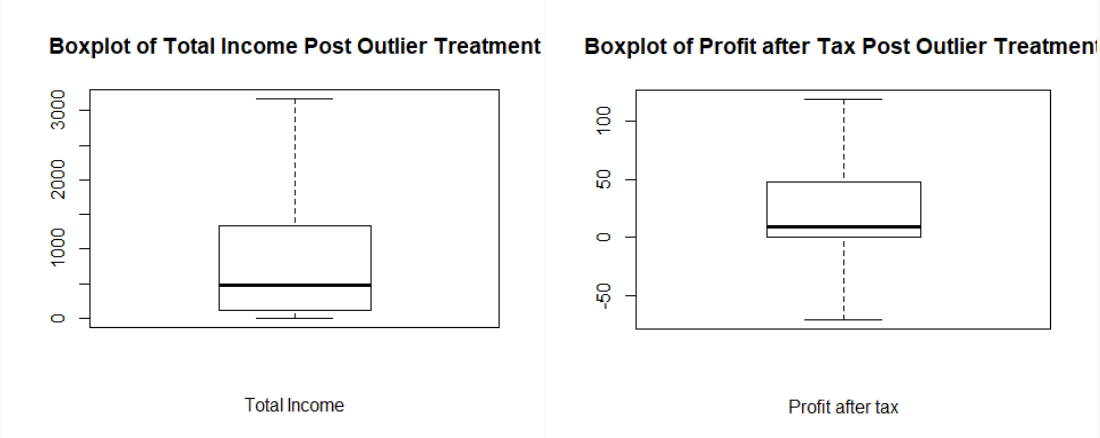
```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
##  -3908.3      0.5      9.5    267.4     48.1 119439.1
```

```
PATOut = subset(TrainData, `Profit after tax` < LLPAT | `Profit after tax`
> ULPAT)
dim(PATOut)
```

```
## [1] 600  45
```

```
TrainData$`Profit after tax`[TrainData$`Profit after tax` >= 119.2] = 119.
2
TrainData$`Profit after tax`[TrainData$`Profit after tax` <= -70.9] = -70.
9
```

Similarly, Profit after tax has 600 outliers present and is treated in the same way as total inc ome. I.E outliers are equated to the maximum and minimum values of the first and third quantile of 119.2 and -70.9 respectively. The box plot below demonstrates the distribution post treatment.




```
summary(TrainData$`PAT as % of net worth`)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -748.72    0.00    7.92   10.27   20.19 2466.67
```

```
PATNWOut = subset(TrainData, `PAT as % of net worth` < LLPATNW | `PAT as % of net worth` > ULPATNW)
dim(PATNWOut)
```

```
## [1] 344  45
```

```
TrainData$`PAT as % of net worth`[TrainData$`PAT as % of net worth` >= 2466.67] = 2466.67
TrainData$`PAT as % of net worth`[TrainData$`PAT as % of net worth` <= -584.44] = -584.44
```

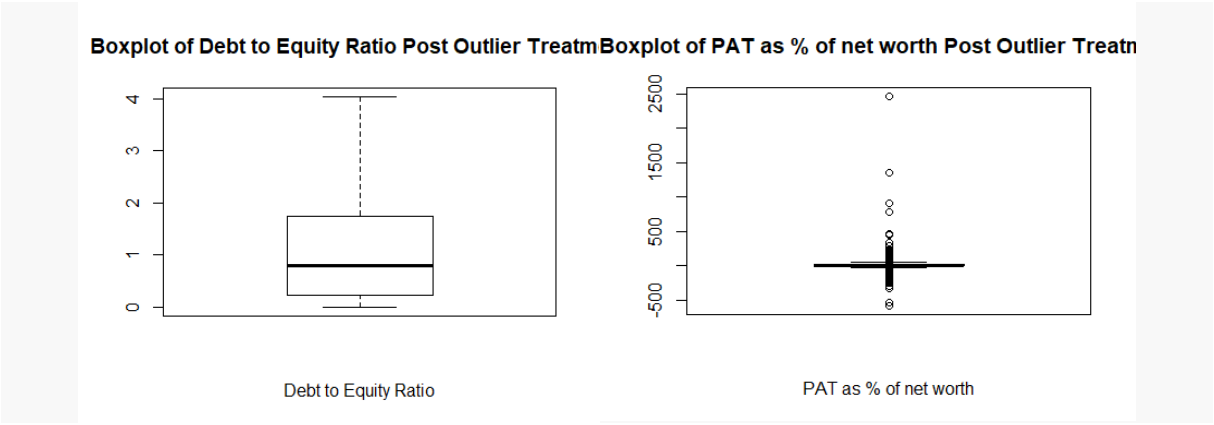
The outliers in PAT as % of net worth are 344 and the minimum and maximum values of first and third quantile of 2466.67 and -584.44 respectively are assigned or equated to make the distribution normal.

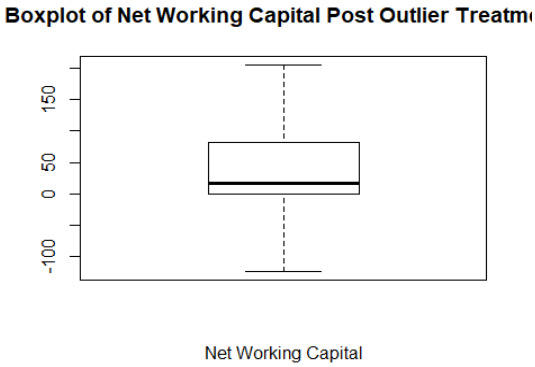
```
DTEOut = subset(TrainData, `Debt to equity ratio (times)` < LLDTE | `Debt to equity ratio (times)` > ULDTE)
dim(DTEOut)
```

```
## [1] 310  45
```

```
TrainData$`Debt to equity ratio (times)`[TrainData$`Debt to equity ratio (times)` >= 4.04] = 4.04
```

Debt to Equity ratio has 310 outliers and upper threshold of 4.04 from the third quantile is s ubstituted to the outlier values. The boxplot below are post the treatment of outliers and are a good indication that the values have now been treated for outliers and represent a fairly normal distribution.





UNIVARIATE ANALYSIS:

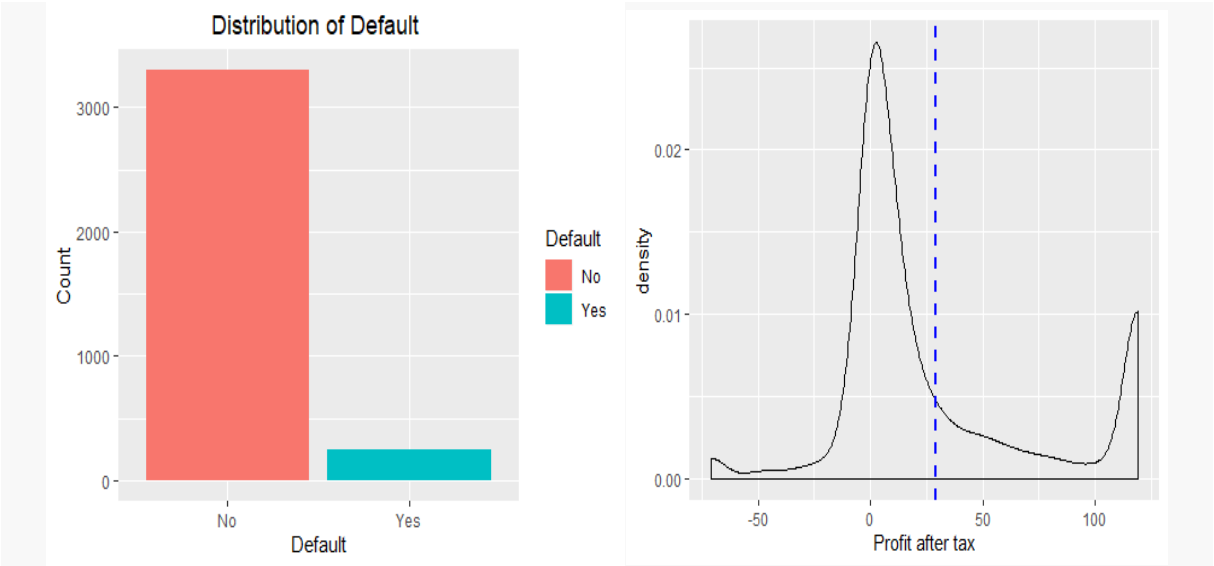


FIG 2.1 **FIG 2.2**

Fig 2.1: This bar chart gives us the number of defaulters and not defaulters. As clearly visible , the number of borrowers or obligators who default are much less than non-defaulters.

Fig 2.2 This distribution in the form of a density graph is for profit after tax. As indicated by the blue line, the mean is around 25-30. Also profit after tax, on the first glimpse, looks normally distributed, with two peaks, one at around 0 and the other post 100.

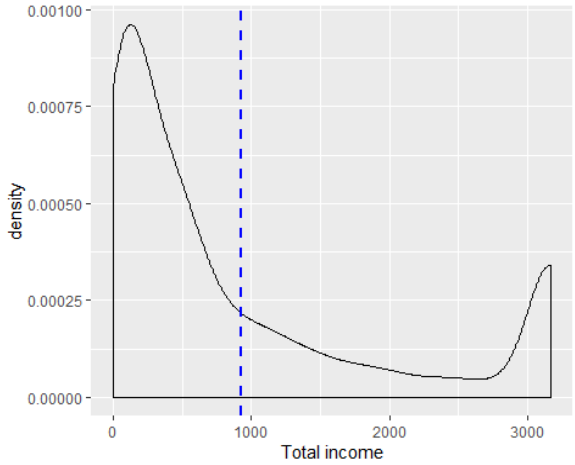


FIG 2.3

Fig 2.3: The density graph of total income gives us the overall mean, indicated by the blue line, at slightly below 1000. Also the density graph is heavily right skewed. The peak is at ust over 100-200 dollars and there seems to be another peak post 3000 dollars.

BIVARIATE ANALYSIS:

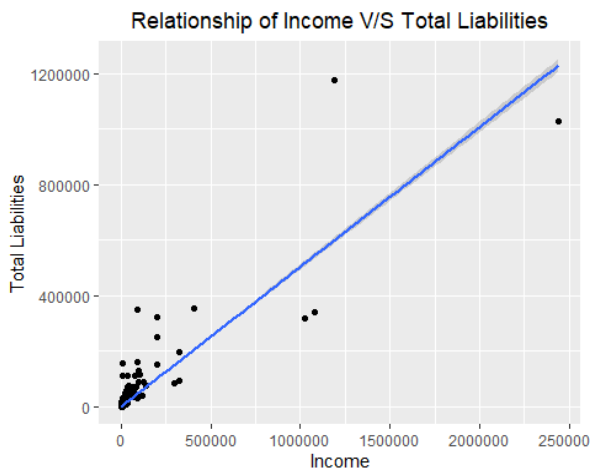


FIG 3.1

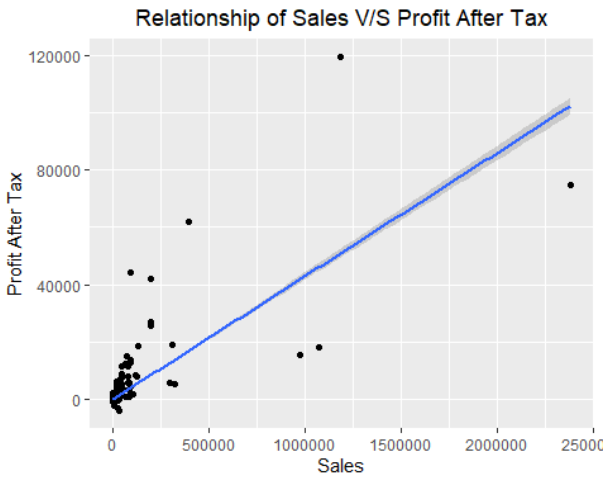


FIG 3.2

Fig 3.1: There seems to be a perfect linear relationship between total liabilities and total income. This signifies a positive correlation between the variables. As the total liabilities increase total income is also bound to increase. This could also be an indication of using either but not both the variables while building our model.

Fig 3.2: Profit after tax and sales have a linear trend which is an indication that as the sales increase profits are also bound to increase. There is a positive correlation between the two, thus we shall consider only one of them towards building our model.

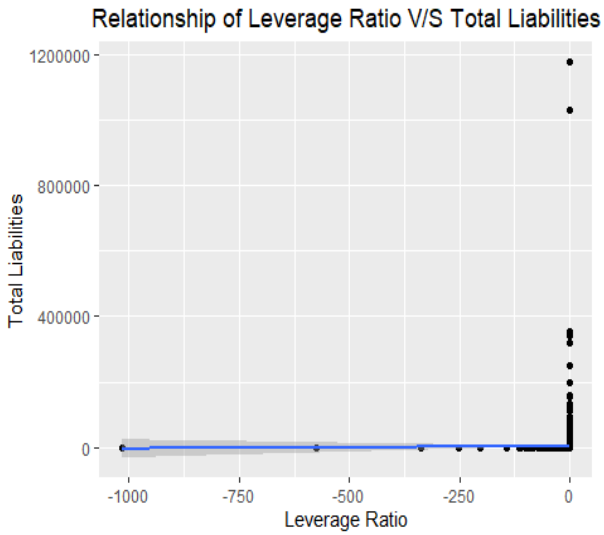


FIG 3.3

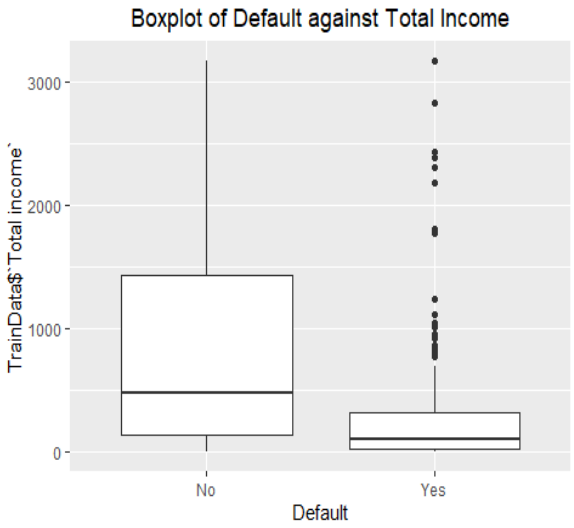


FIG 3.4

Fig 3.3: As the total liabilities increase, the leverage ratio remains relatively straight along the x axis. There is not a linear relationship between them in the truest sense.

Fig 3.3: The number of borrowers who default usually are in the lower income level with some outliers. Similarly, the number of borrowers who do not default have higher mean total income.

Fig 3.5 The number of borrowers who are likely to default have a high debt to equity ratio and borrowers who do not default are at the lower spectrum of debt to equity relatively.

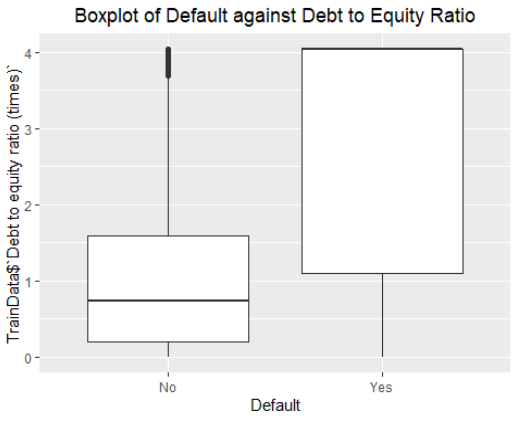


FIG 3.5

LOGISTIC REGRESSION

DATA SLICING

```
prop.table(table(TrainData$Default))
##         No        Yes
## 0.93137532 0.06862468

prop.table(table(TestData$Default))
##          0          1
## 0.92447552 0.07552448
```

For a classification problem, it is imperative that we proceed by ensuring that train and test sets have approximately the same percentages of sample of each target class as their original dataset. Hence, we use stratified sampling here to confirm the same. The percentage of people who will default is 6.8% and the percentage of people who will not default is 93.13% in the training dataset.

FEATURE SELECTION AND APPLICATON

```
Log.Reg.Model <- glm(Default~`Total expenses`+ PBT + `PAT as % of net worth`+ `Net working capital` + `Debt to equity ratio (times)`+ Leverage.Ratio + Liquidity.Ratio.CapitalByLiabilities + Size.Ratio.CapitalAsPerc.Assets, data = TrainData, family = "binomial")

summary(Log.Reg.Model)

## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.9173  -0.2882  -0.2072  -0.1045   6.9219
##
## Coefficients:
##                                    Estimate Std. Error z value
## (Intercept)                       -3.577e+00  1.637e-01 -21.851
## `Total expenses`                  -3.741e-04  8.898e-05  -4.204
## PBT                               -4.196e-03  9.991e-04  -4.200
## `PAT as % of net worth`           -1.393e-02  2.040e-03  -6.827
## `Net working capital`             -3.331e-03  1.527e-03  -2.182
## `Debt to equity ratio (times)`     6.673e-01  5.601e-02  11.914
## Leverage.Ratio                    -4.916e-03  1.126e-03  -4.365
## Liquidity.Ratio.CapitalByLiabilities -1.170e+00  3.713e-01  -3.151
## Size.Ratio.CapitalAsPerc.Assets    2.739e-03  7.328e-04   3.737
##                                   Pr(>|z|)
## (Intercept)                        < 2e-16 ***
## `Total expenses`                  2.63e-05 ***
## PBT                               2.67e-05 ***
## `PAT as % of net worth`           8.67e-12 ***
## `Net working capital`             0.029109 *
## `Debt to equity ratio (times)`     < 2e-16 ***
## Leverage.Ratio                    1.27e-05 ***
## Liquidity.Ratio.CapitalByLiabilities 0.001629 **
## Size.Ratio.CapitalAsPerc.Assets   0.000186 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1771.0  on 3540  degrees of freedom
## Residual deviance: 1152.8  on 3532  degrees of freedom
## AIC: 1170.8
##
## Number of Fisher Scoring iterations: 11
```

After careful consideration of all the variables and iteratively creating several models with different variables, I considered the above model to be the best in terms of prediction, significance of variables (p-values) and AIC score. The variables used in building this model are Total Expenses, PBT (Profit Before Taxes), PAT as a % of net worth, net working capital, debt to equity ratio. These are the original variables provided in our dataset.

Additionally, I will be using the financial ratio variable created earlier and using them in this model building since they are significant predictors with relatively small p-values. These variables are leverage ratio, liquidity ratio (Total capital divided by Total Liabilities) and company size ratio (Capital as a % of Assets)

The AIC here is 1170 which is much better than the other models that were created earlier. To check for multicollinearity between the significant predictors we use the variance inflatio n factor to test out the same as shown below.

```
vif(Log.Reg.Model)
```

```
##                      `Total expenses`                                  P
BT
##                              1.767558                              1.9150
33
##              `PAT as % of net worth`                `Net working capita
l`
##                              1.199129                              1.3200
50
##       `Debt to equity ratio (times)`                       Leverage.Rat
io
##                              1.150383                              1.1574
64
## Liquidity.Ratio.CapitalByLiabilities      Size.Ratio.CapitalAsPerc.Asse
ts
##                              1.445324                              1.0675
14
```

After running the variance inflation factor (vif) on the model it is safe to assume that all the above significant variables we have used in our model are independent or free of any correlation between. This model seems robust and can be used for further analysis.

By calling the vif function on the logistic regression model, we have a check for multicollinearity between the variable used to construct this model. With a threshold of 4 we can positively say that there is no significant correlation between the variables and hence multicollinearity is not an issue between the predictors.

EVALUATING MODEL PERFORMANCE

```
lrtest(Treated.Log.Reg.Model)
```

```
Likelihood ratio test

Model 1: Default ~ `Total expenses` + PBT + `PAT as % of net worth` +
    `Net working capital` + `Debt to equity ratio (times)` +
    Leverage.Ratio + Liquidity.Ratio.CapitalByLiabilities +
Size.Ratio.CapitalAsPerc.Assets
Model 2: Default ~ 1
  #Df  LogLik Df  Chisq Pr(>Chisq)
1   9 -576.41
2   1 -885.49 -8 618.16  < 2.2e-16 ***
```

Null Hypothesis: All betas are zero.
Alternate Hypothesis: At least 1 beta is non zero.

From this likelihood test we can interpret that, intercept only model -885.49 variance was u nknown to us. When we take the full model into consideration, -576.41 variance is unknown to us.

We can say that, 34.90% of the uncertainty inherent in the intercept only model is calibrate d by the full model. Also chi.sq and p-value suggest that we accept the alternate hypothesis that at least one beta is non zero. This model is significant.

```
anova(Treated.Log.Reg.Model)
```

```
## Analysis of Deviance Table


                                  Df Deviance Resid. Df Resid. Dev
NULL                                                3540     1771.0
`Total expenses`                   1   23.282      3539     1747.7
PBT                                1  194.165      3538     1553.5
`PAT as % of net worth`            1  194.802      3537     1358.7
`Net working capital`              1   25.753      3536     1333.0
`Debt to equity ratio (times)`     1  133.764      3535     1199.2
Leverage.Ratio                     1   15.485      3534     1183.7
Liquidity.Ratio.CapitalByLiabilities  1   16.702      3533     1167.0
Size.Ratio.CapitalAsPerc.Assets    1   14.204      3532     1152.8
```

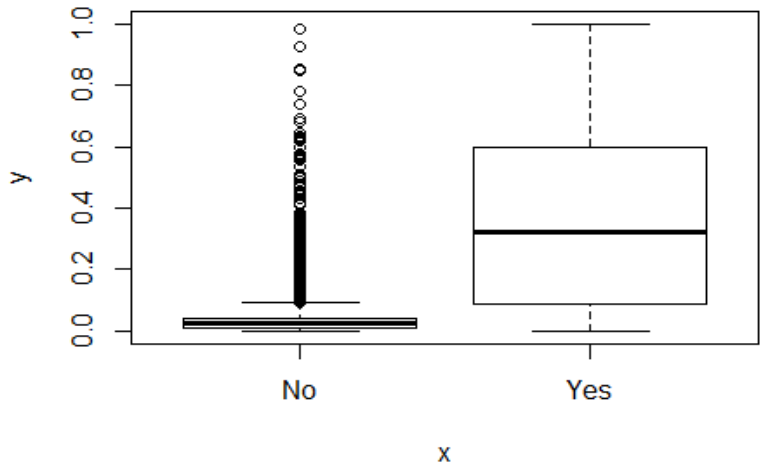
```
pR2(Treated.Log.Reg.Model)
```

```
      llh      llhNull           G2     McFadden         r2ML         r2CU
-576.4079637 -885.4865769  618.1572263    0.3490495    0.1601831 0.4070205
```

Checking for model robustness with McFadden Psuedo R-Squared test. The value of 0.34904 or 34.90 % of the variance is captured by the model, which is relatively good to believe that this model is robust.

DETERMINING A THRESHOLD VALUE

```
plot(TrainData$Default, Log.Reg.Model$fitted.values)
```



With confidence, we can assume, with the help of the above plot, that anything above 0.13 can be classified as a potential defaulter.

PREDICTION ON TRAIN & TEST SAMPLES

```
Log.Reg.Predict <- predict(Log.Reg.Model,data = TrainData, type = "respons
e")
M <- table(TrainData$Default,Log.Reg.Predict > 0.13)
M

# Sensitivity : 92.7228
# Specificity : 70.3703
# Classification Error : 8.81107
# Accuracy : 91.18893
# Precision: 97.699
# FScore : 95.146
######################
#        0     1    #
# No   3058   240   #
# Yes   72    171   #
######################


Predicted.Test <- predict(Log.Reg.Model, newdata = TestData, type = "respo
nse")
M1 <- table(TestData$Default,Predicted.Test > 0.13)
M1

# Sensitivity : 91.905
# Specificity : 88.095
# Classification Error :  7.412
# Accuracy : 92.587
```

```
# Precision: 99.09
# FScore : 95.14

#####################
#        0      1   #
# No    545    48   #
# Yes   5      37   #
#####################
```

The actual values are on the x-axis of the matrix, while the predicted are on the y-axis in this confusion matrix.

CONFUSION MATRIX INTERPRETATION

Performance Measure	*Raw Data*	*Validation Data*
Sensitivity	*92.7228*	*91.905*
Specificity	*70.3703*	*88.095*
Classification Error	*8.81107*	*7.412*
Accuracy	*91.18893*	*92.587*
Precision	*97.699*	*99.09*
FScore	*95.146*	*95.14*

The sensitivity/recall or true positive rate (TPR), i.e the number of borrowers who are not likely to default that we actually predicted correctly is 92.78% when training the model and 91.90% when validating on the test data.

Specificity or true negative rate i.e the number of borrowers that default and are predicted likewise and identified correctly is 70.37% while training the model and 88.09% while validating the model on test data.

The error rate while training the model decreases from 8.81% to 7.31% when validating it on test data.

Similarly, since accuracy is 1 – Error Rate, we can ascertain that accuracy of the model increases slightly by 1.4% (from 91.18% to 92.587%) when training the model and validating on test data, which is a good indication in regards to, robustness and reliability of our model.

RANK ORDERED TABLES (K.S) INTERPRETATION

```
RankTable

##                 Decile Count Defaulters NonDefaulters DRate CumDefaulte
rs
##  1:          (0.157,1]   354        164           190 46.33           1
64
##  2:     (0.0654,0.157]   354         31           323  8.76           1
```

```
95
##  3:    (0.0422,0.0654]   354         17           337  4.80           2
12
##  4:    (0.0316,0.0422]   354          5           349  1.41           2
17
##  5:    (0.0249,0.0316]   354          6           348  1.69           2
23
##  6:    (0.0193,0.0249]   354          6           348  1.69           2
29
##  7:    (0.0135,0.0193]   354          5           349  1.41           2
34
##  8:   (0.00662,0.0135]   354          3           351  0.85           2
37
##  9:  (0.00082,0.00662]   354          3           351  0.85           2
40
## 10: [2.22e-16,0.00082]   355          3           352  0.85           2
43
##     CumNonDefaulters CumRelDefault CumRelNonDefault    KS
##  1:              190         67.49             5.76 61.73
##  2:              513         12.76             9.79  2.97
##  3:              850          7.00            10.22  3.22
##  4:             1199          2.06            10.58  8.52
##  5:             1547          2.47            10.55  8.08
##  6:             1895          2.47            10.55  8.08
##  7:             2244          2.06            10.58  8.52
##  8:             2595          1.23            10.64  9.41
##  9:             2946          1.23            10.64  9.41
## 10:             3298          1.23            10.67  9.44
```

The top four probability deciles contain the majority of the defaulters. The K.S statistic is 61.73on both training dataset and test dataset. The top four deciles cover the highest default rate among the population upto 56%.