Bank Loan Default

– *Capstone Project*

Vompolu Sai Tanuj

PGP-BABI(JUNE 2019) G1

Acknowledgement

I would like to show my gratitude and appreciation to Ashwarya Saraf, who has been at every step of our capstone project helping us in clarification of both business and statistical concepts and without whom this project would have been possible.

And my sincere thanks to the Great Lakes institute of Management for providing me with an opportunity to work on this capstone project.

I would like to thank all the faculty members who have enlightened me with their knowledge and help me understand business analytics and providing me with the right tools to work on this capstone project.

My thanks also goes to our program manager, Shruti Karki, who has been of great help in being an important link between us and management.

Table of Contents

1. Project Introduction
    a. Need for the project
    b. Defining the problem statement
    c. Business Opportunity
    d. Scope and Objectives of the project
2. Exploratory Data Analysis
    a. Data Report
    b. Uni-variate Analysis
    c. Multi-Variate Analysis
    d. Missing Value Treatment
    e. Outlier Treatment
    f. Variable Transformation
    g. Insights from EDA
3. Model Building
    a. Data Preparation
    b. Model building
        i. Logistic Regression
        ii. Naïve Bayes
        iii. KNN
        iv. CART Model
        v. Random Forest
        vi. Bagging
        vii. Adaptive Boosting
        viii. Extreme Gradient Boosting
4. Model Validation
    a. Model Measure Comparison
    b. Best model Interpretation
5. Recommendations
    a. Best model recommendations
    b. Data-Driven recommendations
    c. Business recommendations
6. Conclusion

1. Project Introduction:

The **capstone project** is about creating a **prediction model** for the **bank** to overcome their **Loan Default** problem using the data of the **customers** got from the bank.

a. Need for the project

Yes Bank has faced a **loss** of around **18 thousand** crores in the **third quarter** where the **sole** reason being the bank had to compensate for **loan defaulters.** These kind of cases are not **unique** to **Yes Bank** but also to every other bank in the economy. These **losses** could have been avoided had the banks **scrutinized** its customers properly and avoiding them from becoming victims of this problem**.** This lead to realisation of the importance of **scrutinizing** the **borrowers** before providing them with **loan.**

This requirement for **scrutinizing** the customers has brought us to the need for this **Capstone Project** to create a **prediction model** for a **bank.**

b. Defining the problem statement

Banks are the **most important financial institutions** for any economy. Their **financial** services of ***accepting deposits*** *and* ***provision of loans*** help in **uplifting** economy all while providing revenue for the bank. While latter service of **provision of loan** may be responsible for **majority** of its revenue, it is also the one which involves the **highest amount** of **risk,** in the form of **loan default**

The action of ***loan default*** *could be defined when a* ***borrower*** *accepts* ***loan*** *from the bank but* ***does not*** *pay back the* ***loan taken.*** *This definition could be extended to the cases where the* ***borrower*** *has either* ***not paid the*** *loan* ***in time*** *or* ***not paid*** *the* ***full amount*** *of* ***loan*** *or* ***both.***

A **borrower** can be termed as **non-defaulter** if he/she pays back the **loan** with **interest** on time.

c. Business Opportunity

Analysis of data and **making predictions** present **huge business opportunities** to not only **banks** but every other **small financial institutions** which provides **loans** to earn its revenues.

- Every **business** looks for **expansion** in terms of **size, investments, etc.** This **expansion** is not possible without the bank actually managing the **risk** so that it doesn't fall into any **financial burdens** which cause **hindrances** for **expansion.**

- The **crucial decisions** taken by the **management** for the **bank** require a **proper analysis** of the **current scenario** in order to take the **right decision**.
- **Earning profits** is the main objective of **any business.** A proper **analysis** of the data will help give an idea of the **company's health** on how well it is earning **profits.**
- If a **better picture** is got on the **loan provision**, then only the **capital** that will be **enough** to run the **provisions** for loan can **utilised** instead of using much more than required.
- Since the **predictions** and **analysis** tells us the position in the respect of **debts,** any **essential funds** that might be required to keep the **business** afloat can be arranged and helping the **business** from becoming **money deficient.**

d. Scope and Objectives of the project

The **scope** of this project is **determined** by the **dataset** provided to us by the **management of the bank** and the **information** required by the management to make proper decisions.

5 C's of Loan

Character	**Conditions**
Capital	**Collateral**
Capacity	

The following contribute to the **scope of the project:**

- Out of the **5 C's** of the **Lending** a loan, only **two** of them, **Conditions** and **Collateral** are under the control of the bank. But the **other three C's** solely depend on the **borrower.** Therefore, it is the duty of the bank to **properly evaluate** these factors before lending a loan.
- In our project, the **data** for the project contains these **3 C's, Character, Capacity and Capital** related to the **borrowers.** Our **analysis** and **model building** will solely be based on these **aspects** of the **loan.**
- The project does not take into **consideration** that the **borrowers** are **not existing depositors** in the bank and have come with **sole purpose** of **borrowing** loan.
- The **range** of this **capstone project** is **limited** to **performing predictive analytics** rather than **prescriptive analytics.**

Objectives of Project:

There are **two objectives** of this capstone project which are as follows:

Data Analysis

Analysing the data to gain meaningful insights and give recommendations accordingly

Predictive Modelling

Building a prediction model to churn out defaulters and help management to reduce losses

2. Exploratory Data Analysis:

Exploratory Analysis of data refers to the process of **analysing** the data by exploring the data in every way possible to get **significant** insights which might give useful information on the data. **Getting insights** on the data is not the sole purpose of the **EDA** but also to know the **variables** in the **dataset** better and also making **adjustments** to the data if any required **before** going into the **model building**

a. Data Report

The data of the customers provided to us by the bank is present in an **excel file "loan default_data.xlsx"**. The dataset contains **226786** observations and **41** variables. The **variables** in the current dataset hold the information such as **terms of their current loan, income related information, credit score of those customers, transactions done for the loan, current status of the loan, etc.** We have been provided with a **data dictionary** which holds the **information** about the **variable names** and **what they represent.**

The dataset has variables which are in the wrong data types. Every model building algorithms require the variables to be in the right data type. Therefore the variables must be converted to proper data types. The conversions are done in the following manner:

Date and Time → ***Factor***

Character type with text → ***Factor***

Character type with Numbers → ***Numerical***

No change is to be done for ***Numerical variables***

b. Uni-Variate Analysis:

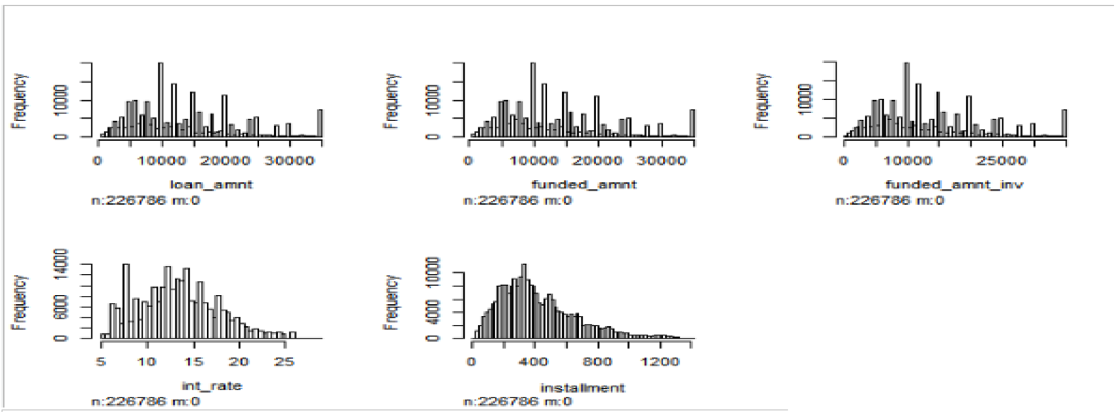
Uni-variate analysis refers to the process of **analysing** each **variable** independently to **summarize** and **finding patterns** in the data. This process can be achieved by plotting

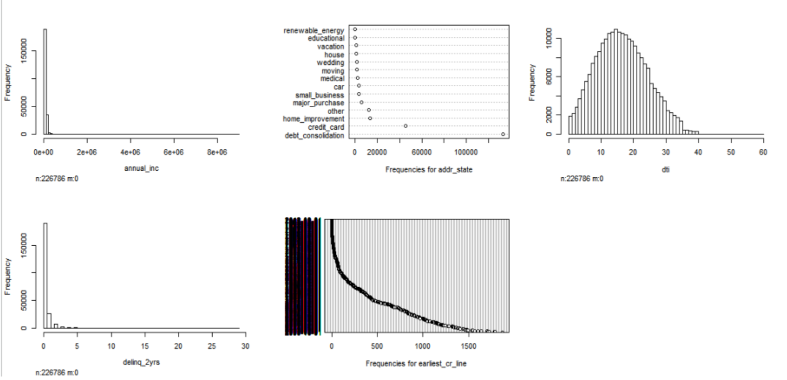
histograms for each of the **numerical variables** and creating **frequency tables** for the **categorical variables.**

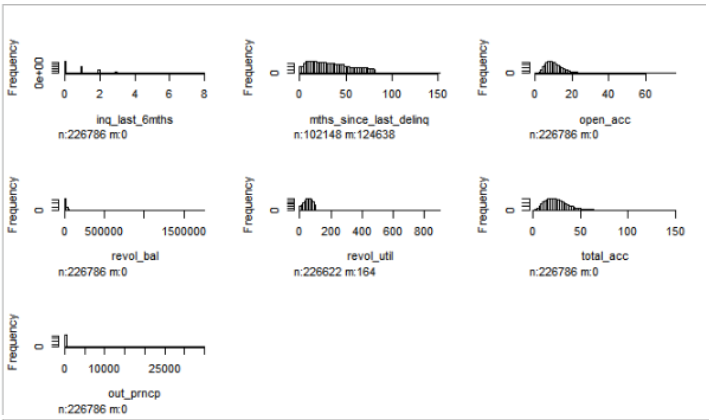
Summary statistics for all the variables:

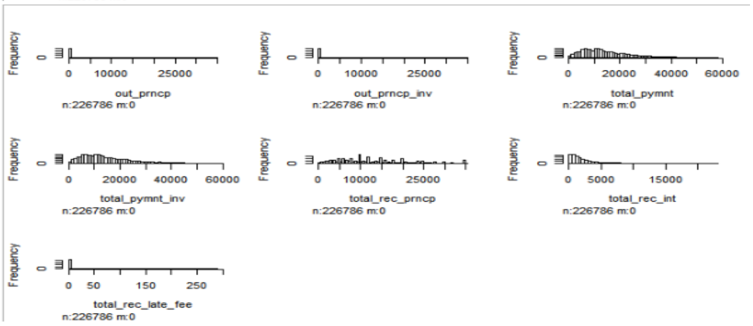
```
  loan_amnt      funded_amnt    funded_amnt_inv        term            int_rate
Min.   :  500   Min.   :  500   Min.   :    0   36 months:179291   Min.   : 5.32
1st Qu.: 7200   1st Qu.: 7200   1st Qu.: 7200   60 months: 47495   1st Qu.:10.25
Median :12000   Median :12000   Median :11975                      Median :13.11
Mean   :13543   Mean   :13507   Mean   :13427                      Mean   :13.49
3rd Qu.:18194   3rd Qu.:18000   3rd Qu.:18000                      3rd Qu.:16.29
Max.   :35000   Max.   :35000   Max.   :35000                      Max.   :28.99

  installment       grade          emp_length       home_ownership     annual_inc
Min.   :  15.69   A:40583   10+ years:69626   ANY     :     1   Min.   :   3000
1st Qu.: 239.55   B:69988   2 years  :21235   MORTGAGE:113338   1st Qu.:  45000
Median : 364.96   C:58264   < 1 year :18660   NONE    :    36   Median :  64000
Mean   : 417.99   D:34627   3 years  :18447   OTHER   :   114   Mean   :  73965
3rd Qu.: 547.43   E:15889   5 years  :15986   OWN     : 19919   3rd Qu.:  90000
Max.   :1409.99   F: 5927   1 year   :15183   RENT    : 93378   Max.   :8900060
                  G: 1508   (Other)  :67649
    verification_status        issue_d         pymnt_plan               purpose
Not Verified   :78058    2014-10-01:  8246   n:226780   debt_consolidation:132971
Source Verified:67981    2014-07-01:  7722   y:     6   credit_card       : 45729
Verified       :80747    2014-04-01:  5860              home_improvement  : 13736
                         2015-01-01:  5830              other             : 12311
                         2013-12-01:  5734              major_purchase    :  5737
                         2014-01-01:  5646              small_business    :  3663
                         (Other)   :187748              (Other)           : 12639
            addr_state          dti           delinq_2yrs        earliest_cr_line
debt_consolidation:132971   Min.   : 0.00   Min.   : 0.000   2000-10-01:  1870
credit_card       : 45729   1st Qu.:10.62   1st Qu.: 0.000   1999-10-01:  1793
home_improvement  : 13736   Median :16.03   Median : 0.000   2001-10-01:  1784
other             : 12311   Mean   :16.44   Mean   : 0.259   2000-11-01:  1725
major_purchase    :  5737   3rd Qu.:21.86   3rd Qu.: 0.000   1999-11-01:  1709
small_business    :  3663   Max.   :59.26   Max.   :29.000   2000-08-01:  1705
(Other)           : 12639                                    (Other)   :216200
 inq_last_6mths   mths_since_last_delinq     open_acc       revol_bal          revol_util
Min.   :0.0000   Min.   :  0.00          Min.   : 0.00   Min.   :      0   Min.   :  0.00
1st Qu.:0.0000   1st Qu.: 17.00          1st Qu.: 7.00   1st Qu.:   5812   1st Qu.: 35.40
Median :0.0000   Median : 32.00          Median :10.00   Median :  10868   Median : 55.00
Mean   :0.8244   Mean   : 35.04          Mean   :10.99   Mean   :  15241   Mean   : 53.67
3rd Qu.:1.0000   3rd Qu.: 51.00          3rd Qu.:14.00   3rd Qu.:  19065   3rd Qu.: 73.20
Max.   :8.0000   Max.   :151.00          Max.   :76.00   Max.   :1743266   Max.   :892.30
                 NA's   :124638                                            NA's   :164
  total_acc        out_prncp        out_prncp_inv      total_pymnt     total_pymnt_inv
Min.   :  2.00   Min.   :    0.0   Min.   :    0.0   Min.   :    0   Min.   :    0
1st Qu.: 17.00   1st Qu.:    0.0   1st Qu.:    0.0   1st Qu.: 7195   1st Qu.: 7110
Median : 24.00   Median :    0.0   Median :    0.0   Median :12290   Median :12208
Mean   : 25.22   Mean   :  982.7   Mean   :  982.3   Mean   :14455   Mean   :14358
3rd Qu.: 32.00   3rd Qu.:    0.0   3rd Qu.:    0.0   3rd Qu.:19728   3rd Qu.:19629
Max.   :150.00   Max.   :35000.0   Max.   :35000.0   Max.   :57778   Max.   :57778

total_rec_prncp total_rec_int     total_rec_late_fee   recoveries collection_recovery_fee
Min.   :    0   Min.   :    0.0   Min.   :  0.0000   Min.   :0    Min.   :0
1st Qu.: 6000   1st Qu.:  629.7   1st Qu.:  0.0000   1st Qu.:0    1st Qu.:0
Median :10500   Median : 1311.1   Median :  0.0000   Median :0    Median :0
Mean   :12503   Mean   : 1951.0   Mean   :  0.5893   Mean   :0    Mean   :0
3rd Qu.:17075   3rd Qu.: 2485.4   3rd Qu.:  0.0000   3rd Qu.:0    3rd Qu.:0
Max.   :35000   Max.   :22777.6   Max.   :286.7476   Max.   :0    Max.   :0

    last_pymnt_d     last_pymnt_amnt        next_pymnt_d       last_credit_pull_d
2015-12-01: 18050   Min.   :    0.0   2016-01-01:  1464   2016-01-01:100820
2015-10-01: 15666   1st Qu.:  732.7   2016-02-01: 17597   2015-12-01: 14828
2015-11-01: 13778   Median : 4956.3   2016-03-01:     2   2015-11-01: 10939
2015-09-01: 13175   Mean   : 7159.7   NA's      :207723   2015-10-01:  9950
2015-07-01: 11335   3rd Qu.:10931.0                       2015-09-01:  8742
(Other)   :154441   Max.   :36475.6                       (Other)   : 81491
NA's      :   341                                         NA's      :    16
  application_type      loan_status
INDIVIDUAL:226780   Default   : 19063
JOINT     :     6   Fully Paid:207723
```

Histograms for Numerical variables:









Frequency Tables for Categorical variables:

Grade

A	B	C	D	E	F	G
40583	69988	58264	34627	15889	5927	1508

Employment Length

< 1 year	1 year	10+ years	2 years
18660	15183	69626	21235
3 years	4 years	5 years	6 years
18447	14525	15986	13055
7 years	8 years	9 years	n/a
12464	10657	8548	8400

Home Ownership

ANY	MORTGAGE	NONE	OTHER
1	113338	36	114
OWN	RENT		
19919	93378		

Verification Status

Not Verified	Source Verified
78058	67981
Verified	
80747	

Application type

INDIVIDUAL	JOINT
226780	6

Date of Issue

```
2007-06-01 2007-07-01 2007-08-01 2007-09-01
         1         30         26         15
2007-10-01 2007-11-01 2007-12-01 2008-01-01
        37         30         67        140
2008-02-01 2008-03-01 2008-04-01 2008-05-01
       149        196        128         61
2008-06-01 2008-07-01 2008-08-01 2008-09-01
        59         66         65         27
2008-10-01 2008-11-01 2008-12-01 2009-01-01
        81        153        190        211
2009-02-01 2009-03-01 2009-04-01 2009-05-01
       226        245        250        277
2009-06-01 2009-07-01 2009-08-01 2009-09-01
       313        327        368        392
2009-10-01 2009-11-01 2009-12-01 2010-01-01
       458        519        536        513
2010-02-01 2010-03-01 2010-04-01 2010-05-01
       564        668        745        795
2010-06-01 2010-07-01 2010-08-01 2010-09-01
       863        982        931        912
2010-10-01 2010-11-01 2010-12-01 2011-01-01
       973        980       1118       1140
2011-02-01 2011-03-01 2011-04-01 2011-05-01
      1027       1100       1196       1198
2011-06-01 2011-07-01 2011-08-01 2011-09-01
      1446       1454       1510       1585
2011-10-01 2011-11-01 2011-12-01 2012-01-01
      1613       1702       1581       2018
2012-02-01 2012-03-01 2012-04-01 2012-05-01
      2010       2325       2577       2637
2012-06-01 2012-07-01 2012-08-01 2012-09-01
      2914       3570       4253       4821
2012-10-01 2012-11-01 2012-12-01 2013-01-01
      4986       4995       4706       4680
2013-02-01 2013-03-01 2013-04-01 2013-05-01
      3717       4089       4544       4784
2013-06-01 2013-07-01 2013-08-01 2013-09-01
      4896       5061       5279       5294
2013-10-01 2013-11-01 2013-12-01 2014-01-01
      5614       5643       5734       5646
2014-02-01 2014-03-01 2014-04-01 2014-05-01
      4964       5380       5860       5639
2014-06-01 2014-07-01 2014-08-01 2014-09-01
      4818       7722       4725       2491
2014-10-01 2014-11-01 2014-12-01 2015-01-01
      8246       4879       1861       5830
2015-02-01 2015-03-01 2015-04-01 2015-05-01
      3307       3235       3969       3283
2015-06-01 2015-07-01 2015-08-01 2015-09-01
      2504       3171       1895       1294
2015-10-01 2015-11-01 2015-12-01
      1335        759        587
```

Payment Plan

```
     n      y
226780      6
```

Purpose for loan

```
               car        credit_card debt_consolidation        educational   home_improvement
              3318              45729             132971                269              13736
             house     major_purchase            medical             moving              other
              1471               5737               2481               1747              12311
  renewable_energy     small_business           vacation            wedding
               230               3663               1422               1701
```

Collection of Recovery fee

```
     0
226786
```

Recoveries

```
     0
226786
```

Loan Status

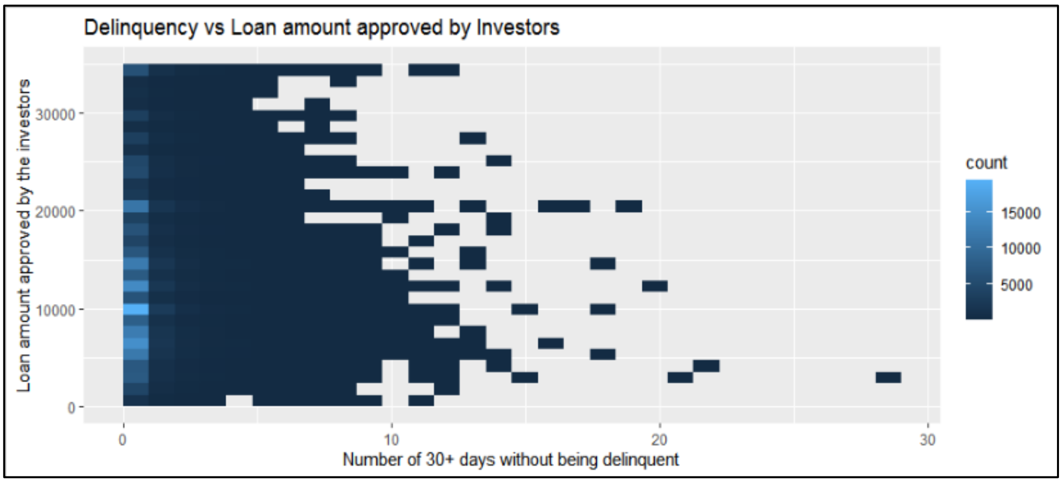
```
Default Fully Paid
  19063     207723
```

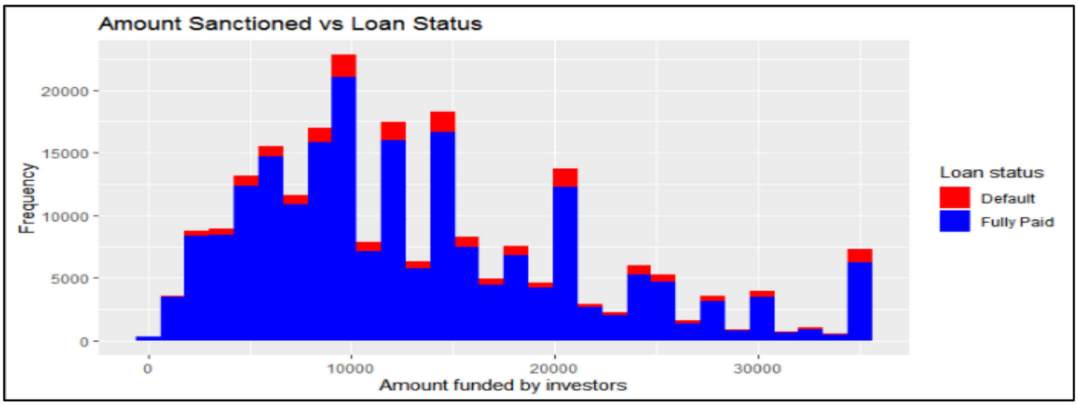
Findings from Univariate Analysis:

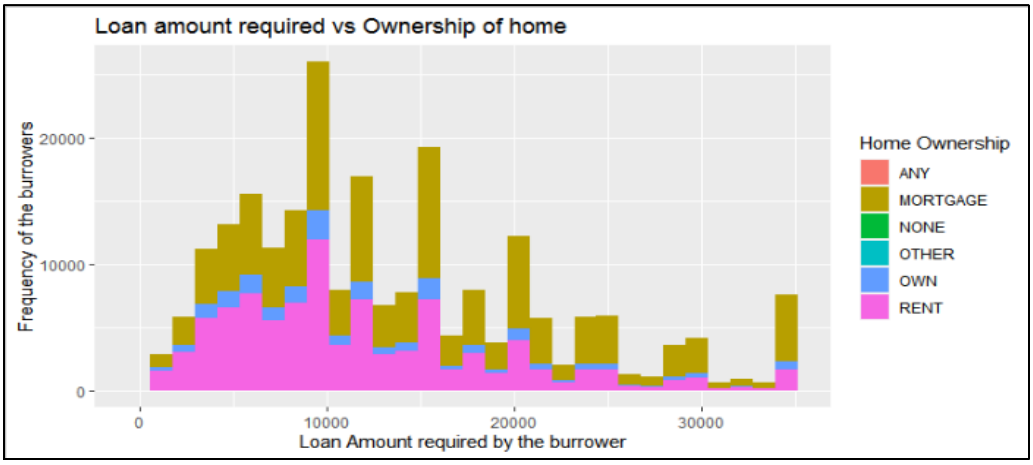
- There are **6 numerical variables** which do not follow a **bell-curve** and deviate from rest of the variables
- **Rest of the variables** are **either highly or slightly skewed** to the **left.**
- The variable **purpose for loan** has the highest number of **factors** which is **14.**
- Variables such as **recoveries, collection recovery fee** have all values as **zeros.** There are **2 variables** which have only **single factor.**
- The dataset is **highly imbalanced** when considered **w.r.t** to the **categorical variables, Payment plan and Loan Status.**

c. Multi-Variate Analysis:

This type of **analysis** will help us give **insights** on **multiple** variables at once and also will help us understand the **relationships** between the **variables** which is very important **model building process. Multi-variate analysis** can be performed by using **correlation plots** and **colored graphs** like **boxplots, bar plots,** etc.





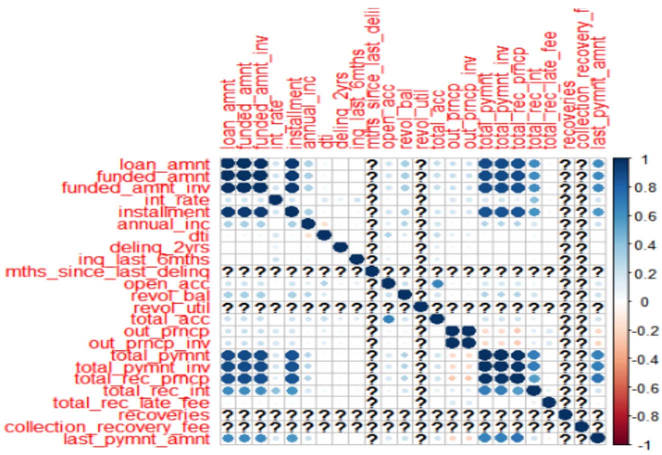


Loan Status vs Verification Status Frequency Table

	Not Verified	Source Verified	Verified
Default	4202	7710	7151
Fully Paid	73856	60271	73596

Correlation Plot:

Correlation between two variables can be defined as the **strength** of **relationship** that exists between two or more **numerical variables.** Below is the **correlation plot** showing the **correlation** values **between** each of these variables.



Findings from Multi-Variate Analysis:

- The **Burrowers** who have **lesser** number of **days** without being **delinquent** have been sanctioned **higher amounts** of loan than the ones the **burrowers** who had **more** number of **days** without being **delinquent.**
- For a given loan, the **rate of interest rises** with respect to **rise in the per month instalment amount.**
- Burrowers asking for higher amount of loans are the ones having either **mortgages** or **living in rent houses.**

- Majority of the **burrowers** have paid back the loan **irrespective** of the **loan amount** sanctioned by the **investors.**
- The **burrowers** who were either **source verified** or **verified** by the **bank** account to **majority** of the **burrowers** who have **fully paid** their loan.
- There are some **variables** for which the **correlation** values could not be found because those variables have **no standard deviation** and hence cannot be **related** with any of the other variables.

d. Missing Value Treatment:

The **missing values** can be termed as the values that are unknown to the analyst when he gets the data. These kind of values also cause hindrances to the **model building process.** Below is the **column wise** distribution of **missing values** before treatment.

```
             loan_amnt              funded_amnt         funded_amnt_inv
                     0                        0                       0
                  term                 int_rate             installment
                     0                        0                       0
                 grade               emp_length          home_ownership
                     0                        0                       0
            annual_inc      verification_status                 issue_d
                     0                        0                       0
            pymnt_plan                  purpose              addr_state
                     0                        0                       0
                   dti              delinq_2yrs        earliest_cr_line
                     0                        0                       0
        inq_last_6mths   mths_since_last_delinq                open_acc
                     0                   124638                       0
             revol_bal               revol_util               total_acc
                     0                      164                       0
             out_prncp            out_prncp_inv             total_pymnt
                     0                        0                       0
       total_pymnt_inv          total_rec_prncp           total_rec_int
                     0                        0                       0
    total_rec_late_fee               recoveries collection_recovery_fee
                     0                        0                       0
          last_pymnt_d          last_pymnt_amnt            next_pymnt_d
                   341                        0                  207723
    last_credit_pull_d         application_type             loan_status
                    16                        0                       0
```

Out of the variables which contain **NAs**, majority of them come from **date and time** columns. Since these columns cannot be used for **model building,** those columns can be omitted altogether. For the remaining **numerical variables** with **NAs**, machine learning algorithms like **KNN** cannot work in eliminating the **NAs** as the dataset has large number of observations. Therefore the process of **median** imputation can be used to replace the **NAs** with the **median** of that variable. In this way, the **normal distribution** of the variable is not disturbed and also will retain its old characteristics

After median imputation:

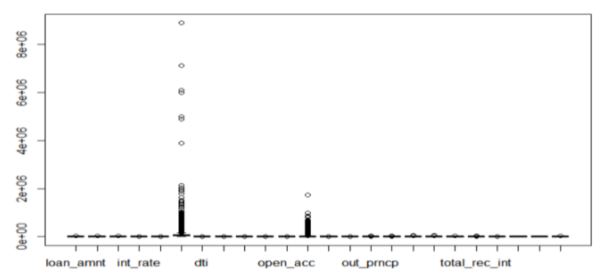
```
             loan_amnt              funded_amnt         funded_amnt_inv
                     0                        0                       0
                  term                 int_rate             installment
                     0                        0                       0
                 grade               emp_length          home_ownership
                     0                        0                       0
            annual_inc      verification_status                 issue_d
                     0                        0                       0
            pymnt_plan                  purpose              addr_state
                     0                        0                       0
                   dti              delinq_2yrs        earliest_cr_line
                     0                        0                       0
        inq_last_6mths   mths_since_last_delinq                open_acc
                     0                        0                       0
             revol_bal               revol_util               total_acc
                     0                        0                       0
             out_prncp            out_prncp_inv             total_pymnt
                     0                        0                       0
       total_pymnt_inv          total_rec_prncp           total_rec_int
                     0                        0                       0
    total_rec_late_fee               recoveries collection_recovery_fee
                     0                        0                       0
          last_pymnt_d          last_pymnt_amnt            next_pymnt_d
                   341                        0                  207723
    last_credit_pull_d         application_type             loan_status
                    16                        0                       0
```

*Now we can see that the **numerical variables** are devoid of **missing values.***

e. Outlier Treatment:

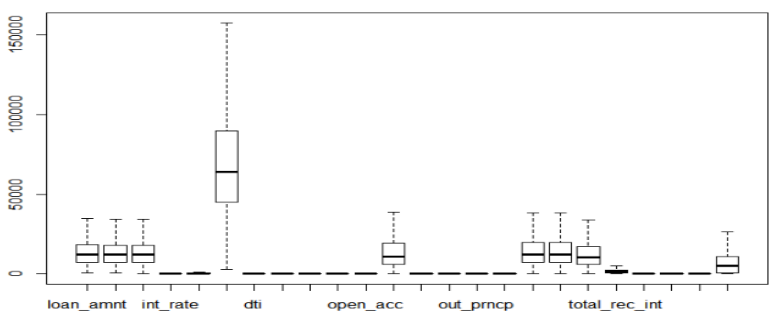
An **outlier** can be defined as those values which are at an abnormal distance from the other values in a **sample distribution.**

Before capping:



The **outliers** are more in number for our dataset. **Hence** omitting them is not an option. **Therefore** to deal with them, we use the method called **capping** where we **impute the outlier values** to the **thresholds** based on whether they are **upper outliers** or **lower outliers.**

After capping:



*Now we can see that all the **outliers** have been eliminated.*

f. Removal of unwanted variables:

While performing **analysis,** we came across **variables** which were found **unsuitable** for model building due to the following reasons.

- Their **standard deviation** was **zero** and hence they might give **abnormal** values while making **predictions.**
- They were in **Time and Date** format and hence could not be used as **variables** in model building.
- Those variables contributed to **majority** of **missing values** and removing them would make the process of model building easier.

- The **desc** and **ID** variable cannot be converted to any of the above data types and hence can be omitted from the analysis.

g. Variable Transformation:

The response variable, loan_status, if required for certain **model building** processes, must be converted to **categorical** variable with **'0'** and '**1'** instead of **"Fully Paid"** and **"Default"** as the variable's **levels.** This can be achieved by using **if else** function.

h. Insights from EDA:

- **Majority** of Burrowers have loans which have been **graded** come under **B-Grade** meaning it is the most **popular** grade **loan.**
- Majority of the **burrowers** have an **employment** period of more than **10 years** indicating that most of the burrowers wait until a period of **10 years** in employment to avail a loan in order to make them financially stable.
- Highest loan amount sanctioned by the **investors** to a **burrower** is **35000 units.**
- The **majority** of the burrowers have a **DTI** value of **16.88** indicating that **burrowers** have been able to balance their **credits** with respect to their **income**.
- The **annual income** of the **burrowers** ranges from **3000** units to **8900060** units with **average income** being **73965** units indicating most of the **burrowers** are able to pay back their loans when compared with the **instalment** amount.
- Only **34.4%** of the burrowers have not undergone **verification** indicating that **bank** has been **doing verifications** regularly.

3. Model Building:

The **second and most important objective** of this capstone project is to **build a prediction model** that will help **bank** decide whether a **borrower** of **loan** will **default** on their loan.

a. Data Preparation:

Since our **dataset** is **highly** imbalanced on the **response variable** where cases of **default** are **19063** and that of **fully paid** are **207723**, the dataset has been **balanced** using the **simultaneous Over-sampling** of **positive class(default)** and **under- sampling** of **negative class (Fully paid)** through **SMOTE.** The **balanced** dataset has brought both the classes to **50%-50% proportion** where **defaulters** were **95315** and **non-defaulters** were **92264.**
We were not provided with a **separate out-of-sample** data and hence we have divided the **existing dataset** in the ratio of **70%-30%** based on the **response variable** to be considered as **training** and **testing** data **respectively** for **model building.**

Dimensions of Training Dataset	Dimensions of Testing Dataset
131305 rows and **34** variables	**56274** rows and **34** variables

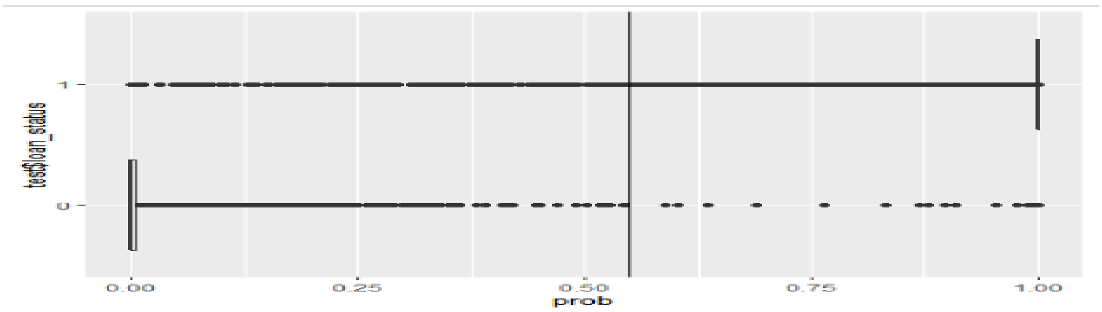
b. Model building:

i. Logistic Regression:

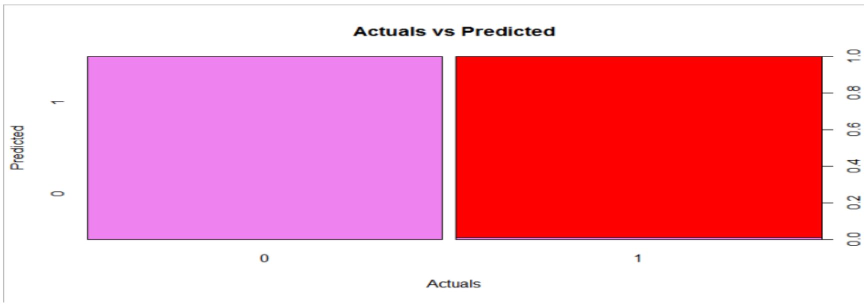
Logistic Regression is a type of predictive model which uses a **Logit Function** to predict the **category** by giving a **probability** of a class **as an output**. We have used only the **significant variables** whose **p-value** are less than **0.05** to make the **model.**

```
Call:
glm(formula = loan_status ~ loan_amnt + funded_amnt + funded_amnt_inv +
    int_rate + term + inq_last_6mths + open_acc + total_pymnt +
    total_pymnt_inv + total_rec_prncp + last_pymnt_amnt, family = "binomial",
    data = train)

Deviance Residuals:
    Min       1Q    Median      3Q      Max
-6.1712  -0.0012   0.0000   0.0010   8.4904

Coefficients:
                  Estimate Std. Error z value Pr(>|z|)
(Intercept)     -3.656e+00  1.494e-01 -24.477  < 2e-16 ***
loan_amnt       -4.035e-04  1.681e-04  -2.400   0.0164 *
funded_amnt      3.707e-03  4.932e-04   7.516 5.64e-14 ***
funded_amnt_inv -1.146e-03  4.627e-04  -2.476   0.0133 *
int_rate         8.418e-02  9.302e-03   9.050  < 2e-16 ***
term60 months    6.572e-01  9.238e-02   7.114 1.13e-12 ***
inq_last_6mths  -1.698e-01  3.988e-02  -4.258 2.06e-05 ***
open_acc         3.864e-02  7.176e-03   5.385 7.26e-08 ***
total_pymnt     -2.215e-03  5.013e-04  -4.418 9.94e-06 ***
total_pymnt_inv  2.578e-03  5.030e-04   5.124 2.99e-07 ***
total_rec_prncp -2.606e-03  4.593e-05 -56.738  < 2e-16 ***
last_pymnt_amnt -1.934e-03  3.738e-05 -51.738  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 181992.7  on 131304  degrees of freedom
Residual deviance:   7437.1  on 131293  degrees of freedom
AIC: 7461.1

Number of Fisher Scoring iterations: 12
```

The **positive coefficient** of the **variables** such as **int_rate** and **funded_amt** points towards the fact that the **model** is **inclined** towards finding the **default** class.



We can see that at the **threshold** of **0.55**, we are able to **predict correctly** more number of **0s** and **1s**. Therefore we can say that anything with a **value** of **above 0.55** is **‘1’** and **below 0.55** is **‘0’**.

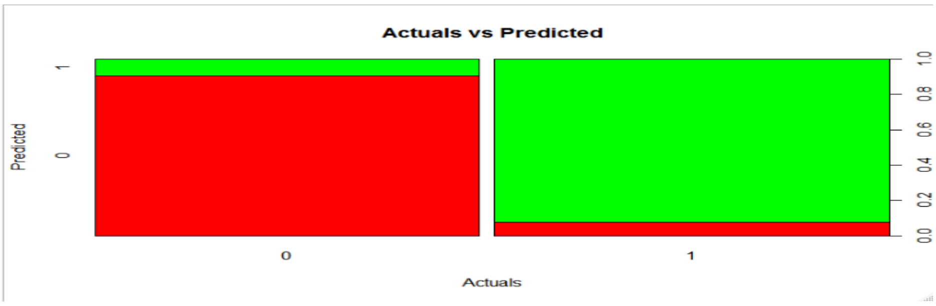


ii. Naïve Bayes:

The **Naïve Bayes** classifier uses an extension of **Bayes** theorem to predict **class outputs** given any number of **independent variables.**

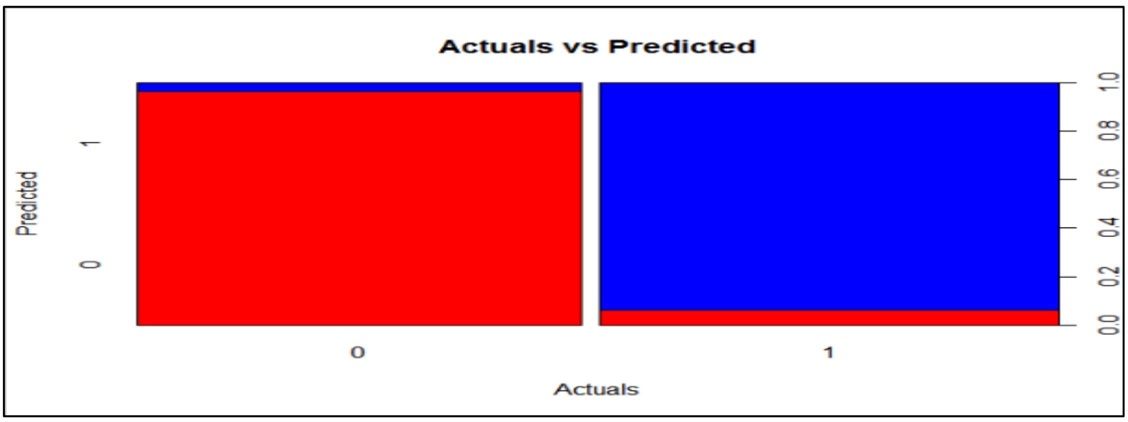
```
Naive Bayes Classifier for Discrete Predictors

Call:
naiveBayes.default(x = X, y = Y, laplace = laplace)

A-priori probabilities:
Y
        0         1 
0.4918701 0.5081299 

Conditional probabilities:
   loan_amnt
Y       [,1]     [,2]
  0 13372.38 8051.505
  1 16855.13 5977.559

   funded_amnt
Y       [,1]     [,2]
  0 13317.70 7991.332
  1 16838.51 5940.375

   funded_amnt_inv
Y       [,1]     [,2]
  0 13225.42 8013.777
  1 16827.03 5936.866

   term
Y    36 months 60 months
  0 0.8050166 0.1949834
  1 0.3695743 0.6304257
```

Above displays the summary of how the **model works.** Since **Naïve Bayes** is easier to implement, we can put all the **predictor variables** for **model building** unlike **logistic regression.**



iii. KNN:

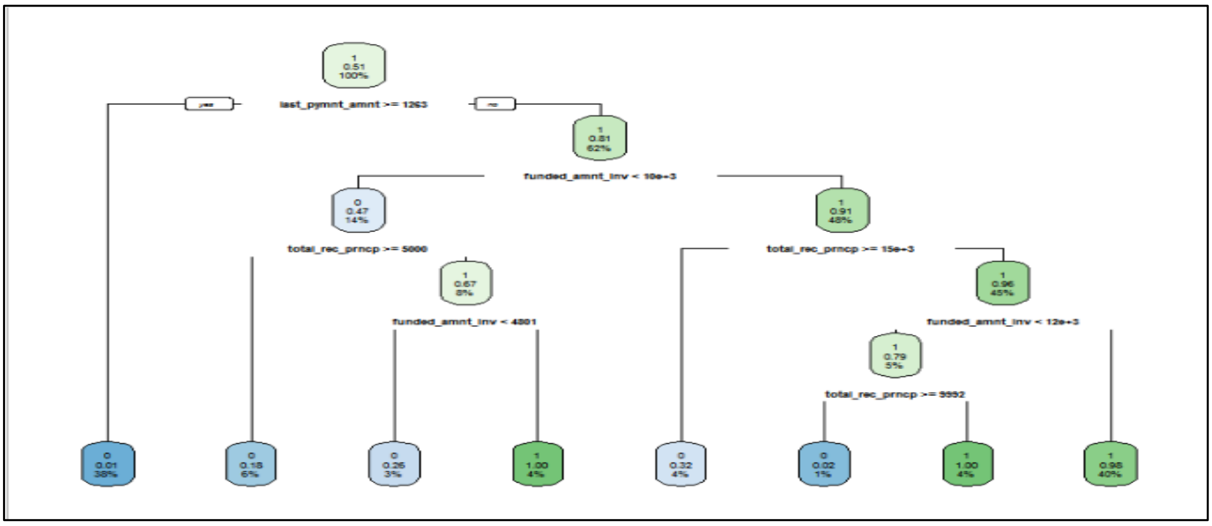
KNN(K Nearest Neighbours) algorithm as a **classifier** uses the **current data** and estimates the **class** of the new data point by using certain **similarity measures of distance.** Since the **algorithm** uses the **distances** between data points to find the **response variable**, all the **predictor variables** that are given to the model must be **numeric.** Since this a **non-parametrical algorithm**, the output of this algorithm is the **set of predictions.**



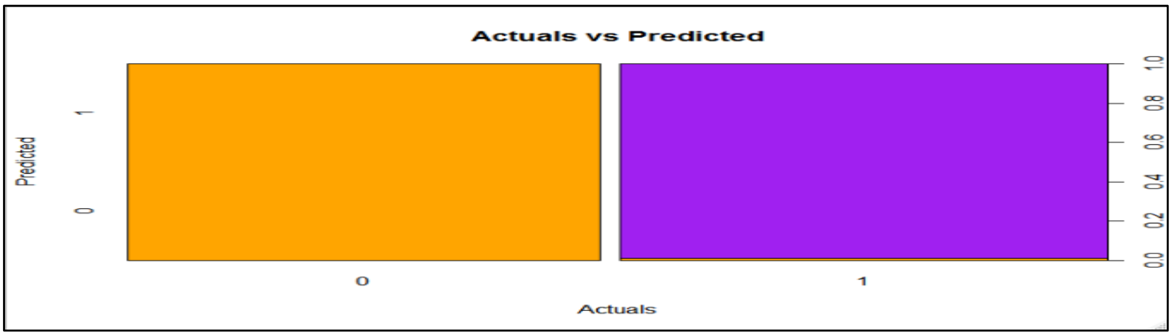
iv. CART Model:

CART (Classification and Regression Tree) is one of the methods of preparing a **Classification predictive model** which uses **Tree structure** to explain the predications of the **Classification or Regression model.**

```
n= 131305

node), split, n, loss, yval, (yprob)
      * denotes terminal node

 1) root 131305 64585 1 (0.491870073 0.508129927)
   2) last_pymnt_amnt>=1262.97 50101    547 0 (0.989082054 0.010917946) *
   3) last_pymnt_amnt< 1262.97 81204 15031 1 (0.185101719 0.814898281)
     6) funded_amnt_inv< 10000.41 17750  8291 0 (0.532901408 0.467098592)
      12) total_rec_prncp>=4999.959 7448  1368 0 (0.816326531 0.183673469) *
      13) total_rec_prncp< 4999.959 10302  3379 1 (0.327994564 0.672005436)
        26) funded_amnt_inv< 4800.779 4519  1166 0 (0.741978314 0.258021686) *
        27) funded_amnt_inv>=4800.779 5783    26 1 (0.004495936 0.995504064) *
     7) funded_amnt_inv>=10000.41 63454  5572 1 (0.087811643 0.912188357)
      14) total_rec_prncp>=14999.32 5023  1608 0 (0.679872586 0.320127414) *
      15) total_rec_prncp< 14999.32 58431  2157 1 (0.036915336 0.963084664)
        30) funded_amnt_inv< 12000.56 6415  1347 1 (0.209976617 0.790023383)
          60) total_rec_prncp>=9992.39 1379    32 0 (0.976794779 0.023205221) *
          61) total_rec_prncp< 9992.39 5036     0 1 (0.000000000 1.000000000) *
        31) funded_amnt_inv>=12000.56 52016   810 1 (0.015572132 0.984427868) *
```



As we can see that the **decision tree** is not **complex** and therefore we don't need to **prune** the **decision tree** as it would result in **loss** of **predictor variables** in the model.

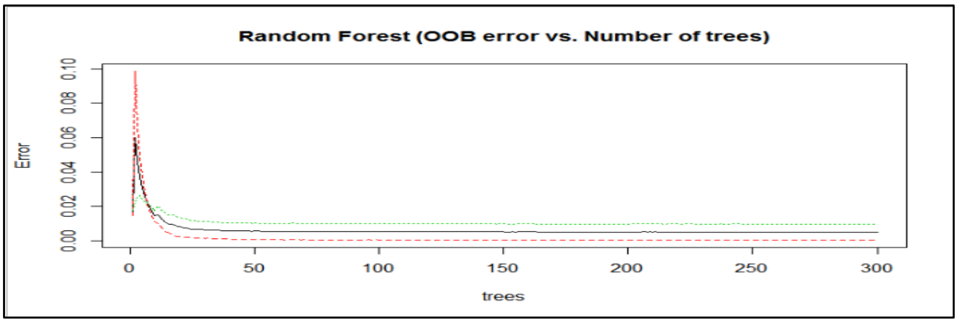


v. Random Forest:

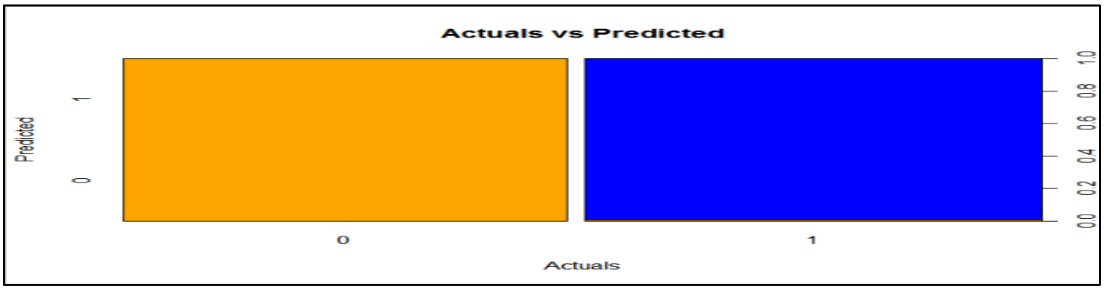
Random Forest is an **ensemble learning technique** where **multiple decision trees** are built and after analysing all the predictions by each of those **multiple trees**, the **best model tree** is selected.

```
Call:
 randomForest(formula = loan_status ~ ., data = train, ntree = 300,      mtry = 3, nodesize = 10, importance = TRUE, do.trace = TRUE)
               Type of random forest: classification
                     Number of trees: 300
No. of variables tried at each split: 3

        OOB estimate of  error rate: 0.5%
Confusion matrix:
      0     1  class.error
0 64567    18 0.0002787025
1   638 66082 0.0095623501
```

	0	1	MeanDecreaseAccuracy
loan_amnt	2.398236e-03	4.131643e-03	2.598487e-03
funded_amnt	4.472017e-03	3.571332e-03	3.289844e-03
funded_amnt_inv	4.368809e-03	3.754159e-03	3.185933e-03
term	1.785441e-03	1.074250e-03	1.122470e-03
int_rate	2.751149e-03	8.613282e-04	1.372204e-03
installment	2.660499e-03	2.257325e-03	1.983543e-03
grade	2.087035e-03	2.247896e-04	1.009188e-03
emp_length	1.032108e-04	1.659995e-04	1.119286e-04
home_ownership	5.574576e-05	3.537735e-05	3.048543e-05
annual_inc	1.040649e-04	5.743826e-04	3.069355e-04
verification_status	3.222780e-04	6.171788e-05	1.717177e-04
pymnt_plan	6.096276e-07	3.584666e-07	3.769772e-07
purpose	1.436478e-03	6.283238e-05	6.971984e-04
addr_state	1.331122e-03	6.400124e-05	6.488471e-04
dti	4.926014e-05	2.607625e-04	1.423820e-04
delinq_2yrs	0.000000e+00	0.000000e+00	0.000000e+00
inq_last_6mths	6.840783e-04	5.759141e-04	5.076284e-04
mths_since_last_delinq	0.000000e+00	0.000000e+00	0.000000e+00
open_acc	1.796648e-04	3.954199e-04	2.429568e-04
revol_bal	1.169631e-04	2.925080e-04	1.627072e-04
revol_util	8.993633e-05	3.534734e-04	1.942376e-04
total_acc	9.635181e-05	2.732800e-04	1.564734e-04
out_prncp	0.000000e+00	0.000000e+00	0.000000e+00
out_prncp_inv	0.000000e+00	0.000000e+00	0.000000e+00
total_pymnt	2.595239e-03	2.318308e-03	1.983890e-03
total_pymnt_inv	2.216468e-03	2.387669e-03	1.779992e-03
total_rec_prncp	4.221479e-03	2.497406e-03	2.864339e-03
total_rec_int	2.867757e-03	4.474845e-03	3.156308e-03
total_rec_late_fee	0.000000e+00	0.000000e+00	0.000000e+00
recoveries	0.000000e+00	0.000000e+00	0.000000e+00
collection_recovery_fee	0.000000e+00	0.000000e+00	0.000000e+00
last_pymnt_amnt	1.558932e-03	4.250130e-03	2.667080e-03
application_type	7.353777e-07	0.000000e+00	3.620683e-07



For 300 **decision trees made,** the **Random forest** algorithm has managed to get the **least OOB(Out of Box)** estimate error rate of **0.5%** which can termed as **good** and will help us give **good predictions.** We can also see that **int_rate** is the most **important predictor.**

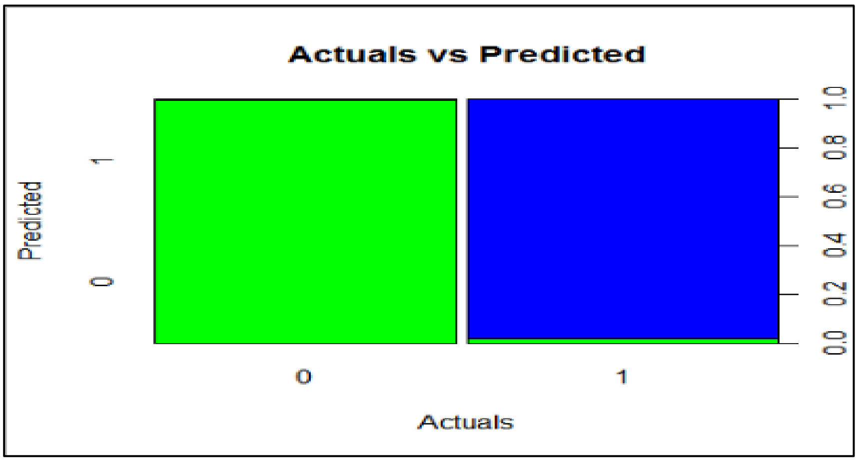


vi. Bagging:

Bagging or Bootstrap Aggregating is an Ensemble learning method where the **classifiers** use random **subsests** of the original **dataset** to make predictions.

```
$btree
n= 131305

node), split, n, loss, yval, (yprob)
      * denotes terminal node

 1) root 131305 64813 1 (0.493606489 0.506393511)
   2) last_pymnt_amnt>=1262.97 50444   510 0 (0.989889779 0.010110221) *
   3) last_pymnt_amnt< 1262.97 80861 14879 1 (0.184007123 0.815992877)
     6) funded_amnt_inv< 10000.41 17693  8305 0 (0.530605324 0.469394676)
      12) total_rec_prncp>=4999.886 7406  1372 0 (0.814744802 0.185255198) *
      13) total_rec_prncp< 4999.886 10287  3354 1 (0.326042578 0.673957422)
        26) funded_amnt_inv< 4800.779 4472  1136 0 (0.745974955 0.254025045) *
        27) funded_amnt_inv>=4800.779 5815    18 1 (0.003095443 0.996904557) *
     7) funded_amnt_inv>=10000.41 63168  5491 1 (0.086926925 0.913073075)
      14) total_rec_prncp>=14999.33 5001  1609 0 (0.678264347 0.321735653) *
      15) total_rec_prncp< 14999.33 58167  2099 1 (0.036085753 0.963914247)
        30) funded_amnt_inv< 12000.56 6377  1334 1 (0.209189274 0.790810726)
          60) total_rec_prncp>=9990.921 1360    26 0 (0.980882353 0.019117647) *
          61) total_rec_prncp< 9990.921 5017     0 1 (0.000000000 1.000000000) *
        31) funded_amnt_inv>=12000.56 51790   765 1 (0.014771191 0.985228809) *

attr(,"class")
   class
"sclass"


$OOB
[1] TRUE

$comb
[1] FALSE

$err
[1] 0.04155211

$call
bagging.data.frame(formula = loan_status ~ ., data = train, coob = TRUE,
    control = rpart.control(maxdepth = 10, minsplit = 3))
```

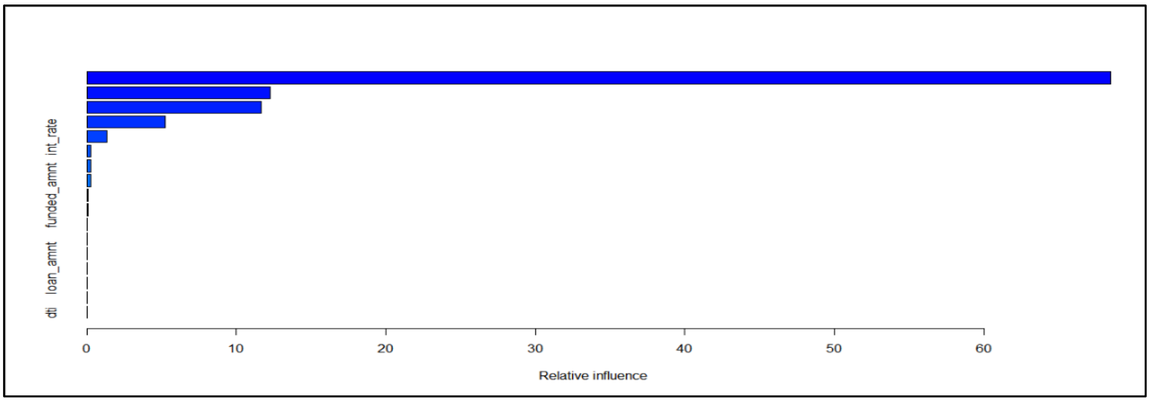
We can see that the **Out of bag** error is **0.0415.**



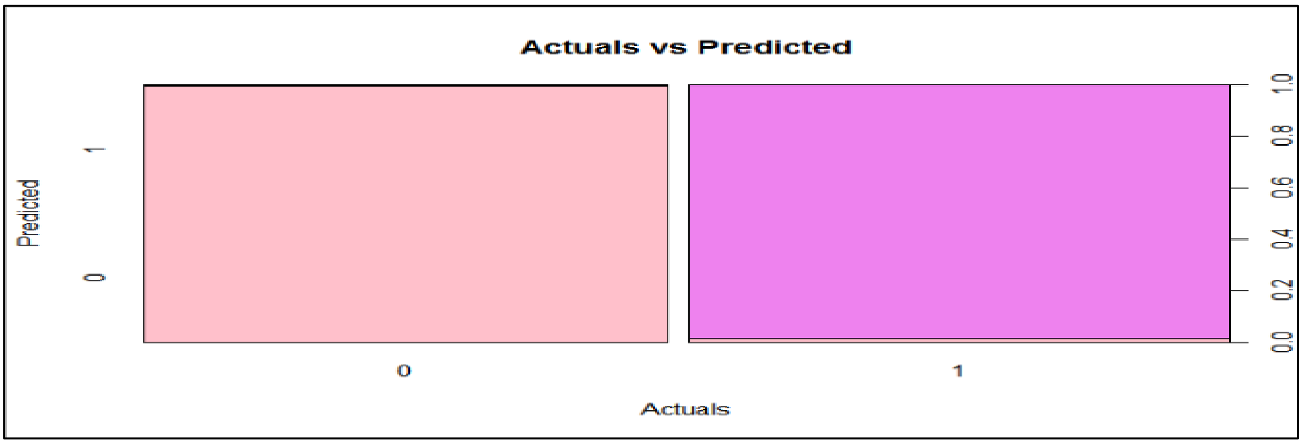
vii. Adaptive Boosting:

Adaptive Boosting or Adaboost is a **boosting algorithm** where the **iteration process** of building a **strong rule** is dependent on the **weightage** given to the **errors** created by the **preceding weak learner.**

```
                                var     rel.inf
last_pymnt_amnt last_pymnt_amnt 68.491413132
total_rec_prncp total_rec_prncp 12.274350907
funded_amnt_inv funded_amnt_inv 11.638873377
total_rec_int     total_rec_int  5.230670787
int_rate               int_rate  1.368792634
inq_last_6mths   inq_last_6mths  0.261706656
open_acc               open_acc  0.254320396
installment         installment  0.240232000
funded_amnt         funded_amnt  0.108647052
total_pymnt         total_pymnt  0.085712289
annual_inc           annual_inc  0.024465272
revol_util           revol_util  0.007644129
total_acc             total_acc  0.004887551
loan_amnt             loan_amnt  0.004871964
total_pymnt_inv total_pymnt_inv  0.001873917
revol_bal             revol_bal  0.001537937
dti                         dti  0.000000000
```



The variables **last_pymnt_amnt, total_rec_prncp** and **funded_amt_inv** are the **most important predictor** variables.

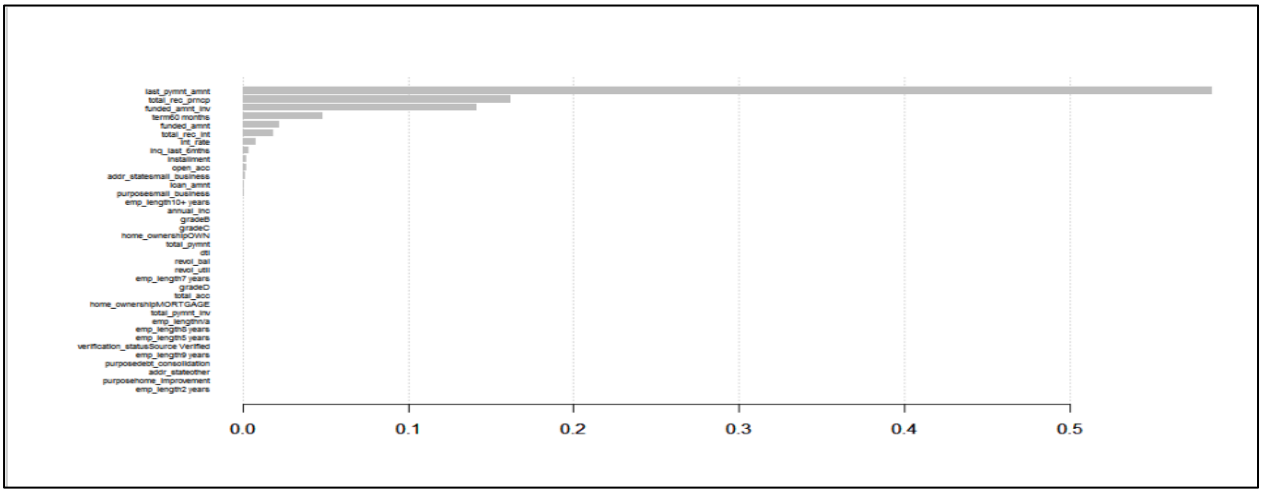


viii. Extreme Gradient Boosting:

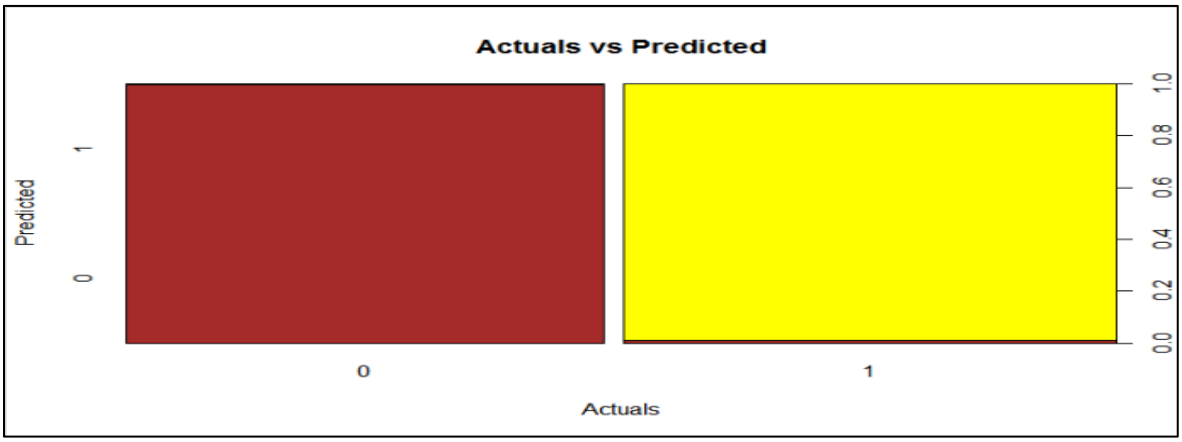
The **xgboost** model tries to make a **strong rule** by trying reduce the **error** obtained from the **loss function** from the **weak learners.**

```
##### xgb.Booster
raw: 123.3 Kb
call:
  xgb.train(params = params, data = dtrain, nrounds = nrounds,
    watchlist = watchlist, verbose = verbose, print_every_n = print_every_n,
    early_stopping_rounds = early_stopping_rounds, maximize = maximize,
    save_period = save_period, save_name = save_name, xgb_model = xgb_model,
    callbacks = callbacks, eta = 0.05, max_depth = 4, nfold = 0,
    objective = "binary:logistic")
params (as set within xgb.train):
  eta = "0.05", max_depth = "4", nfold = "0", objective = "binary:logistic", silent = "1"
xgb.attributes:
  best_iteration, best_msg, best_ntreelimit, best_score, niter
callbacks:
  cb.print.evaluation(period = print_every_n)
  cb.evaluation.log()
  cb.early.stop(stopping_rounds = early_stopping_rounds, maximize = maximize,
    verbose = verbose)
# of features: 78
niter: 100
best_iteration : 100
best_ntreelimit : 100
best_score : 0.006405
nfeatures : 78
evaluation_log:
    iter train_error
       1    0.047721
       2    0.042748
---
      99    0.006557
     100    0.006405
```

The **least error** achieved by the **model** at **100th iteration** is **0.006405.**



We can see that **last payment amount** is the most important variable in this model.



4. <u>Model Validation:</u>

a. <u>Comparison of models using various model measures:</u>

<u>Model Measures used for Comparison:</u>

1) **<u>Confusion Matrix</u>:** We use the confusion matrix to get a frequency of the predicted values vs. the actual values and **also** to find various metrics such as **accuracy, specificity, sensitivity, etc.**

- **Accuracy:** It is the **ratio** of **Correct Predictions** to **Total number of predictions.** ***Higher the better***
- **Sensitivity:** It can be defined as the **ratio** between **actual positives** to **correctly identified positives** by **classifier.**
- **Specificity:** It can be defined as the **ratio** between **actual negatives** to **correctly identified negatives** by **classifier.**

2) **<u>Concordance Ratio:</u>** In this method, the **values of Response variables and Probabilities are taken as pairs** and then tested if the **probabilities** predicted actually hold true. And then the number of pairs are counted with respect to total number of **pairs.** ***Higher the better***

3) **<u>Error Rate:</u>** In this method, the frequency of all the **Falsely Predicted values** are counted and then compared with the total number of **values** in the dataset. This shows to what extent the **predictions done were wrong.** ***Lower the better.***

4) **<u>ROC & AUC:</u>** Both of these measures help in determining the separation of the different Categories present in the **Target and Prediction Variables.** AUC = Area Under the Curve, ROC = Receiver Operating Characteristics. ***Higher the better***

5) **<u>Gini:</u>** The **Gini Coefficient** be determined to test the **purity** of the classes divided in the **Target variable using the prediction model.** ***Higher the better***

6) **<u>KS Value:</u>** The **KS**(Kolmogorov-Smirnov) value is the **highest separation of classes** that has been achieved by the predictive model. ***Higher the better.***

*Out of these, more emphasis was put on **Accuracy, Specificity and Sensitivity.***

Overview of Model Measures of all models

Model	Accuracy	Specificity	Sensitivity	Concordance	Error Rate	ROC	KS	AUC	Gini
Logistic Regression	***0.9956***	0.9993	0.9918	***0.998***	***0.004***	***0.99***	***0.993***	***0.998***	***0.49***
Naïve Bayes	0.916	0.9068	0.9255	0.979	0.083	0.98	0.993	0.998	0.49
KNN	0.9531	0.9670	**0.9997**	0.93	0.083	0.85	0.757	0.937	0.48
CART Model	0.958	0.9867	0.9303	0.973	0.046	0.87	0.917	0.983	0.492
Random Forest	***0.9956***	**0.9997**	0.9917	***0.999***	***0.004***	***0.99***	***0.994***	***0.997***	***0.484***
Bagging	0.958	0.9958	0.9290	0.947	0.041	0.92	0.925	0.972	**0.52**
Adaptive Boosting	0.9917	0.9984	0.9852	0.998	0.008	0.99	0.987	0.998	0.488
XG Boosting	**0.9971**	0.9995	0.9946	**0.999**	**0.002**	**0.99**	**0.994**	**0.999**	0.489

By comparing the **above results** and especially **sensitivity, specificity and accuracy,** we can say that **XG Boost model** has performed **better** over other **models.** This performance is followed by **Logistic Regression** and **Random Forest.**

b. Interpretation of the best model

Inferences of the best model:

- The **XG Boost** performed well due to the reason that its framework contains many **optimization** and **algorithmic advancements.** It also has **Lasso Regression** and **Ridge Regression** to prevent from **over fitting.**
- **Logistic Regression's** performance can be credited to the fact that we have only used **significant variables.** But since we have removed many **variables**, it might **lose** a bit of **variation** of the dataset that is **explained** by the **model.**
- The performance of **Random Forest** is because its algorithm uses **ensemble** of **many pruned decision trees** to build the **best decision tree.**

*This **XG Boost** model was further **tuned** by **increasing** the **number of cross validations** to **2** and **nrounds** value was increased to **1000** but there was **no improvement** in the results.*

***Ensemble method** of **weighted average** was used using the **3** top **models**, **logistic regression, Random Forest** and **XG Boost.** Even though the **results** were **slightly** improved, it could not be used as there is no specific model that can be **deployed** through this method.*

Business understanding of the XG Boost Model:

```
Confusion Matrix and Statistics

          Reference
Prediction     0     1
         0 27673   143
         1     6 28452

               Accuracy : 0.9974
                 95% CI : (0.9969, 0.9978)
    No Information Rate : 0.5081
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.9947

 Mcnemar's Test P-Value : < 2.2e-16

            Sensitivity : 0.9950
            Specificity : 0.9998
         Pos Pred Value : 0.9998
         Neg Pred Value : 0.9949
             Prevalence : 0.5081
         Detection Rate : 0.5056
   Detection Prevalence : 0.5057
      Balanced Accuracy : 0.9974

       'Positive' Class : 1
```

From the above **confusion matrix,** we have obtained **two types** of **errors** which are as follows:

- **False Positive:** This type of **error** occurs when the model predicts that the **borrower** will **default** but in reality he doesn't. In our case, this value is **6.**
  In this type of error, the bank **loses** the money that can be gained from a **potential** customer to whom loan could have been provided. But on the other hand, bank has been saved from **risk** of losing the **loan amount.**

- **False Negative:** This type of **error** occurs when the model predicts that the **borrower** will **not default** on the **loan** but actually **defaults** on the loan. In our case, this value is **143.**

This type of error leads to **huge losses** for the bank as they lose huge **amount of money** through the non-recoverable **loan amounts** provided to such **borrowers** thinking they will pay back. These type of **errors** cost the bank **more money** than the **first error.**

Compared to the **number** of **observations** in the dataset**,** the **values** of **both** the **errors** are **very low.**

Therefore we present the XG Boost as the best model to the management that can be used to classify defaulters and non-defaulters.

5. Recommendations:

a. Model based recommendations:

- The **XG Boost** model which we presented as the best model to the management has the **tendency** to commit more number of **False Negatives** more than **False Positives.** Therefore management must keep in mind the amount of **losses** incurred by each and use the model accordingly.
- The **XG Boost** requires lot of **computational resources** and **time.** Therefore the parameters must be tuned according to the size of the **dataset.**
- The model requires the **dataset** to be in the form of **model matrix** and therefore must be converted to one before using it for predictions.
- Building of **model matrix** requires creation of **dummy** variable mimicking the **response variable** for the **model deployment.**

b. Data based recommendations:

- The **imbalance** in the dataset can be **avoided** by including more number of **positive class(default)** observations.
- The **variables** chosen most probably must follow the **normal distribution** which will make the **job** of **model building easier.**
- By keeping in mind the **process** of **model building,** only those variables which will be used in the **model building process** must be selected in the dataset.
- The **dataset** can be trimmed down further and try to **represent** the **population dataset** as compared to **huge datasets.**
- Inconsistencies such as **missing values** and **outliers** can be avoided so as to improve the **accuracy** of the **models.**

c. Business relevant recommendations:

- The **bank** must find ways to **market** higher **loan grades** such as **F and G** in order to increase their **revenues.**
- The **advertising** must be targeted towards borrowers with **employment period** of **more than 10 years** because that's when they think of going for a **loan.**
- The bank can put emphasis on **housing loans** since most of the **borrowers** are either living in **rented house** or **having mortgages.**
- The bank must put up some offers on **Joint** loans as opposed to **individual** loans to increase the **revenue.**

6. Conclusion:

As a ***business analyst,*** *we have performed* ***analysis*** *on the given dataset to give the* ***management*** *information on their* ***customer base.*** *After analysis, we have* ***prepared various prediction models,*** *compared using* ***model measures*** *and* ***interpreted*** *the best model which could be presented to the management was* ***XG Boost.*** *This* ***model*** *was further improved further by using* ***ensemble method.*** *Further* ***inferences*** *were given for the understanding of the management.*

It is expected that ***management*** *will consider the* ***recommendations*** *provided followed by the* ***usage*** *of* ***best prediction model*** *to* ***minimize*** *their* ***losses*** *and increase their* ***revenue*** *and save themselves from the clutches of* ***loan defaulters.***