26-07-2020

CREDIT RISK MODELLING - FRA

LOGISTIC REGRESSION TO DEVELOP THE CREDIT DEFAULT MODEL

PRESENTED BY: SHILPA GIRIDHAR

CONTENTS

DESCRIPTION

Problem Statement

For this assignment, you are requested to create an India credit risk(default) model, using the data provided in the spreadsheet raw-data.xlsx, and validate it on validation_data.xlsx. Please use the logistic regression framework to develop the credit default model.:

[Hint code]

Data description - After removing variables for multicollinearity, we should try to take at least one variable for creating the model from each of the 4 factors namely -

1) Profitability

2) Leverage

3) Liquidity

4) Company's size

In Dr. Sarkar's video of Default Risk Estimation, he has clearly bifurcated all the variables in different buckets.

Creation of new variables - This is an important step in the project as the company which is the biggest in size, will also have bigger asset size, cash flows etc. (Hint: We need to think in terms of ratios - Equity to asset ratio, debt to equity ratio etc)

Dependent variable - We need to create a default variable which should take the value of 1 when net worth is negative & 0 when net worth is positive.

Validation Dataset - We need to build the model on raw dataset and check the model performance measures on validation dataset.

Requirements

Perform the following :

1. **EDA (40 Marks)**
    - Outlier Treatment (10 marks)
    - Missing Value Treatment (7.5 marks)
    - New Variables Creation (One ration for profitability, leverage, liquidity and company's size each) (7.5 marks)
    - Check for multicollinearity (7.5 marks)
    - Univariate & bivariate analysis (7.5 marks)
2. **Modelling and Predicting (30 Marks)**
    - Build Logistic Regression Model on most important variables (15 marks)
    - Analyze coefficient & their signs (15 marks)
3. **Model Performance Measures (20 Marks)**
    - Predict accuracy of model on dev and validation datasets (10 marks)
    - Sort the data in descending order based on probability of default and then divide into 10 dociles based on probability & check how well the model has performed (10 marks)

DATA EXPLORATION

Reading Data

- The data shows that it's a company balance sheet dataset, with 3541 observations and 52 variables in the Original dataset and 715 observations and 52 variables in the Validation dataset.
- Performed the str and summary function.
- We can observe that there are many missing data (NAs) and that the data set is unbalanced dataset; Few columns are incorrectly detected as of type character;
- Deposits column has all NAs and hence can be dropped
- Below is the snapshot of the Original dataset – which will be pruned to remove NAs, multi-collinearity

```
      Num       Networth Next Year  Total assets        Net worth          Total income
 Min.   :   1   Min.   :-74265.6   Min.   :      0.1   Min.   :      0.0   Min.   :      0.0
 1st Qu.: 886   1st Qu.:     31.7   1st Qu.:     91.3   1st Qu.:     31.3   1st Qu.:    106.4
 Median :1773   Median :    116.3   Median :    309.7   Median :    102.3   Median :    444.9
 Mean   :1772   Mean   :   1616.3   Mean   :   3443.4   Mean   :   1295.9   Mean   :   4582.8
 3rd Qu.:2658   3rd Qu.:    456.1   3rd Qu.:   1098.7   3rd Qu.:    377.3   3rd Qu.:   1440.9
 Max.   :3545   Max.   :805773.4   Max.   :1176509.2   Max.   :613151.6   Max.   :2442828.2
                                                                           NA's   :198
 Change in stock    Total expenses      Profit after tax      PBDITA
 Min.   :-3029.40   Min.   :     -0.1   Min.   : -3908.30   Min.   :  -440.7
 1st Qu.:   -1.80   1st Qu.:     95.8   1st Qu.:     0.50   1st Qu.:     6.9
 Median :    1.60   Median :    407.7   Median :     8.80   Median :    35.4
 Mean   :   41.49   Mean   :   4262.9   Mean   :   277.36   Mean   :   578.1
 3rd Qu.:   18.05   3rd Qu.:   1359.8   3rd Qu.:    52.27   3rd Qu.:   150.2
 Max.   :14185.50   Max.   :2366035.3   Max.   :119439.10   Max.   :208576.5
 NA's   :458        NA's   :139         NA's   :131         NA's   :131
      PBT             Cash profit         PBDITA as % of total income PBT as % of total income
 Min.   : -3894.80   Min.   : -2245.70   Min.   :-6400.000            Min.   :-21340.00
 1st Qu.:     0.70   1st Qu.:     2.90   1st Qu.:    5.000            1st Qu.:     0.55
 Median :    12.40   Median :    18.85   Median :    9.660            Median :     3.31
 Mean   :   383.81   Mean   :   392.07   Mean   :    4.571            Mean   :   -17.28
 3rd Qu.:    71.97   3rd Qu.:    93.20   3rd Qu.:   16.390            3rd Qu.:     8.80
 Max.   :145292.60   Max.   :176911.80   Max.   :  100.000            Max.   :   100.00
 NA's   :131         NA's   :131         NA's   :68                   NA's   :68
 PAT as % of total income Cash profit as % of total income PAT as % of net worth
 Min.   :-21340.00        Min.   :-15020.000               Min.   :-748.72
 1st Qu.:     0.35        1st Qu.:     2.020               1st Qu.:   0.00
 Median :     2.34        Median :     5.640               Median :   7.92
 Mean   :   -19.20        Mean   :    -8.229               Mean   :  10.27
 3rd Qu.:     6.34        3rd Qu.:    10.700               3rd Qu.:  20.19
 Max.   :   150.00        Max.   :   100.000               Max.   :2466.67
 NA's   :68               NA's   :68
```

```
     Sales           Income from financial services  Other income      Total capital
 Min.   :      0.1   Min.   :    0.00               Min.   :    0.00  Min.   :    0.1
 1st Qu.:    112.7   1st Qu.:    0.40               1st Qu.:    0.40  1st Qu.:   13.1
 Median :    453.1   Median :    1.80               Median :    1.40  Median :   42.1
 Mean   :   4549.5   Mean   :   80.84               Mean   :   41.36  Mean   :  216.6
 3rd Qu.:   1433.6   3rd Qu.:    9.68               3rd Qu.:    5.97  3rd Qu.:  100.3
 Max.   :2384984.4   Max.   :51938.20               Max.   :42856.70  Max.   :78273.2
 NA's   :259         NA's   :935                    NA's   :1295      NA's   :4
 Reserves and funds Deposits (accepted by commercial banks)   Borrowings
 Min.   : -6525.9   Mode:logical                            Min.   :     0.10
 1st Qu.:     5.0   NA's:3541                               1st Qu.:    23.95
 Median :    54.8                                           Median :    99.20
 Mean   :  1163.8                                           Mean   :  1122.28
 3rd Qu.:   277.3                                           3rd Qu.:   352.60
 Max.   :625137.8                                           Max.   :278257.30
 NA's   :85                                                 NA's   :366
 Current liabilities & provisions Deferred tax liability Shareholders funds
 Min.   :     0.1                 Min.   :    0.1         Min.   :     0.0
 1st Qu.:    17.8                 1st Qu.:    3.2         1st Qu.:    32.0
 Median :    69.4                 Median :   13.4         Median :   105.6
 Mean   :   940.6                 Mean   :  227.2         Mean   :  1322.1
 3rd Qu.:   261.7                 3rd Qu.:   50.0         3rd Qu.:   393.2
 Max.   :352240.3                 Max.   :72796.6         Max.   :613151.6
 NA's   :96                       NA's   :1140
 Cumulative retained profits Capital employed      TOL/TNW
 Min.   : -6534.3            Min.   :     0.0   Min.   :-350.480
 1st Qu.:     1.1            1st Qu.:    60.8   1st Qu.:   0.600
 Median :    37.1            Median :   214.7   Median :   1.430
 Mean   :   890.5            Mean   :  2328.3   Mean   :   3.994
 3rd Qu.:   202.3            3rd Qu.:   767.3   3rd Qu.:   2.830
 Max.   :390133.8            Max.   :891408.9   Max.   : 473.000
 NA's   :38

 Total term liabilities / tangible net worth Contingent liabilities / Net worth (%)
 Min.   :-325.600                            Min.   :    0.00
 1st Qu.:   0.050                            1st Qu.:    0.00
 Median :   0.340                            Median :    5.33
 Mean   :   1.844                            Mean   :   53.94
 3rd Qu.:   1.000                            3rd Qu.:   30.76
 Max.   : 456.000                            Max.   :14704.27

 Contingent liabilities Net fixed assets    Investments        Current assets
 Min.   :     0.1       Min.   :     0.0   Min.   :     0.00   Min.   :     0.1
 1st Qu.:     6.3       1st Qu.:    26.0   1st Qu.:     1.00   1st Qu.:    36.2
 Median :    38.0       Median :    93.5   Median :     8.35   Median :   145.1
 Mean   :   932.9       Mean   :  1189.7   Mean   :   694.73   Mean   :  1293.4
 3rd Qu.:   192.7       3rd Qu.:   344.9   3rd Qu.:    64.30   3rd Qu.:   502.2
 Max.   :559506.8       Max.   :636604.6   Max.   :199978.60   Max.   :354815.2
 NA's   :1188           NA's   :118        NA's   :1435        NA's   :66
 Net working capital Quick ratio (times) Current ratio (times) Debt to equity ratio (times)
 Min.   :-63839.0    Min.   :  0.000     Min.   :  0.00        Min.   :  0.00
 1st Qu.:    -1.1    1st Qu.:  0.410     1st Qu.:  0.93        1st Qu.:  0.22
 Median :    16.2    Median :  0.670     Median :  1.23        Median :  0.79
 Mean   :   138.6    Mean   :  1.401     Mean   :  2.13        Mean   :  2.78
 3rd Qu.:    84.2    3rd Qu.:  1.030     3rd Qu.:  1.71        3rd Qu.:  1.75
 Max.   : 85782.8    Max.   :341.000     Max.   :505.00        Max.   :456.00
 NA's   :32          NA's   :93          NA's   :93
 Cash to current liabilities (times) Cash to average cost of sales per day Creditors turnover
 Min.   :  0.0000                    Min.   :     0.00                     Length:3541
 1st Qu.:  0.0200                    1st Qu.:     2.79                     Class :character
 Median :  0.0700                    Median :     8.03                     Mode  :character
 Mean   :  0.4904                    Mean   :   158.44
 3rd Qu.:  0.1900                    3rd Qu.:    21.79
 Max.   :165.0000                    Max.   :128040.76
 NA's   :93                          NA's   :85
```

```
 Debtors turnover    Finished goods turnover WIP turnover       Raw material turnover
 Length:3541         Length:3541             Length:3541        Length:3541
 Class :character    Class :character        Class :character   Class :character
 Mode  :character    Mode  :character        Mode  :character   Mode  :character



 Shares outstanding Equity face value       EPS                 Adjusted EPS
 Length:3541         Length:3541        Min.    :-843181.8   Min.    :-843181.8
 Class :character    Class :character   1st Qu.:       0.0   1st Qu.:       0.0
 Mode  :character    Mode  :character   Median :       1.4   Median :       1.2
                                        Mean   :    -220.3   Mean   :    -221.5
                                        3rd Qu.:       9.6   3rd Qu.:       7.5
                                        Max.   :   34522.5   Max.   :   34522.5

 Total liabilities    PE on BSE
 Min.    :      0.1   Length:3541
 1st Qu.:      91.3   Class :character
 Median :     309.7   Mode  :character
 Mean   :    3443.4
 3rd Qu.:    1098.7
 Max.   :1176509.2
```

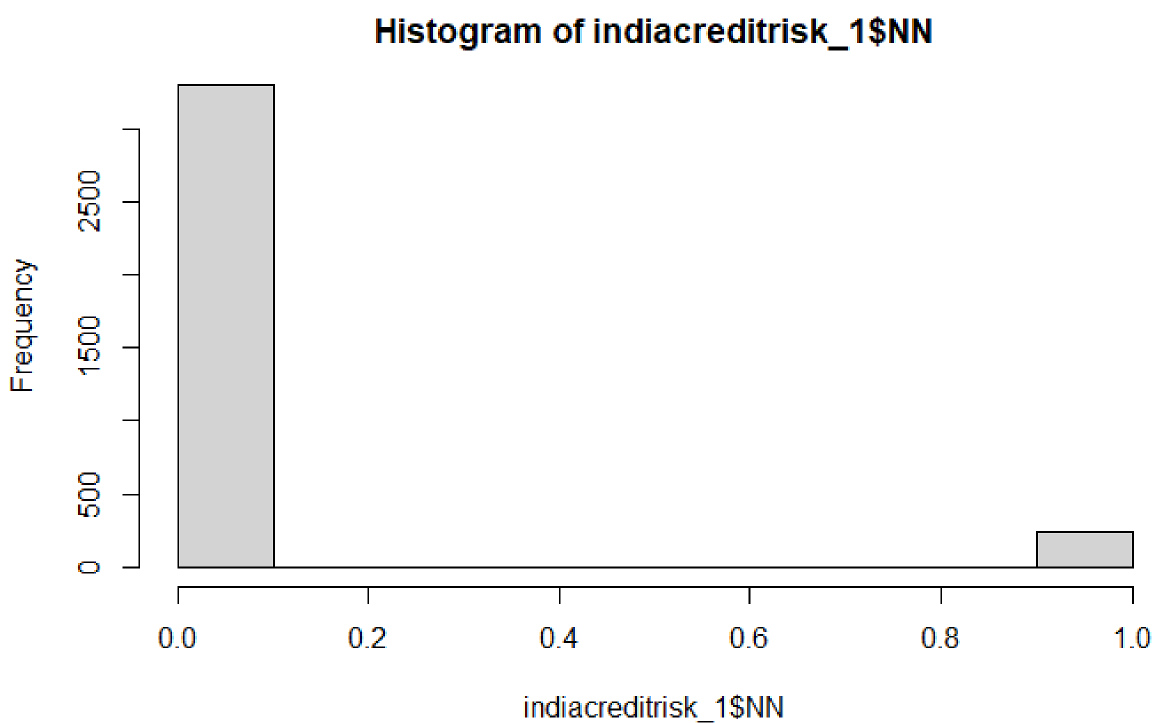
Deriving Y Variable

- Network-Next-Year column is used to derive the Default column, which is the dependent variable or the Y variable
- Derive the Y-variable from NNY column, rename as ‘NN’ and drop the NNY column
- Networth Next Year <0, mark as 1 (Defaulter)
- Networth Next Year >=0, mark as 0 (NonDefaulter).

```
#change to easier column names
colnames(indiacreditrisk_1) = c("Num", "NNY", "TotAssests", "NW", "TotIncome",
"ChangeinStock%","TotExpenses", "PAT", "PBDITA", "PBT",
"CashProfit","PBDITA_TotIncome", "PBD_TotIncome", "PAT_TotIncome",
"CashProfit_TotIncome", "PAT_NW","Sales",
"Income_FinServ","Income_Other","TotCapital","ReserveFunds","Deposits",
"Borrowings",
"Curr_Liabilities","Deferred_TaxLiability","ShareHolderFunds","RetainedProfits_Cumu
lative",
"Capital_employed","TolTNW","TtlTNW","Contingent_LiabilitiesNW","Contingent_Liabili
ties","Net_FixedAssets","Investments","Curr_Assets","Capital_NW","Quick_Ratio","Cur
r_Ratio","Debt-to-Equity_Ratio","Cash-to-Curr_Liabilities","Cash-to-
AvgCost_Sales_PerDay","Creditors_TurnOver","Debtors_TurnOver","FinishedfGoods_TurnO
ver","WIP_TurnOver", "RawMaterial_TurnOver", "Shares_Outstanding",
"EquityFaceValue","EPS","AdjustedEPS","Tot_Liabilities","PE_on_BSE")

names(indiacreditrisk_1)
```

```
#Derive the Y-variable from NNY column and drop the NNY column

indiacreditrisk_1$NN<- ifelse(indiacreditrisk_1$NNY<0,1,0)
hist(indiacreditrisk_1$NN)
#Checking distribution of dependent variable
summary(as.factor(indiacreditrisk_1$NN))

#    0    1
# 3307  234
234/(3307+234)
```

Data shows that 6.6% are defaulters and majority of them are Non-defaulters



Data Preparation

- Removing NAs - We can see that columns Deposits (all), PE_on_BSE (2194) has lot of NAs

  (2194/3541) # 62%

  (1435/3541) # 40.5%

- We can see from this time series that there seems to be seasonal variation in the number of productions per month: there is a crest (or peaks) and trough in the cycle every year. To estimate the trend, seasonal and irregular components of this time series, we decompose the timeseries and plot it. We can also use "autoplot" function from "seasonal" library package which does similar plotting of the components

```
# Columns Deposits (all) and PE_on_BSE (2194) has lot of NAs
(2194/3541) # 62%
(1435/3541) # 40.5%
# Since ratio of NAs is very high we drop these columns (22,52) and proceed
with treating outliers and imputing NAs in other columns.
# we can also remove the Num as well as NNY column from which we have the deriv
ed Y-variable from this column

indiacreditrisk_2 = indiacreditrisk_1[,-c(1,2,22,52)]

# after removal there are 3541 obs and 49 columns

#Creating Default ('NN') as factor type variable
indiacreditrisk_2$NN=as.factor(indiacreditrisk_2$NN)

str(indiacreditrisk_2)
#Creating other column that are as character as numeric type variable
indiacreditrisk_2$Creditors_TurnOver = as.numeric(indiacreditrisk_2$Creditors_T
urnOver)
indiacreditrisk_2$Debtors_TurnOver = as.numeric(indiacreditrisk_2$Debtors_TurnO
ver)
indiacreditrisk_2$FinishedfGoods_TurnOver = as.numeric(indiacreditrisk_2$Finish
edfGoods_TurnOver)
indiacreditrisk_2$WIP_TurnOver = as.numeric(indiacreditrisk_2$WIP_TurnOver)
indiacreditrisk_2$RawMaterial_TurnOver = as.numeric(indiacreditrisk_2$RawMateri
al_TurnOver)
indiacreditrisk_2$Shares_Outstanding = as.numeric(indiacreditrisk_2$Shares_Outs
tanding)
indiacreditrisk_2$EquityFaceValue = as.numeric(indiacreditrisk_2$EquityFaceValu
e)
str(indiacreditrisk_2)
```

- Outlier treatment for all columns in the train (original raw) and test (validation) dataset using the function below.

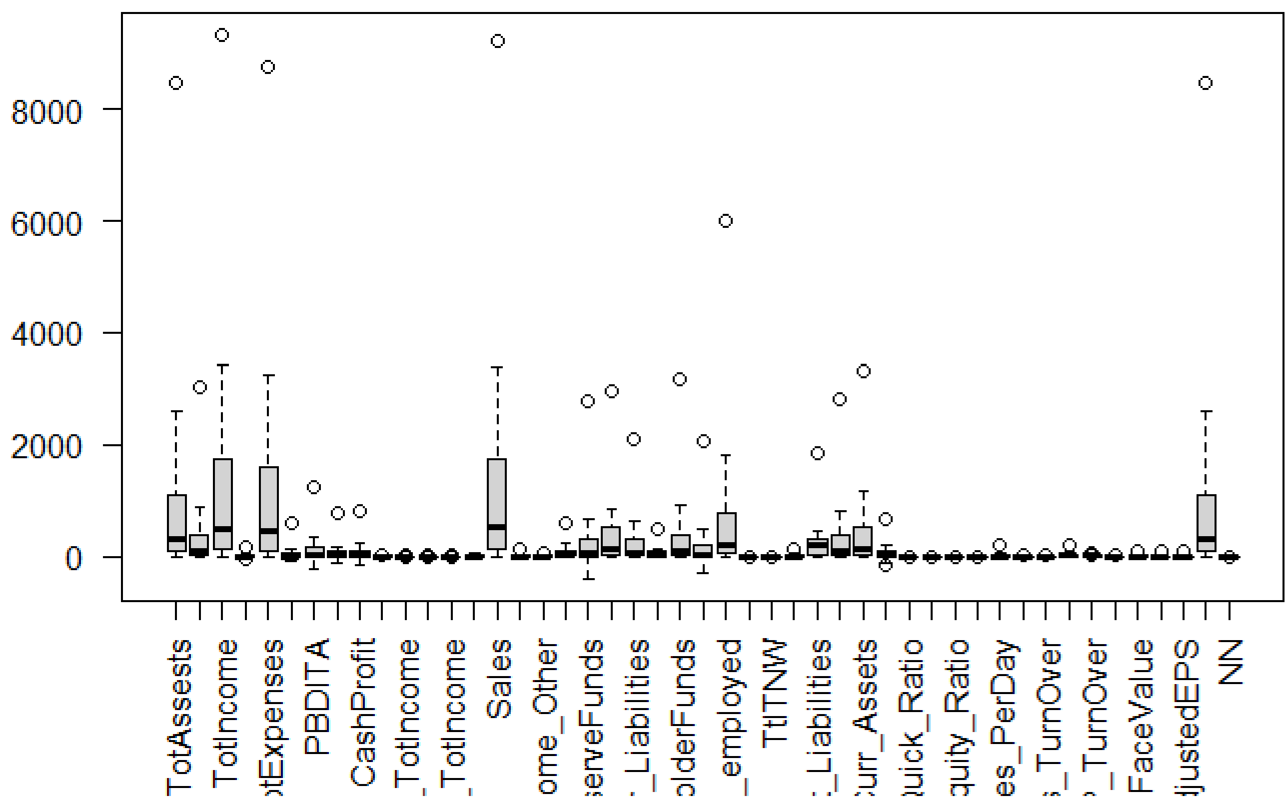
```
#Outlier treatment

outlier_capping = function(x){
  qnt = quantile(x, probs=c(.25, .75), na.rm = T)
 caps = quantile(x, probs=c(.05, .95), na.rm = T)
    H = 1.5 * IQR(x, na.rm = T)
   x[x < (qnt[1] - H)] <- caps[1]
   x[x > (qnt[2] + H)] <- caps[2]
   return(x) }

#boxplot all variables and verify

boxplot(indiacreditrisk_2[,-c(44,49)],las=2)
# excluding investments, share outstanding and default column as the value range is very
 high

boxplot(indiacreditrisk_val2[,-c(1,32,45)],las=2)
# excluding investments, share outstanding and default column as the value range is very
 high
```

- Removing NAs in Train & Test Dataset

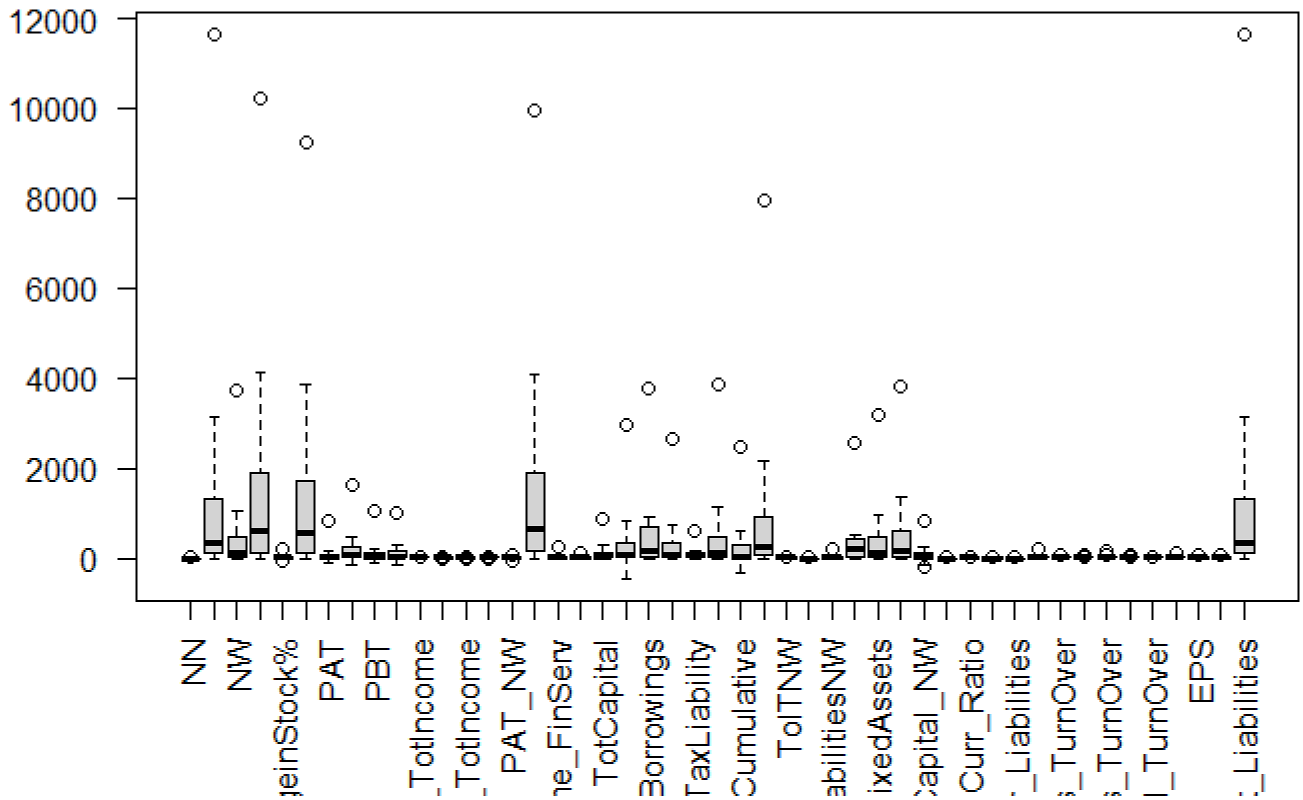
```
# Function replaces NA by mean:
replace_by_mean <- function(x) {
  x[is.na(x)] <- mean(x, na.rm = TRUE)
  return(x)
}


# A function imputes NA observations for categorical variables:

replace_na_categorical <- function(x) {
  x %>%
    table() %>%
    as.data.frame() %>%
    arrange(-Freq) ->> my_df

  n_obs <- sum(my_df$Freq)
  pop <- my_df$. %>% as.character()
  set.seed(29)
  x[is.na(x)] <- sample(pop, sum(is.na(x)), replace = TRUE, prob = my_df$Freq)
  return(x)
}

# Use the two functions in train dataset:
train_data <- indiacreditrisk_2 %>%
  mutate_if(is.numeric, replace_by_mean) %>%
  mutate_if(is.factor, replace_na_categorical)

summary(train_data)
dim(train_data)
```

```
sort(colSums(is.na(train_data)))
boxplot(train_data[,-c(31,44)],las=2)


# Use the two functions in test dataset:
test_data <- indiacreditrisk_val2 %>%
  mutate_if(is.numeric, replace_by_mean) %>%
  mutate_if(is.factor, replace_na_categorical)

summary(test_data)
dim(test_data)
sort(colSums(is.na(test_data)))
boxplot(test_data)
boxplot(test_data[,-c(32,45)],las=2)
```

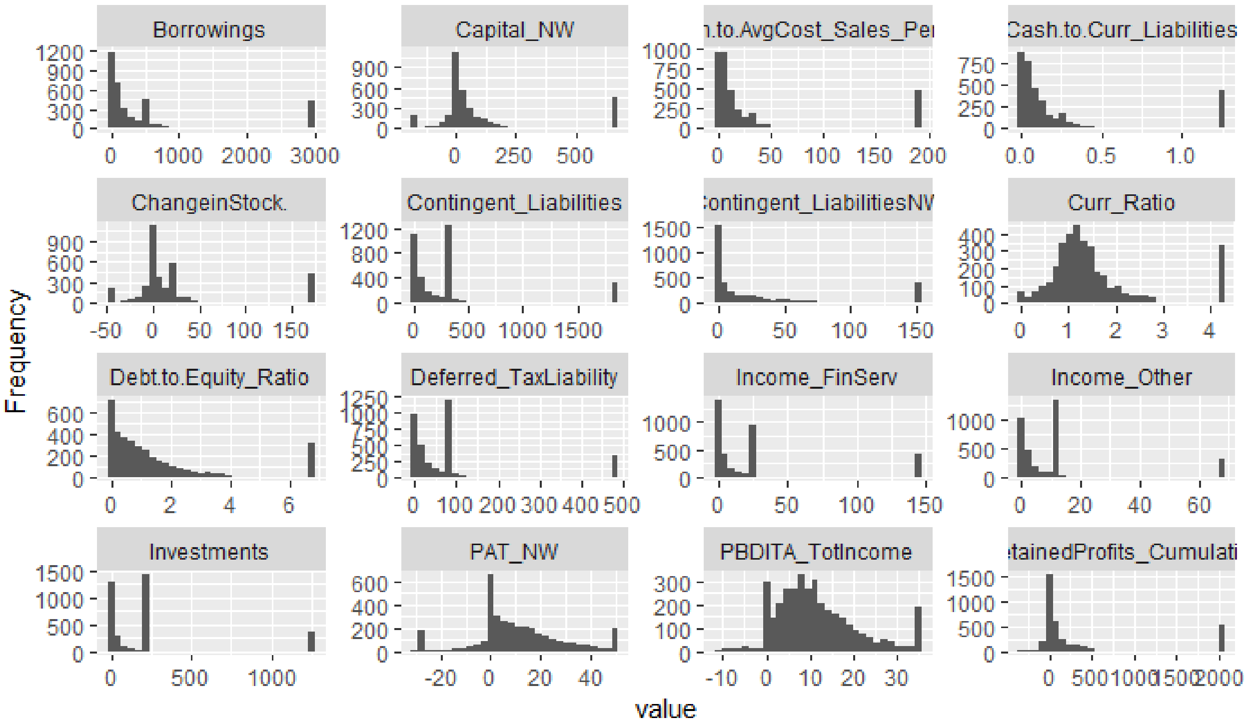
Boxplot of train dataset (original raw) after treatment of NAs and outliers -



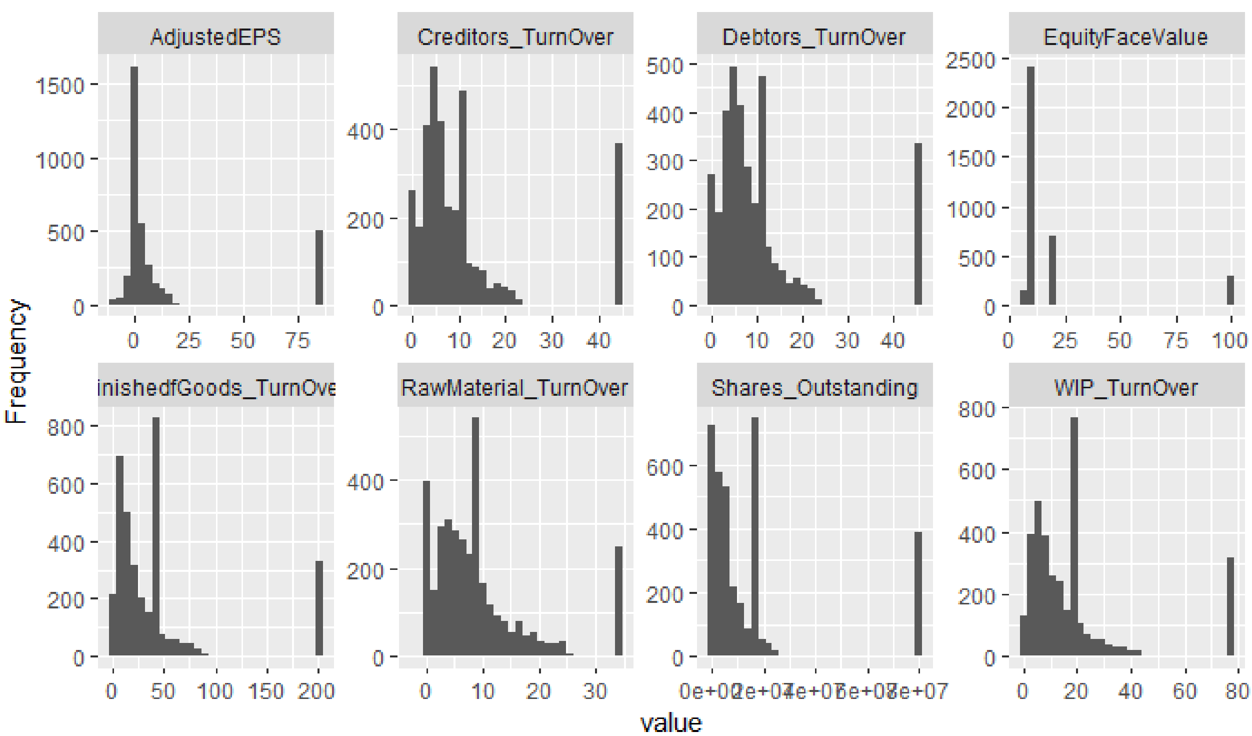
Boxplot of test dataset (validation) after treatment of NAs and outliers -



Histogram plot after removal of highly collinear variables on reduced train dataset



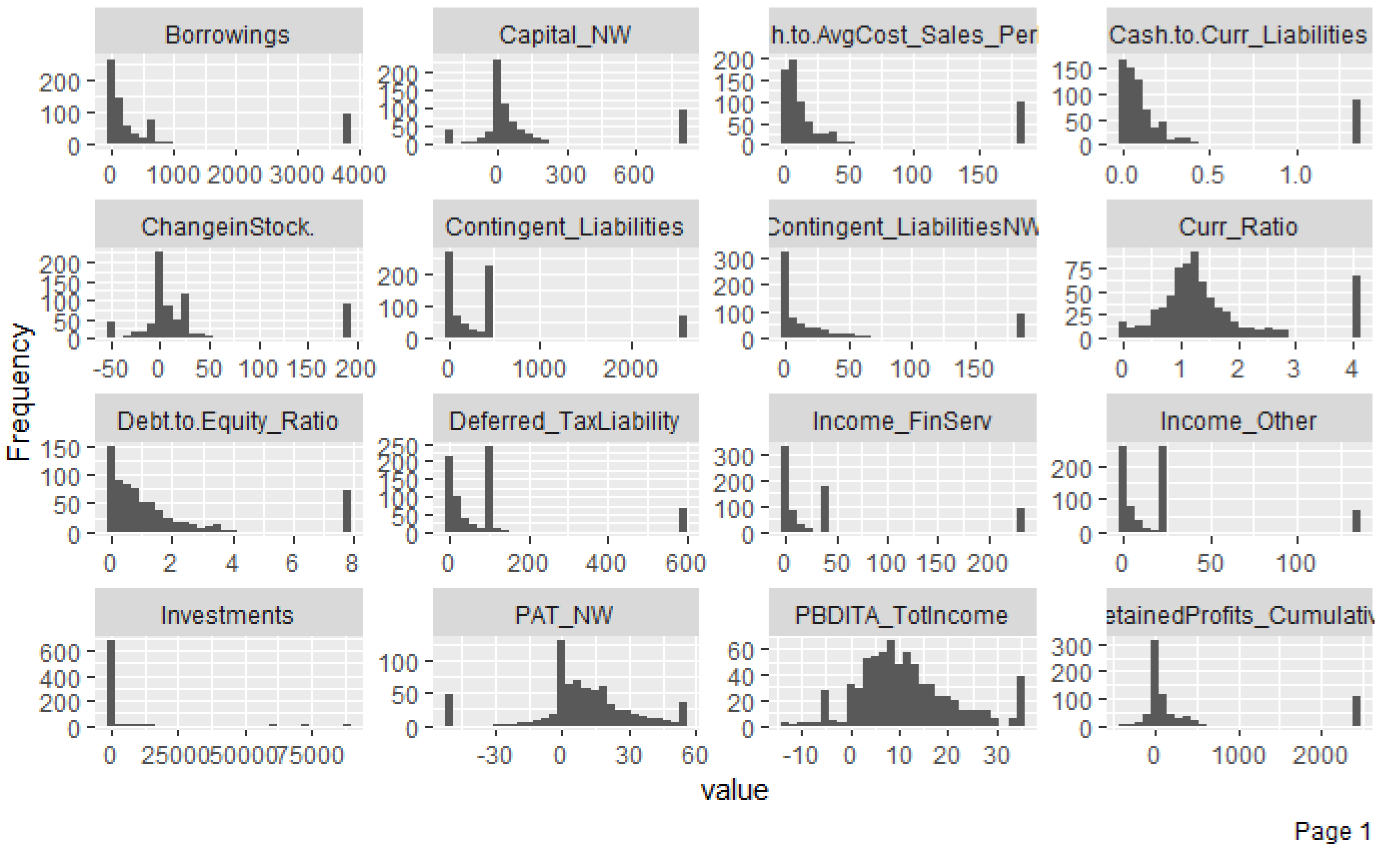
Page 1

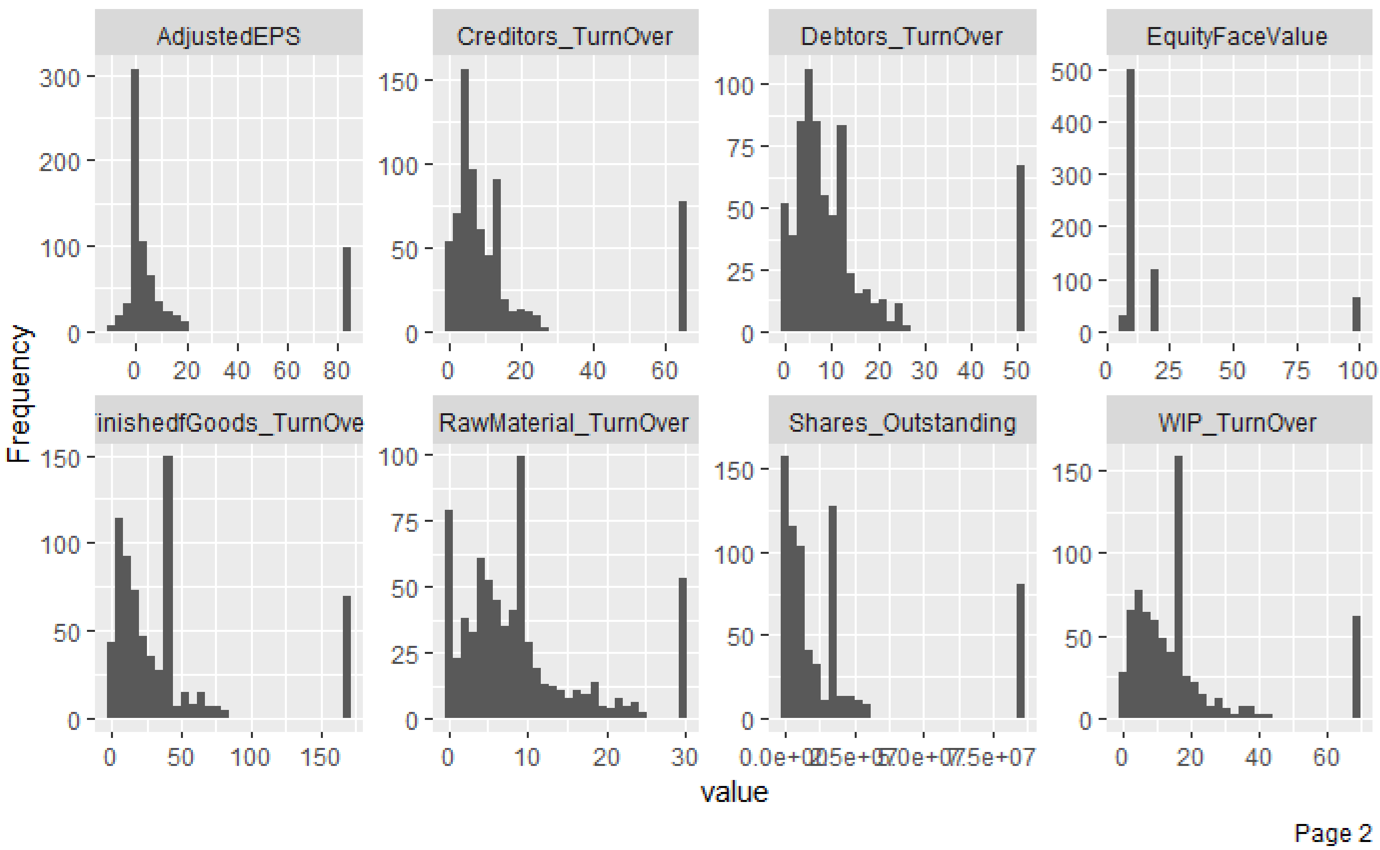


Page 2

```
> dim(reduced_test_data)
[1] 715  25

names(reduced_test_data)

[1] "NN"                          "ChangeinStock%"              "PBDITA_TotIncome"
[4] "PAT_NW"                      "Income_FinServ"              "Income_Other"
[7] "Borrowings"                  "Deferred_TaxLiability"       "RetainedProfits_Cumul
ative"
[10] "Contingent_LiabilitiesNW"    "Contingent_Liabilities"      "Investments"
[13] "Capital_NW"                  "Curr_Ratio"                  "Debt-to-Equity_Ratio
"
[16] "Cash-to-Curr_Liabilities"    "Cash-to-AvgCost_Sales_PerDay" "Creditors_TurnOver"
[19] "Debtors_TurnOver"            "FinishedfGoods_TurnOver"     "WIP_TurnOver"
[22] "RawMaterial_TurnOver"        "Shares_Outstanding"          "EquityFaceValue"
[25] "AdjustedEPS"
```

Histogram plot after removal of highly collinear variables on reduced test dataset





AdjustedEPS
Creditors_TurnOver
Debtors_TurnOver
EquityFaceValue
inishedfGoods_TurnOve
RawMaterial_TurnOver
Shares_Outstanding
WIP_TurnOver
Frequency
value
Page 2

MODELLING AND PREDICTING

LOGISTIC REGRESSION

- Build a Logistic Regression model on the important variables.
- Additional ratios for Profitability, Liquidity and Leverage is calculated as per the requirement.

CREATING ADDITIONAL VARIABLES

PROFITABILITY RATIO

- PBDITA Profit before Depreciation, Income, Tax and Amortisation. Its stands for Total Revenue.

- PBT - PROFIT BEFORE TAX (Total Revenue - Total Expenses)

PBT = PBDITA – (INTEREST + DEPRECIATION)

This is the actual income earned by the entity.

- PAT PROFIT AFTER TAX (PROFIT BEFORE TAX – TAX EXPENSES)

PAT = PBT - TAX EXPENSES

The tax amount when subtracted from PBT is called as Profit after Tax.

- All the above (PBDITA, PBT, PAT) as a percentage of Total Income indicates the financial health of the organization. The Higher the ratio, the better the financial health.
- SALES - Income generating by selling the product or goods

- Gross Profitability Ratio = PROFIT AFTER TAX /Sales

```
train_data$ProfitabilityRatio_new = (train_data$PAT)/(train_data$Sales)
test_data$ProfitabilityRatio_new = (test_data$PAT)/(test_data$Sales)
```

- Net Profit Margin Ratio = = Net Income /Sales; Higher the percentage the more profitable the business is

```
train_data$NetProfitMargin_new = (train_data$TotIncome)/(train_data$Sales)
test_data$NetProfitMargin_new = (test_data$TotIncome)/(test_data$Sales)
```

LIQUIDITY RATIO

- Liquidity Ratio = Networking Capital / Total Assets

```
train_data$LiquidityRatio_new = (train_data$Capital_NW)/(train_data$TotAssests)
test_data$LiquidityRatio_new = (test_data$Capital_NW)/(test_data$TotAssests)
```

- Current Liquidity Ratio = This is the ratio of Current assets and current liabilities. The higher the value the better the capacity of the organization to clear its debts.

```
# train_data$'Curr_Ratio'   - this already available in the dataset
```

- Quick ratio- It is the total cash equivalent divided by the current liabilities. If the ratio is high, it indicates the organization has capabilities to repay its liabilities. In fact, it has the ability to repay in faster or quick time. If the ratio is low or less than one, the management has to work towards making it closer to one which will help them during contingencies and reduce their liabilities as low as possible.

```
#train_data$'Curr_Ratio'   - this already available in the dataset
```

- Equity Multiplier = Total Equity / Total Assets

LEVERAGE RATIO

Used to assess the ability of a company to meet its financial obligations. Main factors considered are debt, equity, assets, and interest expenses. Common leverage ratios include the debt-equity ratio, equity multiplier, degree of financial leverage, and consumer leverage ratio.

- Gearing Ratio: It is calculated as total borrowings divided by net worth of the business
  Gearing Ratio = total borrowings/net worth

```
train_data$GearingRatio_new = (train_data$Tot_Liabilities)/(train_data$NW)
test_data$GearingRatio_new = (test_data$Tot_Liabilities)/(test_data$NW)
```

- Total Assets Turnover = Sales/Total Assets

```
train_data$TotAssetsTurnover_new = (train_data$Sales)/(train_data$TotAssests)
test_data$TotAssetsTurnover_new = (test_data$Sales)/(test_data$TotAssests)
```

- Debt-to-Equity (D/E) Ratio = Total Shareholders' Equity / Total Liabilities

```
#train_data$'Debt-to-Equity_Ratio'   - this already available in the dataset
```

- PAT as % of net worth (Return on Equity Ratio) = Profit after Tax (PAT) ÷ Net worth (NW)

```
#train_data$PAT_NW - this is already there in the dataset
```

- Current ratio = This is the ratio of Current assets and current liabilities. Current ratio (times)

```
# train_data$'Curr_Ratio'   - this already available in the dataset
```

ACTIVITY (OR TURNOVER) RATIOS

Gearing Ratio: It is calculated as total borrowings divided by net worth of the business
Gearing Ratio = total borrowings/net worth

```
train_data$'GearingRatio_new' = (train_data$Borrowings)/(train_data$NW)
```

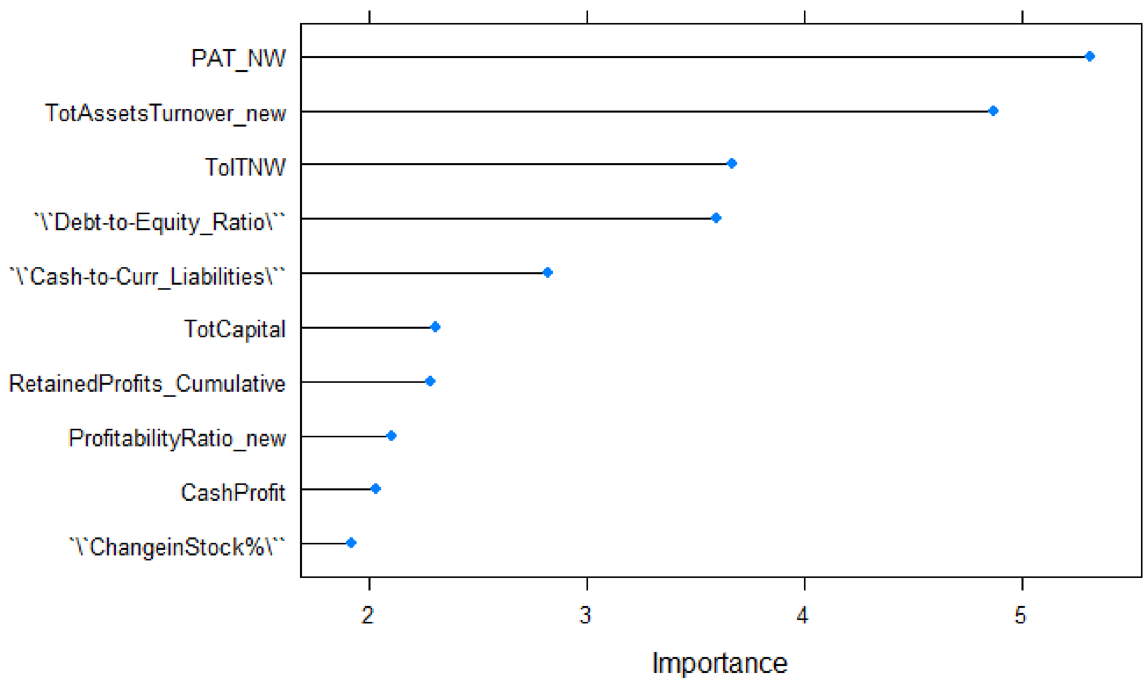
Debtors Turnover Ratio: indicates the number of times your debtors pay you over a year; can be used to determine if a company is having difficulties collecting sales made on credit. Low debtors turnover ratio implies inefficient management of debtors or less liquid debtors.

#Creditors Turnover Ratio: A high creditors turnover ratio signifies that the creditors are being paid promptly. This shows that your business is highly creditworthy.

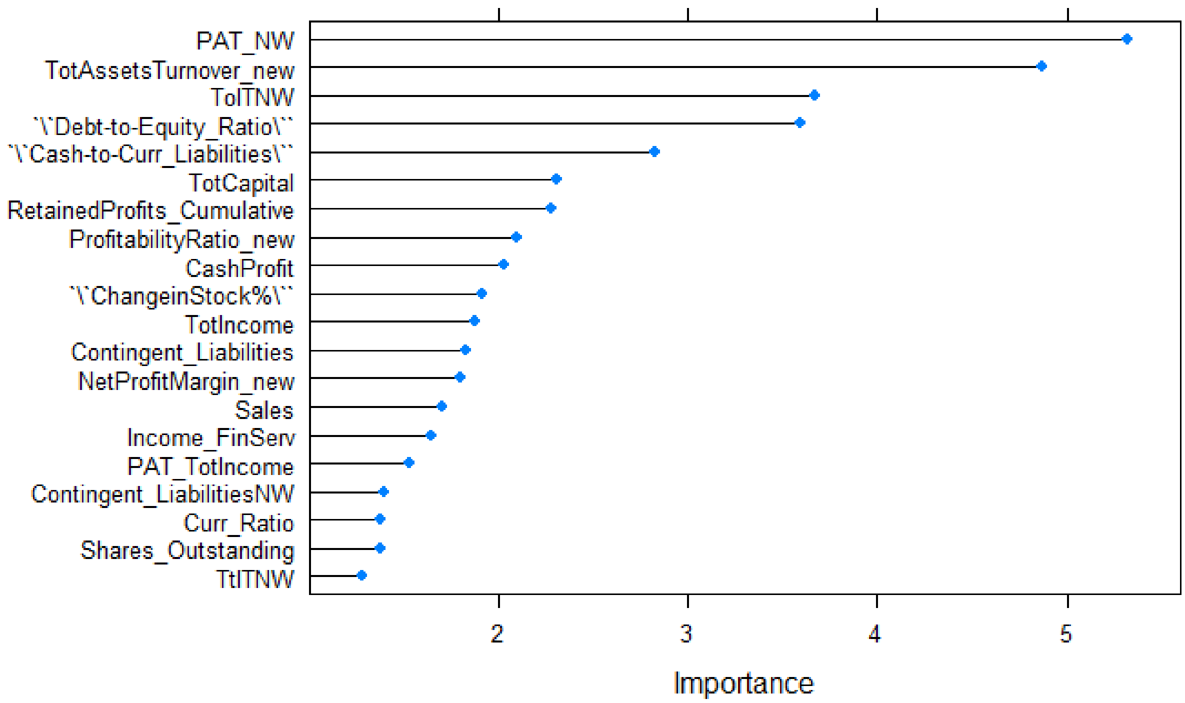
IDENTIFY HIGHLY CORRELATED & IMPORTANT VARIABLES

```
# ensure results are repeatable
set.seed(1234)
# load the library
library(mlbench)
library(caret)
# prepare training scheme
control <- trainControl(method="repeatedcv", number=10, repeats=3)
# train the model
model <- train(as.factor(NN)~., data=train_data[,-c(53)], method="glm",
preProcess="scale", trControl=control)
# estimate variable importance
importance <- varImp(model, scale=FALSE)
# summarize importance
print(importance)
# plot importance
plot(importance)
```

10 most important variables shown (out of 50)



20 most important variables shown (out of 50)

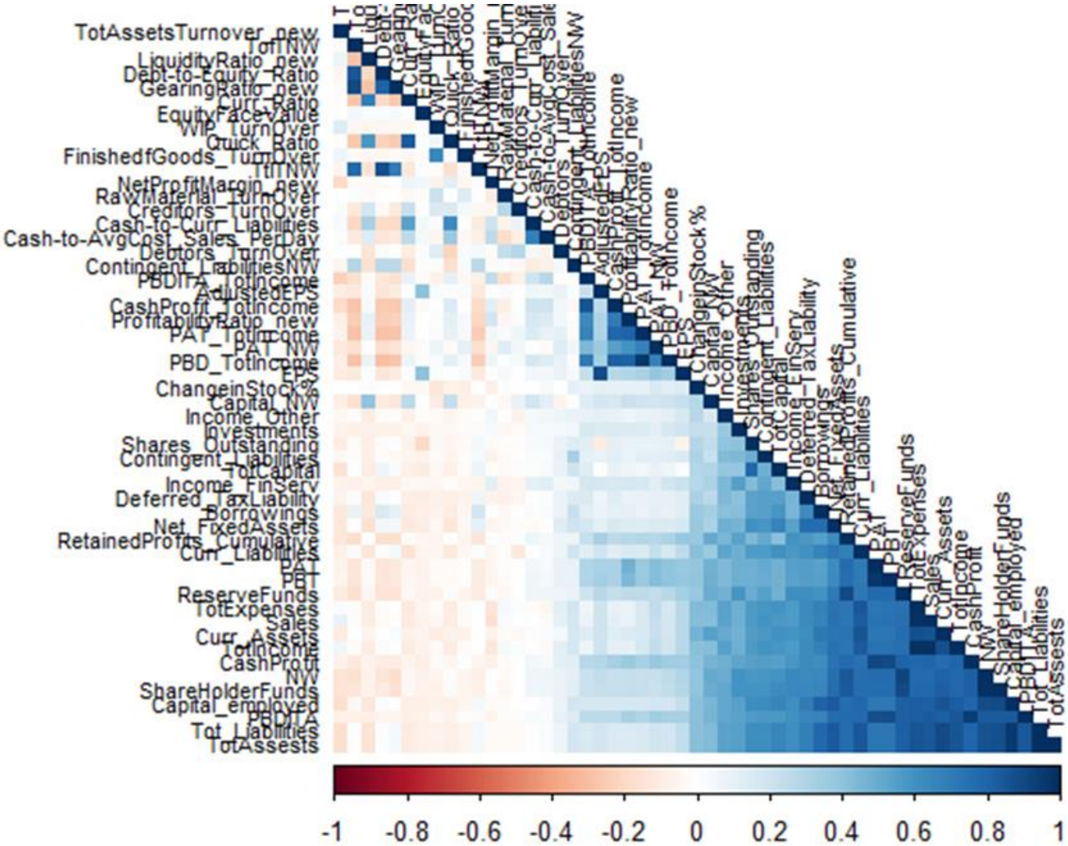



```
# Following variables are identified as most important

"TotAssetsTurnover_new", "PAT_NW", "Debt-to-Equity_Ratio", "TolTNW", "Cash-to-Curr_Liabi
lities", "TotCapital",""RetainedProfits_Cumulative","ProfitabilityRatio_new","CashProfit
","ChangeinStock%", "Totincome","Contingent_Liabilities", "NetProfitMargin_new","Sales",
"Income_FinServ", "PAT_totincome",",""Contingent_LiabilitiesNW", "Curr_Ratio", "Shares_Ou
tstanding","TtlNW"
```

- Check for Multi-collinearity

```
#checking for Multi-collinearity

numeric.var <- sapply(train_data, is.numeric)
corr.matrix <- cor(train_data[,numeric.var])

corrplot(corr.matrix, order = "FPC", method = "color", type = "lower", tl.cex = 0.7, tl.
col = rgb(0, 0, 0))
corrplot(corr.matrix, order = "FPC", method =  "number", type = "lower", tl.cex = 0.7, t
l.col = rgb(0, 0, 0))
highlyCorrelated <- caret::findCorrelation(cor(train_data[,numeric.var]),cutoff = 0.7,na
mes = T, verbose = T)
highlyCorrelated
```

Corrplot for train dataset and finding highly correlated variables (before reducing variables in train dataset)



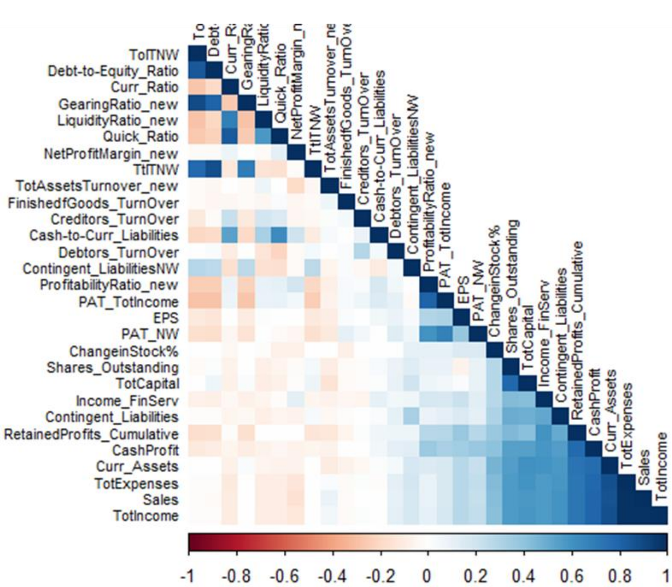

```
highlyCorrelated <- caret::findCorrelation(cor(train_data[,numeric.var]),cutoff = 0.7,na
mes = T, verbose = T)

> highlyCorrelated
# [1] "PBDITA"          "NW"                 "ShareHolderFunds"     "CashProfit"
# [5] "TotAssests"      "Tot_Liabilities"    "Capital_employed"     "ReserveFunds"
# [9] "PBT"             "PAT"                "Curr_Assets"          "TotIncome"
#[13] "TotExpenses"     "Sales"              "Curr_Liabilities"     "Net_FixedAssets"
#[17] "TotCapital"   "PBD_TotIncome"      "PAT_TotIncome"        "CashProfit_TotIncome"
#[21] "EPS"             "Debt-to-Equity_Ratio" "EquityMultiplier_new" "TolTNW"
#[25] "Quick_Ratio"
```

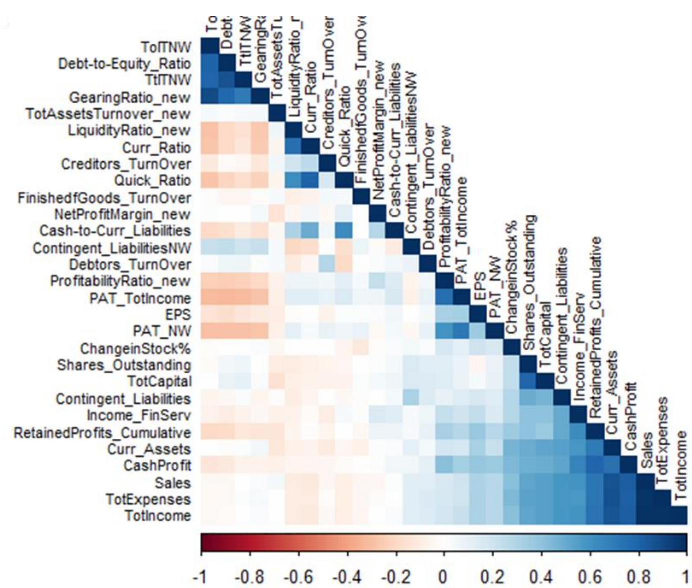
- Reducing certain columns in train and test dataset to include important variables and exclude variables that are highly correlated in test data

```
reduced_train_data <-train_data[,c("NN", "TotAssetsTurnover_new", “PAT_NW", "Debt-to-Equ
ity_Ratio", "Cash-to-Curr_Liabilities", "TolTNW", "TotCapital","RetainedProfits_Cumulati
ve", "ProfitabilityRatio_new","CashProfit", "ChangeinStock%", "TotIncome", "Contingent_L
iabilities",  "NetProfitMargin_new","Sales", "Income_FinServ", "PAT_TotIncome","Continge
nt_LiabilitiesNW", "Curr_Ratio", "Shares_Outstanding",","TtlTNW","TotExpenses", "Curr_As
sets", "Quick_Ratio", "Debtors_TurnOver","Creditors_TurnOver", "FinishedfGoods_TurnOver"
, "LiquidityRatio_new", "GearingRatio_new")]


> dim(reduced_train_data)
[1] 3541   30

reduced_test_data <-test_data[,c("NN","TotAssetsTurnover_new","PAT_NW","Debt-to-Equity_R
atio","Cash-to-Curr_Liabilities", "TolTNW","TotCapital","RetainedProfits_Cumulative","Pr
ofitabilityRatio_new","CashProfit","ChangeinStock%","TotIncome", "Contingent_Liabilities
", "NetProfitMargin_new","Sales","Income_FinServ","PAT_TotIncome","Contingent_Liabilitie
sNW","Curr_Ratio", "Shares_Outstanding","TtlTNW","TotExpenses", "Curr_Assets", "Quick_Ra
tio","EPS", "Debtors_TurnOver","Creditors_TurnOver", "FinishedfGoods_TurnOver", "Liquidi
tyRatio_new","GearingRatio_new")]

> dim(reduced_test_data)
[1] 715   30
```

Correlation plot on reduced train and test dataset –

Positive correlations are displayed in blue and negative correlations in red color. Color intensity and the size of the circle are proportional to the correlation coefficients.





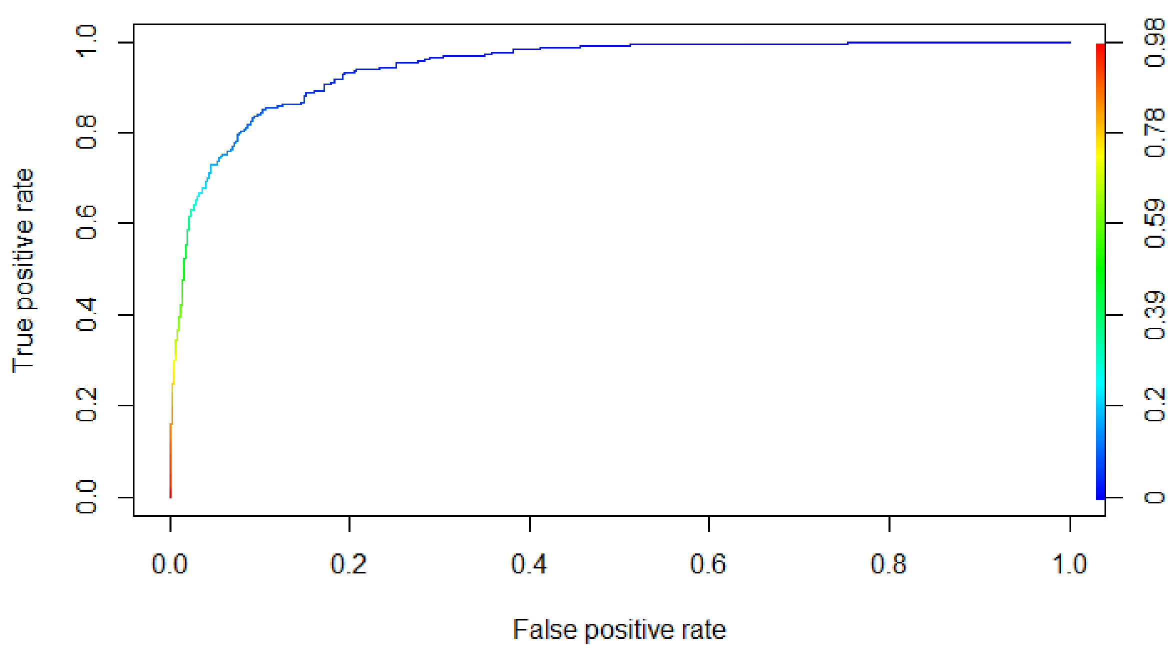
MODELS

MODEL #1

```
#MODEL1
glm.model1 = glm(NN~ . , reduced_train_data , family = binomial(link = 'logit'))
tidy(glm.model1)
summary(glm.model1)
vif(glm.model1)
```

```
> summary(glm.model1)

Call:
glm(formula = NN ~ ., family = binomial(link = "logit"), data =
reduced_train_data)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.3276  -0.2137  -0.1209  -0.0266   3.6678

Coefficients:
                             Estimate Std. Error z value Pr(>|z|)
(Intercept)                -7.308e+00  1.741e+00  -4.197 2.70e-05 ***
TotAssetsTurnover_new       4.263e-02  7.828e-03   5.446 5.15e-08 ***
PAT_NW                     -4.101e-02  7.353e-03  -5.578 2.44e-08 ***
`Debt-to-Equity_Ratio`      1.712e-01  7.660e-02   2.235 0.025446 *
`Cash-to-Curr_Liabilities`  9.955e-01  3.211e-01   3.101 0.001931 **
TolTNW                      1.849e-02  5.797e-02   0.319 0.749814
TotCapital                 -2.705e-03  1.043e-03  -2.595 0.009467 **
RetainedProfits_Cumulative -2.468e-03  1.086e-03  -2.274 0.022995 *
ProfitabilityRatio_new      7.391e+00  2.299e+00   3.215 0.001304 **
CashProfit                 -2.679e-03  1.514e-03  -1.770 0.076695 .
`ChangeinStock%`            5.007e-03  2.465e-03   2.031 0.042240 *
TotIncome                   5.094e-04  2.067e-04   2.465 0.013717 *
Contingent_Liabilities     -4.843e-04  3.395e-04  -1.426 0.153725
NetProfitMargin_new         3.268e+00  1.623e+00   2.013 0.044086 *
Sales                      -4.214e-04  1.842e-04  -2.288 0.022126 *
Income_FinServ              6.686e-03  3.701e-03   1.807 0.070836 .
PAT_TotIncome              -9.147e-02  1.774e-02  -5.155 2.53e-07 ***
Contingent_LiabilitiesNW    1.995e-03  1.789e-03   1.115 0.264700
Curr_Ratio                 -2.502e-01  1.916e-01  -1.306 0.191450
Shares_Outstanding          1.254e-08  7.841e-09   1.599 0.109714
TtlTNW                     -7.723e-02  1.097e-01  -0.704 0.481404
TotExpenses                -1.500e-04  1.950e-04  -0.769 0.441614
Curr_Assets                -1.852e-04  2.430e-04  -0.762 0.445874
Quick_Ratio                -3.476e-01  2.694e-01  -1.290 0.196964
EPS                        -2.958e-02  1.564e-02  -1.891 0.058612 .
Debtors_TurnOver           -4.989e-03  8.769e-03  -0.569 0.569416
Creditors_TurnOver          1.368e-03  9.755e-03   0.140 0.888492
```

```
FinishedfGoods_TurnOver      1.199e-04  2.015e-03   0.059 0.952565
LiquidityRatio_new           3.075e-01  6.632e-01   0.464 0.642945
GearingRatio_new             1.600e-01  4.242e-02   3.773 0.000162 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1723.67  on 3540  degrees of freedom
Residual deviance:  884.56  on 3511  degrees of freedom
AIC: 944.56

Number of Fisher Scoring iterations: 10
#Confusion matrix
pred.glm.model1 <- predict(glm.model1, newdata = reduced_train_data, type =
"response")
cm1= table(ActualValue=reduced_train_data$NN, PredictedValue=pred.glm.model1>0.5)
print("Confusion Matrix for Logistic Regression")
calc(cm1)

"Confusion Matrix for Logistic Regression"
[1] "Accuracy :- 95.1708556904829"
[1] "FNR :- 52.991452991453"
[1] "FPR :- 1.42122769882068"
[1] "precision :-  70.0636942675159"
[1] "recall//TPR :-  70.0636942675159"
[1] "Sensitivity :-  47.008547008547"
[1] "Specificity :-  98.5787723011793"

roc.pred<- prediction(pred.glm.model1, reduced_train_data$NN)
roc.perf1 = performance(roc.pred, measure = "tpr", x.measure = "fpr")
plot(roc.perf1, colorize = TRUE, text.adj = c(-0.2,1.7))
```

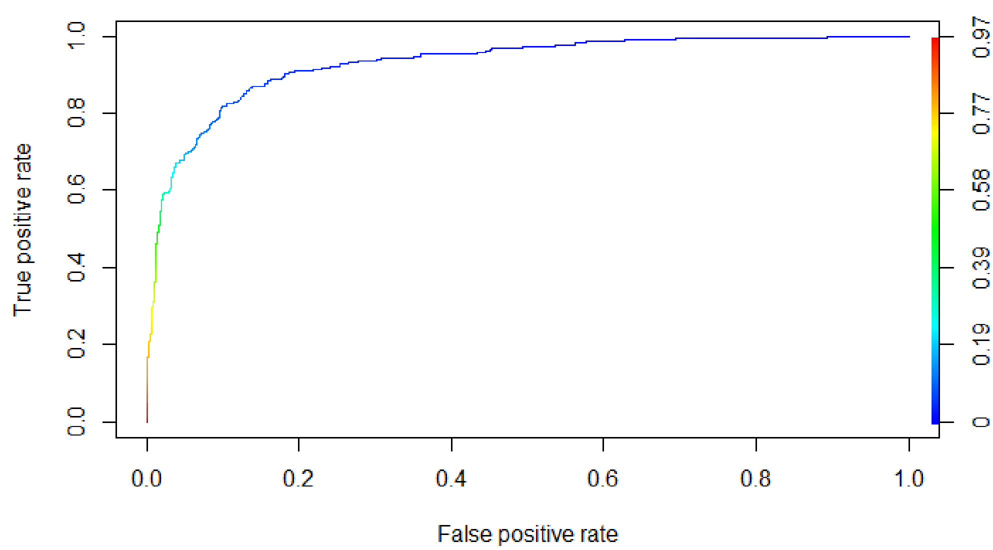


```
#AUC should be more than 0.7 in both the training and validation samples. Should
not be a significant difference between AUC score of both these samples. If it is
more than 0.8, it is considered as an excellent score.

auc.perf1=as.numeric(performance(roc.pred,"auc")@y.values)
print(paste('Area Under the Curve for test Dataset:',auc.perf1))

[1] "Area Under the Curve for test Dataset: 0.945480190944363"


#KS - KS Test measures to check whether model is able to separate events and non-
events. In probability of default (bank defaulters) model, it checks whether the
credit risk model is able to distinguish between good and bad customers.
# Ideally, max KS value should be in first three deciles and score lies between
40 and 70. And there should not be more than 10 points (in absolute) difference
between training and validation KS score. Score above 70 is susceptible and might
be overfitting so rigorous validation is required.

KS2 <-max(attr(roc.perf2, 'y.values')[[1]]-attr(roc.perf2, 'x.values')[[1]])
print(paste('K-S Value for test Dataset',KS2))
[1] "K-S Value for test Dataset 0.748259971725348"
```

MODEL #2

```
glm.model2 <- glm(NN~PBDITA_TotIncome + PAT_NW + Curr_Ratio + `Debt-to-
Equity_Ratio`+`Cash-to-Curr_Liabilities`+TolTNW
+TotCapital+`ChangeinStock%`+Investments+Contingent_Liabilities+EPS+GearingRatio_
new, family = "binomial", data = reduced_train_data)
tidy(glm.model2)
summary(glm.model2)
vif(glm.model2)

Call:
glm(formula = NN ~ PBDITA_TotIncome + PAT_NW + Curr_Ratio + `Debt-to-
Equity_Ratio` + `Cash-to-Curr_Liabilities` + Contingent_Liabilities + EPS +
    GearingRatio_new + TotAssetsTurnover_new + Income_FinServ +
    Debtors_TurnOver, family = "binomial", data = reduced_train_data)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.0290  -0.2311  -0.1433  -0.0586   4.0933

Coefficients:
                             Estimate Std. Error z value Pr(>|z|)
(Intercept)                -3.4658772  0.2573488 -13.468  < 2e-16 ***
PBDITA_TotIncome           -0.0187670  0.0105495  -1.779  0.07525 .
PAT_NW                     -0.0590338  0.0067691  -8.721  < 2e-16 ***
Curr_Ratio                 -0.4328969  0.1112521  -3.891 9.98e-05 ***
`Debt-to-Equity_Ratio`      0.1468118  0.0542288   2.707  0.00678 **
`Cash-to-Curr_Liabilities`  1.0661069  0.2708473   3.936 8.28e-05 ***
Contingent_Liabilities     -0.0006671  0.0002756  -2.421  0.01549 *
EPS                        -0.0476092  0.0207381  -2.296  0.02169 *
GearingRatio_new            0.1687049  0.0275830   6.116 9.58e-10 ***
TotAssetsTurnover_new       0.0336815  0.0058579   5.750 8.94e-09 ***
```

```
Income_FinServ              0.0022443  0.0029835   0.752  0.45192
Debtors_TurnOver           -0.0088114  0.0076085  -1.158  0.24683
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1723.53  on 3539  degrees of freedom
Residual deviance:  949.22  on 3528  degrees of freedom
AIC: 973.22

Number of Fisher Scoring iterations: 9


#Confusion matrix
pred.glm.model2 <- predict(glm.model2, newdata = reduced_train_data, type =
"response")
cm2= table(ActualValue=reduced_train_data$NN, PredictedValue=pred.glm.model2>0.5)
print("Confusion Matrix for Logistic Regression")
calc(cm2)

[1] "Confusion Matrix for Logistic Regression"

> calc(cm2)
[1] "Accuracy :- 95.3107344632768"
[1] "FNR :- 51.2820512820513"
[1] "FPR :- 1.39140955837871"
[1] "precision :-  71.25"
[1] "recall//TPR :-  71.25"
[1] "Sensitivity :-  48.7179487179487"
[1] "Specificity :-  98.6085904416213"

roc.pred<- prediction(pred.glm.model2, reduced_train_data$NN)
roc.perf2 = performance(roc.pred, measure = "tpr", x.measure = "fpr")
plot(roc.perf2, colorize = TRUE, text.adj = c(-0.2,1.7))
```

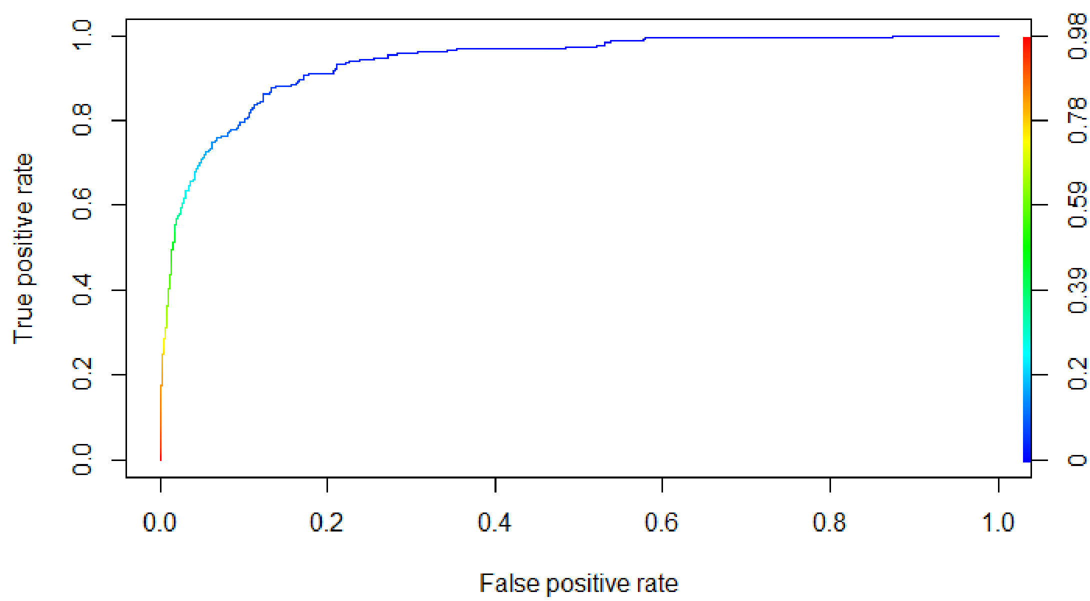


```
auc.perf2=as.numeric(performance(roc.pred,"auc")@y.values)
print(paste('Area Under the Curve for test Dataset:',auc.perf2))
[1] "Area Under the Curve for test Dataset: 0.936067368105491"


#KS

KS2 <-max(attr(roc.perf2, 'y.values')[[1]]-attr(roc.perf2, 'x.values')[[1]])
print(paste('K-S Value for test Dataset',KS2))
[1] "K-S Value for test Dataset 0.733561356973335"
```

MODEL #3

```
glm.model3 <- glm(NN~TotAssetsTurnover_new + ProfitabilityRatio_new +
GearingRatio_new +LiquidityRatio_new + ProfitabilityRatio_new +
NetProfitMargin_new + PAT_NW + PAT_TotIncome + `Debt-to-Equity_Ratio`+`Cash-to-
Curr_Liabilities`+ Curr_Ratio + NW +EPS +TolTNW, family = "binomial", data =
reduced_train_data)

tidy(glm.model3)
summary(glm.model3)

Call:
glm(formula = NN ~ TotAssetsTurnover_new + ProfitabilityRatio_new +
    GearingRatio_new + LiquidityRatio_new + ProfitabilityRatio_new +
    NetProfitMargin_new + PAT_NW + PAT_TotIncome + `Debt-to-Equity_Ratio` +
    `Cash-to-Curr_Liabilities` + Curr_Ratio + NW + EPS + TolTNW,
    family = "binomial", data = reduced_train_data)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.4015  -0.2155  -0.1322  -0.0579   3.8182

Coefficients:
                             Estimate Std. Error z value Pr(>|z|)
(Intercept)                -8.2085681  1.6343817  -5.022 5.10e-07 ***
TotAssetsTurnover_new       0.0406946  0.0065718   6.192 5.93e-10 ***
ProfitabilityRatio_new      4.3989528  2.0726316   2.122  0.03380 *
GearingRatio_new            0.1338862  0.0386968   3.460  0.00054 ***
LiquidityRatio_new         -0.1519352  0.6282923  -0.242  0.80892
NetProfitMargin_new         4.0921732  1.5315983   2.672  0.00754 **
PAT_NW                     -0.0434740  0.0073404  -5.923 3.17e-09 ***
PAT_TotIncome              -0.0746153  0.0156534  -4.767 1.87e-06 ***
`Debt-to-Equity_Ratio`      0.1255379  0.0581850   2.158  0.03096 *
`Cash-to-Curr_Liabilities`  0.7977170  0.2865573   2.784  0.00537 **
Curr_Ratio                 -0.3612989  0.1409540  -2.563  0.01037 *
NW                         -0.0005214  0.0002348  -2.220  0.02639 *
EPS                        -0.0339051  0.0159695  -2.123  0.03374 *
TolTNW                      0.0489526  0.0496474   0.986  0.32413
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)
```

```
    Null deviance: 1723.67  on 3540  degrees of freedom
Residual deviance:  920.69  on 3527  degrees of freedom
AIC: 948.69

Number of Fisher Scoring iterations: 9

#Confusion matrix
pred.glm.model3 <- predict(glm.model3, newdata = reduced_train_data, type =
"response")
cm3=table(ActualValue=reduced_train_data$NN, PredictedValue=pred.glm.model3>0.5)
print("Confusion Matrix for Logistic Regression")
calc(cm3)

"Confusion Matrix for Logistic Regression"
[1] "Accuracy :- 95.340299350466"
[1] "FNR :- 50.4273504273504"
[1] "FPR :- 1.42122769882068"
[1] "precision :-  71.1656441717791"
[1] "recall//TPR :-  71.1656441717791"
[1] "Sensitivity :-  49.5726495726496"
[1] "Specificity :-  98.5787723011793"

roc.pred<- prediction(pred.glm.model2, reduced_train_data$NN)
roc.perf2 = performance(roc.pred, measure = "tpr", x.measure = "fpr")
plot(roc.perf2, colorize = TRUE, text.adj = c(-0.2,1.7))
```




```
auc.perf2=as.numeric(performance(roc.pred,"auc")@y.values)
print(paste('Area Under the Curve for test Dataset:',auc.perf2))
[1] "Area Under the Curve for test Dataset: 0.937792535388556"

#KS
KS3 <-max(attr(roc.perf3, 'y.values')[[1]]-attr(roc.perf3, 'x.values')[[1]])
print(paste('K-S Value for test Dataset',KS3))
[1] "K-S Value for test Dataset 0.742754943541155"
```

#MODEL VALIDATION ON TEST DATA

```
PredictTest = predict(glm.model1, newdata=reduced_test_data,type="response")
summary(PredictTest)
cm4=table(ActualValue=reduced_test_data$NN, PredictedValue=PredictTest>0.5)
calc(cm4)

[1] "Accuracy :- 95.9440559440559"
[1] "FNR :- 22.2222222222222"
[1] "FPR :- 2.57186081694402"
[1] "precision :-  71.1864406779661"
[1] "recall//TPR :-  71.1864406779661"
[1] "Sensitivity :-  77.7777777777778"
[1] "Specificity :-  97.428139183056"
```

MODEL COMPARISON USING - CONFUSION MATRIX INTERPRETATION FOR ALL MODELS

```
library(performance)
compare_performance(glm.model1, glm.model2, glm.model3, rank = TRUE)
```

```
Model      | Type |    AIC |     BIC | R2_Tjur | RMSE | LOGLOSS | SCORE_LOG | SCORE_SPHERICAL |  PCP | Performance_Score
------------------------------------------------------------------------------------------------------------------------
glm.model1 |  glm | 944.56 | 1129.72 |    0.43 | 0.50 |    0.12 |      -Inf |            0.01 | 0.93 |            71.43%
glm.model3 |  glm | 948.69 | 1035.10 |    0.42 | 0.51 |    0.13 |      -Inf |            0.01 | 0.93 |            60.54%
glm.model2 |  glm | 973.22 | 1047.28 |    0.40 | 0.52 |    0.13 |      -Inf |            0.01 | 0.93 |            26.73%

Model glm.model1 (of class glm) performed best with an overall performance score of 71.43%.
```

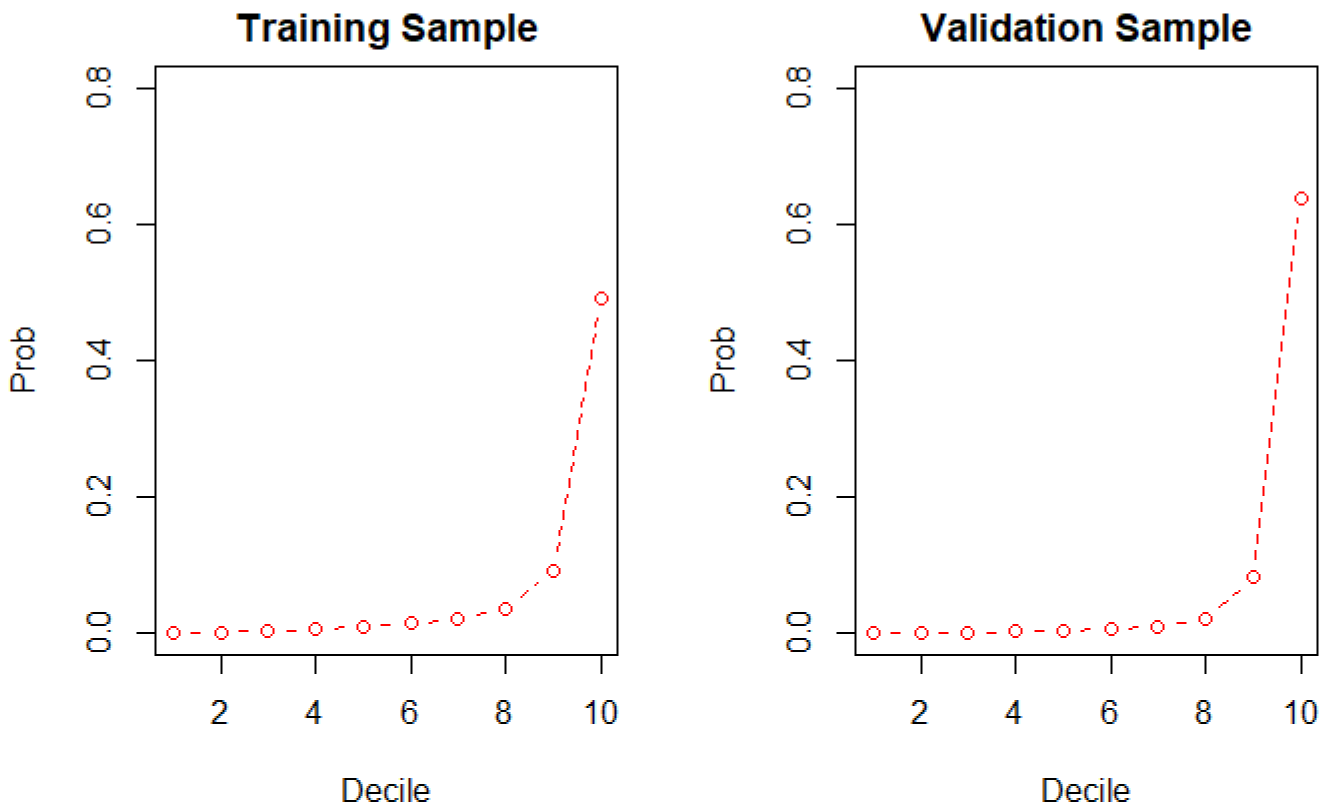
CONFUSION MATRIX	LOGISTIC REGRESSION (Model 1)	LOGISTIC REGRESSION (Model 2)	LOGISTIC REGRESSION (Model 2)
Accuracy	95.2%	95.31%	95.34%
FNR	52.2%	51.3%	50.42%
FPR	1.36%	1.39%	1.42%
PRECISION	71.3%	71.3%	71.1%
RECALL (TPR)	71.3%	71.3%	71.1%
SENSITIVITY (TNR)	47.8%	48.7%	49.6%
SPECIFICITY	98.6%	98.6%	98.6%
KS	0.735	0.733	0.742
AUC	0.93809	0.93044	0.937

Logistic regression Models gives variables such as TotAssetsTurnover_new, ProfitabilityRatio_new, GearingRatio_new, LiquidityRatio_new, NetProfitMargin_new, PAT_NW, PAT_TotIncome, `Debt-to-Equity_Ratio`, `Cash-to-Curr_Liabilities` and Curr_Ratio are most significant variables for predicting companies that are likely to default. Accuracy and Specificity are good in all the models.

RANK ORDERING

```
library(data.table)
library(scales)
# Rank Ordering
decile <- function(x)
{
 deciles <- vector(length=10)
 for (i in seq(0.1,1,.1))
 {
 deciles[i*10] <- quantile(x, i, na.rm=T)
 }
 return (
 ifelse(x<deciles[1], 1,
 ifelse(x<deciles[2], 2,
 ifelse(x<deciles[3], 3,
 ifelse(x<deciles[4], 4,
 ifelse(x<deciles[5], 5,
 ifelse(x<deciles[6], 6,
 ifelse(x<deciles[7], 7,
 ifelse(x<deciles[8], 8,
 ifelse(x<deciles[9], 9, 10
 ))))))))))
}

#calculate deciles for train and test data
reduced_train_data$deciles <- decile(pred.glm.model1)
tmp_DT1 = data.table(reduced_train_data)
reduced_test_data$deciles <- decile(pred.glm.model4)
tmp_DT2 = data.table(reduced_test_data)


# After the deciles are created, they are ranked.
rank1 <- tmp_DT1[, list(cnt=length(NN),
 cnt_resp=sum(NN==1),
 cnt_non_resp=sum(NN==0)
), by=deciles][order(-deciles)]

rank1$rrate <- round(rank1$cnt_resp / rank1$cnt,4);
rank1$cum_resp <- cumsum(rank1$cnt_resp)
rank1$cum_non_resp <- cumsum(rank1$cnt_non_resp)
rank1$cum_rel_resp <- round(rank1$cum_resp / sum(rank1$cnt_resp),4);
rank1$cum_rel_non_resp <- round(rank1$cum_non_resp / sum(rank1$cnt_non_resp),4);
rank1$ks <- abs(rank1$cum_rel_resp - rank1$cum_rel_non_resp) * 100;
rank1$rrate <- percent(rank1$rrate)
rank1$cum_rel_resp <- percent(rank1$cum_rel_resp)
rank1$cum_rel_non_resp <- percent(rank1$cum_rel_non_resp)
newtrainRank <- rank1

rank2 <- tmp_DT2[, list(cnt=length(NN),
 cnt_resp=sum(NN==1),
 cnt_non_resp=sum(NN==0)
), by=deciles][order(-deciles)]
```

```
rank2$rrate <- round(rank2$cnt_resp / rank2$cnt,4);
rank2$cum_resp <- cumsum(rank2$cnt_resp)
rank2$cum_non_resp <- cumsum(rank2$cnt_non_resp)
rank2$cum_rel_resp <- round(rank2$cum_resp / sum(rank2$cnt_resp),4);
rank2$cum_rel_non_resp <- round(rank2$cum_non_resp / sum(rank2$cnt_non_resp),4);
rank2$ks <- abs(rank2$cum_rel_resp - rank2$cum_rel_non_resp) * 100;
rank2$rrate <- percent(rank2$rrate)
rank2$cum_rel_resp <- percent(rank2$cum_rel_resp)
rank2$cum_rel_non_resp <- percent(rank2$cum_rel_non_resp)
newtestRank <- rank2

# Decile Comparison
# cut_p returns the cut internal for each observation
cut_ptrain = with(newtrainRank,
cut(pred.glm.model1, breaks = quantile(pred.glm.model1, prob=seq(0,1,0.1)),
include.lowest = T))
cut_ptest = with(newtestRank,
cut(pred.glm.model4, breaks = quantile(pred.glm.model4, prob=seq(0,1,0.1)),
include.lowest = T))
levels(cut_ptrain)
levels(cut_ptest)
reduced_train_data$rank1 = factor(cut_ptrain, labels = 1:10)
reduced_test_data$rank2 = factor(cut_ptest, labels = 1:10)

#Get aggregated data

mean.obs.train = aggregate(NN ~ reduced_train_data$rank1, data = reduced_train_data,
mean)
mean.pred.train = aggregate(pred.glm.model1 ~ reduced_train_data$rank1, data =
reduced_train_data, mean)
mean.obs.val = aggregate( NN ~ reduced_test_data$rank2, data = reduced_test_data,
mean)
mean.pred.val = aggregate(pred.glm.model4 ~ reduced_test_data$rank2, data =
reduced_test_data, mean)

mean.obs.train = aggregate(NN ~ newtrainRank$rank1, data = newtrainRank, mean)
mean.pred.train = aggregate(pred.glm.model1 ~ newtrainRank$rank1, data =
newtrainRank, mean)
mean.obs.val = aggregate( NN ~ newtestRank$rank2, data = newtestRank, mean)
mean.pred.val = aggregate(pred.glm.model4 ~ newtestRank$rank2, data = newtestRank,
mean)

# plot the mean vs deciles
par(mfrow=c(1,2))

plot(mean.obs.train[,2], type="b", col="black", ylim=c(0,0.8),
xlab="Decile",ylab="Prob")
lines(mean.pred.train[,2], type="b", col="red", lty=2)
title(main="Training Sample")
plot(mean.obs.val[,2], type="b", col="black", ylim=c(0,0.8), xlab="Decile",
ylab="Prob")
lines(mean.pred.val[,2], type="b", col="red", lty=2)
title(main="Validation Sample")
```


Training Sample
Prob
0.0
0.2
0.4
0.6
0.8
2
4
6
8
10
Decile
Validation Sample
Prob
0.0
0.2
0.4
0.6
0.8
2
4
6
8
10
Decile