Introduction to Data Science

Default of Credit Card Clients

Presented by:
Group ID: 1

Student Reg#	Student Name
L1F16BSCS0049	**Mohsin Basti**
L1F16BSCS0159	**Abdullah Shafique**
L1S16BSCS0021	**Rehmat Ali Haider**

Faculty of Information Technology

University of Central Punjab

Table of Contents

1. Data Source

1.1 Data set Selection

We have selected this dataset because it was related to our interest. Moreover, using this dataset we can make an application to predict the further defaults of credit cards of clients. Using this dataset, we will build such an application which will consider the variables of dataset as input and will predict the defaults of credit cards of further clients in future.

1.2 Location Site of Data Source

We have taken this dataset from the famous website which is UCI. This site provides bulk of datasets.
Following is the link from where we get this dataset.
http://archive.ics.uci.edu/ml/datasets/default+of+credit+card+clients

1.3 Packages and Libraries Requirements

The packages we have identified to use in this project till now are following:

Packages:

1: dplyr
2: ggplot2
3: plyr
4: tidyr
5: stringr

Libraries:

1: dplyr
2: plyr
3: ggplot2
4: tidyr
5: stringr
6: randomforest
7: naivebayes
8: neuralnet
9: caret
10: rpart
11: rpart.plot
12: class
13: caTools
14: naniar
15: e1070

We may use more libraries and packages after exploring more about the project. If we will use more libraries and packages then will be mention/update those in next document.

1.4 Data set Information

Following is the information regarding the data set we are using in our project:
This research aimed at the case of customers default payments in Taiwan and compares the predictive accuracy of probability of default among six data mining methods. From the perspective of risk management, the result of predictive accuracy of the estimated probability of default will be more valuable than the binary result of classification - credible or not

credible clients. Because the real probability of default is unknown, this study presented the novel Sorting Smoothing Method to estimate the real probability of default. With the real probability of default as the response variable (Y), and the predictive probability of default as the independent variable (X), the simple linear regression result (Y = A + BX) shows that the forecasting model produced by artificial neural network has the highest coefficient of determination; its regression intercept (A) is close to zero, and regression coefficient (B) to one. Therefore, among the six data mining techniques, artificial neural network is the only one that can accurately estimate the real probability of default.

2. Data Cleansing

2.1 Dataset Exploration

We have done Exploration of Our Dataset:

```
library(readxl)
myData <- read_excel("~/Desktop/default of credit card clients.xls")
View(myData)
```

head(myData)

```
> head(myData)
# A tibble: 6 x 25
     ID LIMIT_BAL   SEX EDUCATION MARRIAGE   AGE PAY_0 PAY_2 PAY_3 PAY_4 PAY_5 PAY_6 BILL_AMT1 BILL_AMT2
  <dbl>     <dbl> <dbl>     <dbl>    <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>     <dbl>     <dbl>
1     1     20000     2         2        1    24     2     2    -1    -1    -2    -2      3913      3102
2     2    120000     2         2        2    26    -1     2     0     0     0     2      2682      1725
3     3     90000     2         2        2    34     0     0     0     0     0     0     29239     14027
4     4     50000     2         2        1    37     0     0     0     0     0     0     46990     48233
5     5     50000     1         2        1    57    -1     0    -1     0     0     0      8617      5670
6     6     50000     1         1        2    37     0     0     0     0     0     0     64400     57069
# … with 11 more variables: BILL_AMT3 <dbl>, BILL_AMT4 <dbl>, BILL_AMT5 <dbl>, BILL_AMT6 <dbl>,
#   PAY_AMT1 <dbl>, PAY_AMT2 <dbl>, PAY_AMT3 <dbl>, PAY_AMT4 <dbl>, PAY_AMT5 <dbl>, PAY_AMT6 <dbl>,
#   `default payment next month` <dbl>
>
```

tail(myData)

```
> tail(myData)
# A tibble: 6 x 25
     ID LIMIT_BAL   SEX EDUCATION MARRIAGE   AGE PAY_0 PAY_2 PAY_3 PAY_4 PAY_5 PAY_6 BILL_AMT1 BILL_AMT2
  <dbl>     <dbl> <dbl>     <dbl>    <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>     <dbl>     <dbl>
1 29995     80000     1         2        2    34     2     2     2     2     2     2     72557     77708
2 29996    220000     1         3        1    39     0     0     0     0     0     0    188948    192815
3 29997    150000     1         3        2    43    -1    -1    -1    -1     0     0      1683      1828
4 29998     30000     1         2        2    37     4     3     2    -1     0     0      3565      3356
5 29999     80000     1         3        1    41     1    -1     0     0     0    -1     -1645     78379
6 30000     50000     1         2        1    46     0     0     0     0     0     0     47929     48905
# … with 11 more variables: BILL_AMT3 <dbl>, BILL_AMT4 <dbl>, BILL_AMT5 <dbl>, BILL_AMT6 <dbl>,
#   PAY_AMT1 <dbl>, PAY_AMT2 <dbl>, PAY_AMT3 <dbl>, PAY_AMT4 <dbl>, PAY_AMT5 <dbl>, PAY_AMT6 <dbl>,
#   `default payment next month` <dbl>
>
```

class(myData)

```
> class(myData)
[1] "tbl_df"     "tbl"        "data.frame"
>
```

dim(myData)

```
> dim(myData)
[1] 30000    25
>
```

summary(myData)

```
> summary(myData)
       ID          LIMIT_BAL           SEX          EDUCATION        MARRIAGE          AGE
 Min.   :    1   Min.   :  10000   Min.   :1.000   Min.   :0.000   Min.   :0.000   Min.   :21.00
 1st Qu.: 7501   1st Qu.:  50000   1st Qu.:1.000   1st Qu.:1.000   1st Qu.:1.000   1st Qu.:28.00
 Median :15000   Median : 140000   Median :2.000   Median :2.000   Median :2.000   Median :34.00
 Mean   :15000   Mean   : 167484   Mean   :1.604   Mean   :1.853   Mean   :1.552   Mean   :35.49
 3rd Qu.:22500   3rd Qu.: 240000   3rd Qu.:2.000   3rd Qu.:2.000   3rd Qu.:2.000   3rd Qu.:41.00
 Max.   :30000   Max.   :1000000   Max.   :2.000   Max.   :6.000   Max.   :3.000   Max.   :79.00
                                                                   NA's   :1
     PAY_0             PAY_2             PAY_3             PAY_4             PAY_5
 Min.   :-2.0000   Min.   :-2.0000   Min.   :-2.0000   Min.   :-2.0000   Min.   :-2.0000
 1st Qu.:-1.0000   1st Qu.:-1.0000   1st Qu.:-1.0000   1st Qu.:-1.0000   1st Qu.:-1.0000
 Median : 0.0000   Median : 0.0000   Median : 0.0000   Median : 0.0000   Median : 0.0000
 Mean   :-0.0167   Mean   :-0.1338   Mean   :-0.1662   Mean   :-0.2207   Mean   :-0.2662
 3rd Qu.: 0.0000   3rd Qu.: 0.0000   3rd Qu.: 0.0000   3rd Qu.: 0.0000   3rd Qu.: 0.0000
 Max.   : 8.0000   Max.   : 8.0000   Max.   : 8.0000   Max.   : 8.0000   Max.   : 8.0000

     PAY_6           BILL_AMT1         BILL_AMT2         BILL_AMT3         BILL_AMT4         BILL_AMT5
 Min.   :-2.0000   Min.   :-165580   Min.   :-69777   Min.   :-157264   Min.   :-170000   Min.   :-81334
 1st Qu.:-1.0000   1st Qu.:   3559   1st Qu.:  2985   1st Qu.:   2666   1st Qu.:   2327   1st Qu.:  1763
 Median : 0.0000   Median :  22382   Median : 21200   Median :  20088   Median :  19052   Median : 18104
 Mean   :-0.2911   Mean   :  51223   Mean   : 49179   Mean   :  47013   Mean   :  43263   Mean   : 40311
 3rd Qu.: 0.0000   3rd Qu.:  67091   3rd Qu.: 64006   3rd Qu.:  60165   3rd Qu.:  54506   3rd Qu.: 50190
 Max.   : 8.0000   Max.   : 964511   Max.   :983931   Max.   :1664089   Max.   : 891586   Max.   :927171

   BILL_AMT6          PAY_AMT1         PAY_AMT2          PAY_AMT3         PAY_AMT4         PAY_AMT5
 Min.   :-339603   Min.   :     0   Min.   :      0   Min.   :     0   Min.   :     0   Min.   :     0.0
 1st Qu.:   1256   1st Qu.:  1000   1st Qu.:    833   1st Qu.:   390   1st Qu.:   296   1st Qu.:   252.5
 Median :  17071   Median :  2100   Median :   2009   Median :  1800   Median :  1500   Median :  1500.0
 Mean   :  38872   Mean   :  5664   Mean   :   5921   Mean   :  5226   Mean   :  4826   Mean   :  4799.4
 3rd Qu.:  49198   3rd Qu.:  5006   3rd Qu.:   5000   3rd Qu.:  4505   3rd Qu.:  4013   3rd Qu.:  4031.5
 Max.   : 961664   Max.   :873552   Max.   :1684259   Max.   :896040   Max.   :621000   Max.   :426529.0

    PAY_AMT6        default payment next month
 Min.   :     0.0   Min.   :0.0000
 1st Qu.:   117.8   1st Qu.:0.0000
 Median :  1500.0   Median :0.0000
 Mean   :  5215.5   Mean   :0.2212
 3rd Qu.:  4000.0   3rd Qu.:0.0000
 Max.   :528666.0   Max.   :1.0000

>
```

names(myData)

```
> names(myData)
 [1] "ID"                      "LIMIT_BAL"               "SEX"
 [4] "EDUCATION"               "MARRIAGE"                "AGE"
 [7] "PAY_0"                   "PAY_2"                   "PAY_3"
[10] "PAY_4"                   "PAY_5"                   "PAY_6"
[13] "BILL_AMT1"               "BILL_AMT2"               "BILL_AMT3"
[16] "BILL_AMT4"               "BILL_AMT5"               "BILL_AMT6"
[19] "PAY_AMT1"                "PAY_AMT2"                "PAY_AMT3"
[22] "PAY_AMT4"                "PAY_AMT5"                "PAY_AMT6"
[25] "default payment next month"
>
```

str(myData)

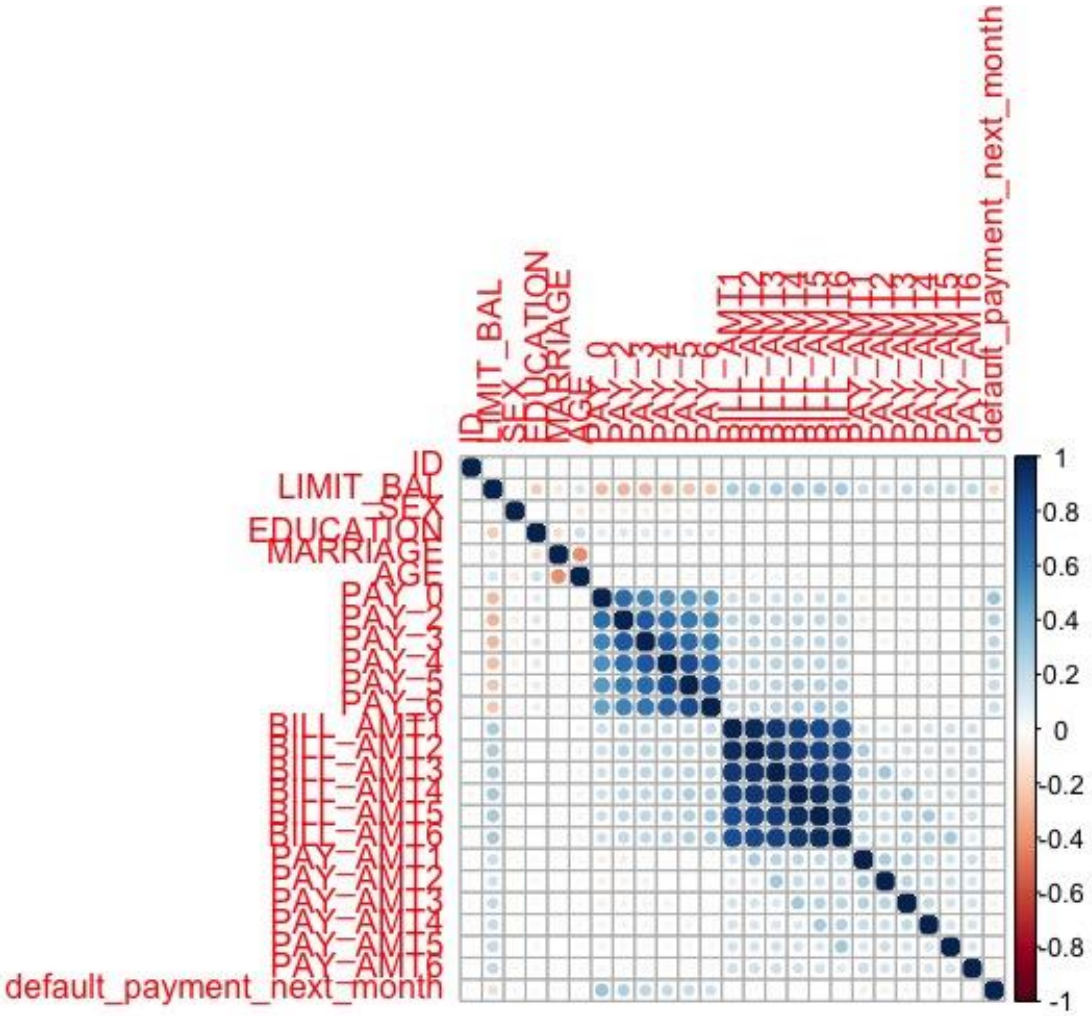
```
> str(myData)
tibble [30,000 × 25] (S3: tbl_df/tbl/data.frame)
 $ ID                        : num [1:30000] 1 2 3 4 5 6 7 8 9 10 ...
 $ LIMIT_BAL                 : num [1:30000] 20000 120000 90000 50000 50000 50000 500000 100000 140000 20000 ...
 $ SEX                       : num [1:30000] 2 2 2 2 1 1 1 2 2 1 ...
 $ EDUCATION                 : num [1:30000] 2 2 2 2 2 1 1 2 3 3 ...
 $ MARRIAGE                  : num [1:30000] 1 2 2 1 1 2 2 2 1 2 ...
 $ AGE                       : num [1:30000] 24 26 34 37 57 37 29 23 28 35 ...
 $ PAY_0                     : num [1:30000] 2 -1 0 0 -1 0 0 0 0 -2 ...
 $ PAY_2                     : num [1:30000] 2 2 0 0 0 0 0 -1 0 -2 ...
 $ PAY_3                     : num [1:30000] -1 0 0 0 -1 0 0 -1 2 -2 ...
 $ PAY_4                     : num [1:30000] -1 0 0 0 0 0 0 0 0 -2 ...
 $ PAY_5                     : num [1:30000] -2 0 0 0 0 0 0 0 0 -1 ...
 $ PAY_6                     : num [1:30000] -2 2 0 0 0 0 0 -1 0 -1 ...
 $ BILL_AMT1                 : num [1:30000] 3913 2682 29239 46990 8617 ...
 $ BILL_AMT2                 : num [1:30000] 3102 1725 14027 48233 5670 ...
 $ BILL_AMT3                 : num [1:30000] 689 2682 13559 49291 35835 ...
 $ BILL_AMT4                 : num [1:30000] 0 3272 14331 28314 20940 ...
 $ BILL_AMT5                 : num [1:30000] 0 3455 14948 28959 19146 ...
 $ BILL_AMT6                 : num [1:30000] 0 3261 15549 29547 19131 ...
 $ PAY_AMT1                  : num [1:30000] 0 0 1518 2000 2000 ...
 $ PAY_AMT2                  : num [1:30000] 689 1000 1500 2019 36681 ...
 $ PAY_AMT3                  : num [1:30000] 0 1000 1000 1200 10000 657 38000 0 432 0 ...
 $ PAY_AMT4                  : num [1:30000] 0 1000 1000 1100 9000 ...
 $ PAY_AMT5                  : num [1:30000] 0 0 1000 1069 689 ...
 $ PAY_AMT6                  : num [1:30000] 0 2000 5000 1000 679 ...
 $ default payment next month: num [1:30000] 1 1 0 0 0 0 0 0 0 0 ...
>
```

glimpse(myData)

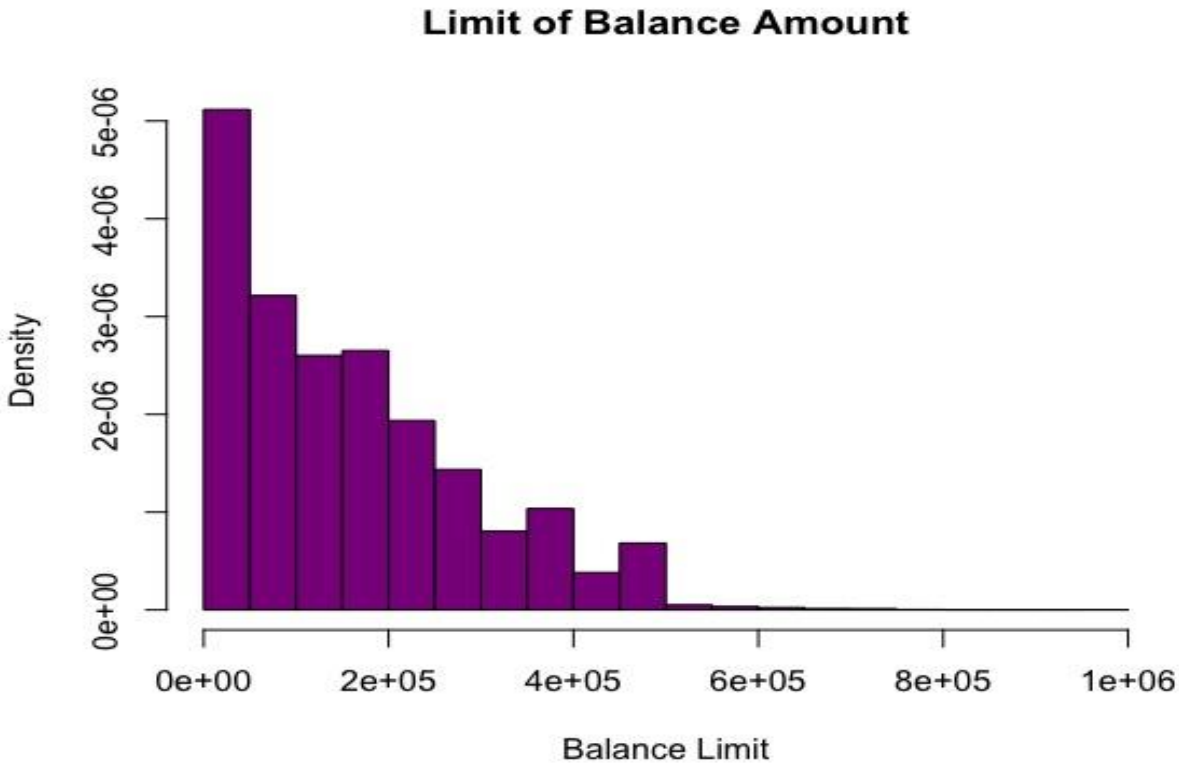
```
> glimpse(myData)
Rows: 30,000
Columns: 25
$ ID                        <dbl> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 2…
$ LIMIT_BAL                 <dbl> 20000, 120000, 90000, 50000, 50000, 50000, 500000, 100000, 140000, 2…
$ SEX                       <dbl> 2, 2, 2, 2, 1, 1, 1, 2, 2, 1, 2, 2, 2, 1, 1, 2, 1, 1, 2, 2, 2, 2, 2,…
$ EDUCATION                 <dbl> 2, 2, 2, 2, 2, 1, 1, 2, 3, 3, 3, 1, 2, 2, 1, 3, 1, 1, 1, 1, 3, 2, 2,…
$ MARRIAGE                  <dbl> 1, 2, 2, 1, 1, 2, 2, 2, 1, 2, 2, 2, NA, 2, 2, 3, 2, 1, 1, 2, 2, 1, 2…
$ AGE                       <dbl> 24, 26, 34, 37, 57, 37, 29, 23, 28, 35, 34, 51, 41, 30, 29, 23, 24, …
$ PAY_0                     <dbl> 2, -1, 0, 0, -1, 0, 0, 0, 0, -2, 0, -1, -1, 1, 0, 1, 0, 0, 1, 1, 0, …
$ PAY_2                     <dbl> 2, 2, 0, 0, 0, 0, 0, -1, 0, -2, 0, -1, 0, 2, 0, 2, 0, 0, -2, -2, 0, …
$ PAY_3                     <dbl> -1, 0, 0, 0, -1, 0, 0, -1, 2, -2, 2, -1, -1, 2, 0, 0, 2, 0, -2, -2, …
$ PAY_4                     <dbl> -1, 0, 0, 0, 0, 0, 0, 0, 0, -2, 0, -1, -1, 0, 0, 0, 2, -1, -2, -2, 0…
$ PAY_5                     <dbl> -2, 0, 0, 0, 0, 0, 0, 0, 0, -1, 0, -1, -1, 0, 0, 0, 2, -1, -2, -2, 0…
$ PAY_6                     <dbl> -2, 2, 0, 0, 0, 0, 0, -1, 0, -1, -1, 2, -1, 2, 0, 0, 2, -1, -2, -2, …
$ BILL_AMT1                 <dbl> 3913, 2682, 29239, 46990, 8617, 64400, 367965, 11876, 11285, 0, 1107…
$ BILL_AMT2                 <dbl> 3102, 1725, 14027, 48233, 5670, 57069, 412023, 380, 14096, 0, 9787, …
$ BILL_AMT3                 <dbl> 689, 2682, 13559, 49291, 35835, 57608, 445007, 601, 12108, 0, 5535, …
$ BILL_AMT4                 <dbl> 0, 3272, 14331, 28314, 20940, 19394, 542653, 221, 12211, 0, 2513, 85…
$ BILL_AMT5                 <dbl> 0, 3455, 14948, 28959, 19146, 19619, 483003, -159, 11793, 13007, 182…
$ BILL_AMT6                 <dbl> 0, 3261, 15549, 29547, 19131, 20024, 473944, 567, 3719, 13912, 3731,…
$ PAY_AMT1                  <dbl> 0, 0, 1518, 2000, 2000, 2500, 55000, 380, 3329, 0, 2306, 21818, 1000…
$ PAY_AMT2                  <dbl> 689, 1000, 1500, 2019, 36681, 1815, 40000, 601, 0, 0, 12, 9966, 6500…
$ PAY_AMT3                  <dbl> 0, 1000, 1000, 1200, 10000, 657, 38000, 0, 432, 0, 50, 8583, 6500, 3…
$ PAY_AMT4                  <dbl> 0, 1000, 1000, 1100, 9000, 1000, 20239, 581, 1000, 13007, 300, 22301…
$ PAY_AMT5                  <dbl> 0, 0, 1000, 1069, 689, 1000, 13750, 1687, 1000, 1122, 3738, 0, 2870,…
$ PAY_AMT6                  <dbl> 0, 2000, 5000, 1000, 679, 800, 13770, 1542, 1000, 0, 66, 3640, 0, 0,…
$ `default payment next month` <dbl> 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 1, 1,…
>
```

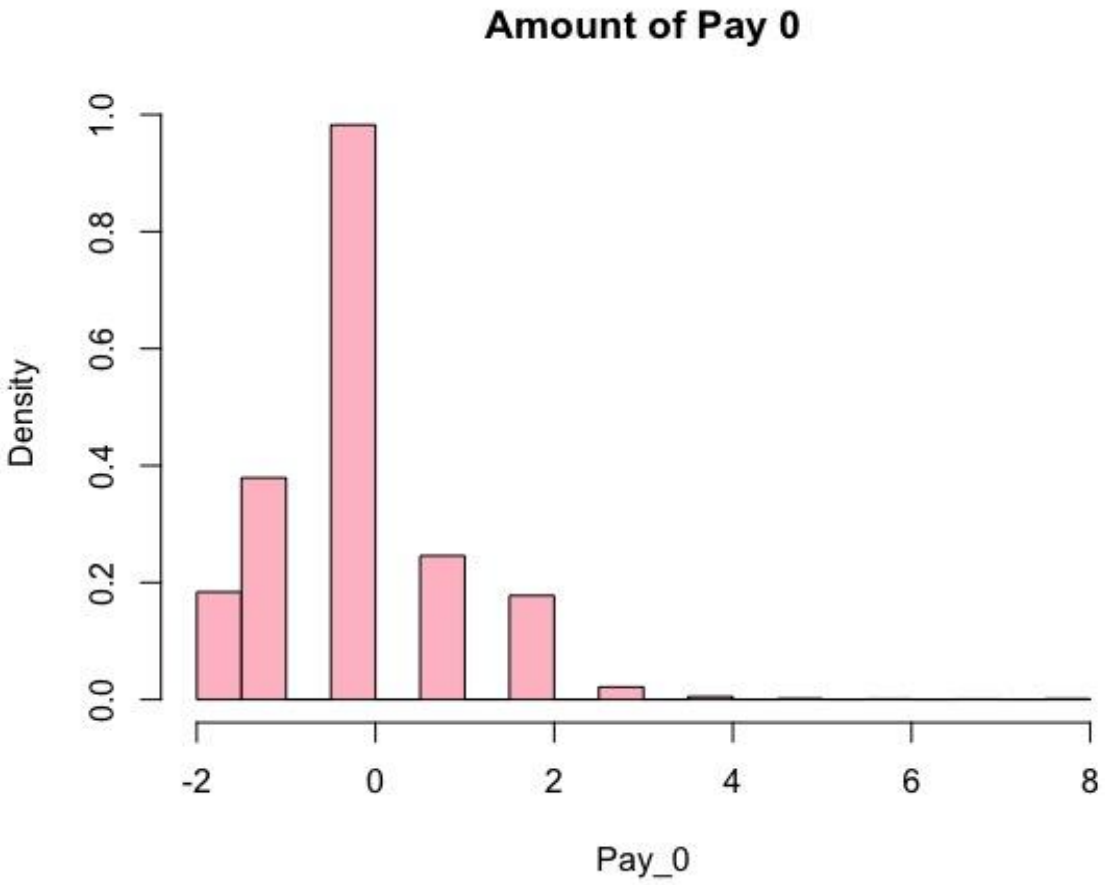
3. Exploratory Data Analysis

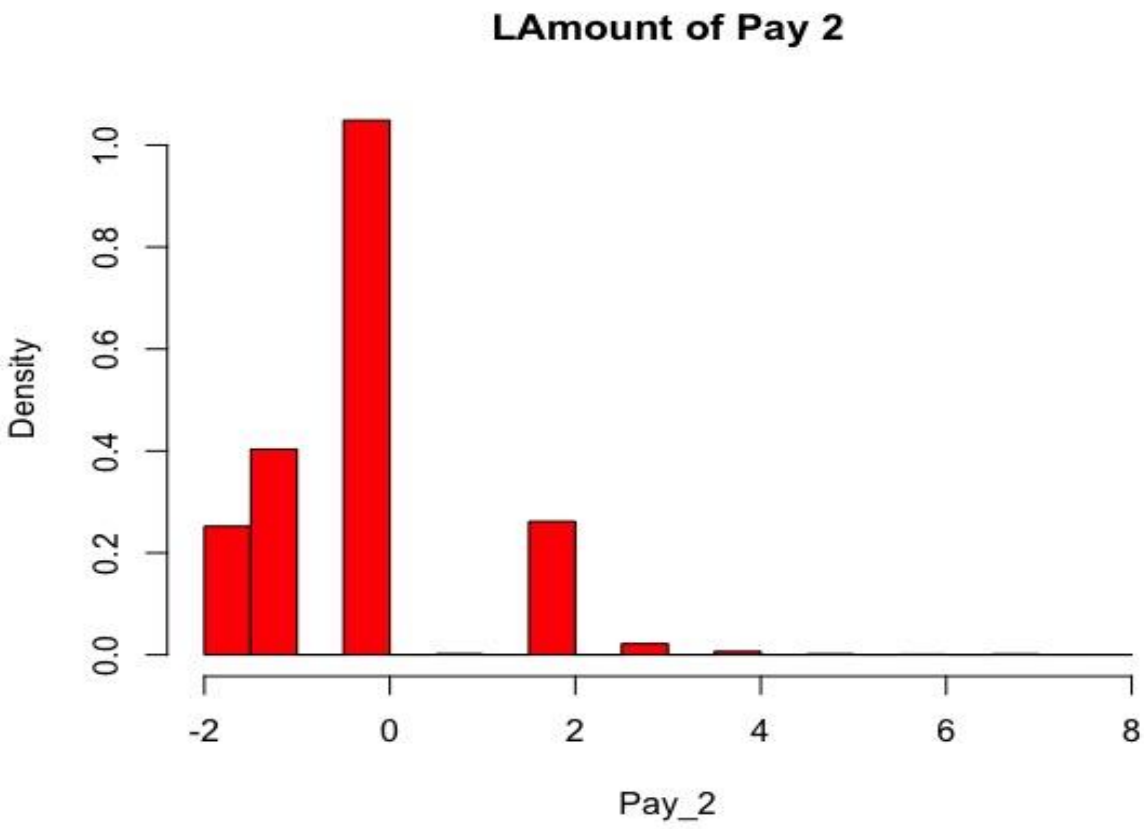
Correlation Plot:

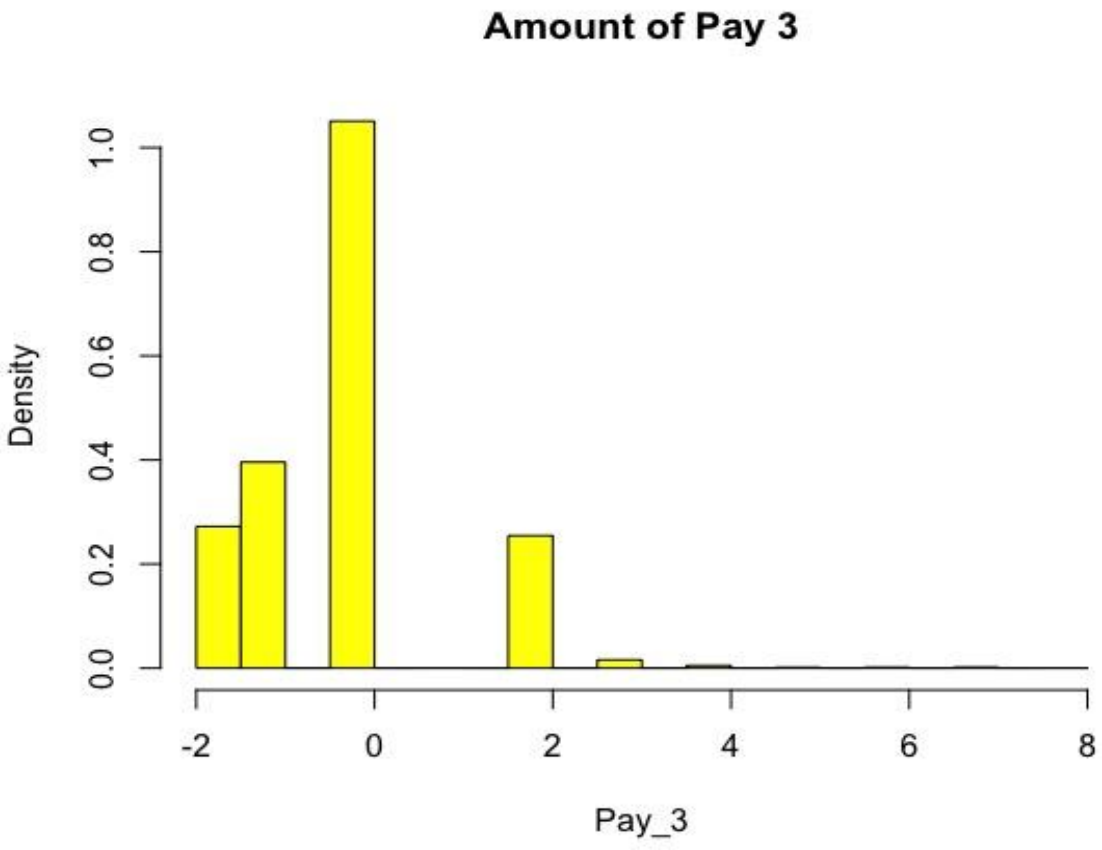


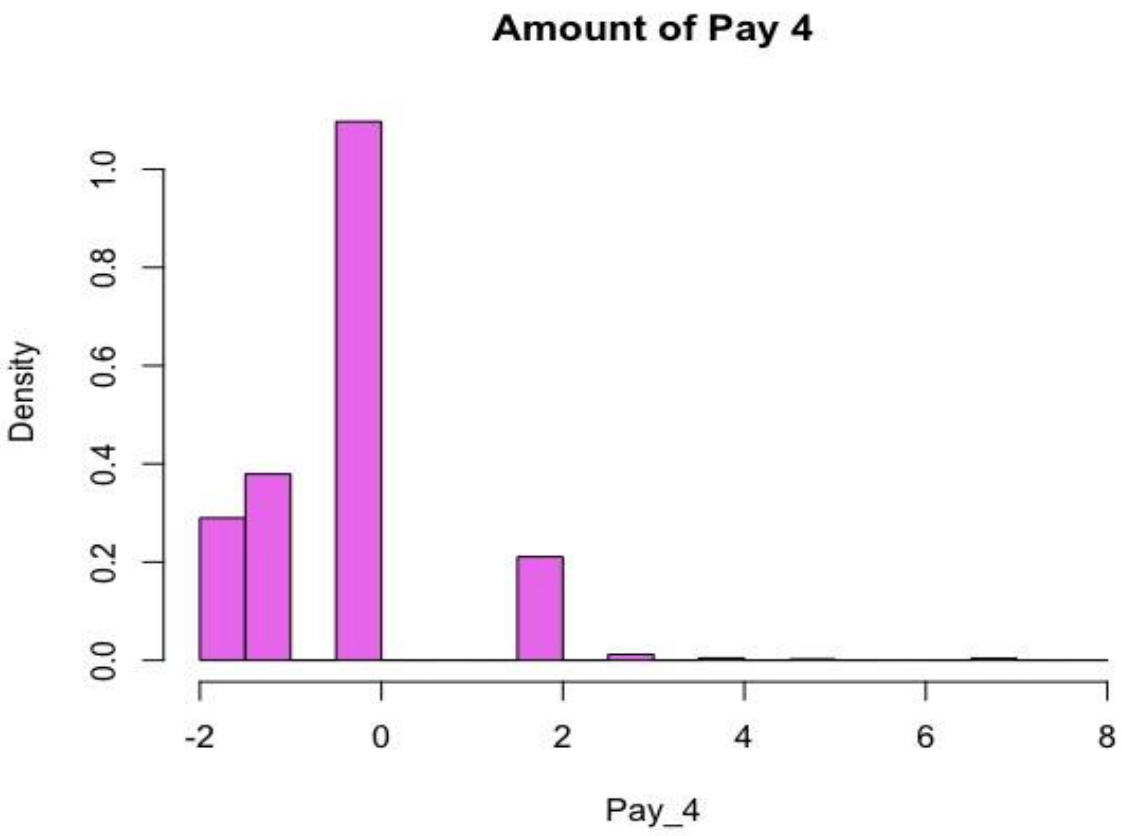
Histogram Plot:

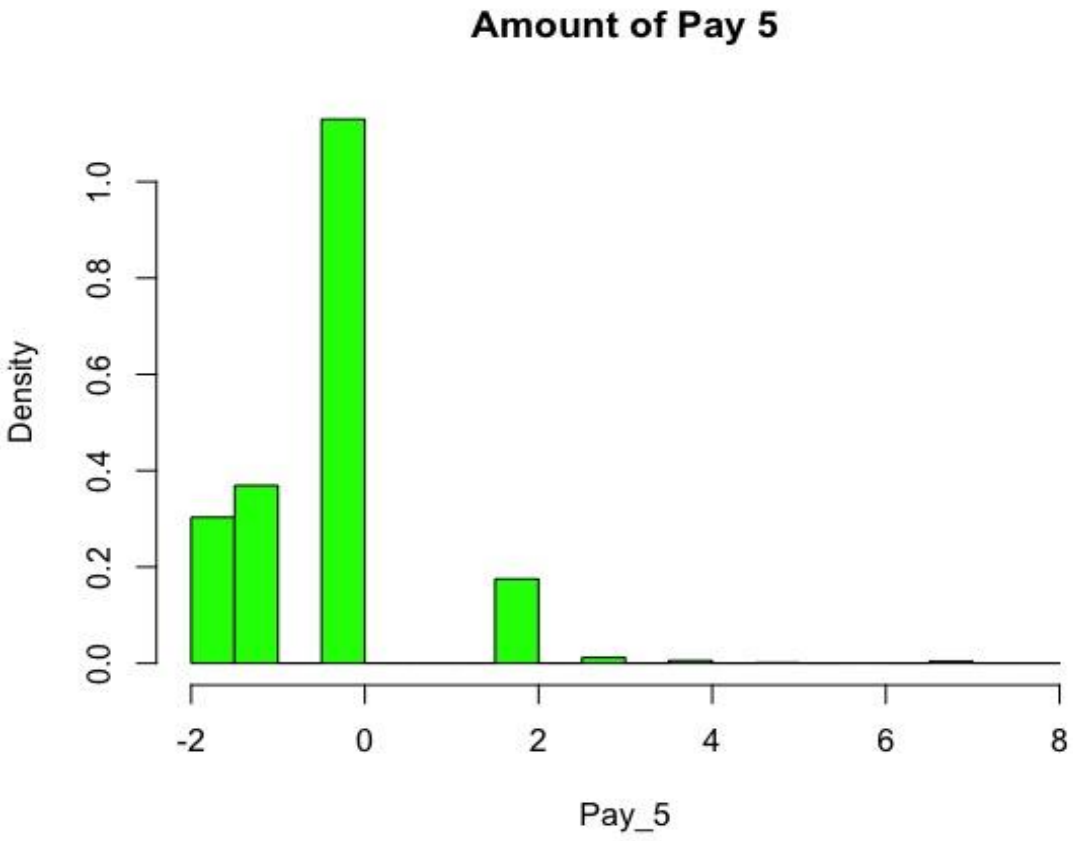

Limit of Balance Amount
Density
0e+00
2e-06
3e-06
4e-06
5e-06
0e+00
2e+05
4e+05
6e+05
8e+05
1e+06
Balance Limit

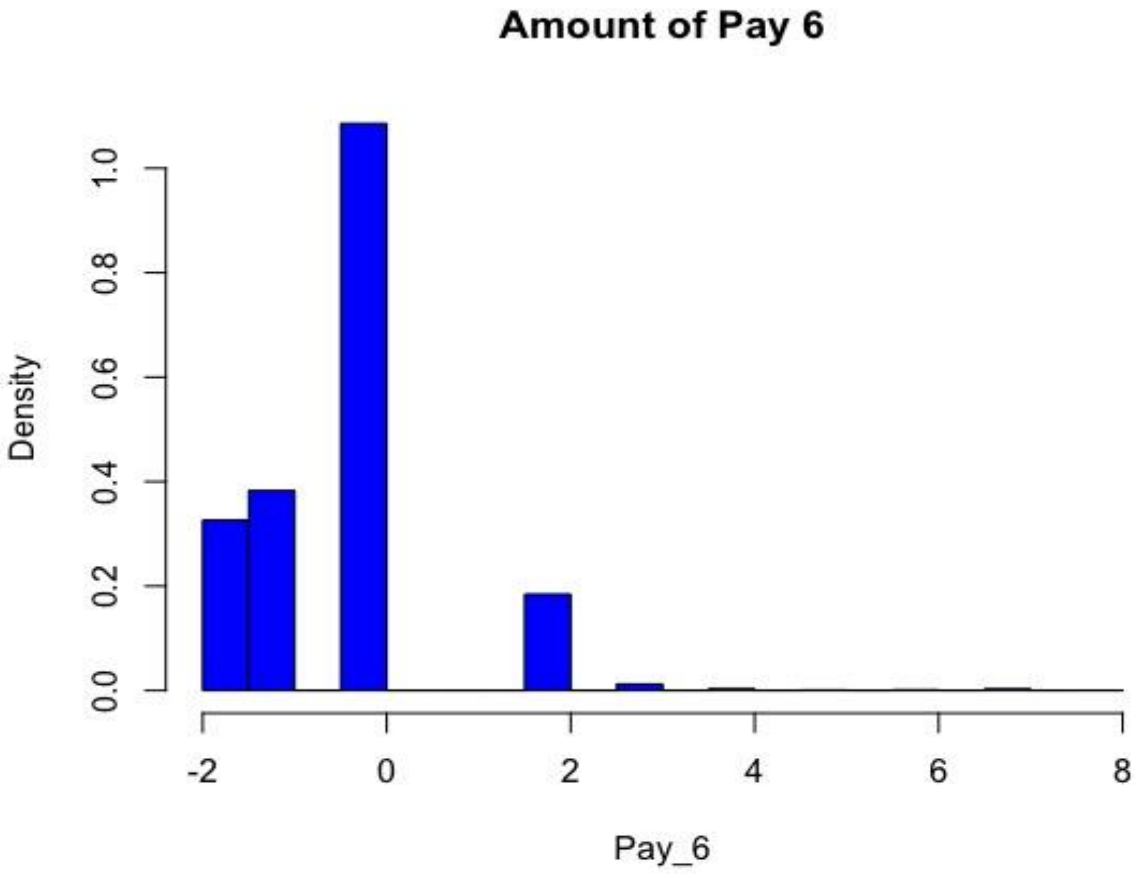

Amount of Pay 0
Density
0.0
0.2
0.4
0.6
0.8
1.0
-2
0
2
4
6
8
Pay_0

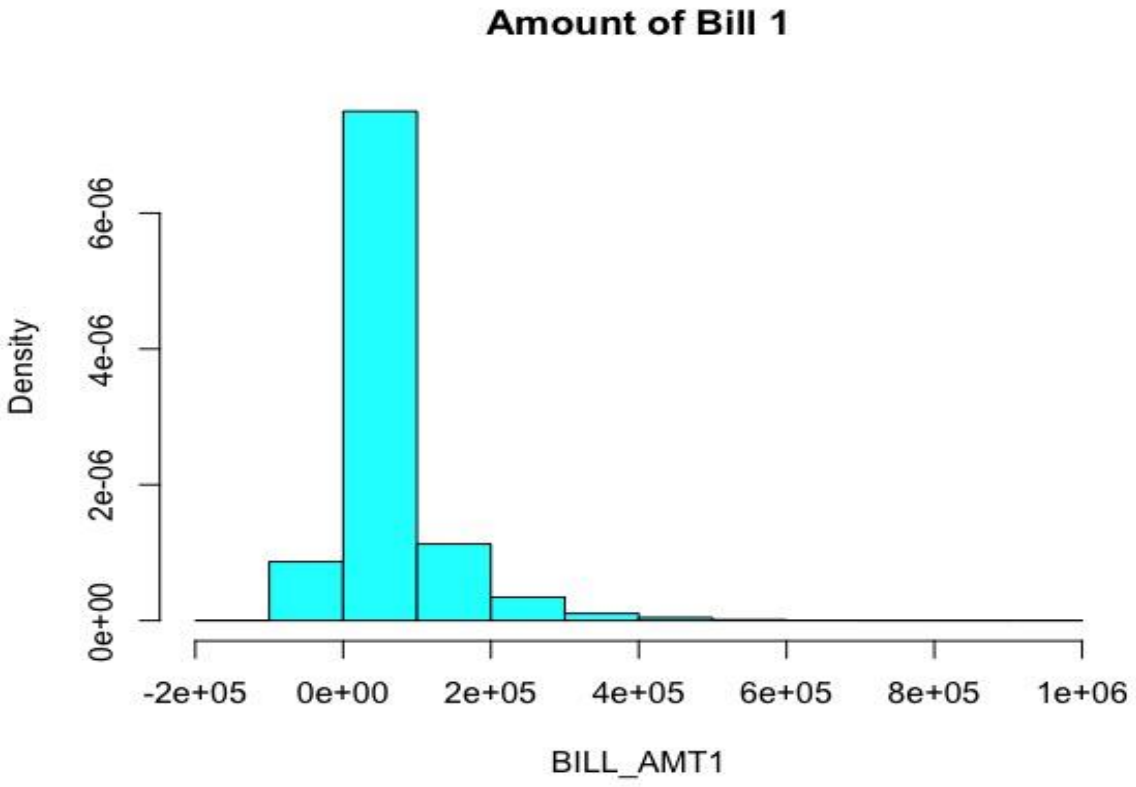

LAmount of Pay 2
Density
0.0
0.2
0.4
0.6
0.8
1.0
-2
0
2
4
6
8
Pay_2

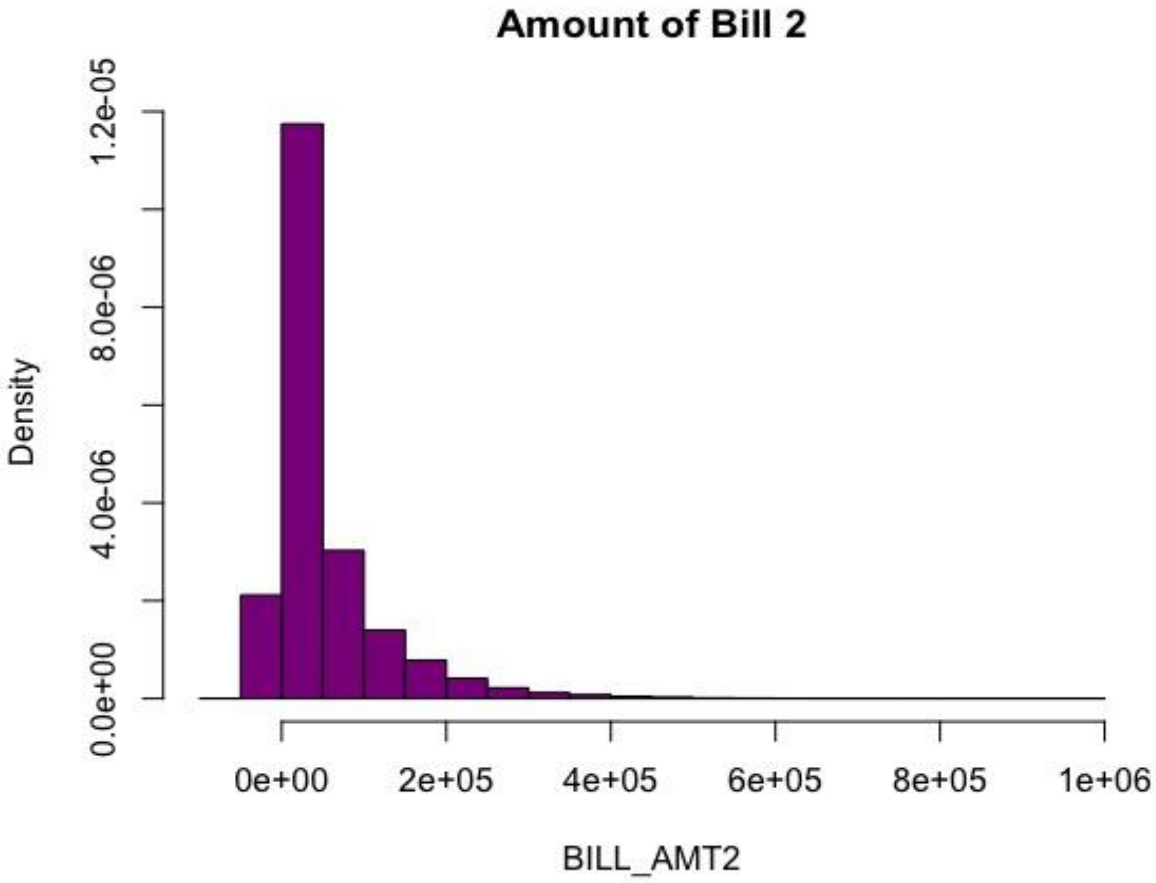

Amount of Pay 3
Density
0.0
0.2
0.4
0.6
0.8
1.0
-2
0
2
4
6
8
Pay_3

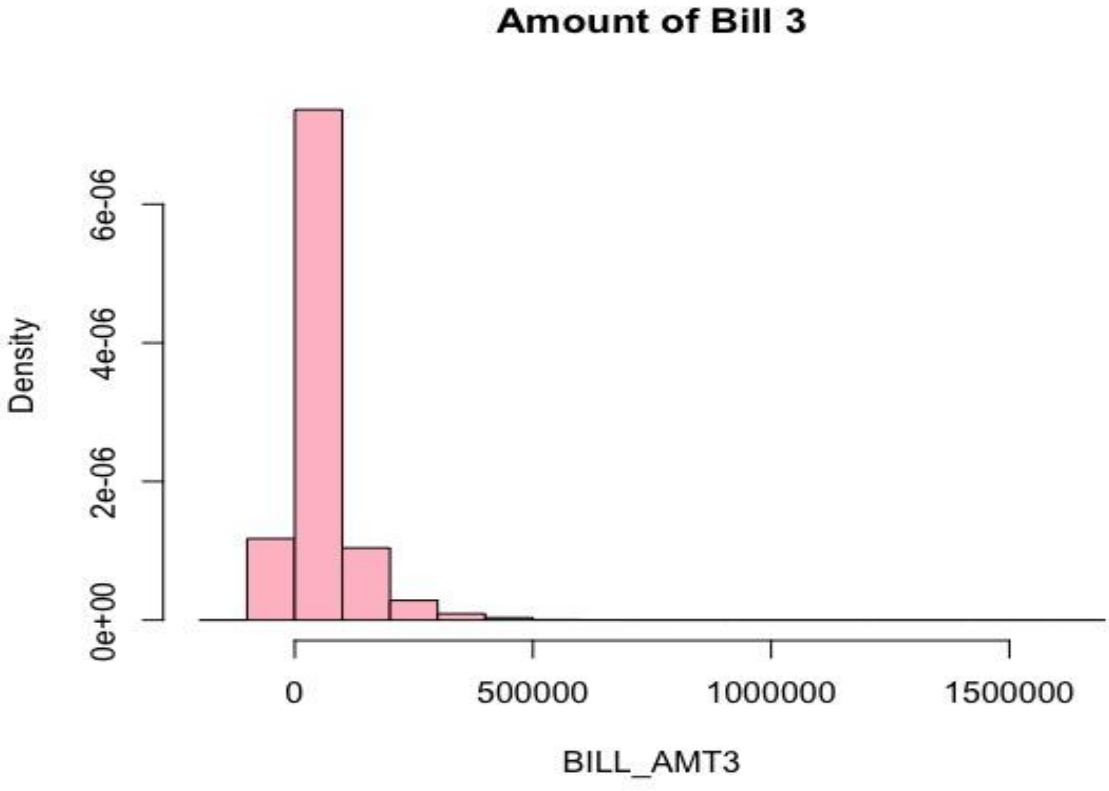

Amount of Pay 4
Density
0.0
0.2
0.4
0.6
0.8
1.0
-2
0
2
4
6
8
Pay_4

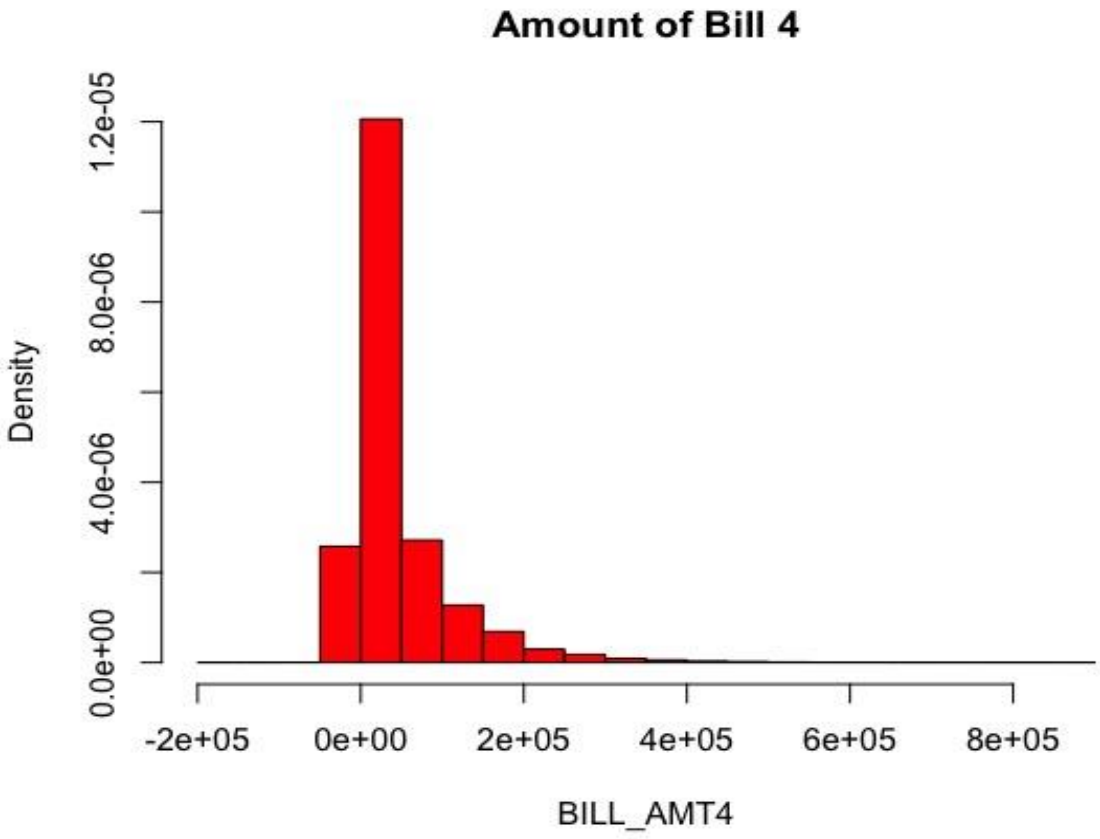

Amount of Pay 5
Density
0.0
0.2
0.4
0.6
0.8
1.0
-2
0
2
4
6
8
Pay_5

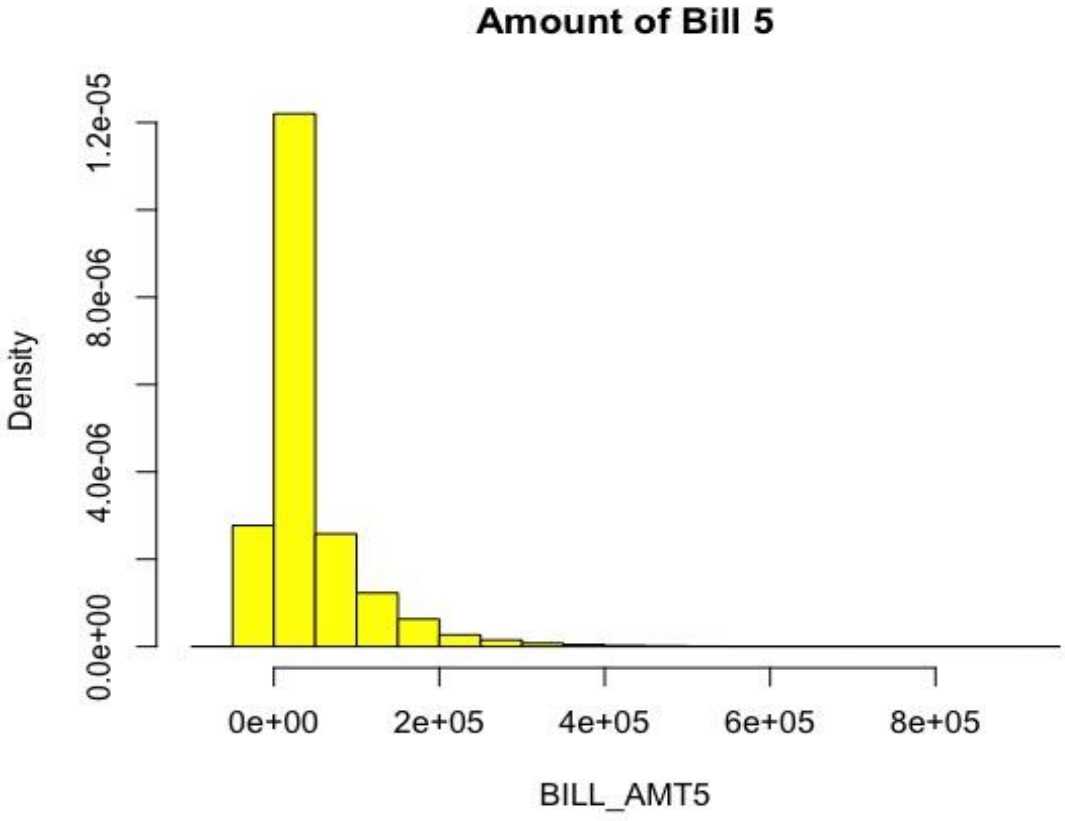

Amount of Pay 6
Density
0.0
0.2
0.4
0.6
0.8
1.0
-2
0
2
4
6
8
Pay_6

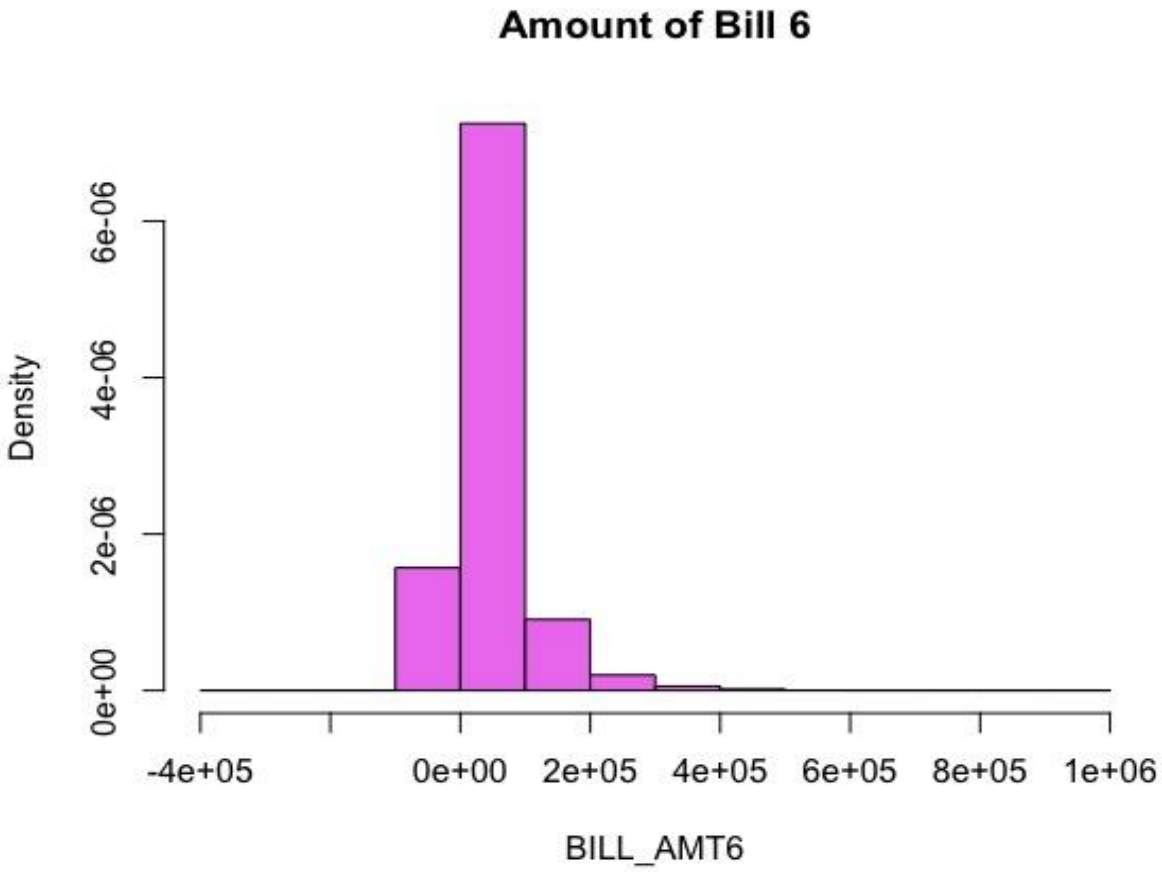

Amount of Bill 1
Density
0e+00
2e-06
4e-06
6e-06
-2e+05
0e+00
2e+05
4e+05
6e+05
8e+05
1e+06
BILL_AMT1

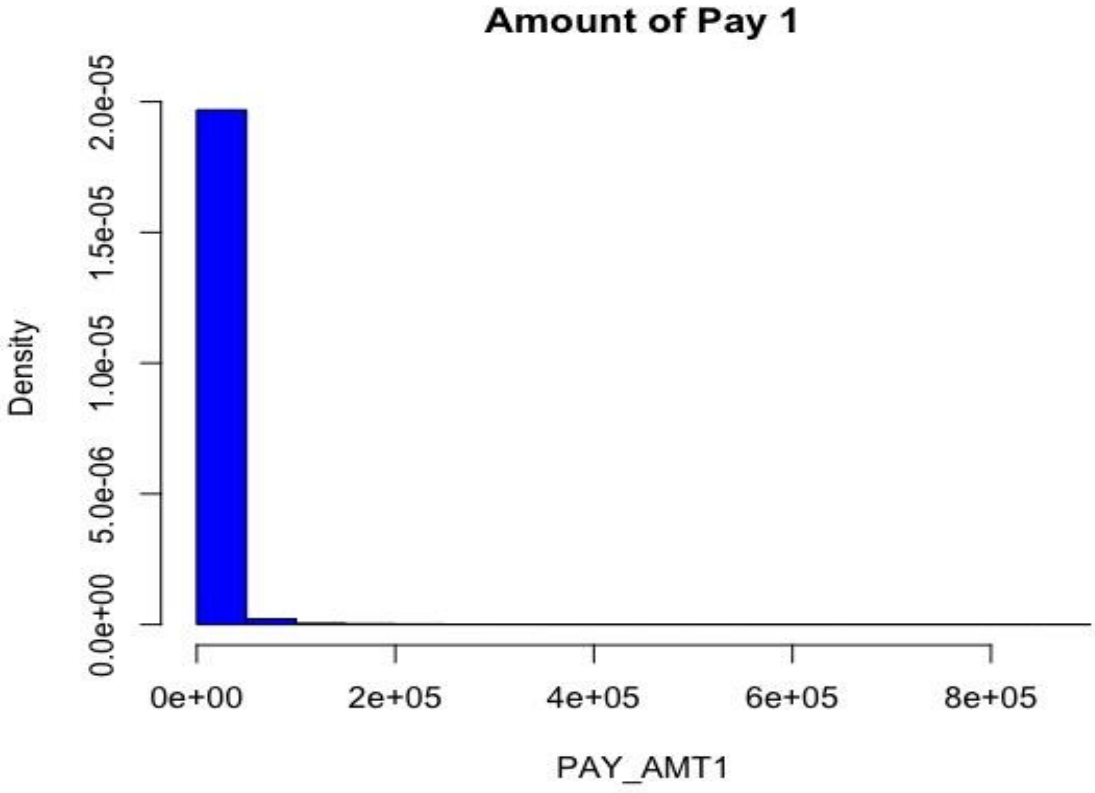

Amount of Bill 2
Density
0.0e+00
4.0e-06
8.0e-06
1.2e-05
0e+00
2e+05
4e+05
6e+05
8e+05
1e+06
BILL_AMT2

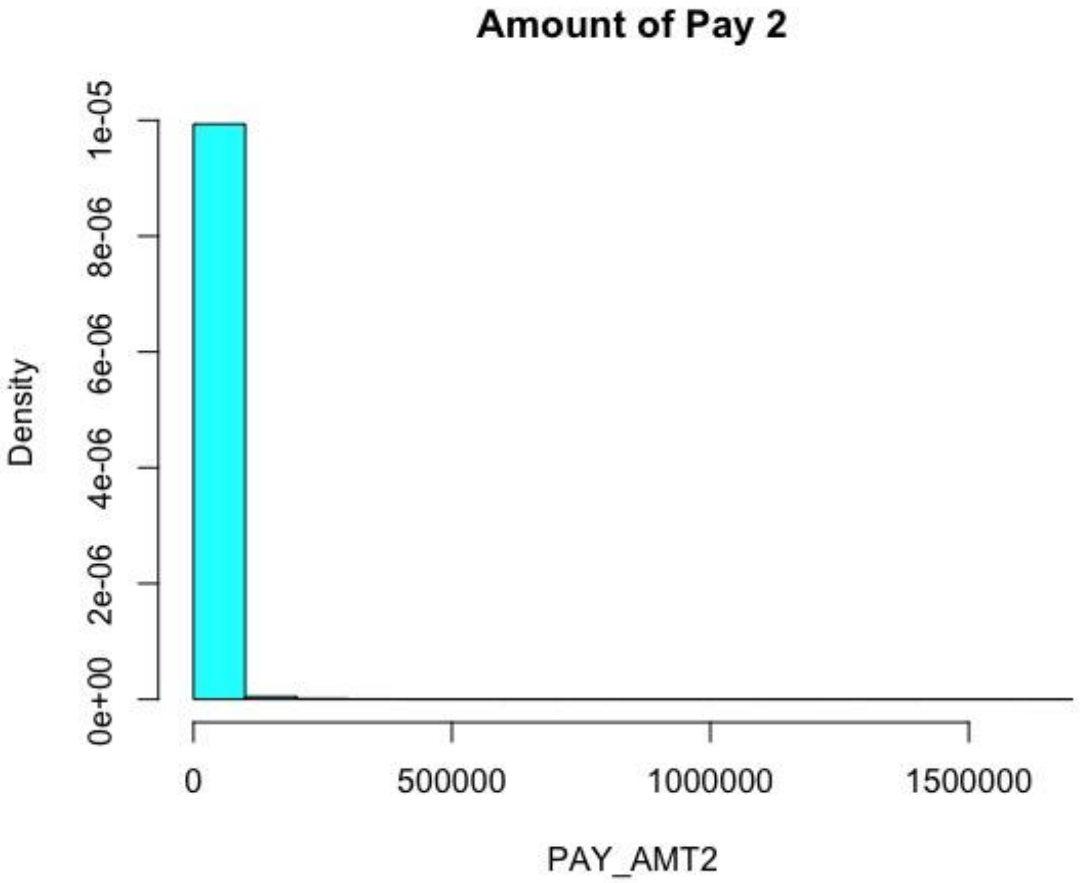

Amount of Bill 3
Density
0e+00
2e-06
4e-06
6e-06
0
500000
1000000
1500000
BILL_AMT3

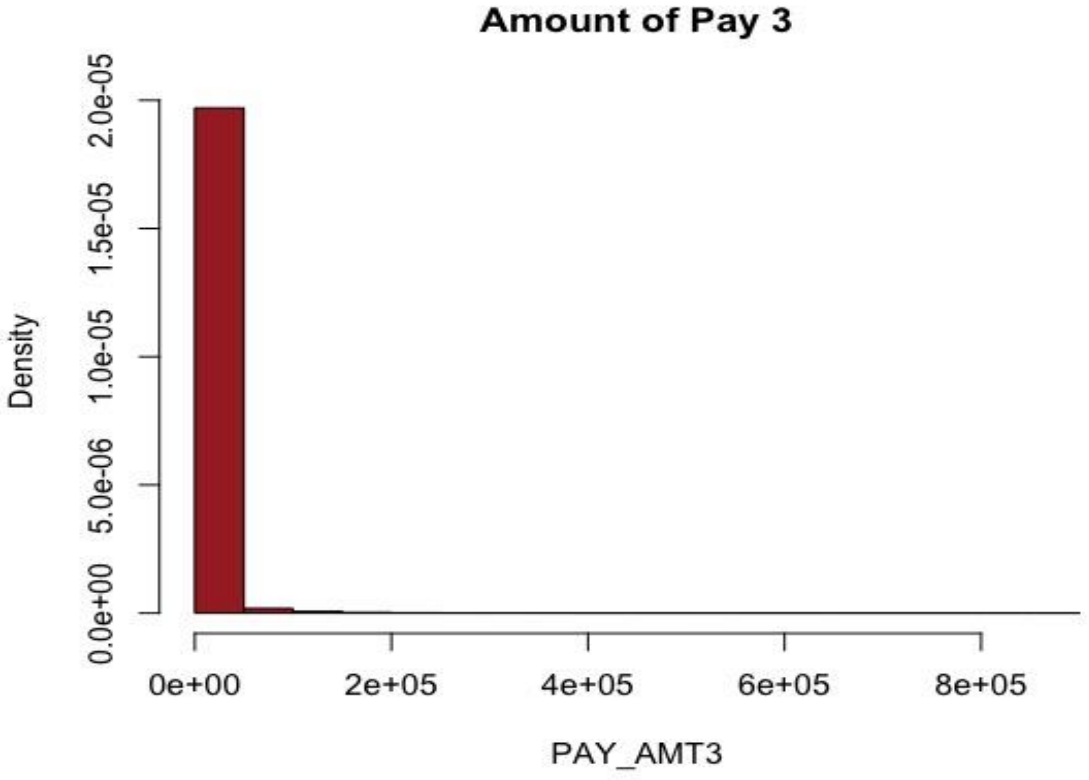

Amount of Bill 4
Density
0.0e+00
4.0e-06
8.0e-06
1.2e-05
-2e+05
0e+00
2e+05
4e+05
6e+05
8e+05
BILL_AMT4

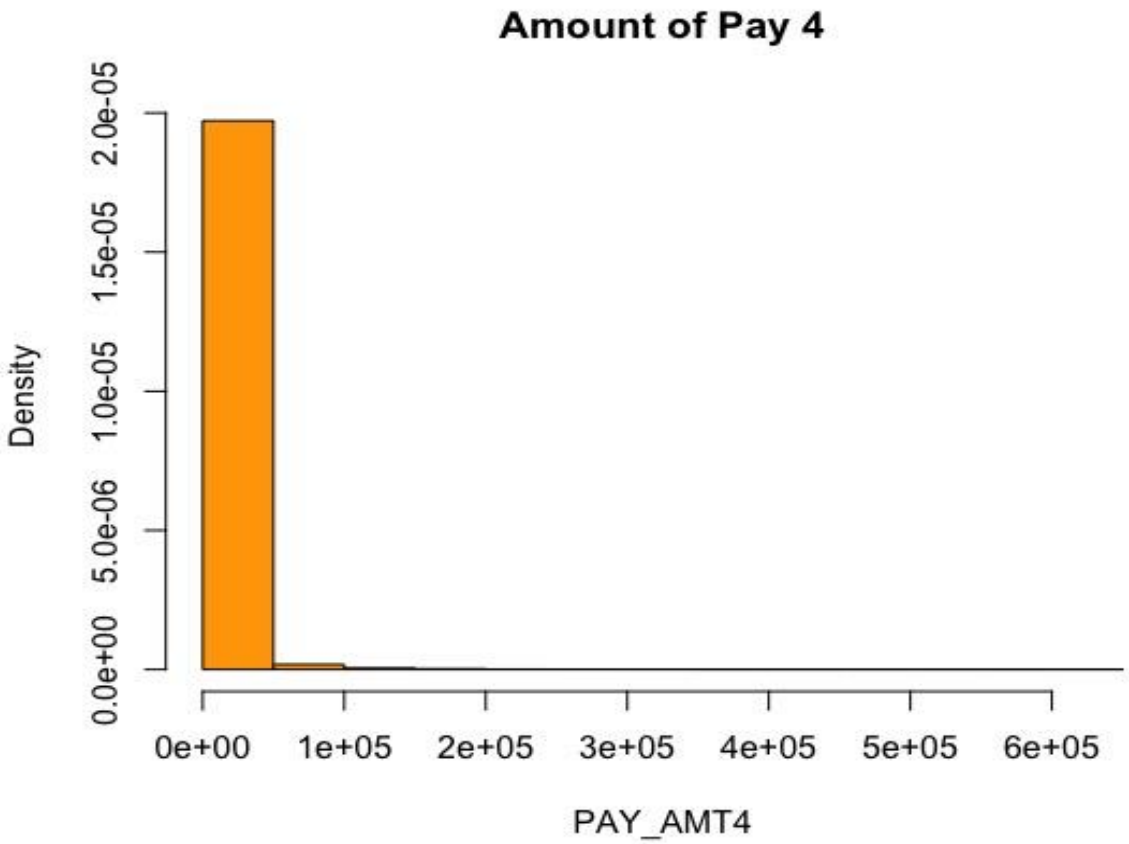

Amount of Bill 5
Density
0.0e+00
4.0e-06
8.0e-06
1.2e-05
0e+00
2e+05
4e+05
6e+05
8e+05
BILL_AMT5

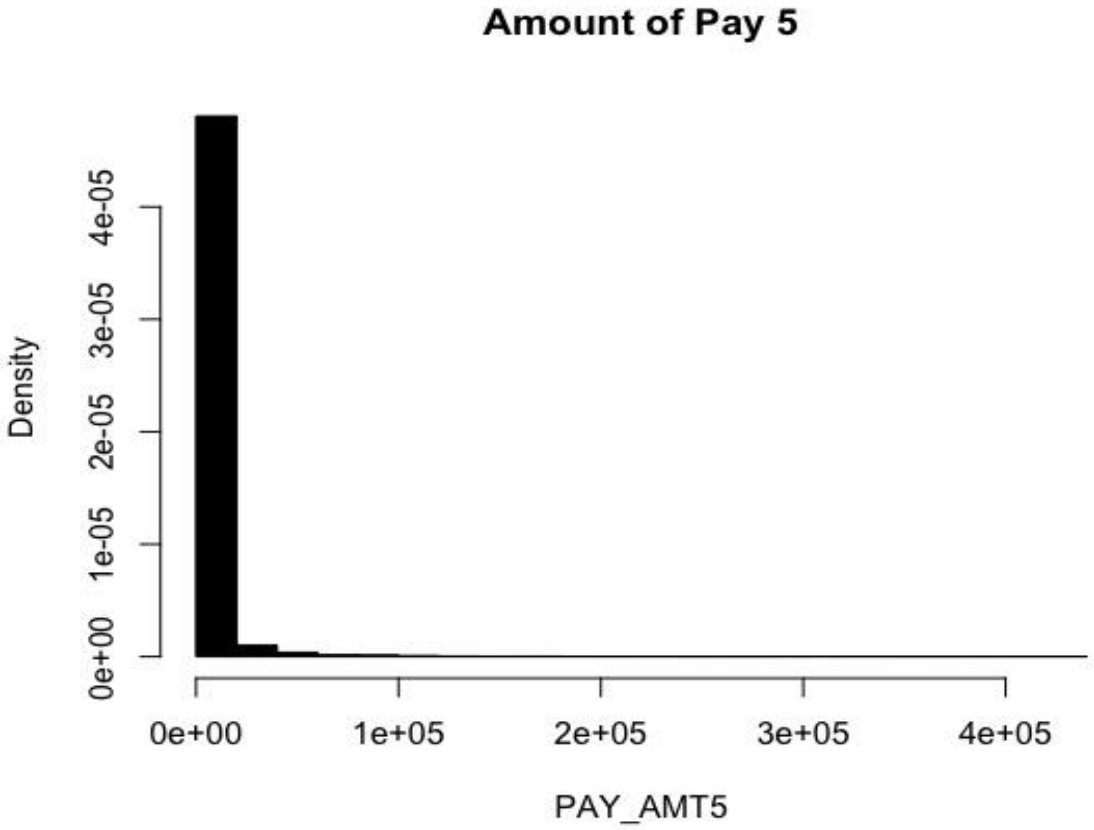

Amount of Bill 6
Density
0e+00
2e-06
4e-06
6e-06
-4e+05
0e+00
2e+05
4e+05
6e+05
8e+05
1e+06
BILL_AMT6

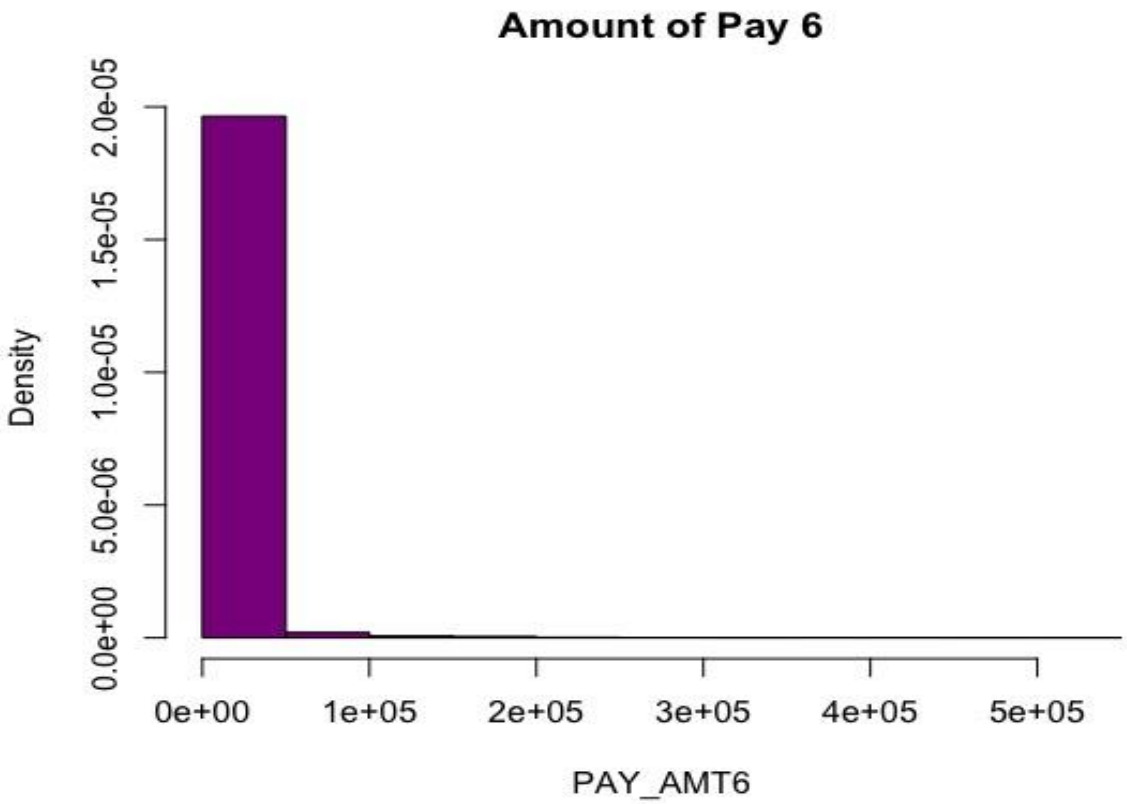

Amount of Pay 1
Density
0.0e+00
5.0e-06
1.0e-05
1.5e-05
2.0e-05
0e+00
2e+05
4e+05
6e+05
8e+05
PAY_AMT1

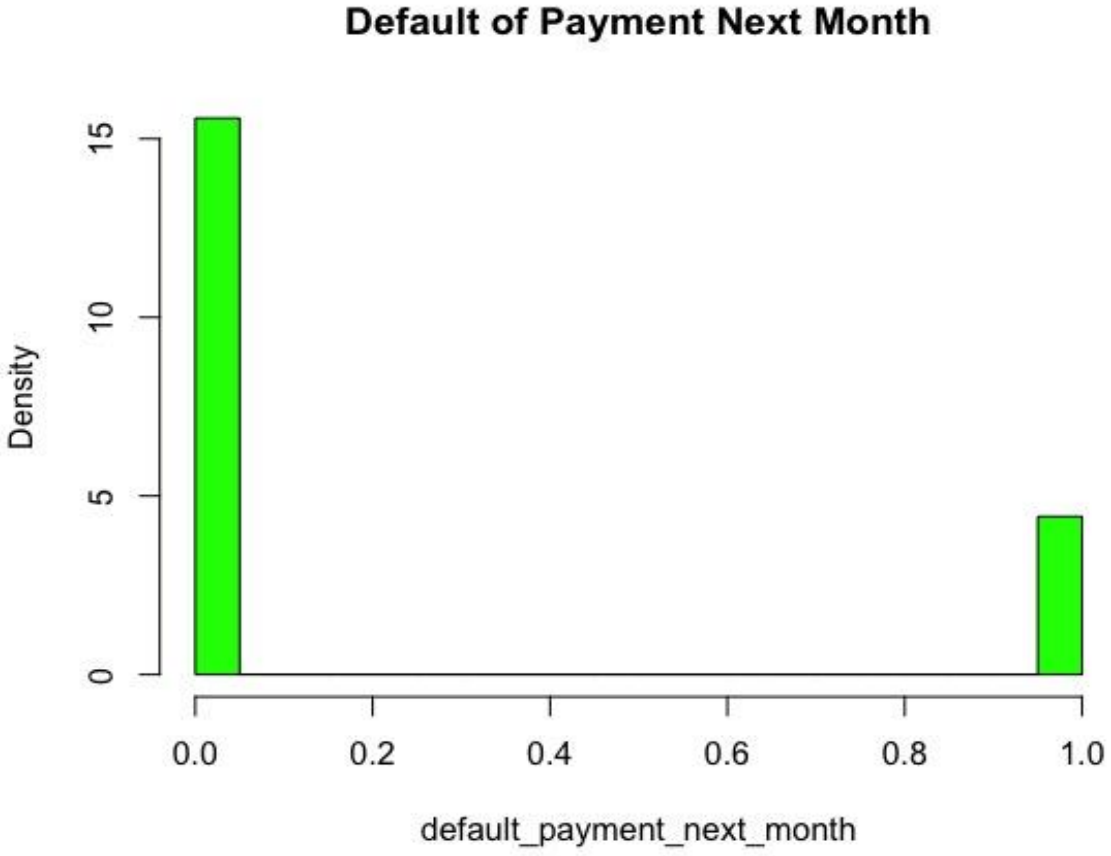

Amount of Pay 2
Density
0e+00
2e-06
4e-06
6e-06
8e-06
1e-05
0
500000
1000000
1500000
PAY_AMT2


Amount of Pay 3
Density
0.0e+00
5.0e-06
1.0e-05
1.5e-05
2.0e-05
0e+00
2e+05
4e+05
6e+05
8e+05
PAY_AMT3

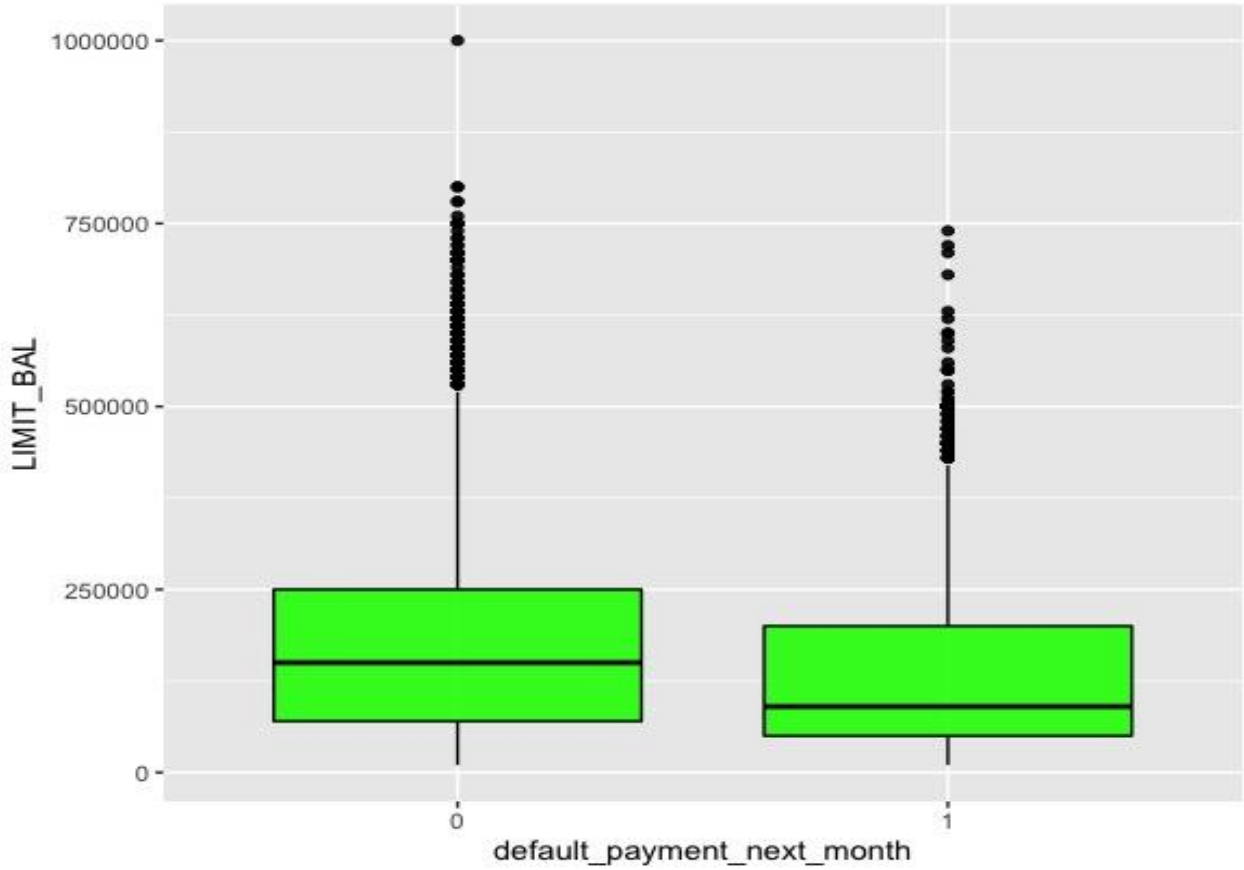

Amount of Pay 4
Density
0.0e+00
5.0e-06
1.0e-05
1.5e-05
2.0e-05
0e+00
1e+05
2e+05
3e+05
4e+05
5e+05
6e+05
PAY_AMT4

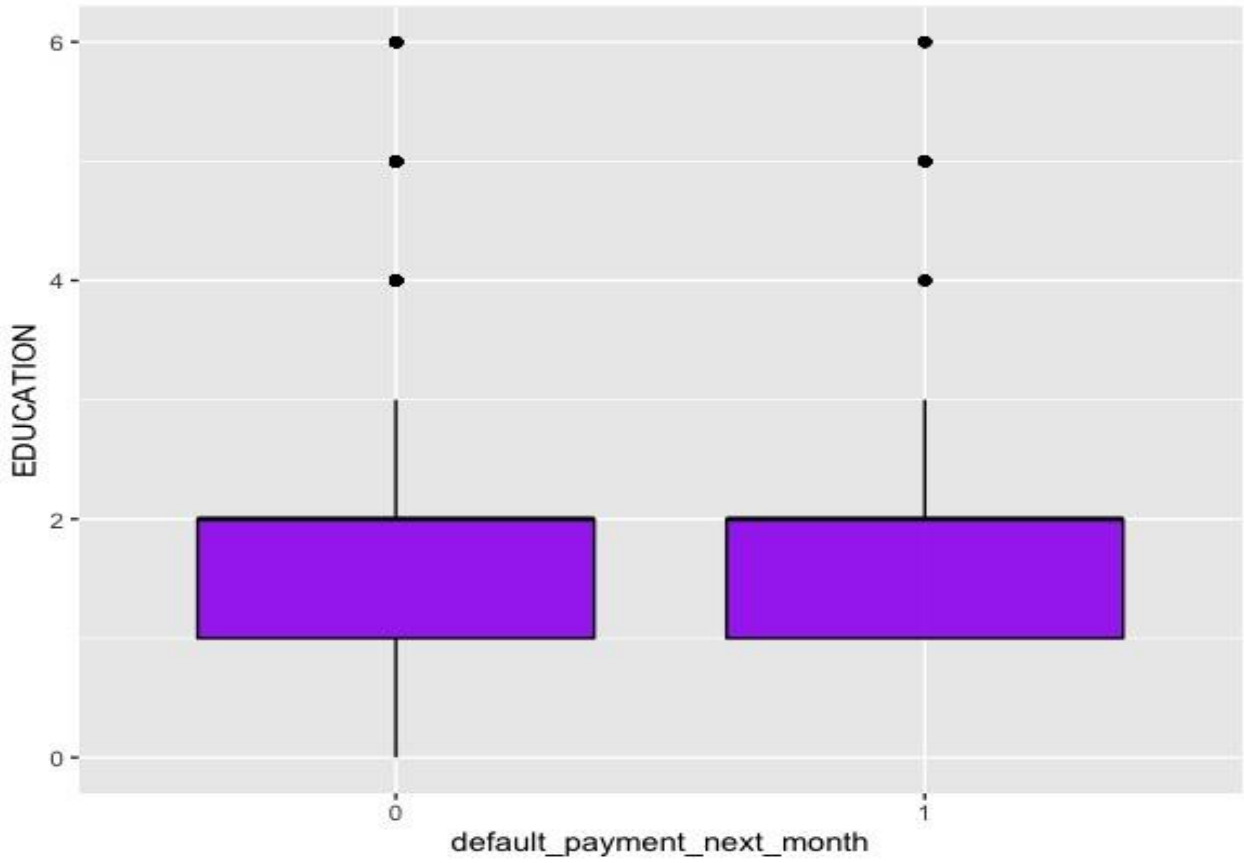

Amount of Pay 5
Density
0e+00
1e-05
2e-05
3e-05
4e-05
0e+00
1e+05
2e+05
3e+05
4e+05
PAY_AMT5

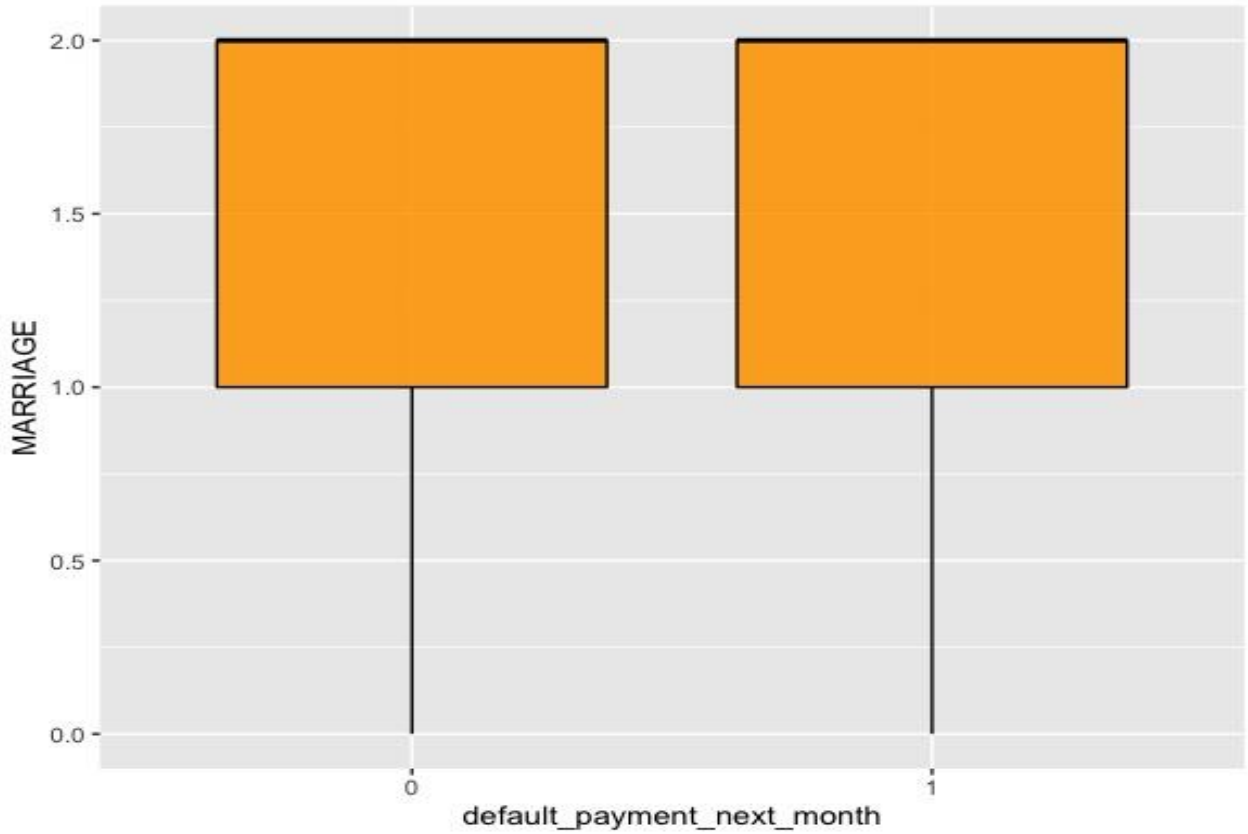

Amount of Pay 6
Density
0.0e+00
5.0e-06
1.0e-05
1.5e-05
2.0e-05
0e+00
1e+05
2e+05
3e+05
4e+05
5e+05
PAY_AMT6

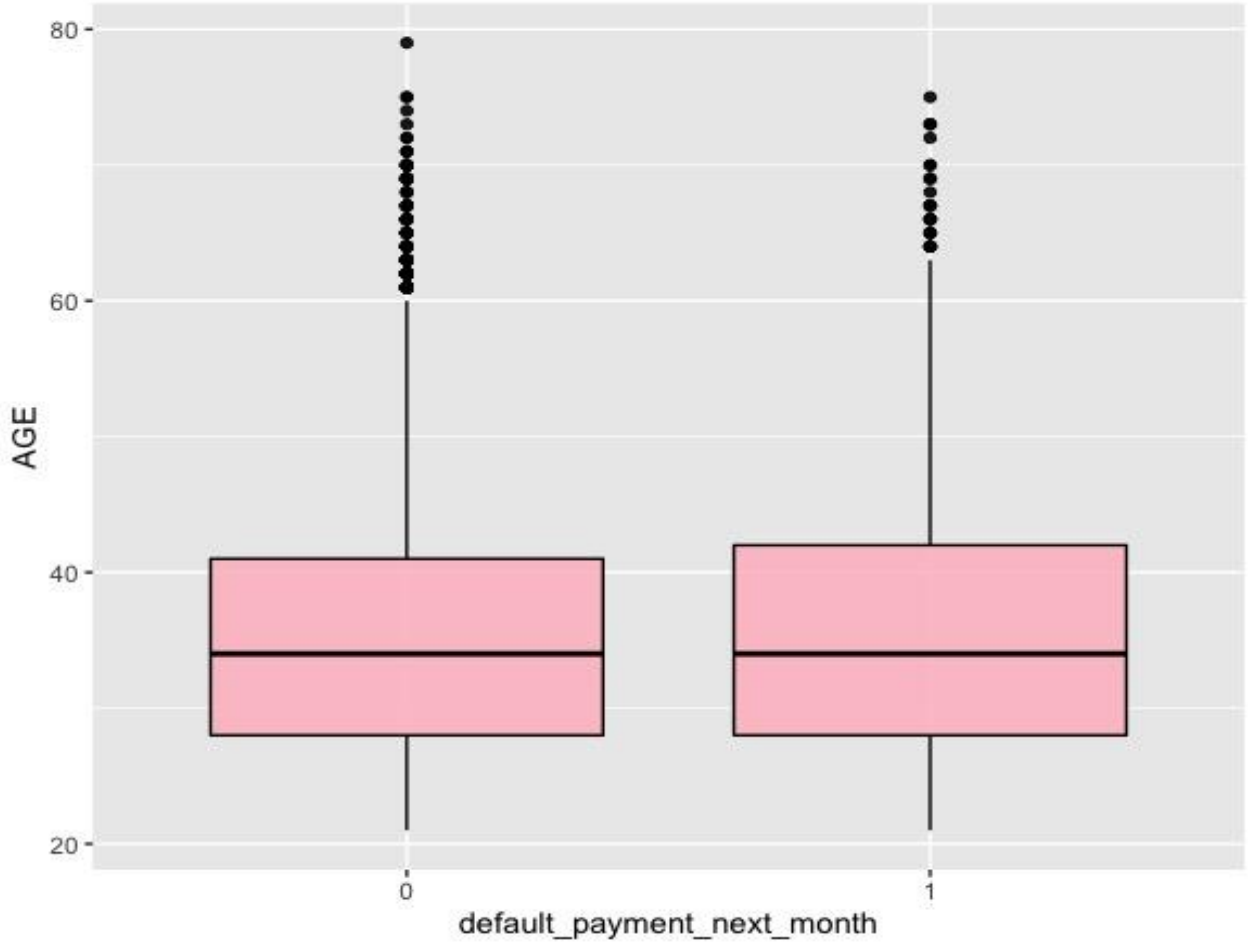

Default of Payment Next Month
Density
0
5
10
15
0.0
0.2
0.4
0.6
0.8
1.0
default_payment_next_month

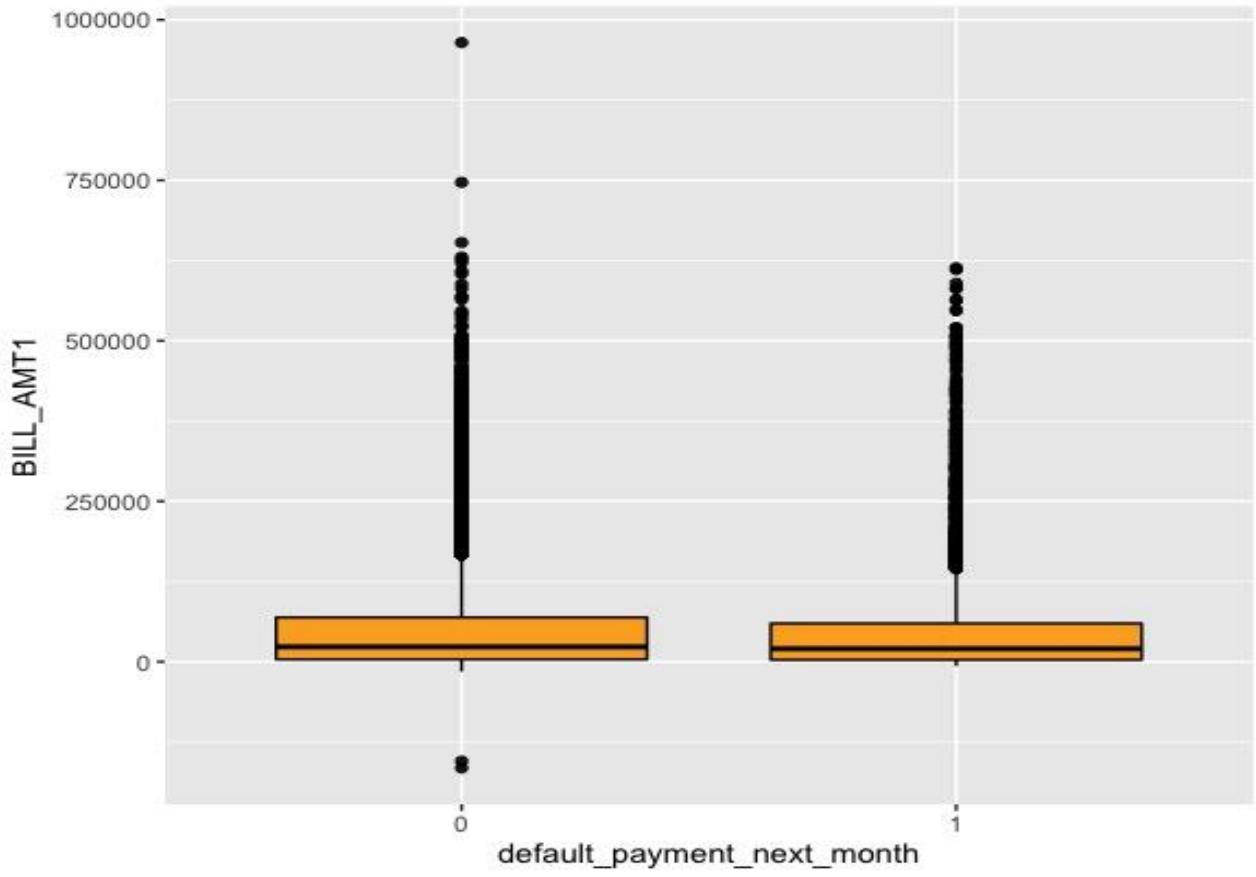
Box Plot:

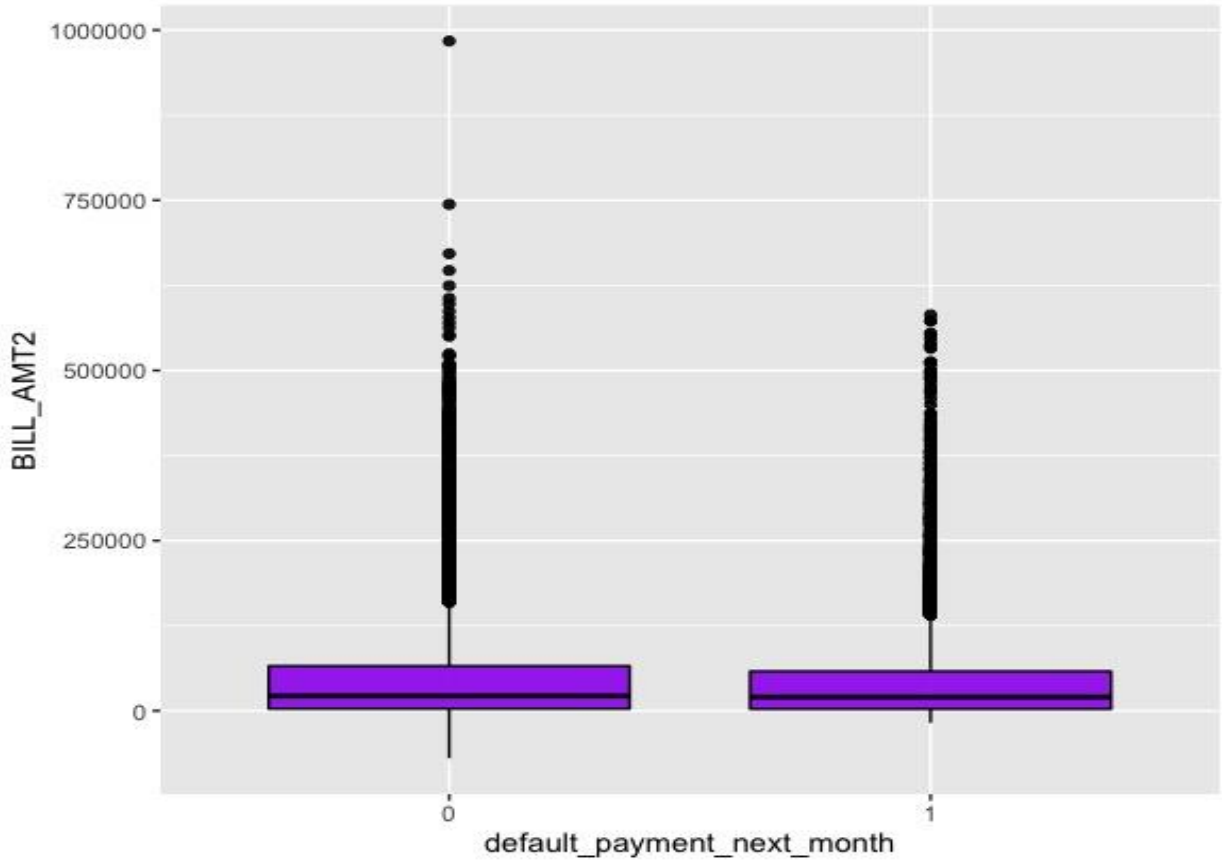

1000000
750000
500000
250000
0
LIMIT_BAL
0
1
default_payment_next_month

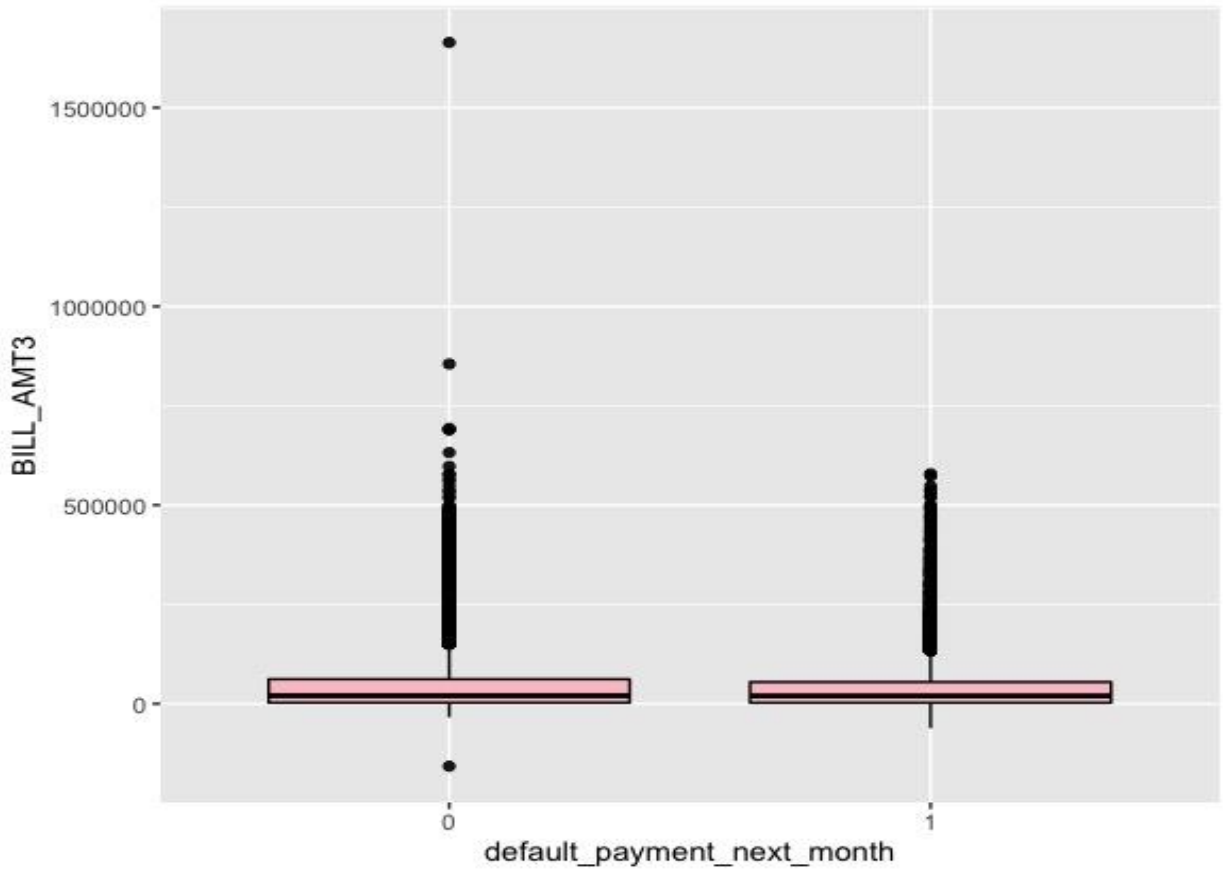

6
4
2
0
EDUCATION
0
1
default_payment_next_month

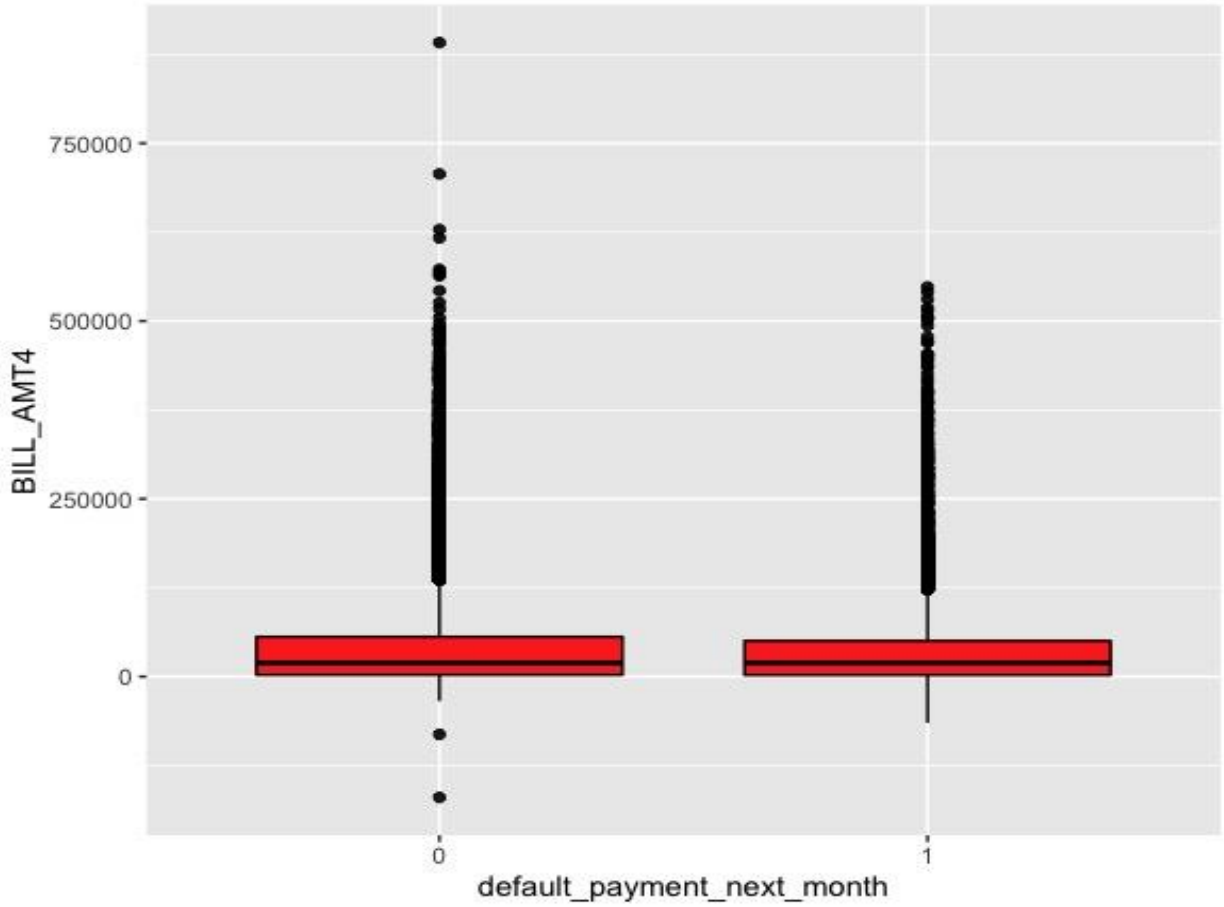

2.0
1.5
1.0
0.5
0.0
MARRIAGE
0
1
default_payment_next_month

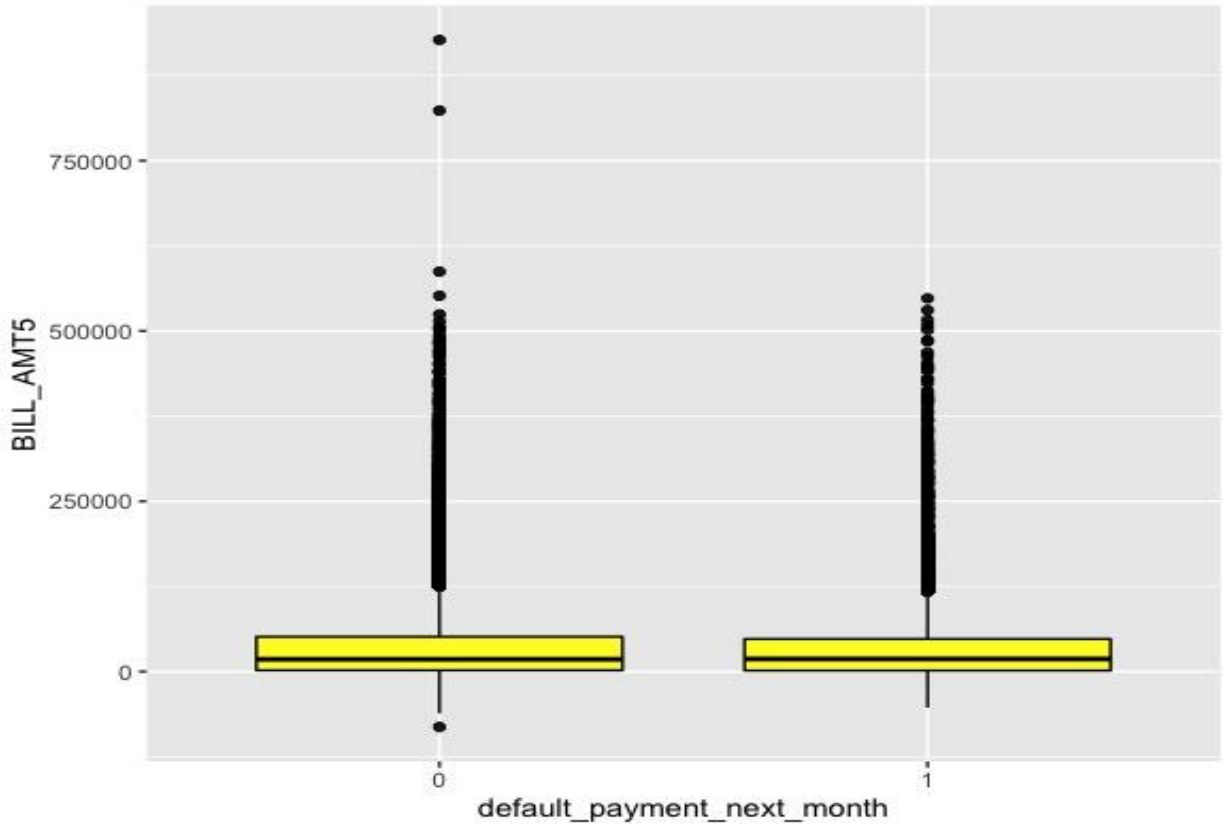

80
60
40
20
AGE
0
1
default_payment_next_month

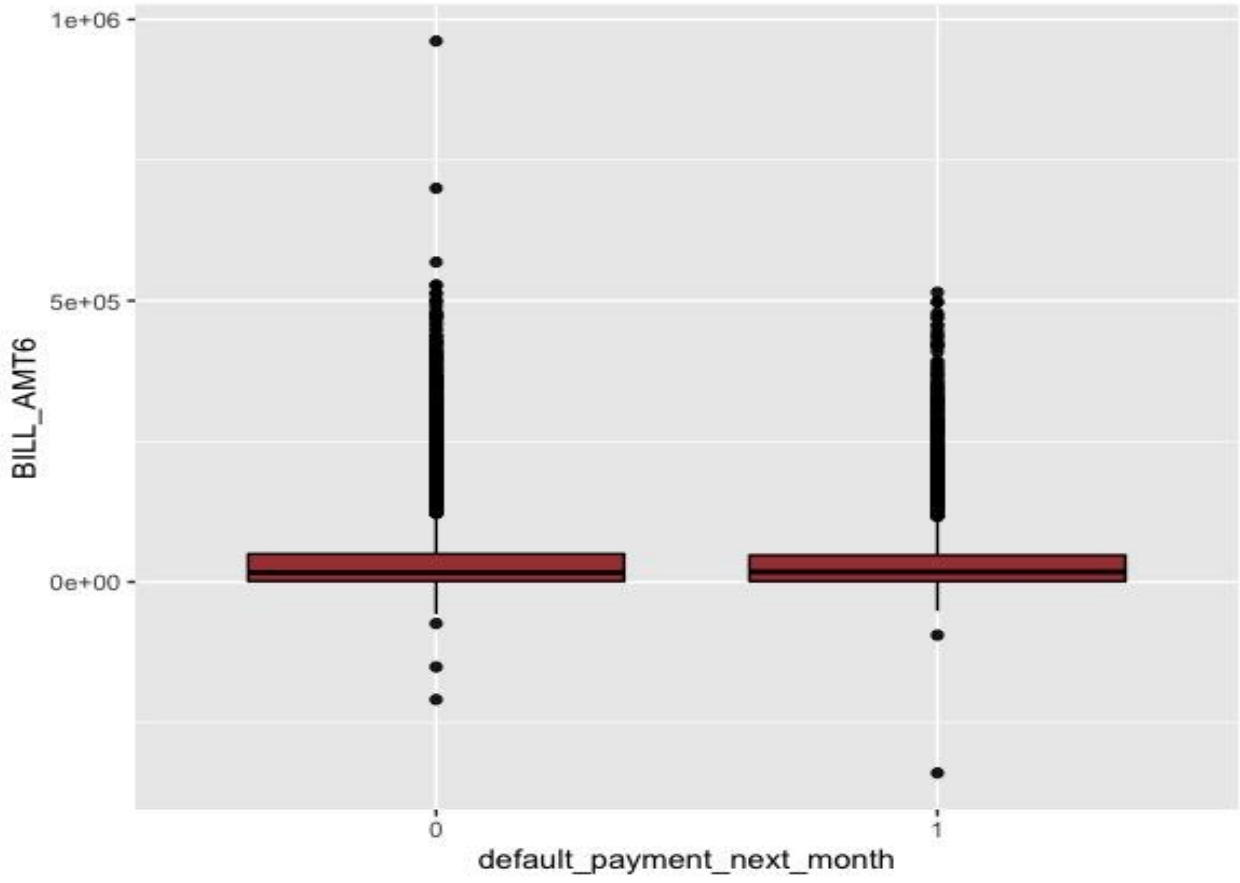

1000000
750000
500000
250000
0
BILL_AMT1
0
1
default_payment_next_month

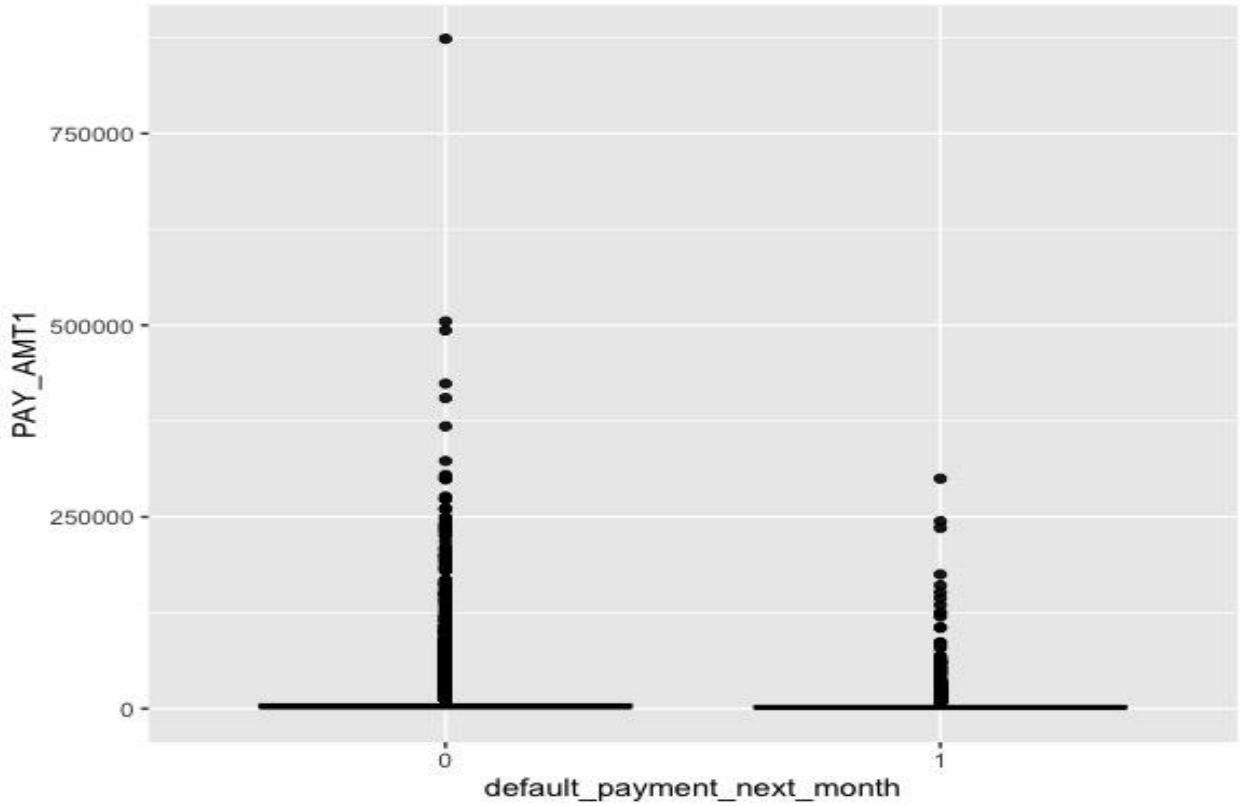

1000000
750000
500000
250000
0
BILL_AMT2
0
1
default_payment_next_month

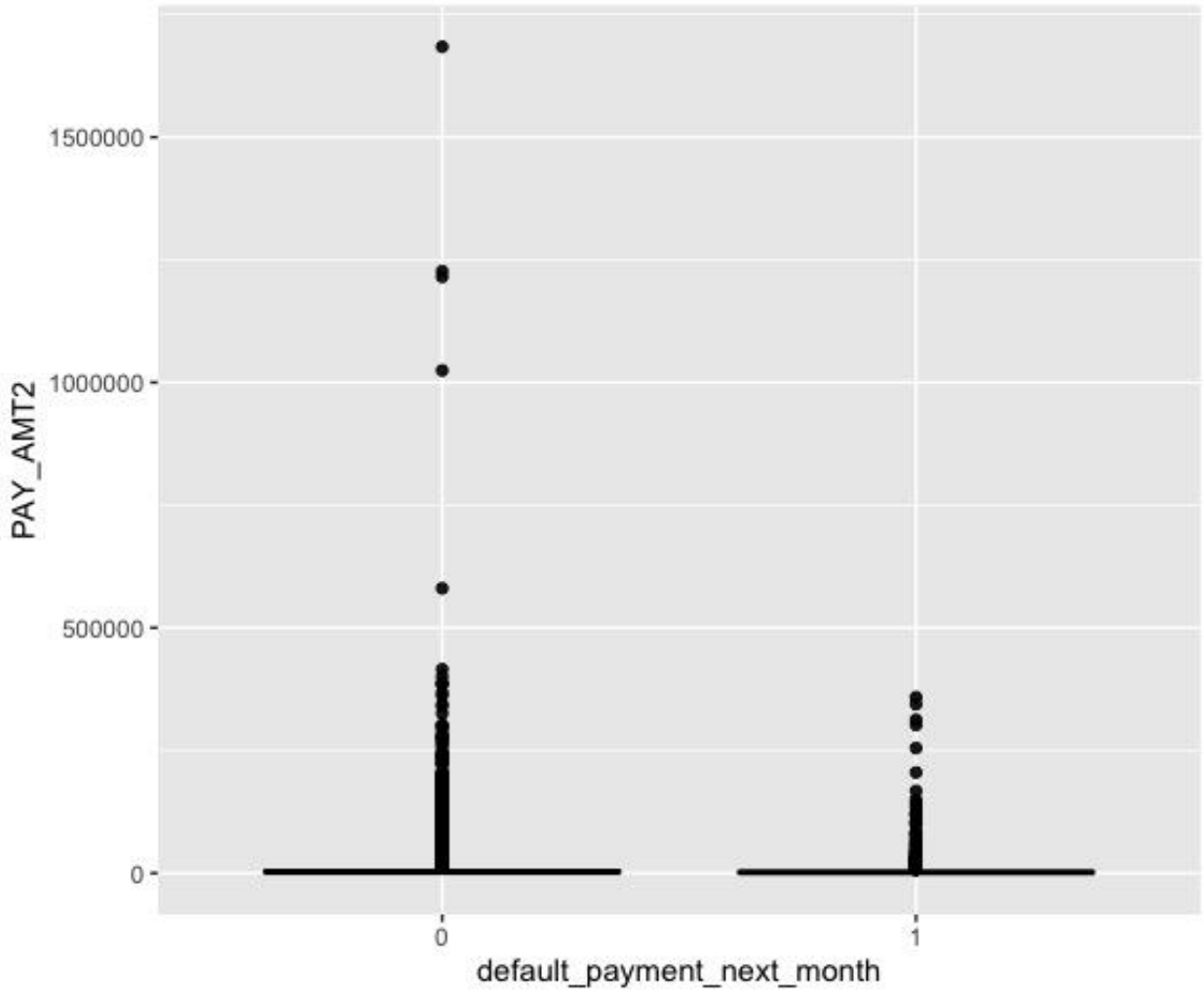

1500000
1000000
500000
0
BILL_AMT3
0
1
default_payment_next_month

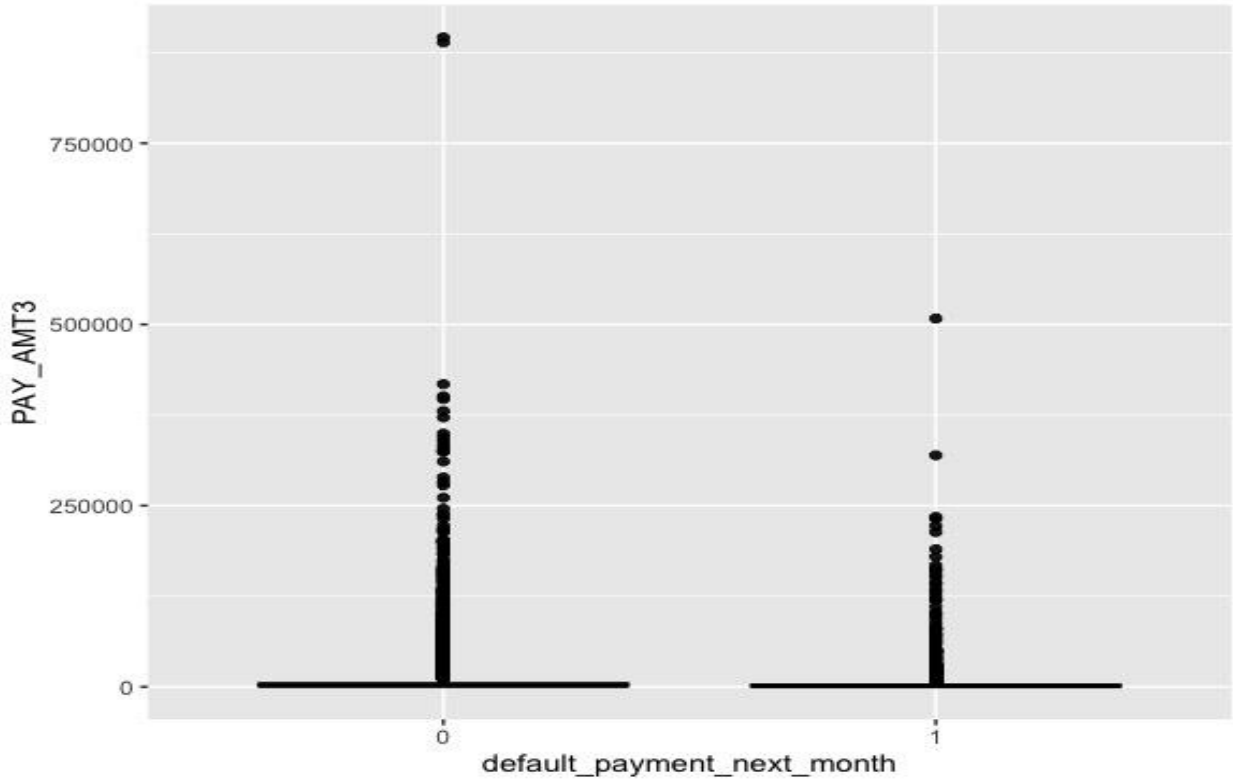

750000
500000
250000
0
BILL_AMT4
0
1
default_payment_next_month

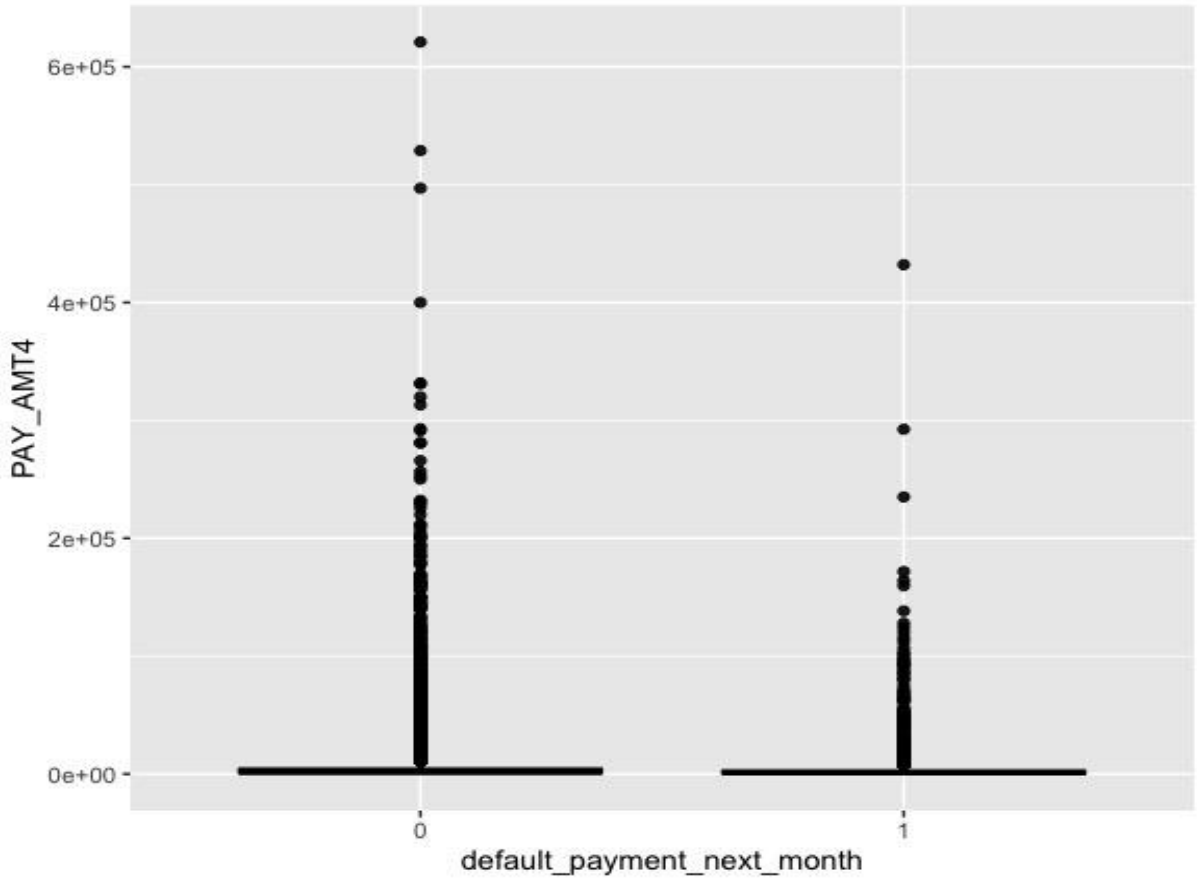

750000
500000
250000
0
BILL_AMT5
0
1
default_payment_next_month

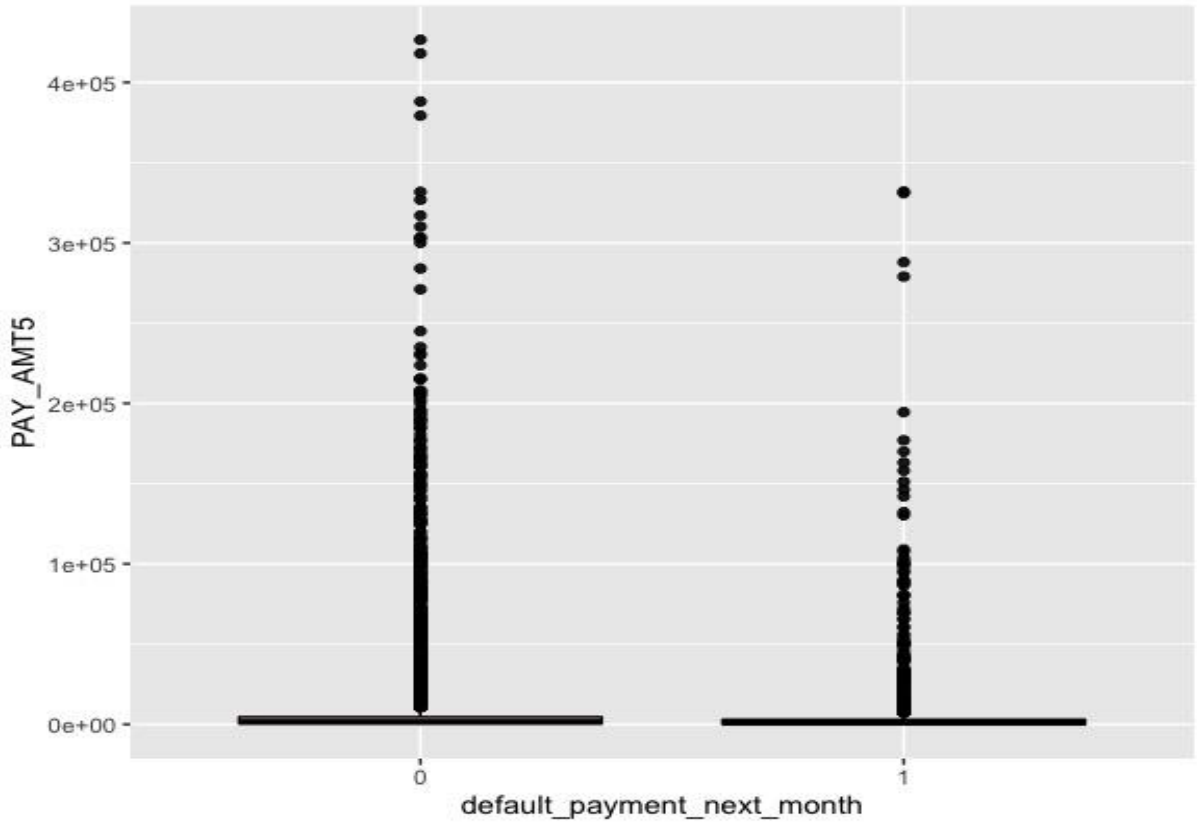

1e+06
5e+05
0e+00
BILL_AMT6
0
1
default_payment_next_month

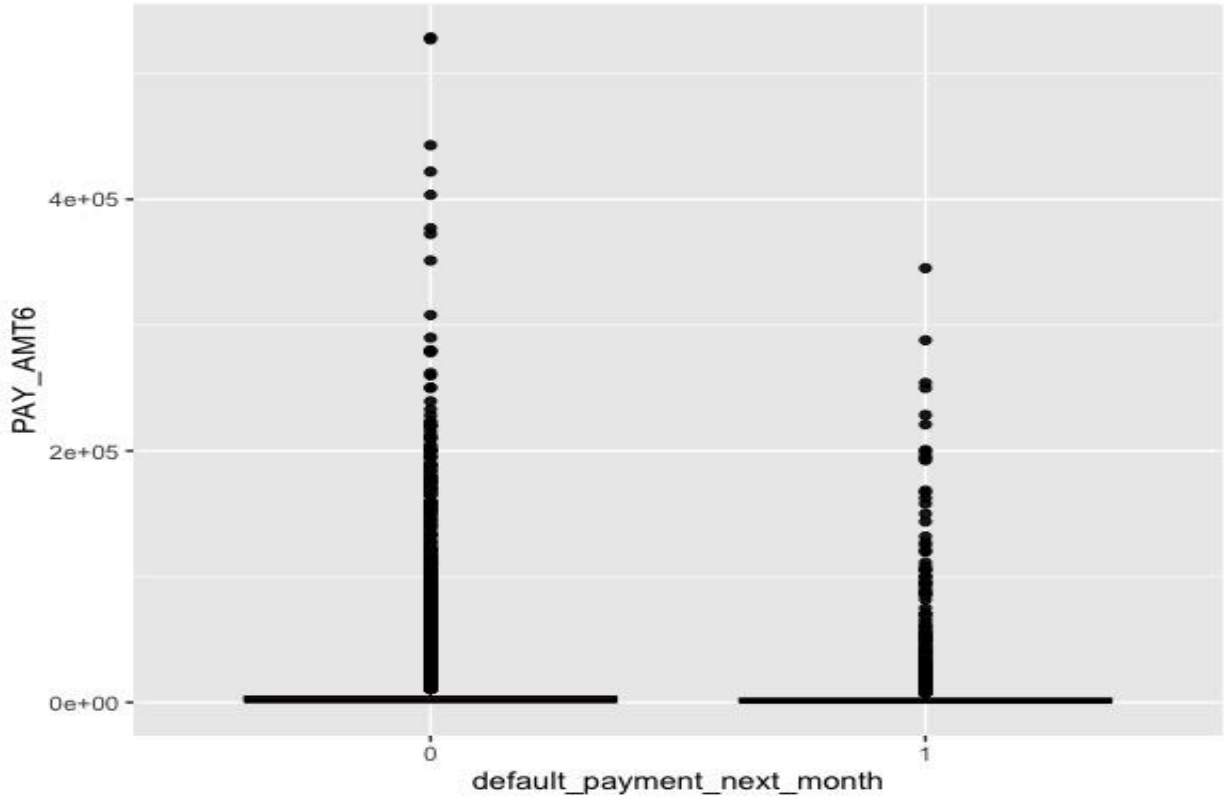

750000
500000
250000
0
PAY_AMT1
0
1
default_payment_next_month


1500000
1000000
500000
0
PAY_AMT2
0
1
default_payment_next_month

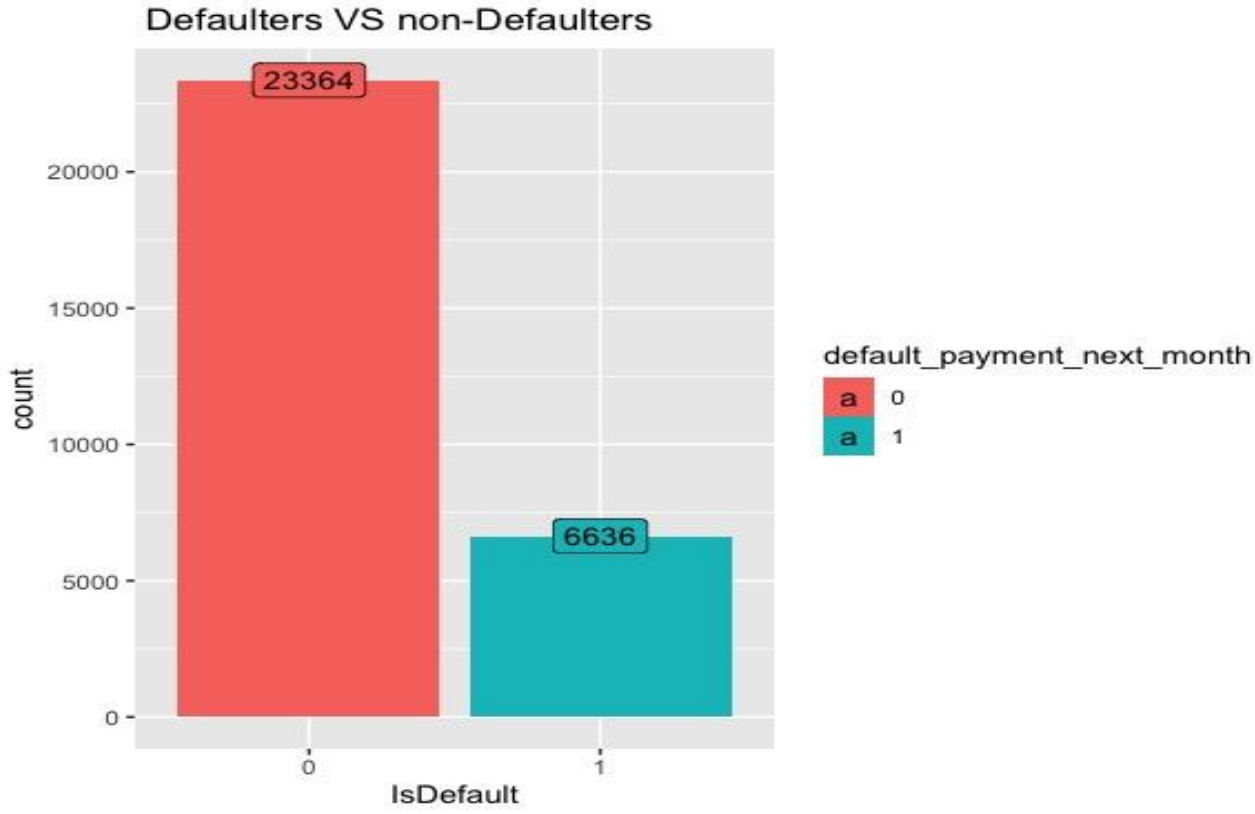

750000
500000
250000
0
PAY_AMT3
0
1
default_payment_next_month

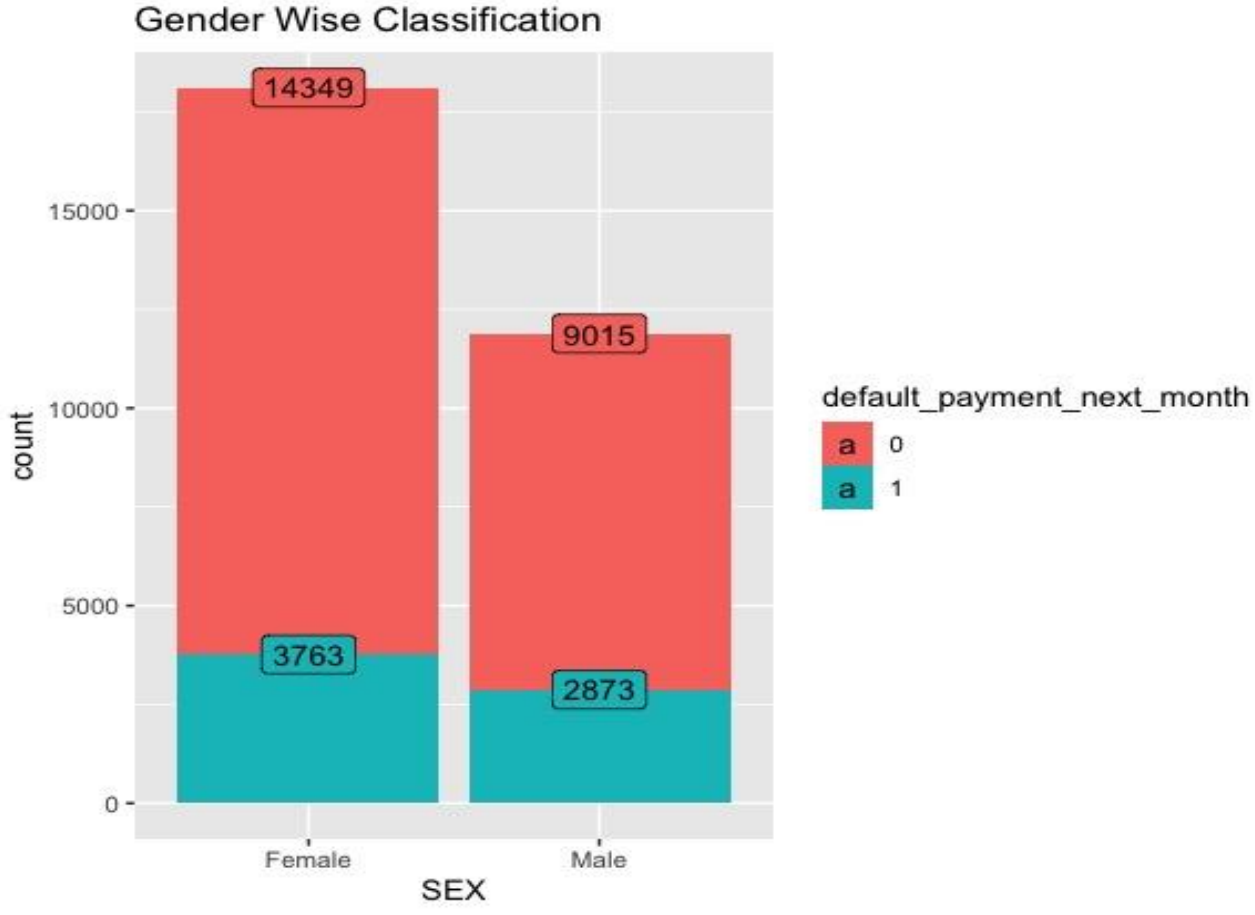

6e+05
4e+05
2e+05
0e+00
PAY_AMT4
0
1
default_payment_next_month

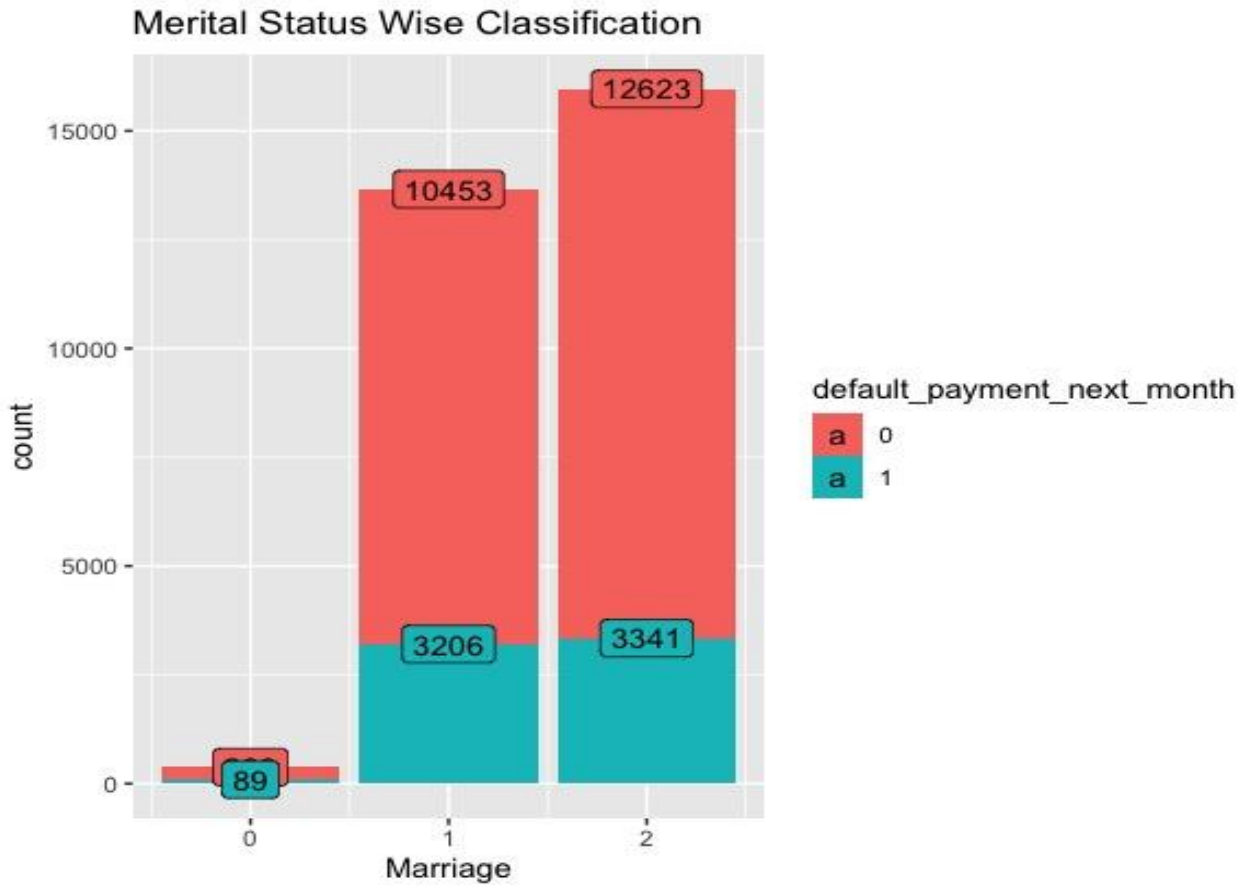

4e+05
3e+05
2e+05
1e+05
0e+00
PAY_AMT5
0
1
default_payment_next_month

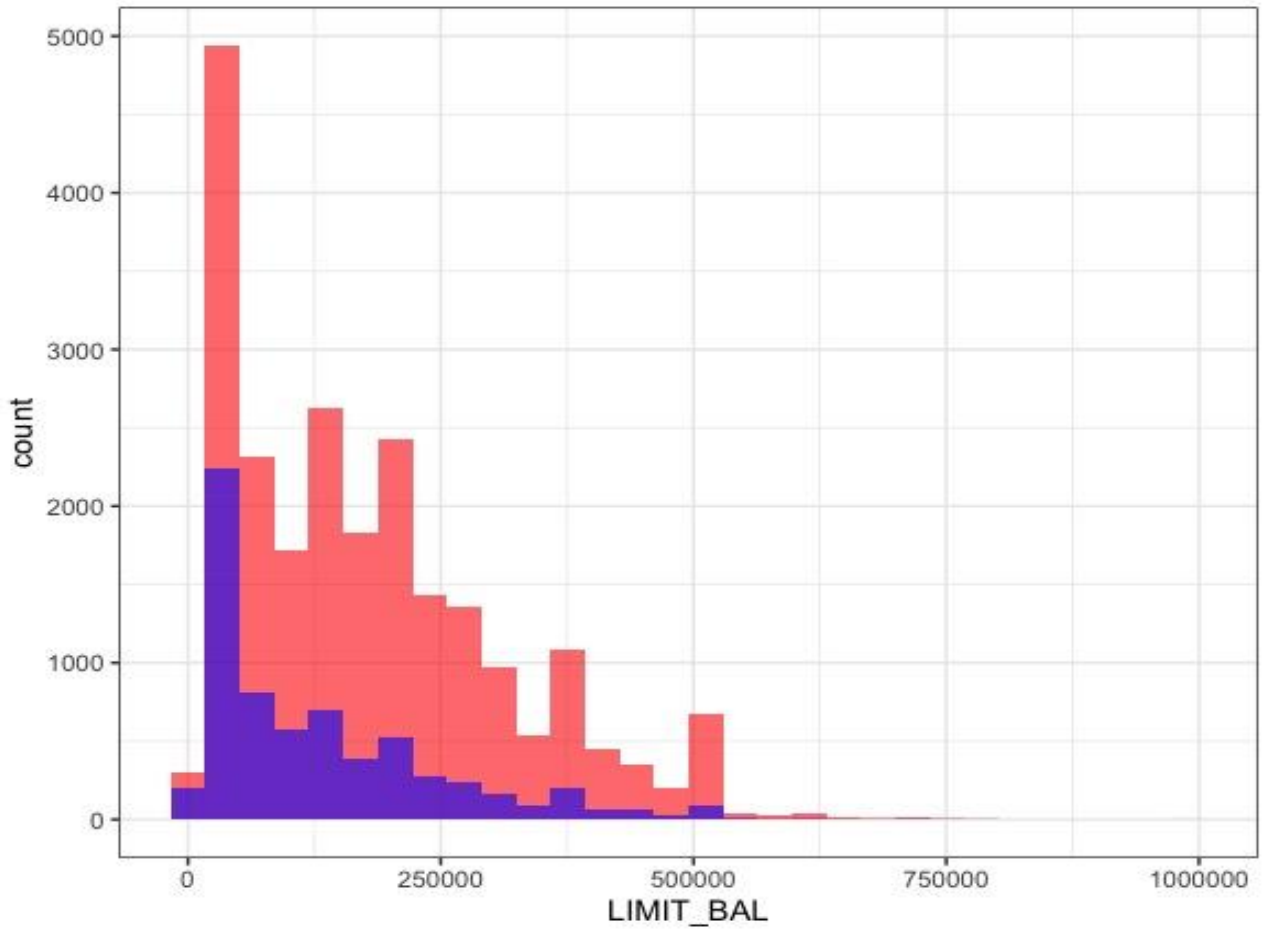

4e+05
2e+05
0e+00
PAY_AMT6
0
1
default_payment_next_month

Bar Plotting for Variables with Each other:

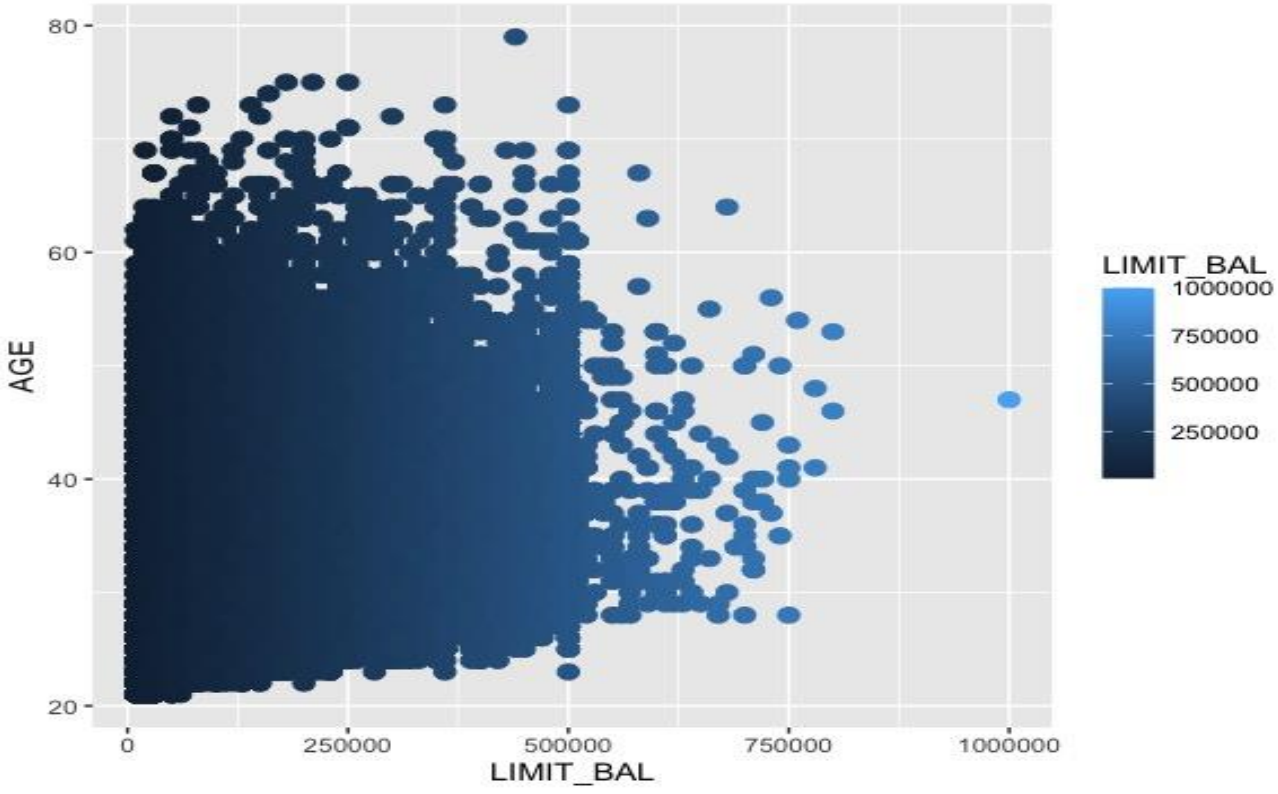


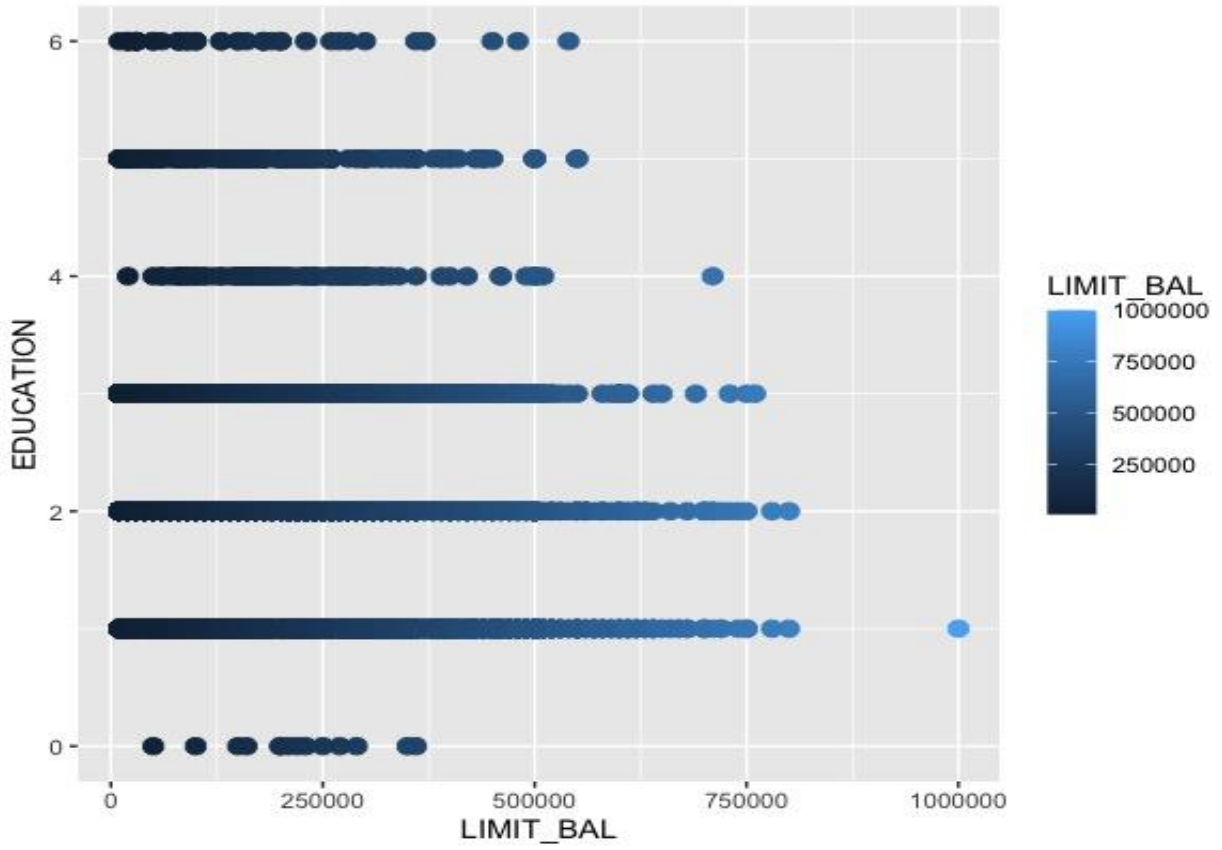


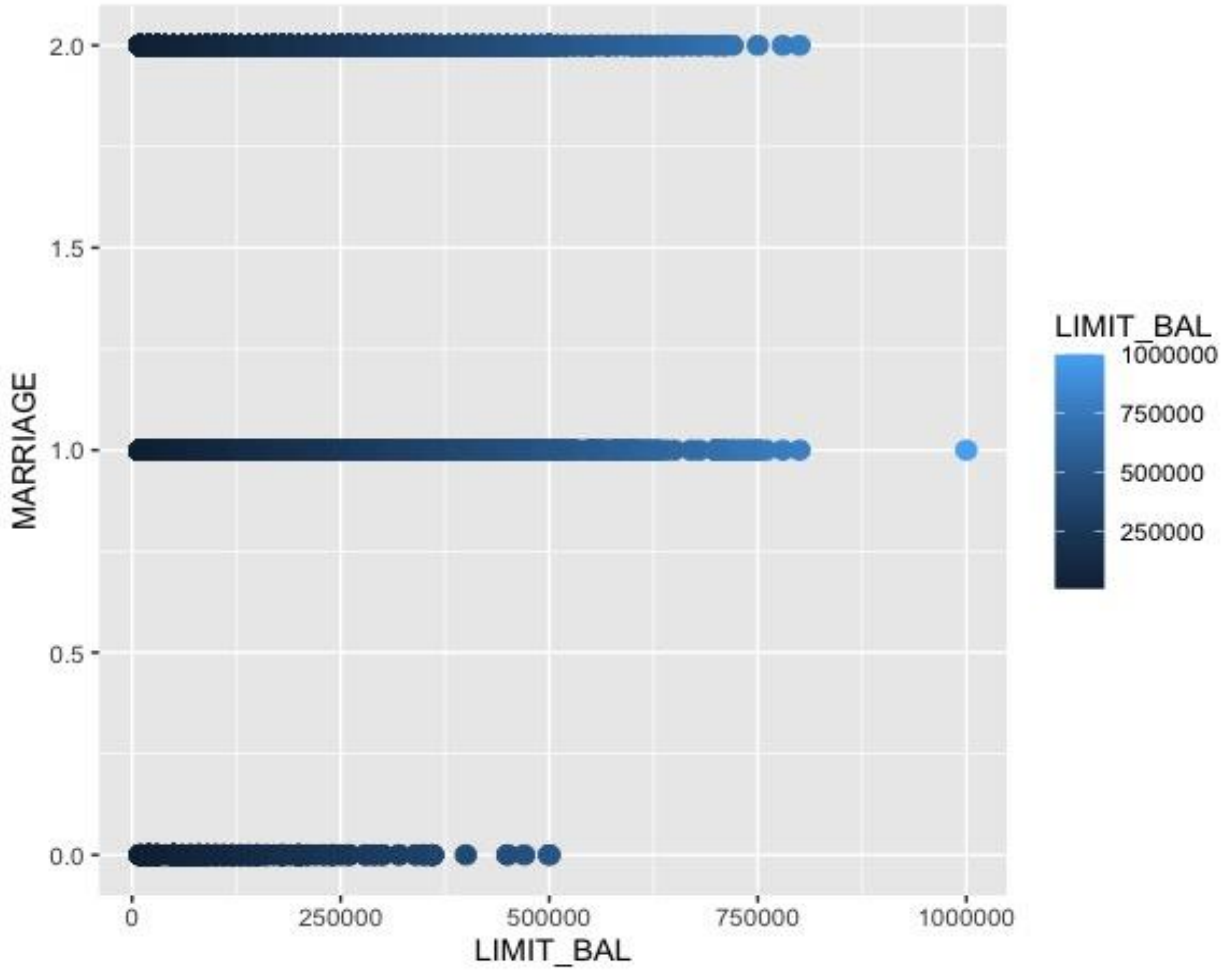

Merital Status Wise Classification
15000
10000
5000
0
count
12623
10453
3206
3341
89
0
1
2
Marriage
default_payment_next_month
a 0
a 1

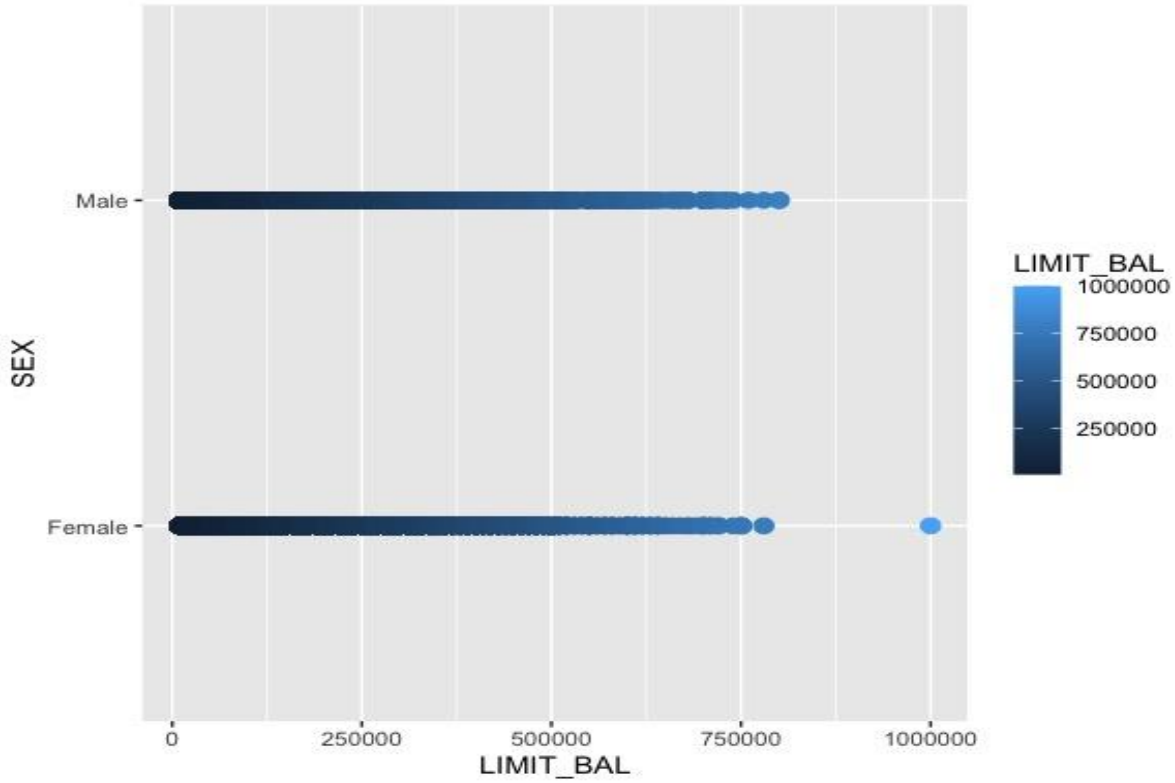


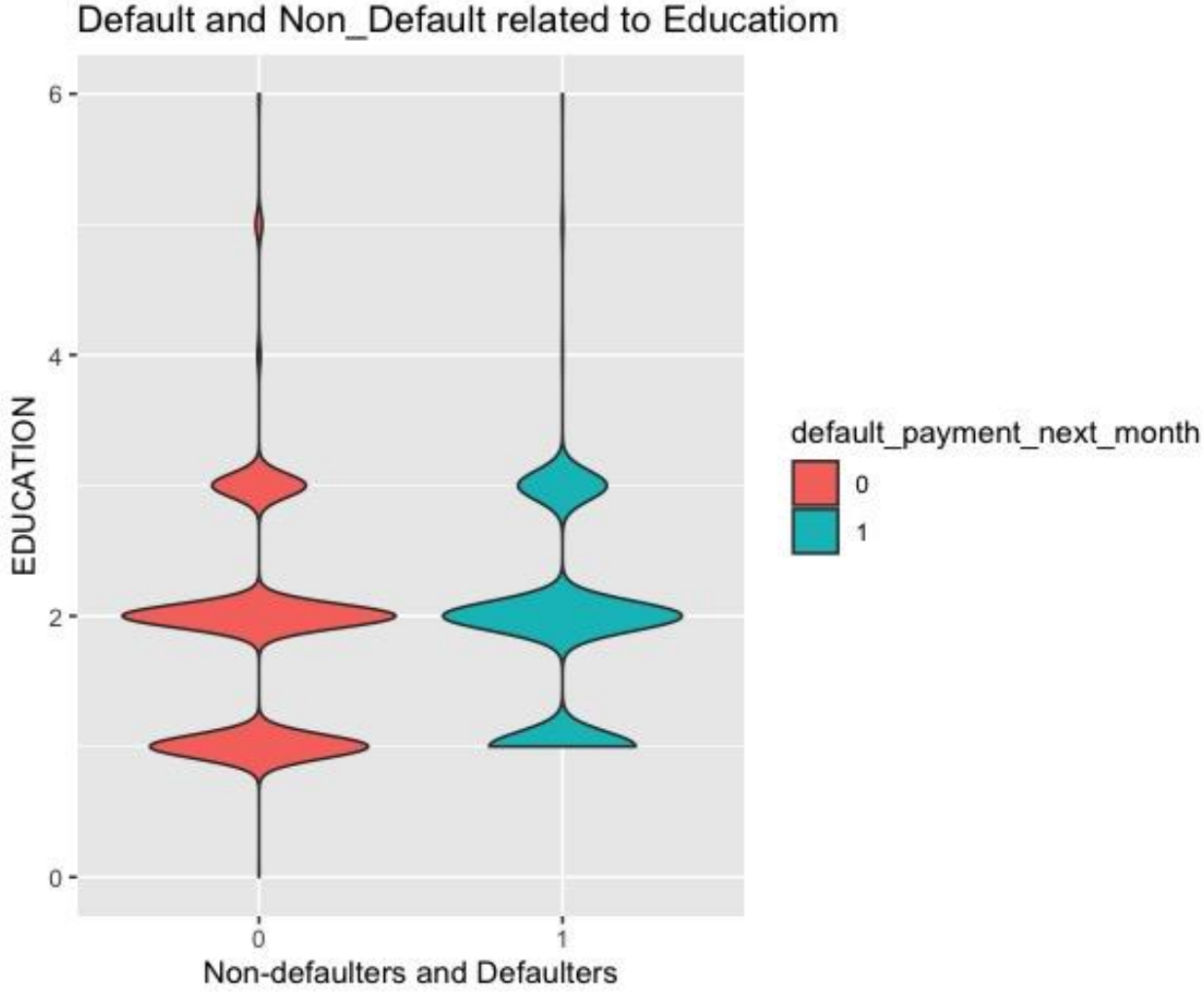
Age VS Limit Balance:

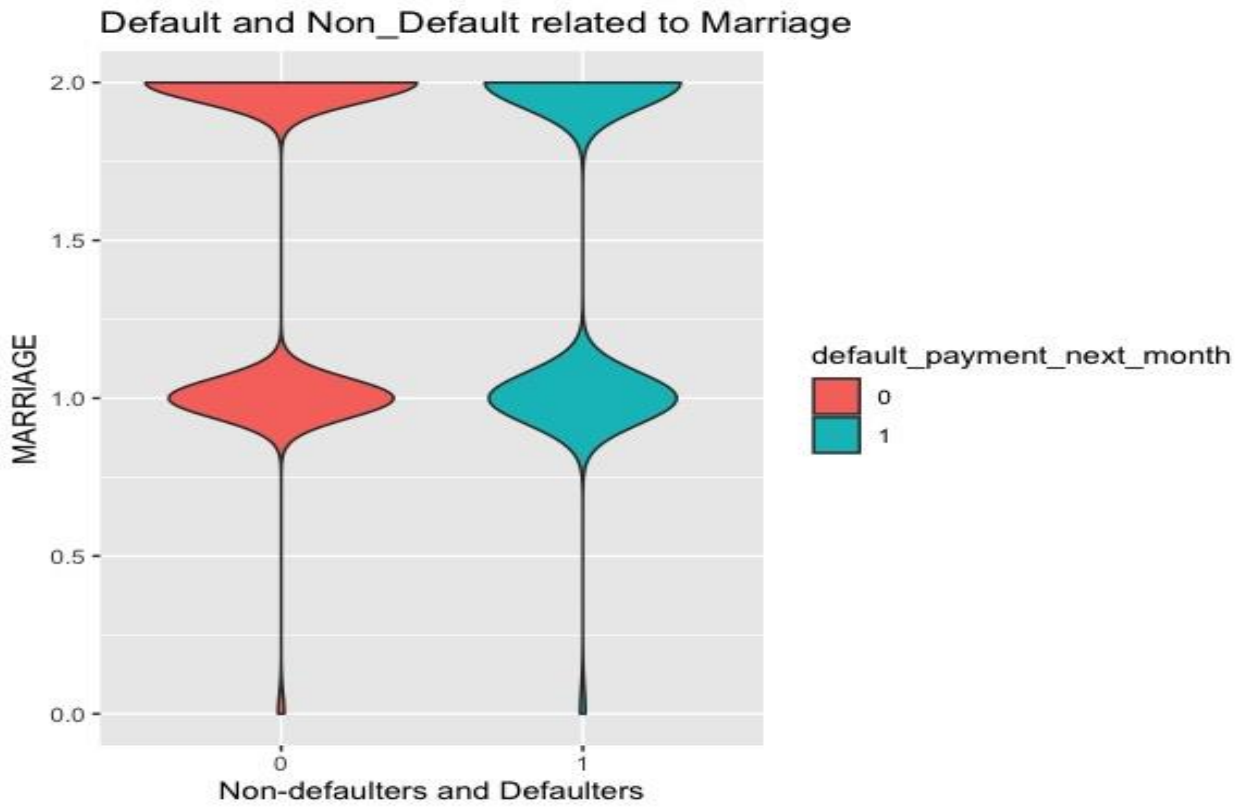



EDUCATION
6
4
2
0
0
250000
500000
750000
1000000
LIMIT_BAL
LIMIT_BAL
1000000
750000
500000
250000

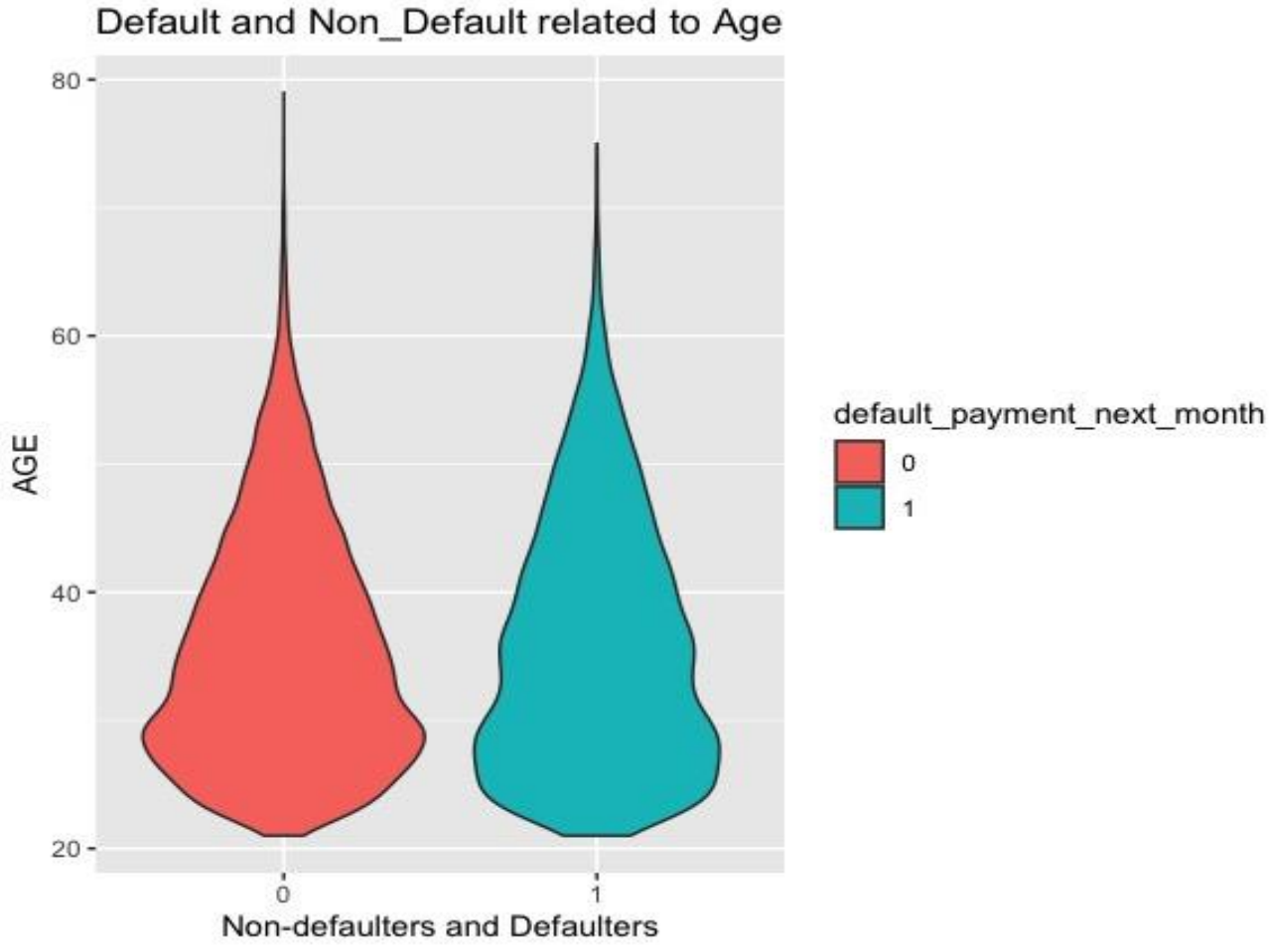

MARRIAGE
2.0
1.5
1.0
0.5
0.0
0
250000
500000
750000
1000000
LIMIT_BAL
LIMIT_BAL
1000000
750000
500000
250000


Male
Female
SEX
0
250000
500000
750000
1000000
LIMIT_BAL
LIMIT_BAL
1000000
750000
500000
250000


Default and Non_Default related to Educatiom
6
4
2
0
EDUCATION
0
1
Non-defaulters and Defaulters
default_payment_next_month
0
1



Defaulters and Non-Defaulters WRT Age:



3.1 Dataset Preprocessing

3.1.1 Loading/Extraction of the source Dataset

```
library(readxl)
myData <- read_excel("~/Desktop/default of credit card clients.xls")
myData
```

```
> myData
# A tibble: 30,000 x 25
      ID LIMIT_BAL   SEX EDUCATION MARRIAGE   AGE PAY_0 PAY_2 PAY_3 PAY_4 PAY_5 PAY_6 BILL_AMT1 BILL_AMT2
   <dbl>     <dbl> <dbl>     <dbl>    <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>     <dbl>     <dbl>
 1     1     20000     2         2        1    24     2     2    -1    -1    -2    -2      3913      3102
 2     2    120000     2         2        2    26    -1     2     0     0     0     2      2682      1725
 3     3     90000     2         2        2    34     0     0     0     0     0     0     29239     14027
 4     4     50000     2         2        1    37     0     0     0     0     0     0     46990     48233
 5     5     50000     1         2        1    57    -1     0    -1     0     0     0      8617      5670
 6     6     50000     1         1        2    37     0     0     0     0     0     0     64400     57069
 7     7    500000     1         1        2    29     0     0     0     0     0     0    367965    412023
 8     8    100000     2         2        2    23     0    -1    -1     0     0    -1     11876       380
 9     9    140000     2         3        1    28     0     0     2     0     0     0     11285     14096
10    10     20000     1         3        2    35    -2    -2    -2    -2    -1    -1         0         0
# … with 29,990 more rows, and 11 more variables: BILL_AMT3 <dbl>, BILL_AMT4 <dbl>, BILL_AMT5 <dbl>,
#   BILL_AMT6 <dbl>, PAY_AMT1 <dbl>, PAY_AMT2 <dbl>, PAY_AMT3 <dbl>, PAY_AMT4 <dbl>, PAY_AMT5 <dbl>,
#   PAY_AMT6 <dbl>, `default payment next month` <dbl>
>
```

3.1.2 Extract/Filter Desired Attributes/Fields

Desired attributes are following:

```
> Desired
# A tibble: 30,000 x 23
   LIMIT_BAL SEX    EDUCATION MARRIAGE   AGE PAY_1 PAY_2 PAY_3 PAY_4 PAY_5 PAY_6 BILL_AMT1 BILL_AMT2
       <dbl> <fct>      <dbl>    <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>     <dbl>     <dbl>
 1     20000 Fema…          2        1    24     2     2    -1    -1    -2    -2      3913      3102
 2    120000 Fema…          2        2    26    -1     2     0     0     0     2      2682      1725
 3     90000 Fema…          2        2    34     0     0     0     0     0     0     29239     14027
 4     50000 Fema…          2        1    37     0     0     0     0     0     0     46990     48233
 5     50000 Male           2        1    57    -1     0    -1     0     0     0      8617      5670
 6     50000 Male           1        2    37     0     0     0     0     0     0     64400     57069
 7    500000 Male           1        2    29     0     0     0     0     0     0    367965    412023
 8    100000 Fema…          2        2    23     0    -1    -1     0     0    -1     11876       380
 9    140000 Fema…          3        1    28     0     0     2     0     0     0     11285     14096
10     20000 Male           3        2    35    -2    -2    -2    -2    -1    -1         0         0
# … with 29,990 more rows, and 10 more variables: BILL_AMT3 <dbl>, BILL_AMT4 <dbl>, BILL_AMT5 <dbl>,
#   BILL_AMT6 <dbl>, PAY_AMT1 <dbl>, PAY_AMT2 <dbl>, PAY_AMT3 <dbl>, PAY_AMT4 <dbl>, PAY_AMT5 <dbl>,
#   PAY_AMT6 <dbl>
>
```

3.1.3 Removing Duplicates Records

Checking duplicate records in our dataset:

```
> duplicated(myData)
   [1] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
  [18] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
  [35] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
  [52] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
  [69] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
  [86] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
 [103] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
 [120] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
 [137] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
 [154] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
 [171] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
 [188] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
 [205] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
 [222] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
 [239] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
 [256] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
 [273] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
 [290] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
 [307] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
 [324] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
 [341] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
 [358] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
 [375] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
 [392] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
 [409] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
 [426] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
 [443] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
 [460] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
 [477] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
 [494] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
 [511] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
 [528] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
 [545] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
 [562] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
 [579] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
 [596] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
 [613] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
 [630] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
 [647] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
 [664] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
 [681] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
 [698] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
 [715] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
 [732] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
 [749] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
 [766] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
 [783] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
 [800] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
 [817] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
 [834] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
 [851] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
 [868] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
 [885] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
 [902] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
 [919] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
 [936] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
 [953] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
 [970] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
 [987] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
 [ reached getOption("max.print") -- omitted 29001 entries ]
>
```

checkingDuplicate <- duplicated(myData)
length(checkingDuplicate[checkingDuplicate== TRUE])

```
> checkingDuplicate <- duplicated(myData)
> length(checkingDuplicate[checkingDuplicate== TRUE])
[1] 0
>
```

So, we found out that there are no duplicate records in our dataset.

3.1.4 Structuring Date and Time Transformation

There are not any dates and times in our dataset. So, we cannot structure date and time transformation.

3.1.5 Replacing NA for Date-Time Observation/Records

There are not any dates and times in our dataset. So, there will be no NA in date-time.

3.1.6 Reformatting and Imputation of Date-Time Attributes

There are not any dates and times in our dataset. So, we cannot impute and reformat Date-Time attributes.

3.1.7 Imputation on Data-Time column

There are not any dates and times in our dataset. So, we cannot do imputation on Date-Time column.

3.1.8 Creation Classifiers

- **Random Forest Classifier:**

```
> plot(classifier.rf)
> classifier.rf

Call:
 randomForest(formula = default_payment_next_month ~ ., data = training_setnew,      ntree = 10)
               Type of random forest: classification
                     Number of trees: 10
No. of variables tried at each split: 4

        OOB estimate of  error rate: 22.93%
Confusion matrix:
      0    1 class.error
0 16461 2046   0.1105528
1  3406 1865   0.6461772
>
```

Predictions Done by Random Forest:

```
> our.predict.rf
  1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18  19  20  21
  0   0   0   0   0   0   0   0   0   0   0   0   0   1   1   0   0   1   0   1   0
 22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42
  0   0   1   0   0   0   0   1   0   0   0   1   0   0   0   0   0   0   0   0   0
 43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63
  1   0   1   0   0   1   0   1   0   0   0   0   0   0   1   0   0   0   0   0   0
 64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84
  0   0   0   1   0   1   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
 85  86  87  88  89  90  91  92  93  94  95  96  97  98  99 100 101 102 103 104 105
  0   0   0   0   0   0   0   0   0   0   0   1   0   0   0   0   1   0   0   0   1
106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126
  0   0   0   0   0   0   0   0   0   0   0   1   0   0   0   0   0   0   0   1   0
127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144 145 146 147
  0   0   0   0   0   0   0   0   0   0   0   1   0   0   0   0   0   0   0   0   0
148 149 150 151 152 153 154 155 156 157 158 159 160 161 162 163 164 165 166 167 168
  0   1   0   0   0   1   0   0   0   0   0   0   0   0   1   0   0   0   0   0   1
169 170 171 172 173 174 175 176 177 178 179 180 181 182 183 184 185 186 187 188 189
  0   0   1   0   0   1   0   0   0   0   0   0   1   1   0   0   1   0   0   0   0
190 191 192 193 194 195 196 197 198 199 200 201 202 203 204 205 206 207 208 209 210
  0   0   0   0   0   0   0   0   0   0   1   0   0   0   0   0   1   0   0   0   0
211 212 213 214 215 216 217 218 219 220 221 222 223 224 225 226 227 228 229 230 231
  1   0   0   0   0   1   0   0   0   0   0   0   0   0   1   1   1   0   0   0   0
232 233 234 235 236 237 238 239 240 241 242 243 244 245 246 247 248 249 250 251 252
  0   0   0   0   0   0   1   0   0   0   0   0   0   0   1   0   0   0   0   0   0
253 254 255 256 257 258 259 260 261 262 263 264 265 266 267 268 269 270 271 272 273
  0   1   1   0   0   1   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
274 275 276 277 278 279 280 281 282 283 284 285 286 287 288 289 290 291 292 293 294
  0   0   0   1   0   0   1   0   1   1   0   0   0   0   0   0   0   0   0   0   0
295 296 297 298 299 300 301 302 303 304 305 306 307 308 309 310 311 312 313 314 315
  0   1   0   0   0   0   0   0   0   0   0   0   0   1   0   0   0   0   0   0   0
316 317 318 319 320 321 322 323 324 325 326 327 328 329 330 331 332 333 334 335 336
  0   0   1   0   0   0   0   0   0   0   0   1   0   0   0   0   1   0   0   0   0
337 338 339 340 341 342 343 344 345 346 347 348 349 350 351 352 353 354 355 356 357
  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   1   0   0   0
358 359 360 361 362 363 364 365 366 367 368 369 370 371 372 373 374 375 376 377 378
  0   0   0   0   0   0   0   0   1   0   0   0   0   0   0   0   1   0   0   0   0
379 380 381 382 383 384 385 386 387 388 389 390 391 392 393 394 395 396 397 398 399
  0   0   0   0   0   0   0   0   1   0   1   0   0   0   0   0   0   0   0   0   0
400 401 402 403 404 405 406 407 408 409 410 411 412 413 414 415 416 417 418 419 420
  0   0   0   0   0   0   0   1   1   0   1   0   0   0   0   1   1   0   0   0   0
421 422 423 424 425 426 427 428 429 430 431 432 433 434 435 436 437 438 439 440 441
  1   0   0   1   0   1   0   1   1   1   0   0   0   0   0   0   0   0   0   0   0
```

```
442 443 444 445 446 447 448 449 450 451 452 453 454 455 456 457 458 459 460 461 462
  0   0   1   0   0   0   0   0   0   0   0   1   1   0   0   0   0   1   0   0   1
463 464 465 466 467 468 469 470 471 472 473 474 475 476 477 478 479 480 481 482 483
  0   0   0   1   0   0   0   0   0   1   0   0   0   1   0   0   0   0   0   0   0
484 485 486 487 488 489 490 491 492 493 494 495 496 497 498 499 500 501 502 503 504
  0   1   0   0   0   0   0   0   0   1   0   0   0   0   0   1   0   0   0   0   0
505 506 507 508 509 510 511 512 513 514 515 516 517 518 519 520 521 522 523 524 525
  0   1   0   0   0   0   0   0   0   0   0   0   0   1   0   0   0   0   1   0   0
526 527 528 529 530 531 532 533 534 535 536 537 538 539 540 541 542 543 544 545 546
  0   0   0   0   0   0   1   0   0   0   0   0   1   0   0   0   0   0   0   0   1
547 548 549 550 551 552 553 554 555 556 557 558 559 560 561 562 563 564 565 566 567
  0   0   0   0   0   0   0   0   1   0   0   1   0   0   0   0   0   0   0   0   0
568 569 570 571 572 573 574 575 576 577 578 579 580 581 582 583 584 585 586 587 588
  0   0   0   0   0   1   0   0   0   0   0   0   0   0   0   1   0   0   0   0   0
589 590 591 592 593 594 595 596 597 598 599 600 601 602 603 604 605 606 607 608 609
  0   0   1   0   0   0   0   0   0   1   0   0   0   0   0   1   0   0   0   0   0
610 611 612 613 614 615 616 617 618 619 620 621 622 623 624 625 626 627 628 629 630
  0   1   0   0   0   0   0   1   0   0   0   1   0   0   1   0   1   0   0   0   1
631 632 633 634 635 636 637 638 639 640 641 642 643 644 645 646 647 648 649 650 651
  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
652 653 654 655 656 657 658 659 660 661 662 663 664 665 666 667 668 669 670 671 672
  0   1   0   0   0   0   0   0   0   0   0   0   1   0   0   1   0   0   0   0   0
673 674 675 676 677 678 679 680 681 682 683 684 685 686 687 688 689 690 691 692 693
  1   0   0   0   1   0   0   1   0   0   0   0   0   0   0   0   0   1   0   0   0
694 695 696 697 698 699 700 701 702 703 704 705 706 707 708 709 710 711 712 713 714
  0   1   0   0   0   0   0   0   1   0   0   1   0   0   0   0   0   0   0   0   1
715 716 717 718 719 720 721 722 723 724 725 726 727 728 729 730 731 732 733 734 735
  0   0   0   0   0   0   0   1   0   0   0   1   0   1   1   0   0   0   0   0   0
736 737 738 739 740 741 742 743 744 745 746 747 748 749 750 751 752 753 754 755 756
  0   0   0   0   0   1   0   0   1   0   0   1   1   0   0   0   0   0   0   0   0
757 758 759 760 761 762 763 764 765 766 767 768 769 770 771 772 773 774 775 776 777
  1   0   0   0   0   0   1   0   0   0   0   0   0   0   0   0   0   0   0   0   0
778 779 780 781 782 783 784 785 786 787 788 789 790 791 792 793 794 795 796 797 798
  0   0   0   0   1   0   0   0   0   0   0   1   0   0   0   0   0   0   0   0   0
799 800 801 802 803 804 805 806 807 808 809 810 811 812 813 814 815 816 817 818 819
  1   0   0   0   0   0   1   0   0   0   0   0   1   0   0   0   0   0   0   0   0
820 821 822 823 824 825 826 827 828 829 830 831 832 833 834 835 836 837 838 839 840
  0   0   0   0   0   0   0   0   1   0   0   0   0   0   1   0   0   0   0   0   0
841 842 843 844 845 846 847 848 849 850 851 852 853 854 855 856 857 858 859 860 861
  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
862 863 864 865 866 867 868 869 870 871 872 873 874 875 876 877 878 879 880 881 882
  0   0   0   0   0   0   1   0   0   0   0   1   0   0   0   0   1   0   1   0   0
883 884 885 886 887 888 889 890 891 892 893 894 895 896 897 898 899 900 901 902 903
  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
904 905 906 907 908 909 910 911 912 913 914 915 916 917 918 919 920 921 922 923 924
  0   0   0   0   0   0   0   0   1   0   0   0   0   0   0   0   0   0   0   0   0
925 926 927 928 929 930 931 932 933 934 935 936 937 938 939 940 941 942 943 944 945
  0   0   0   0   0   0   0   0   0   0   1   0   0   0   1   0   0   1   0   0   0
946 947 948 949 950 951 952 953 954 955 956 957 958 959 960 961 962 963 964 965 966
  0   0   0   0   0   0   0   0   1   0   0   0   0   0   0   0   0   0   0   0   0
967 968 969 970 971 972 973 974 975 976 977 978 979 980 981 982 983 984 985 986 987
  1   0   0   0   0   0   1   0   0   0   0   0   0   0   0   1   0   0   0   0   0
988 989 990 991 992 993 994 995 996 997 998 999 1000
  0   1   0   0   0   0   0   0   0   0   0   0   0
 [ reached getOption("max.print") -- omitted 5000 entries ]
Levels: 0 1
>
```

Confusion Matrix of Random Forest:

```
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 4335  875
         1  338  452

               Accuracy : 0.7978
                 95% CI : (0.7874, 0.8079)
    No Information Rate : 0.7788
    P-Value [Acc > NIR] : 0.0001818

                  Kappa : 0.3137

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

            Sensitivity : 0.9277
            Specificity : 0.3406
         Pos Pred Value : 0.8321
         Neg Pred Value : 0.5722
             Prevalence : 0.7788
         Detection Rate : 0.7225
   Detection Prevalence : 0.8683
      Balanced Accuracy : 0.6341

       'Positive' Class : 0

> |
```

- **SVM Classifier:**

Splitting of Dataset for Training and Testing:

```
> split = sample.split(myData$default_payment_next_month, SplitRatio = 0.75)
> split
   [1]  TRUE FALSE  TRUE FALSE FALSE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE  TRUE FALSE
  [18] FALSE  TRUE FALSE FALSE FALSE  TRUE  TRUE  TRUE  TRUE FALSE FALSE FALSE FALSE  TRUE  TRUE FALSE  TRUE
  [35]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE  TRUE FALSE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE
  [52]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE FALSE  TRUE  TRUE  TRUE  TRUE FALSE
  [69]  TRUE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE FALSE  TRUE
  [86] FALSE FALSE FALSE FALSE FALSE  TRUE FALSE  TRUE  TRUE FALSE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE
 [103]  TRUE  TRUE FALSE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE FALSE
 [120]  TRUE  TRUE FALSE FALSE FALSE  TRUE  TRUE FALSE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE FALSE FALSE  TRUE
 [137]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE
 [154]  TRUE  TRUE  TRUE  TRUE FALSE  TRUE  TRUE FALSE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
 [171]  TRUE  TRUE FALSE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE  TRUE  TRUE
 [188]  TRUE FALSE  TRUE FALSE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
 [205]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE  TRUE  TRUE FALSE FALSE FALSE  TRUE  TRUE  TRUE
 [222]  TRUE  TRUE  TRUE  TRUE FALSE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
 [239]  TRUE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE FALSE  TRUE
 [256]  TRUE  TRUE  TRUE  TRUE FALSE  TRUE FALSE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE
 [273]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE
 [290] FALSE  TRUE FALSE  TRUE FALSE FALSE  TRUE  TRUE  TRUE  TRUE FALSE FALSE  TRUE  TRUE  TRUE  TRUE FALSE
 [307]  TRUE  TRUE FALSE  TRUE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
 [324]  TRUE  TRUE  TRUE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE  TRUE FALSE
 [341]  TRUE  TRUE  TRUE  TRUE FALSE  TRUE FALSE  TRUE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE FALSE FALSE
 [358]  TRUE  TRUE  TRUE  TRUE FALSE  TRUE  TRUE FALSE FALSE  TRUE FALSE  TRUE FALSE FALSE FALSE  TRUE  TRUE
 [375] FALSE  TRUE  TRUE  TRUE FALSE FALSE  TRUE  TRUE  TRUE  TRUE FALSE FALSE FALSE  TRUE  TRUE  TRUE  TRUE
 [392]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE  TRUE
 [409]  TRUE  TRUE FALSE  TRUE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE
 [426]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
 [443]  TRUE  TRUE FALSE  TRUE FALSE FALSE  TRUE  TRUE FALSE  TRUE  TRUE FALSE  TRUE  TRUE FALSE  TRUE FALSE
 [460]  TRUE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE  TRUE
 [477] FALSE  TRUE  TRUE FALSE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE FALSE FALSE FALSE  TRUE
 [494]  TRUE  TRUE  TRUE FALSE FALSE FALSE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE
 [511] FALSE  TRUE  TRUE FALSE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE
 [528]  TRUE  TRUE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
 [545]  TRUE  TRUE  TRUE FALSE  TRUE FALSE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE  TRUE  TRUE FALSE FALSE
 [562]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE
 [579] FALSE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE  TRUE FALSE  TRUE FALSE  TRUE FALSE
 [596] FALSE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE FALSE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE
 [613]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE
 [630]  TRUE  TRUE  TRUE  TRUE  TRUE FALSE FALSE FALSE FALSE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
 [647] FALSE  TRUE FALSE  TRUE FALSE FALSE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
 [664]  TRUE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE  TRUE  TRUE FALSE  TRUE  TRUE FALSE  TRUE FALSE
 [681]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE  TRUE FALSE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE
```

SVM Model:

```
> SVMclassifier = svm(formula = default_payment_next_month ~ .,
+                     data = training_set,
+                     type = 'C-classification',
+                     kernel = 'linear')
> SVMclassifier

Call:
svm(formula = default_payment_next_month ~ ., data = training_set, type = "C-classification",
    kernel = "linear")


Parameters:
   SVM-Type:  C-classification
 SVM-Kernel:  linear
       cost:  1

Number of Support Vectors:  12757

>
```

Prediction Done by SVM Model:

```
> prediction = predict(SVMclassifier, newdata = test_set[-25])
> prediction
  1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18  19  20  21
  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   1   0
 22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42
  1   0   0   0   0   0   0   1   0   0   0   1   0   0   0   0   0   0   0   0   0
 43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63
  0   0   0   0   1   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
 64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84
  0   0   0   0   0   0   0   0   0   0   0   0   0   1   0   0   0   0   0   0   0
 85  86  87  88  89  90  91  92  93  94  95  96  97  98  99 100 101 102 103 104 105
  0   0   0   0   0   0   1   1   0   0   0   0   0   0   0   0   0   0   0   0   0
106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126
  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   1   0   0   0   0
127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144 145 146 147
  0   0   1   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
148 149 150 151 152 153 154 155 156 157 158 159 160 161 162 163 164 165 166 167 168
  0   0   1   0   0   0   0   0   0   0   0   1   0   0   0   0   0   0   0   0   0
169 170 171 172 173 174 175 176 177 178 179 180 181 182 183 184 185 186 187 188 189
  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
190 191 192 193 194 195 196 197 198 199 200 201 202 203 204 205 206 207 208 209 210
  0   0   0   0   0   0   0   0   1   0   0   0   0   0   0   0   0   0   0   1   0
211 212 213 214 215 216 217 218 219 220 221 222 223 224 225 226 227 228 229 230 231
  0   0   0   0   0   0   0   0   1   0   0   0   0   0   0   1   0   0   0   0   1
232 233 234 235 236 237 238 239 240 241 242 243 244 245 246 247 248 249 250 251 252
  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   1   0   0   0   0   0
253 254 255 256 257 258 259 260 261 262 263 264 265 266 267 268 269 270 271 272 273
  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
274 275 276 277 278 279 280 281 282 283 284 285 286 287 288 289 290 291 292 293 294
  0   0   0   0   0   0   1   1   0   0   0   0   0   0   0   0   0   0   0   0   0
295 296 297 298 299 300 301 302 303 304 305 306 307 308 309 310 311 312 313 314 315
  0   0   1   0   0   0   0   0   0   0   0   0   0   0   1   0   0   0   0   0   0
316 317 318 319 320 321 322 323 324 325 326 327 328 329 330 331 332 333 334 335 336
  0   0   0   0   0   1   0   0   0   0   0   1   0   0   1   0   0   0   0   0   0
337 338 339 340 341 342 343 344 345 346 347 348 349 350 351 352 353 354 355 356 357
  1   1   0   0   0   0   0   0   0   0   0   0   0   0   1   0   0   1   0   0   0
358 359 360 361 362 363 364 365 366 367 368 369 370 371 372 373 374 375 376 377 378
  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
379 380 381 382 383 384 385 386 387 388 389 390 391 392 393 394 395 396 397 398 399
  0   0   0   0   0   0   0   0   0   0   0   1   0   0   0   0   0   0   0   0   0
400 401 402 403 404 405 406 407 408 409 410 411 412 413 414 415 416 417 418 419 420
  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
```

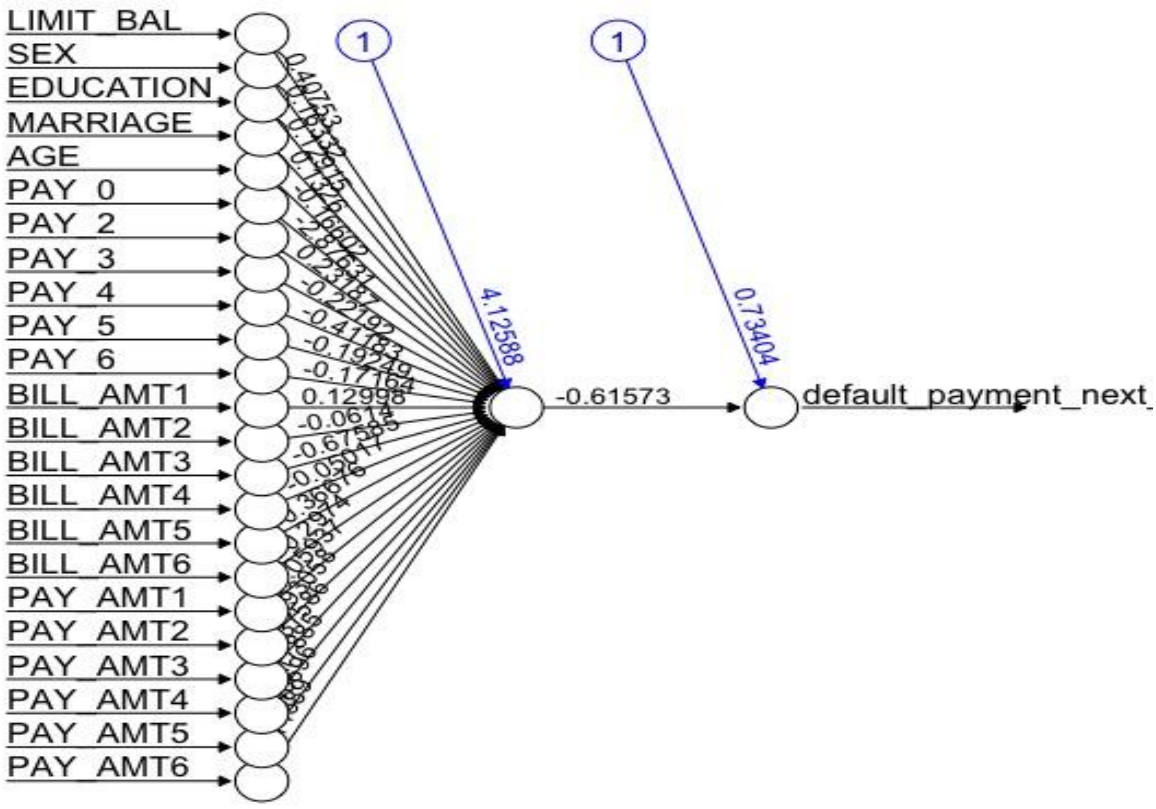
```
  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
421 422 423 424 425 426 427 428 429 430 431 432 433 434 435 436 437 438 439 440 441
  0   0   0   1   0   0   0   0   0   0   0   1   0   0   0   0   0   0   0   0   0
442 443 444 445 446 447 448 449 450 451 452 453 454 455 456 457 458 459 460 461 462
  0   0   0   0   0   0   0   0   0   0   0   0   0   1   0   0   1   0   0   0   0
463 464 465 466 467 468 469 470 471 472 473 474 475 476 477 478 479 480 481 482 483
  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   1   0
484 485 486 487 488 489 490 491 492 493 494 495 496 497 498 499 500 501 502 503 504
  1   0   0   0   0   0   0   0   0   0   1   0   0   1   0   0   0   0   0   0   0
505 506 507 508 509 510 511 512 513 514 515 516 517 518 519 520 521 522 523 524 525
  0   0   0   0   0   1   1   0   0   1   0   0   0   0   0   0   0   0   0   0   0
526 527 528 529 530 531 532 533 534 535 536 537 538 539 540 541 542 543 544 545 546
  0   0   0   0   0   0   0   1   0   0   0   0   0   0   0   1   1   0   0   0   1
547 548 549 550 551 552 553 554 555 556 557 558 559 560 561 562 563 564 565 566 567
  0   0   1   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   1   0
568 569 570 571 572 573 574 575 576 577 578 579 580 581 582 583 584 585 586 587 588
  0   0   0   1   0   0   0   0   1   0   0   0   0   1   0   0   0   0   1   1   1
589 590 591 592 593 594 595 596 597 598 599 600 601 602 603 604 605 606 607 608 609
  0   0   1   0   0   0   0   0   0   0   0   1   0   0   0   1   0   0   0   0   0
610 611 612 613 614 615 616 617 618 619 620 621 622 623 624 625 626 627 628 629 630
  0   0   0   1   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
631 632 633 634 635 636 637 638 639 640 641 642 643 644 645 646 647 648 649 650 651
  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
652 653 654 655 656 657 658 659 660 661 662 663 664 665 666 667 668 669 670 671 672
  0   1   0   1   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   1   1
673 674 675 676 677 678 679 680 681 682 683 684 685 686 687 688 689 690 691 692 693
  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   1   1   1   0   0
694 695 696 697 698 699 700 701 702 703 704 705 706 707 708 709 710 711 712 713 714
  0   0   0   0   0   0   0   0   0   0   0   0   1   0   0   0   0   0   1   0   0
715 716 717 718 719 720 721 722 723 724 725 726 727 728 729 730 731 732 733 734 735
  0   0   0   0   0   0   0   1   0   0   0   0   0   0   0   0   0   0   0   0   0
736 737 738 739 740 741 742 743 744 745 746 747 748 749 750 751 752 753 754 755 756
  0   0   0   0   0   0   0   0   0   0   0   0   1   0   0   0   0   0   0   0   1
757 758 759 760 761 762 763 764 765 766 767 768 769 770 771 772 773 774 775 776 777
  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
778 779 780 781 782 783 784 785 786 787 788 789 790 791 792 793 794 795 796 797 798
  0   0   0   0   0   0   0   0   1   0   1   0   0   0   1   0   0   0   0   0   0
799 800 801 802 803 804 805 806 807 808 809 810 811 812 813 814 815 816 817 818 819
  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
820 821 822 823 824 825 826 827 828 829 830 831 832 833 834 835 836 837 838 839 840
  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   1   0   0   0   1   0
841 842 843 844 845 846 847 848 849 850 851 852 853 854 855 856 857 858 859 860 861
  0   0   0   0   1   0   0   0   0   1   0   0   0   0   0   0   0   0   0   0   0

862 863 864 865 866 867 868 869 870 871 872 873 874 875 876 877 878 879 880 881 882
  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
883 884 885 886 887 888 889 890 891 892 893 894 895 896 897 898 899 900 901 902 903
  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
904 905 906 907 908 909 910 911 912 913 914 915 916 917 918 919 920 921 922 923 924
  0   0   0   0   0   0   1   0   0   1   0   0   0   0   0   0   0   0   0   0   0
925 926 927 928 929 930 931 932 933 934 935 936 937 938 939 940 941 942 943 944 945
  0   0   0   0   0   0   0   0   0   0   0   0   0   1   0   0   0   0   0   0   0
946 947 948 949 950 951 952 953 954 955 956 957 958 959 960 961 962 963 964 965 966
  0   0   0   1   0   0   1   0   0   0   0   0   0   0   0   1   0   0   0   0   0
967 968 969 970 971 972 973 974 975 976 977 978 979 980 981 982 983 984 985 986 987
  1   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
988 989 990 991 992 993 994 995 996 997 998 999 1000
  0   0   0   0   0   1   0   0   0   0   0   0   0
 [ reached getOption("max.print") -- omitted 6500 entries ]
Levels: 0 1
> 
```

SVM confusion Matrix:

```
> g <- table(test_set$default_payment_next_month, prediction)
> confusionMatrix(g)
Confusion Matrix and Statistics

   prediction
      0    1
  0 5673  168
  1 1257  402

               Accuracy : 0.81
                 95% CI : (0.8009, 0.8188)
    No Information Rate : 0.924
    P-Value [Acc > NIR] : 1

                  Kappa : 0.2791

 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 0.8186
            Specificity : 0.7053
         Pos Pred Value : 0.9712
         Neg Pred Value : 0.2423
             Prevalence : 0.9240
         Detection Rate : 0.7564
   Detection Prevalence : 0.7788
      Balanced Accuracy : 0.7619

       'Positive' Class : 0

>
```

- **Neural Network Classifier:**

Neural Network Diagram:



Confusion Matrix of Neural Network:

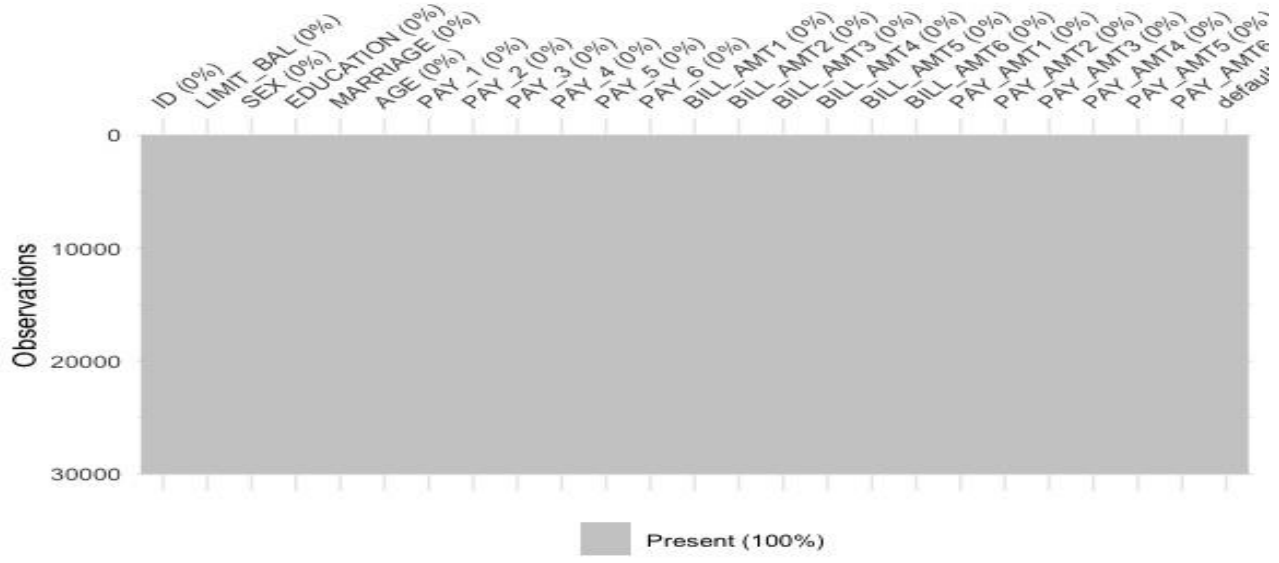
```
> cm4 <- confusionMatrix(gx)
> cm4
Confusion Matrix and Statistics

      prediction
actual    0    1
     0 4403  270
     1  841  486

               Accuracy : 0.8148
                 95% CI : (0.8048, 0.8246)
    No Information Rate : 0.874
    P-Value [Acc > NIR] : 1

                  Kappa : 0.3646

 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 0.8396
            Specificity : 0.6429
         Pos Pred Value : 0.9422
         Neg Pred Value : 0.3662
             Prevalence : 0.8740
         Detection Rate : 0.7338
   Detection Prevalence : 0.7788
      Balanced Accuracy : 0.7412

       'Positive' Class : 0

>
```

3.1.9 Removing Others NA Records of Dataset

```
> is.na(myData)
           ID LIMIT_BAL   SEX EDUCATION MARRIAGE   AGE PAY_0 PAY_2 PAY_3 PAY_4 PAY_5 PAY_6 BILL_AMT1
    [1,] FALSE     FALSE FALSE     FALSE    FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE     FALSE
    [2,] FALSE     FALSE FALSE     FALSE    FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE     FALSE
    [3,] FALSE     FALSE FALSE     FALSE    FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE     FALSE
    [4,] FALSE     FALSE FALSE     FALSE    FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE     FALSE
    [5,] FALSE     FALSE FALSE     FALSE    FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE     FALSE
    [6,] FALSE     FALSE FALSE     FALSE    FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE     FALSE
    [7,] FALSE     FALSE FALSE     FALSE    FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE     FALSE
    [8,] FALSE     FALSE FALSE     FALSE    FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE     FALSE
    [9,] FALSE     FALSE FALSE     FALSE    FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE     FALSE
   [10,] FALSE     FALSE FALSE     FALSE    FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE     FALSE
   [11,] FALSE     FALSE FALSE     FALSE    FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE     FALSE
   [12,] FALSE     FALSE FALSE     FALSE    FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE     FALSE
   [13,] FALSE     FALSE FALSE     FALSE     TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE     FALSE
   [14,] FALSE     FALSE FALSE     FALSE    FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE     FALSE
   [15,] FALSE     FALSE FALSE     FALSE    FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE     FALSE
   [16,] FALSE     FALSE FALSE     FALSE    FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE     FALSE
   [17,] FALSE     FALSE FALSE     FALSE    FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE     FALSE
   [18,] FALSE     FALSE FALSE     FALSE    FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE     FALSE
   [19,] FALSE     FALSE FALSE     FALSE    FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE     FALSE
   [20,] FALSE     FALSE FALSE     FALSE    FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE     FALSE
   [21,] FALSE     FALSE FALSE     FALSE    FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE     FALSE
   [22,] FALSE     FALSE FALSE     FALSE    FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE     FALSE
   [23,] FALSE     FALSE FALSE     FALSE    FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE     FALSE
   [24,] FALSE     FALSE FALSE     FALSE    FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE     FALSE
   [25,] FALSE     FALSE FALSE     FALSE    FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE     FALSE
   [26,] FALSE     FALSE FALSE     FALSE    FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE     FALSE
   [27,] FALSE     FALSE FALSE     FALSE    FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE     FALSE
   [28,] FALSE     FALSE FALSE     FALSE    FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE     FALSE
   [29,] FALSE     FALSE FALSE     FALSE    FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE     FALSE
   [30,] FALSE     FALSE FALSE     FALSE    FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE     FALSE
   [31,] FALSE     FALSE FALSE     FALSE    FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE     FALSE
   [32,] FALSE     FALSE FALSE     FALSE    FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE     FALSE
   [33,] FALSE     FALSE FALSE     FALSE    FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE     FALSE
   [34,] FALSE     FALSE FALSE     FALSE    FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE     FALSE
   [35,] FALSE     FALSE FALSE     FALSE    FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE     FALSE
   [36,] FALSE     FALSE FALSE     FALSE    FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE     FALSE
   [37,] FALSE     FALSE FALSE     FALSE    FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE     FALSE
   [38,] FALSE     FALSE FALSE     FALSE    FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE     FALSE
   [39,] FALSE     FALSE FALSE     FALSE    FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE     FALSE
   [40,] FALSE     FALSE FALSE     FALSE    FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE     FALSE
```

	BILL_AMT2	BILL_AMT3	BILL_AMT4	BILL_AMT5	BILL_AMT6	PAY_AMT1	PAY_AMT2	PAY_AMT3	PAY_AMT4	PAY_AMT5
[1,]	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
[2,]	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
[3,]	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
[4,]	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
[5,]	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
[6,]	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
[7,]	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
[8,]	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
[9,]	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
[10,]	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
[11,]	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
[12,]	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
[13,]	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
[14,]	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
[15,]	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
[16,]	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
[17,]	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
[18,]	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
[19,]	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
[20,]	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
[21,]	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
[22,]	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
[23,]	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
[24,]	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
[25,]	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
[26,]	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
[27,]	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
[28,]	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
[29,]	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
[30,]	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
[31,]	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
[32,]	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
[33,]	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
[34,]	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
[35,]	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
[36,]	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
[37,]	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
[38,]	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
[39,]	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
[40,]	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE

```
      PAY_AMT6 default payment next month
 [1,]    FALSE                       FALSE
 [2,]    FALSE                       FALSE
 [3,]    FALSE                       FALSE
 [4,]    FALSE                       FALSE
 [5,]    FALSE                       FALSE
 [6,]    FALSE                       FALSE
 [7,]    FALSE                       FALSE
 [8,]    FALSE                       FALSE
 [9,]    FALSE                       FALSE
[10,]    FALSE                       FALSE
[11,]    FALSE                       FALSE
[12,]    FALSE                       FALSE
[13,]    FALSE                       FALSE
[14,]    FALSE                       FALSE
[15,]    FALSE                       FALSE
[16,]    FALSE                       FALSE
[17,]    FALSE                       FALSE
[18,]    FALSE                       FALSE
[19,]    FALSE                       FALSE
[20,]    FALSE                       FALSE
[21,]    FALSE                       FALSE
[22,]    FALSE                       FALSE
[23,]    FALSE                       FALSE
[24,]    FALSE                       FALSE
[25,]    FALSE                       FALSE
[26,]    FALSE                       FALSE
[27,]    FALSE                       FALSE
[28,]    FALSE                       FALSE
[29,]    FALSE                       FALSE
[30,]    FALSE                       FALSE
[31,]    FALSE                       FALSE
[32,]    FALSE                       FALSE
[33,]    FALSE                       FALSE
[34,]    FALSE                       FALSE
[35,]    FALSE                       FALSE
[36,]    FALSE                       FALSE
[37,]    FALSE                       FALSE
[38,]    FALSE                       FALSE
[39,]    FALSE                       FALSE
[40,]    FALSE                       FALSE
[ reached getOption("max.print") -- omitted 29960 rows ]
```



So, there are no NA records in our dataset.

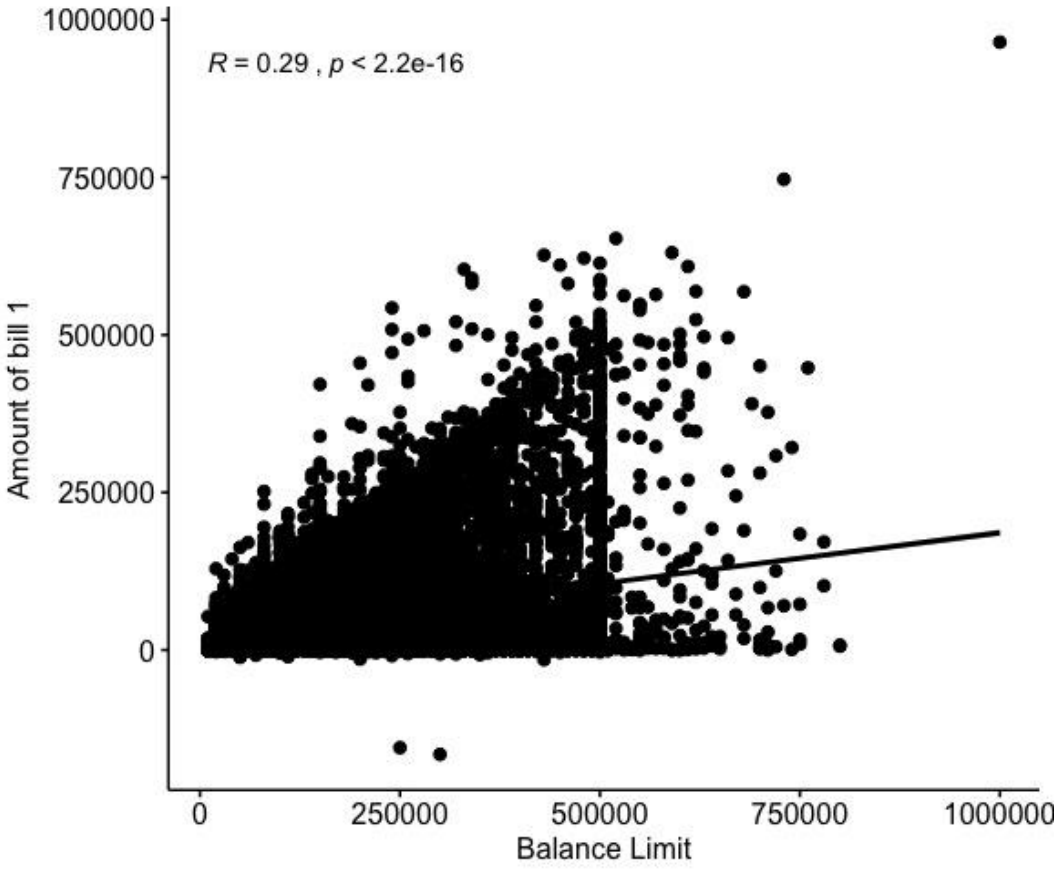
4. Identified Questions

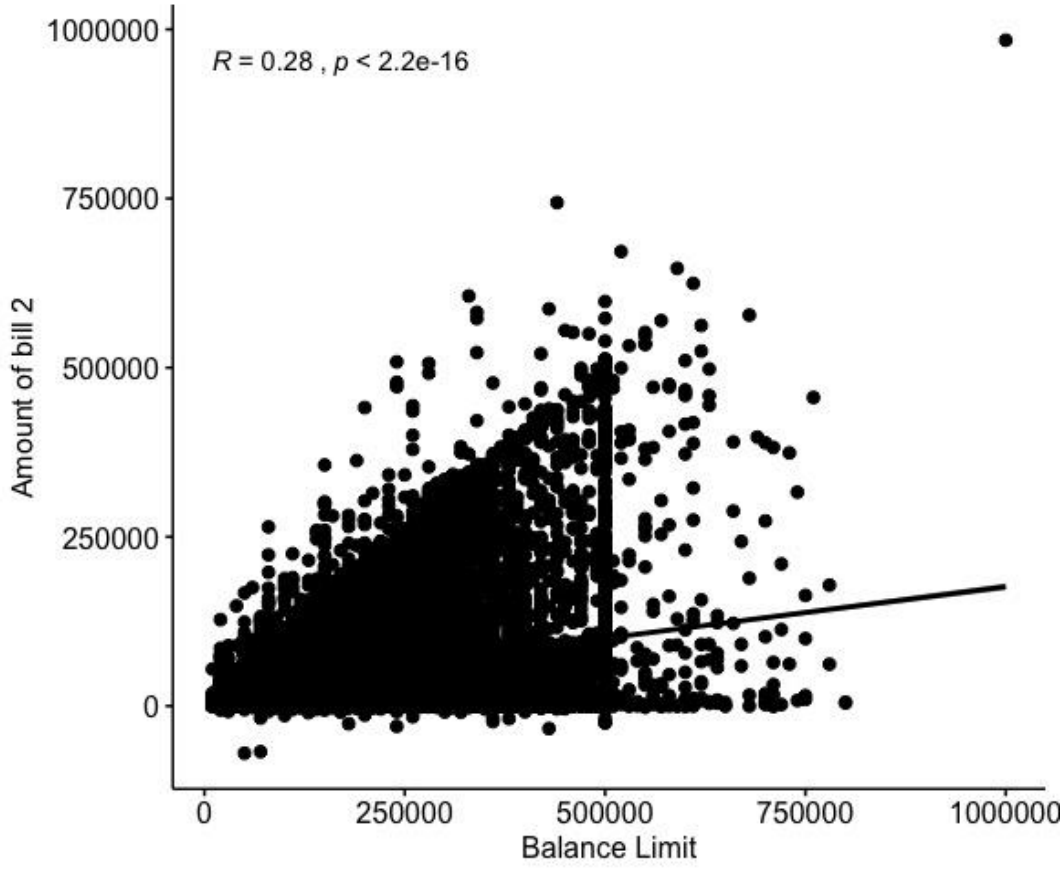
1: What are the factors that affect the most towards the default of payment next month?
2: Will I face default of payment next month? (can be answered by predicting)
3: What are the chances of not having default of payment next month?

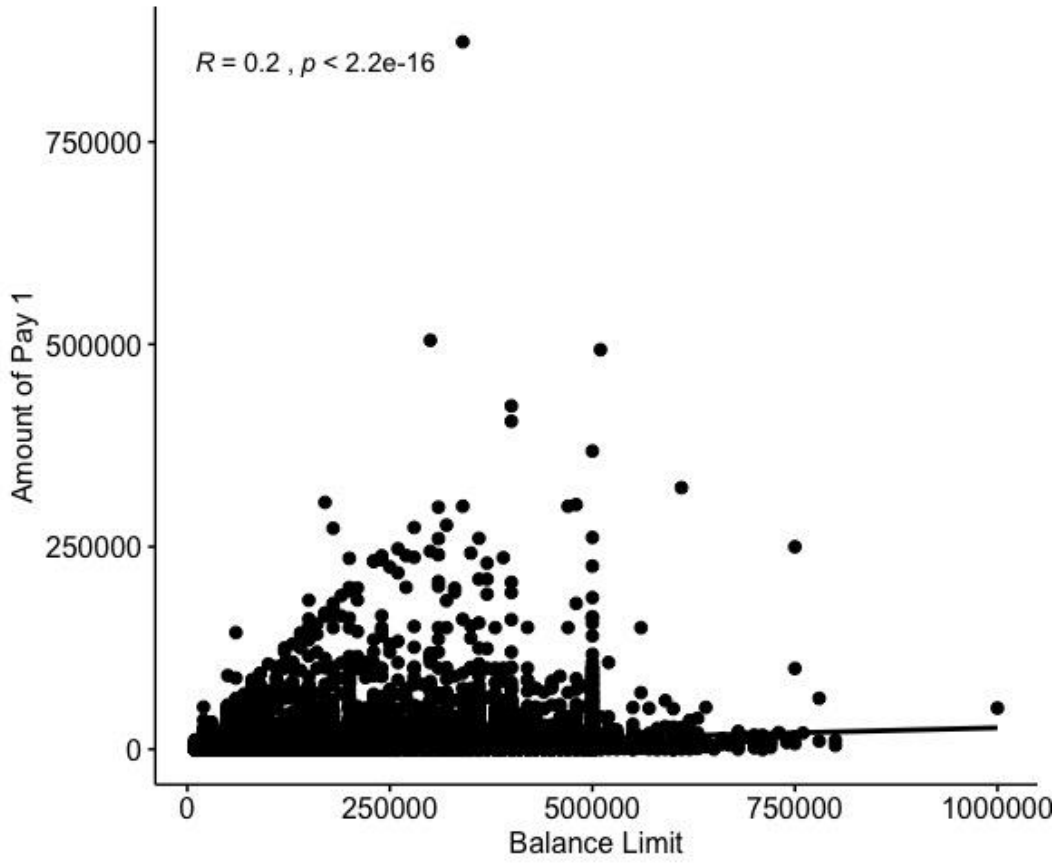
5. Machine Learning Modeling

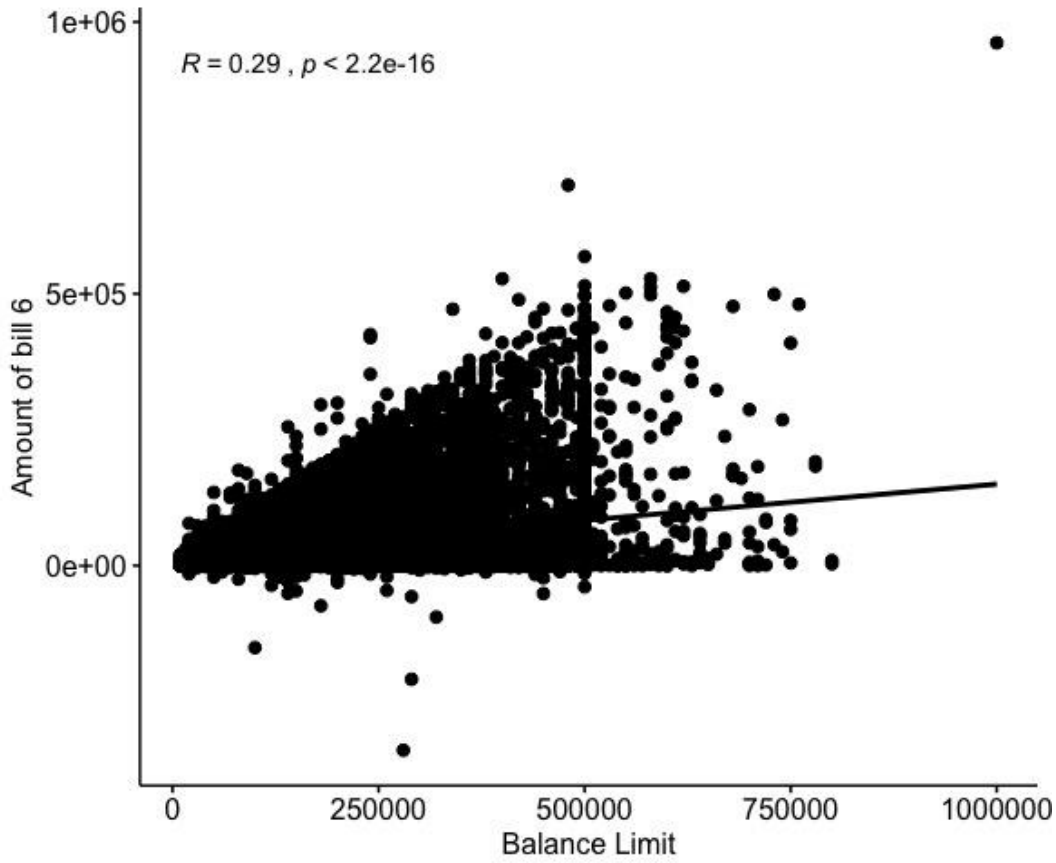
5.1 Statistics and Regression Modeling

5.1.1 Correlation




R = 0.28 , p < 2.2e-16
Amount of bill 2
1000000
750000
500000
250000
0
0
250000
500000
750000
1000000
Balance Limit


R = 0.2 , p < 2.2e-16
Amount of Pay 1
750000
500000
250000
0
0
250000
500000
750000
1000000
Balance Limit



5.2 Classification Modeling

5.2.1 Decision Tree

Predictions Done by Decision Tree:

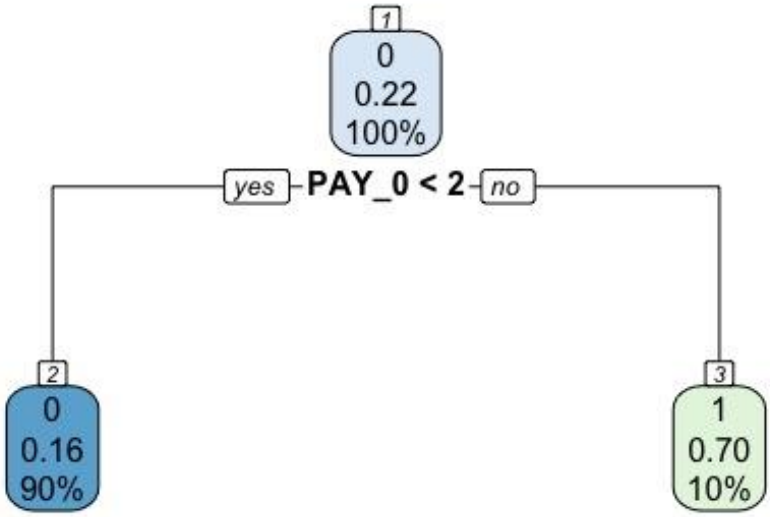
```
> pred
  1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18  19  20  21
  0   0   0   0   0   0   0   1   0   0   0   0   0   0   0   0   0   0   1   0   1
 22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42
  0   0   0   0   0   0   0   0   0   1   0   0   1   0   0   0   0   0   0   0   0
 43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63
  1   1   1   0   0   0   0   0   0   0   0   0   0   1   0   0   0   0   0   0   0
 64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84
  1   1   0   0   1   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
 85  86  87  88  89  90  91  92  93  94  95  96  97  98  99 100 101 102 103 104 105
  0   1   0   0   0   0   0   0   0   0   0   1   0   0   0   1   0   0   0   0   0
106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126
  0   0   1   0   1   0   0   0   1   0   0   0   0   0   0   0   0   0   0   0   0
127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144 145 146 147
  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
148 149 150 151 152 153 154 155 156 157 158 159 160 161 162 163 164 165 166 167 168
  0   0   0   0   0   0   0   0   1   0   0   0   0   0   0   0   0   1   0   0   0
169 170 171 172 173 174 175 176 177 178 179 180 181 182 183 184 185 186 187 188 189
  0   0   0   0   0   0   0   0   0   0   0   0   1   0   0   0   1   0   0   0   0
190 191 192 193 194 195 196 197 198 199 200 201 202 203 204 205 206 207 208 209 210
  0   1   0   0   0   0   1   0   0   0   0   0   0   0   0   0   0   0   0   0   0
211 212 213 214 215 216 217 218 219 220 221 222 223 224 225 226 227 228 229 230 231
  0   1   0   0   0   1   1   0   0   0   0   0   0   0   0   0   0   0   0   0   0
232 233 234 235 236 237 238 239 240 241 242 243 244 245 246 247 248 249 250 251 252
  0   0   0   0   0   0   0   0   0   0   0   0   1   0   1   0   0   0   0   1   0
253 254 255 256 257 258 259 260 261 262 263 264 265 266 267 268 269 270 271 272 273
  0   0   0   0   0   0   0   1   0   0   0   0   0   0   0   0   0   0   0   0   0
274 275 276 277 278 279 280 281 282 283 284 285 286 287 288 289 290 291 292 293 294
  0   0   0   0   0   0   1   0   0   0   0   0   0   0   0   0   0   0   0   0   0
295 296 297 298 299 300 301 302 303 304 305 306 307 308 309 310 311 312 313 314 315
  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   1
316 317 318 319 320 321 322 323 324 325 326 327 328 329 330 331 332 333 334 335 336
  0   0   0   1   0   0   0   0   0   0   0   0   0   0   0   0   0   1   1   0   0
337 338 339 340 341 342 343 344 345 346 347 348 349 350 351 352 353 354 355 356 357
  1   0   1   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
358 359 360 361 362 363 364 365 366 367 368 369 370 371 372 373 374 375 376 377 378
  0   0   0   0   1   0   0   0   0   0   1   1   0   0   1   1   0   0   0   0   0
379 380 381 382 383 384 385 386 387 388 389 390 391 392 393 394 395 396 397 398 399
  0   1   0   0   1   0   0   0   0   0   0   0   1   0   0   0   0   0   0   0   0
400 401 402 403 404 405 406 407 408 409 410 411 412 413 414 415 416 417 418 419 420
  0   0   0   0   1   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
```

```
 421  422  423  424  425  426  427  428  429  430  431  432  433  434  435  436  437  438  439  440  441
   0    0    0    0    0    0    1    0    0    0    0    0    0    0    0    0    0    1    0    0    0
 442  443  444  445  446  447  448  449  450  451  452  453  454  455  456  457  458  459  460  461  462
   0    0    0    0    0    0    0    0    1    0    1    0    0    0    0    0    0    0    1    1    1
 463  464  465  466  467  468  469  470  471  472  473  474  475  476  477  478  479  480  481  482  483
   0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    1
 484  485  486  487  488  489  490  491  492  493  494  495  496  497  498  499  500  501  502  503  504
   0    0    0    0    0    0    1    0    1    1    1    1    0    1    0    0    0    0    0    0    0
 505  506  507  508  509  510  511  512  513  514  515  516  517  518  519  520  521  522  523  524  525
   0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
 526  527  528  529  530  531  532  533  534  535  536  537  538  539  540  541  542  543  544  545  546
   0    0    0    0    0    0    0    0    0    0    0    0    1    0    0    0    0    0    0    0    0
 547  548  549  550  551  552  553  554  555  556  557  558  559  560  561  562  563  564  565  566  567
   1    0    0    0    1    0    0    0    0    0    0    0    0    0    0    0    0    0    0    1    0
 568  569  570  571  572  573  574  575  576  577  578  579  580  581  582  583  584  585  586  587  588
   0    0    0    0    0    0    0    0    0    0    0    0    0    1    0    0    0    0    0    1    0
 589  590  591  592  593  594  595  596  597  598  599  600  601  602  603  604  605  606  607  608  609
   0    0    1    0    0    0    0    0    0    0    0    0    0    0    0    0    0    1    0    0    0
 610  611  612  613  614  615  616  617  618  619  620  621  622  623  624  625  626  627  628  629  630
   0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    1
 631  632  633  634  635  636  637  638  639  640  641  642  643  644  645  646  647  648  649  650  651
   0    1    0    0    0    0    1    0    0    1    0    0    0    1    0    0    0    0    0    0    0
 652  653  654  655  656  657  658  659  660  661  662  663  664  665  666  667  668  669  670  671  672
   0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
 673  674  675  676  677  678  679  680  681  682  683  684  685  686  687  688  689  690  691  692  693
   0    0    0    0    0    0    1    1    0    0    0    0    0    0    0    0    0    0    0    0    0
 694  695  696  697  698  699  700  701  702  703  704  705  706  707  708  709  710  711  712  713  714
   0    0    0    0    1    0    0    0    0    0    0    0    0    1    0    0    0    0    1    0    0
 715  716  717  718  719  720  721  722  723  724  725  726  727  728  729  730  731  732  733  734  735
   0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    1
 736  737  738  739  740  741  742  743  744  745  746  747  748  749  750  751  752  753  754  755  756
   1    0    0    0    0    0    0    0    1    0    0    0    0    0    1    1    0    0    0    0    0
 757  758  759  760  761  762  763  764  765  766  767  768  769  770  771  772  773  774  775  776  777
   0    0    0    0    0    0    0    0    0    1    1    0    0    0    1    0    0    0    1    0    1
 778  779  780  781  782  783  784  785  786  787  788  789  790  791  792  793  794  795  796  797  798
   0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
 799  800  801  802  803  804  805  806  807  808  809  810  811  812  813  814  815  816  817  818  819
   0    1    0    0    1    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
 820  821  822  823  824  825  826  827  828  829  830  831  832  833  834  835  836  837  838  839  840
   0    1    0    1    0    0    0    0    0    0    1    0    0    0    0    0    0    0    0    0    0
 841  842  843  844  845  846  847  848  849  850  851  852  853  854  855  856  857  858  859  860  861
 862  863  864  865  866  867  868  869  870  871  872  873  874  875  876  877  878  879  880  881  882
   0    0    1    0    0    1    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
 883  884  885  886  887  888  889  890  891  892  893  894  895  896  897  898  899  900  901  902  903
   0    0    0    1    0    0    0    0    0    0    0    1    0    0    0    0    0    0    0    0    0
 904  905  906  907  908  909  910  911  912  913  914  915  916  917  918  919  920  921  922  923  924
   0    0    0    0    0    0    0    0    1    0    0    0    0    0    0    0    0    0    0    0    0
 925  926  927  928  929  930  931  932  933  934  935  936  937  938  939  940  941  942  943  944  945
   0    0    0    0    0    0    0    0    1    1    1    1    0    0    0    0    0    0    0    0    0
 946  947  948  949  950  951  952  953  954  955  956  957  958  959  960  961  962  963  964  965  966
   0    0    0    0    0    0    0    0    0    1    0    0    0    1    1    0    0    0    0    0    0
 967  968  969  970  971  972  973  974  975  976  977  978  979  980  981  982  983  984  985  986  987
   0    1    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
 988  989  990  991  992  993  994  995  996  997  998  999 1000
   0    0    0    0    0    0    0    0    0    0    0    0    0
 [ reached getOption("max.print") -- omitted 8000 entries ]
Levels: 0 1
```

Description of Tree:

```
> tree
n= 21000

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

1) root 21000 4575 0 (0.7821429 0.2178571)
  2) PAY_0< 1.5 18840 3066 0 (0.8372611 0.1627389) *
  3) PAY_0>=1.5 2160  651 1 (0.3013889 0.6986111) *
>
```

Tree Diagram:



Confusion Matrix of Tree Model:

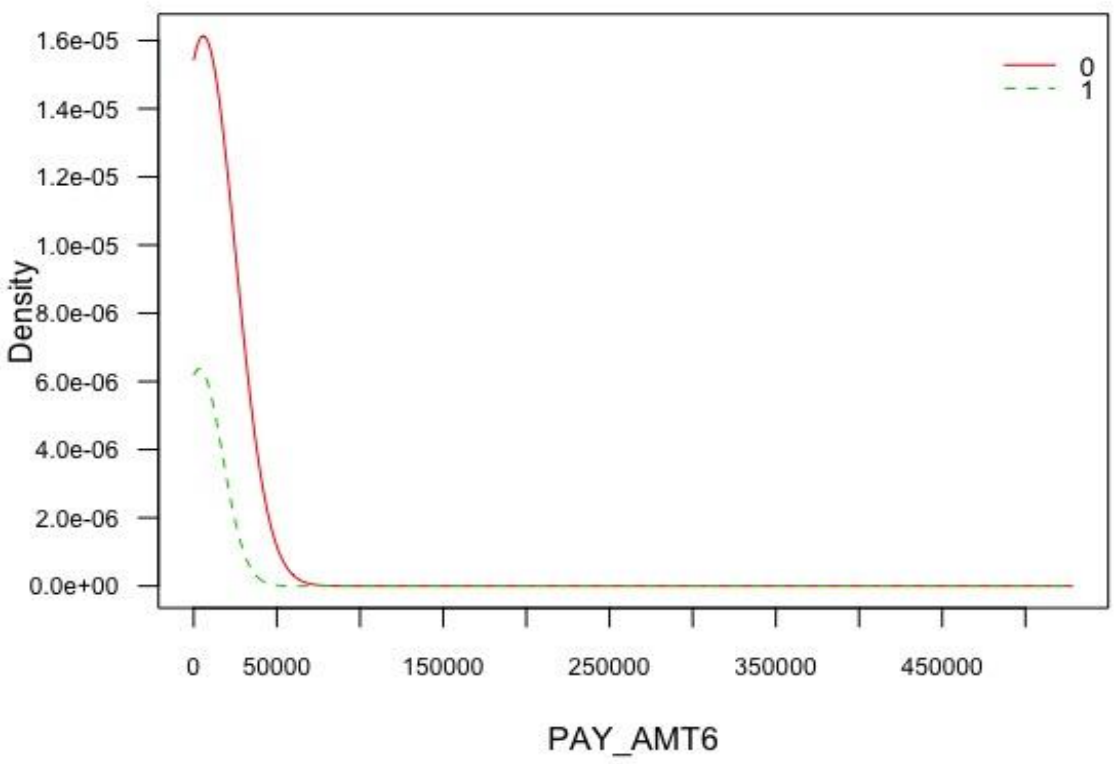
```
> confusionMatrix(t)
Confusion Matrix and Statistics

   pred
      0    1
  0 6637  302
  1 1393  668

               Accuracy : 0.8117
                 95% CI : (0.8034, 0.8197)
    No Information Rate : 0.8922
    P-Value [Acc > NIR] : 1

                  Kappa : 0.3447

 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 0.8265
            Specificity : 0.6887
         Pos Pred Value : 0.9565
         Neg Pred Value : 0.3241
             Prevalence : 0.8922
         Detection Rate : 0.7374
   Detection Prevalence : 0.7710
      Balanced Accuracy : 0.7576

       'Positive' Class : 0

>
```

5.2.2 Naïve Bayes

```
============================================== Naive Bayes ==============================================

 Call:
naive_bayes.formula(formula = default_payment_next_month ~ .,
    data = train)

-----------------------------------------------------------------------------------------------------------

Laplace smoothing: 0

-----------------------------------------------------------------------------------------------------------

 A priori probabilities:

        0         1
0.7792935 0.2207065

-----------------------------------------------------------------------------------------------------------

 Tables:

-----------------------------------------------------------------------------------------------------------
 ::: ID (Gaussian)
-----------------------------------------------------------------------------------------------------------

ID             0         1
  mean 15046.953 14748.813
  sd    8678.196  8570.251

-----------------------------------------------------------------------------------------------------------
 ::: LIMIT_BAL (Gaussian)
-----------------------------------------------------------------------------------------------------------

LIMIT_BAL        0        1
     mean 178652.7 130006.7
     sd   131633.4 115579.3

-----------------------------------------------------------------------------------------------------------
```

```
 ::: SEX (Gaussian)
----------------------------------------------------------------------------------------------------------------

SEX            0         1
  mean 1.6149931 1.5663420
  sd   0.4866085 0.4956208

----------------------------------------------------------------------------------------------------------------
 ::: EDUCATION (Gaussian)
----------------------------------------------------------------------------------------------------------------

EDUCATION         0         1
     mean 1.8425534 1.8936849
     sd   0.8101268 0.7211269

----------------------------------------------------------------------------------------------------------------
 ::: MARRIAGE (Gaussian)
----------------------------------------------------------------------------------------------------------------

MARRIAGE         0         1
    mean 1.5580555 1.5268727
    sd   0.5209389 0.5252247

----------------------------------------------------------------------------------------------------------------

# ... and 19 more tables

----------------------------------------------------------------------------------------------------------------

> |
```

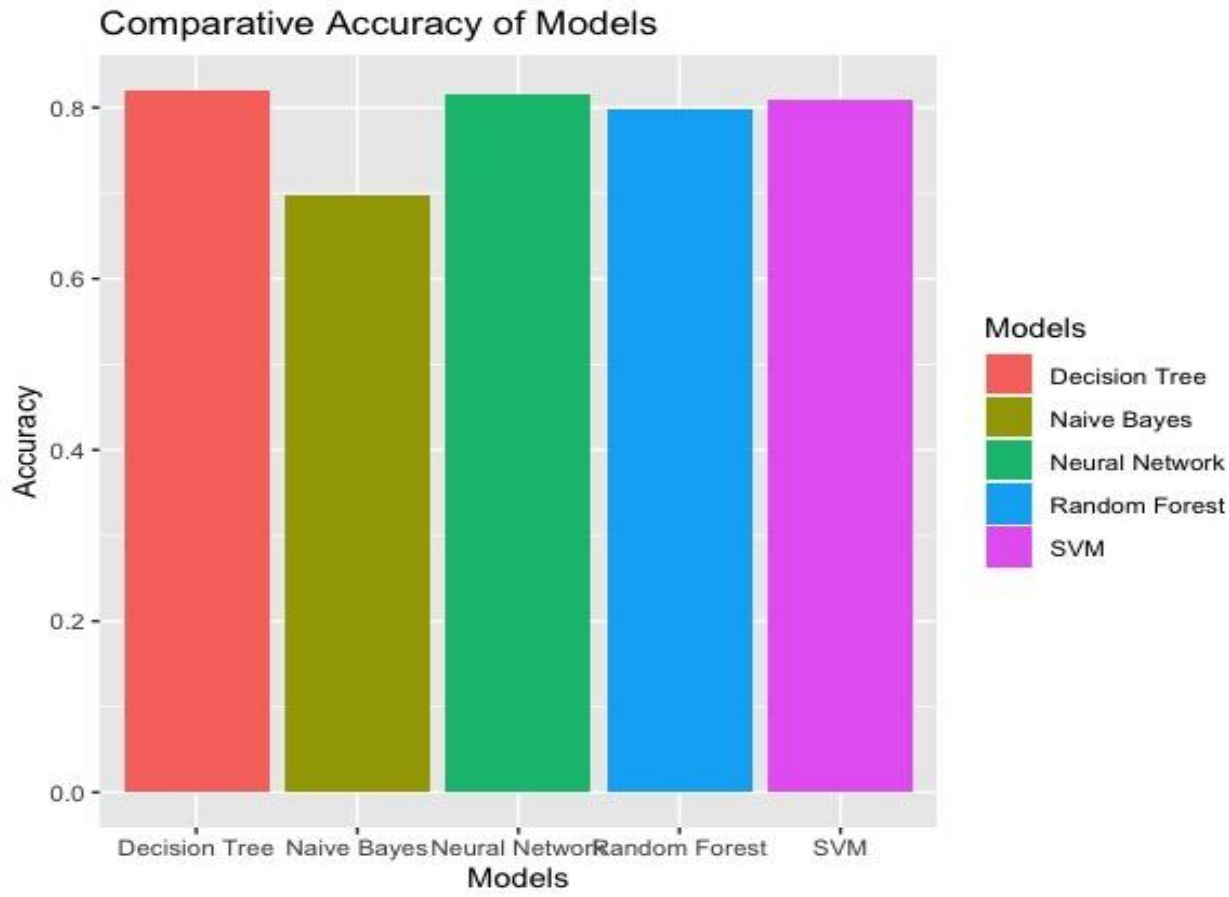
Plotting of Naïve Bayes Model:



Confusion Matrix of Naïve Bayes Algorithm:

```
> confusionMatrix(x)
Confusion Matrix and Statistics

   ap
       0    1
  0 4089 1752
  1  540 1119

               Accuracy : 0.6944
                 95% CI : (0.6838, 0.7048)
    No Information Rate : 0.6172
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.2969

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

            Sensitivity : 0.8833
            Specificity : 0.3898
         Pos Pred Value : 0.7001
         Neg Pred Value : 0.6745
             Prevalence : 0.6172
         Detection Rate : 0.5452
   Detection Prevalence : 0.7788
      Balanced Accuracy : 0.6366

       'Positive' Class : 0

> |
```

5.2.3 Comparison of Accuracy



6. Members Roles and Responsibilities

6.1 Group Rules

- Every group member has to listen to the group leader.
- Tasks will be divided by the group leader to all members and it will be mandatory for all to complete the assigned tasks.
- There will be a meeting regarding the project twice a week and it will be compulsory to attend the meeting.
- All group members will check and understand the work done by other members of the group.
- If any of group member is unable to do the assigned task then he should let the group leader know about it before the deadline.
- Regular update regarding the progress of project is compulsory for everyone.
- All must be punctual for the meeting.

6.2 Task List

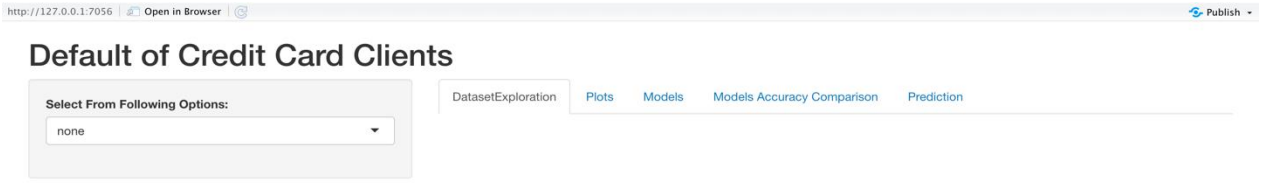
No. Activities
N1 Finding Dataset
N2 Understanding/Exploring Dataset
N3 Learning and applying cleaning process on Dataset
N4 Task Distribution
N5 Designing the application for Dataset
N6 Finding/Designing best Machine Learning Algorithm for the application
N7 Implementing Machine Learning Algorithms on Application
N8 Turning project to shiny application
N9 System Testing
N10 Final Touchups
N11 Preparing for Project Presentation

6.3 Matrix Score Weighted

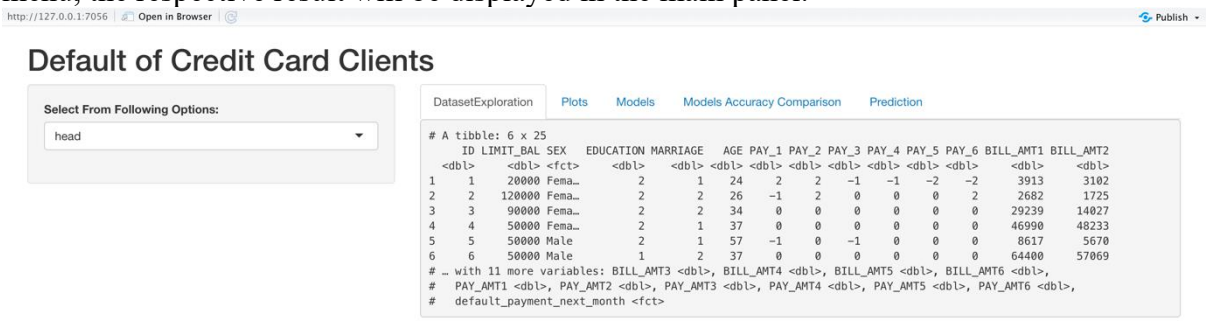
Members	Report	Dataset Cleaning	Machine Learning Modelling	Data Analysis and Visualization	Shiny Application
Mohsin Basti	80%	75%	75%	75%	Pending
Abdullah Shafique	20%	25%	25%	25%	Pending
Rehmat Ali Haider	0%	0%	0%	0%	Pending

7. Shiny Application Screen shots and published URL

Shiny application we have made for our project is interactive. It consists of 5 tabs name as “DataExploration”, “Plots”, “Models”, “Models Accuracy Comparison”, and “Prediction”.

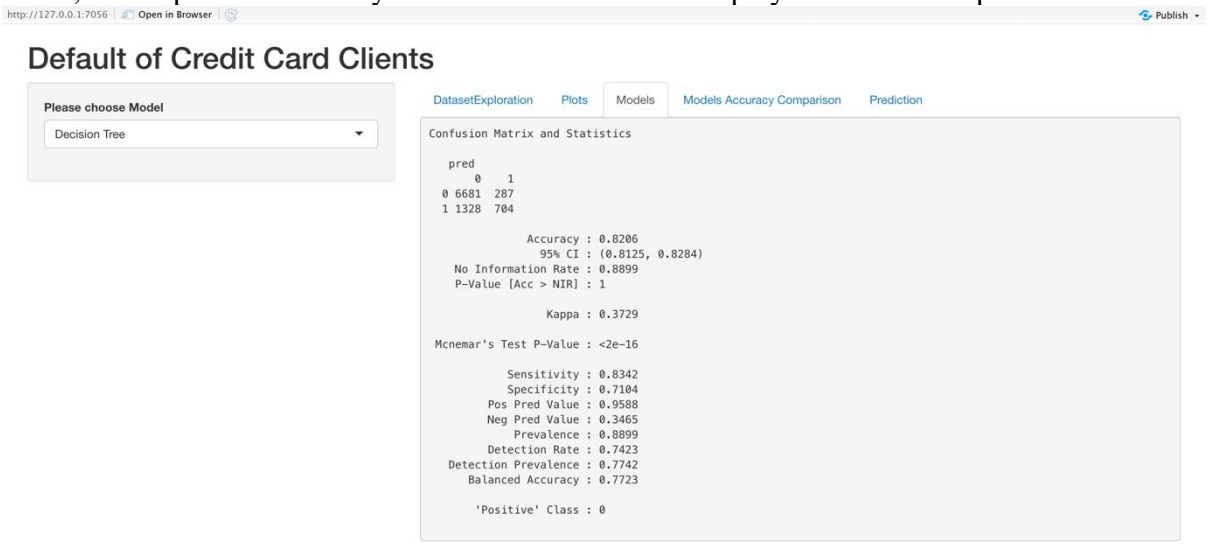


First tab “DataExploration” have a drop-down menu in the sidebar, drop-down menu has many options for data exploration. After selecting data exploration type from drop-down menu, the respective result will be displayed in the main panel.



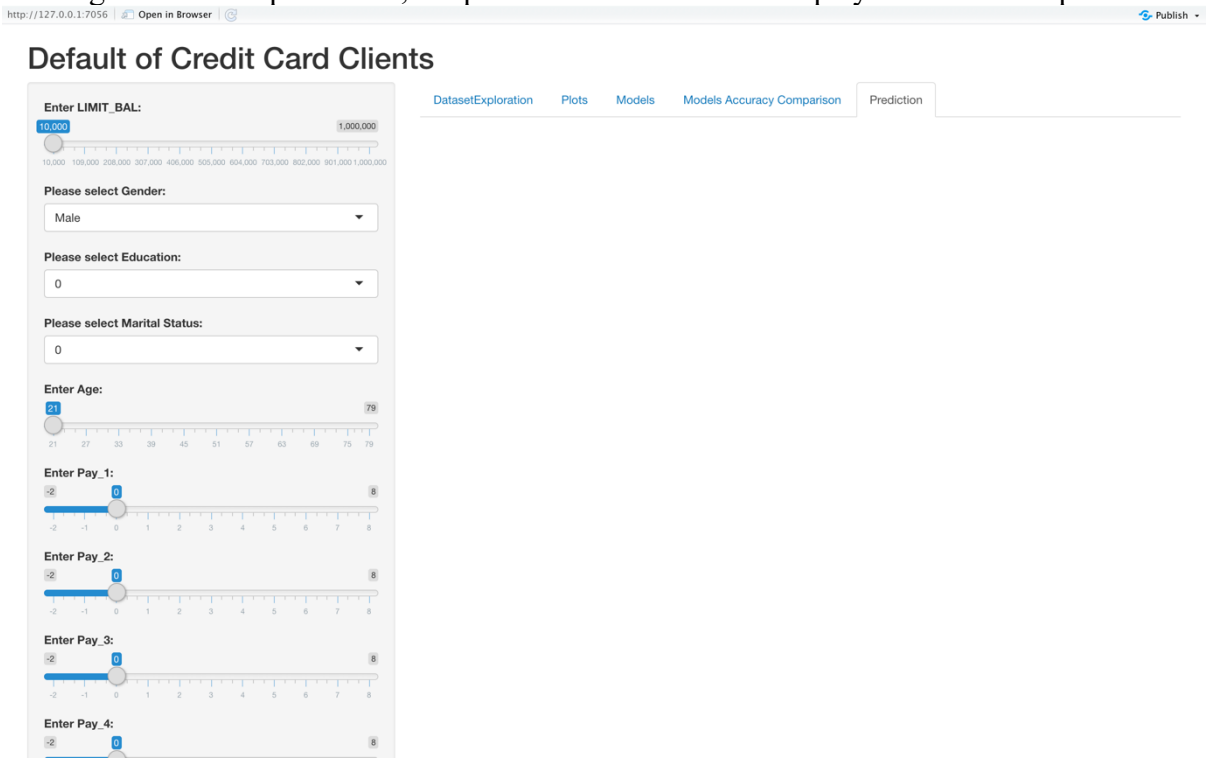
There comes “Plot” tab, which have a drop-down menu in the sidebar, drop-down menu has many options for different type of graphs. After selecting plot type from drop-down menu, the respective plot will be displayed in the main panel. Some graph types have further plot options for attributes.

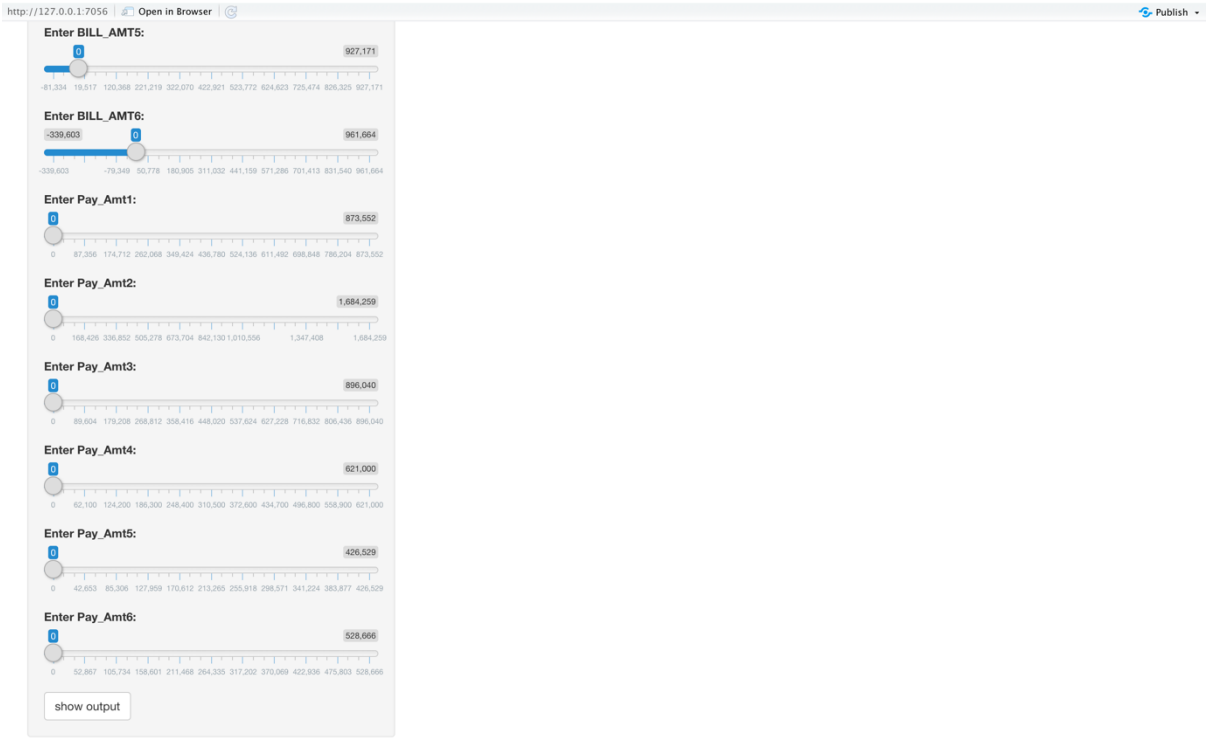
Then it comes “Models” tab, which have a drop-down menu in the sidebar, drop-down menu has many options for different type of models. After selecting model type from drop-down menu, the respective accuracy detail of model will be displayed in the main panel.

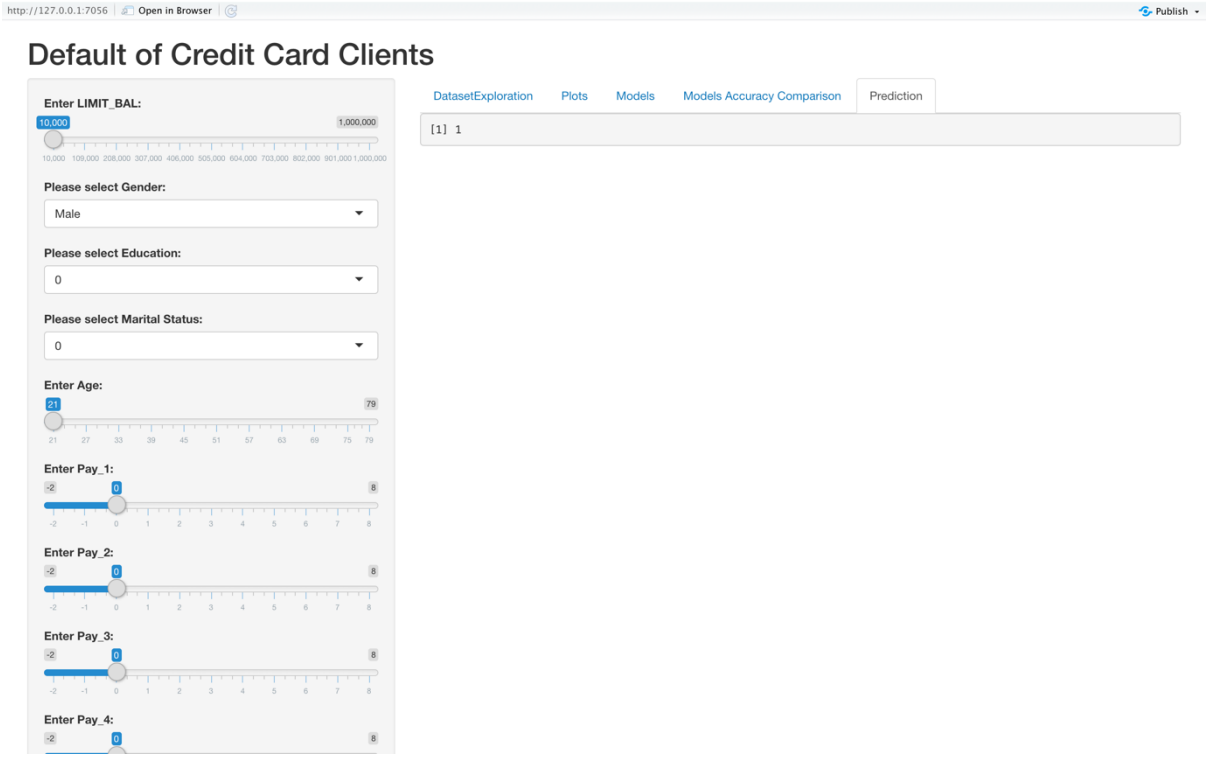


There comes “Models Accuracy Comparison” tab, which will display the accuracy comparison graph of all models trained in our project.

There comes the “Prediction” tab, in which we will adjust different inputs according to our requirements. Then according to selected inputs, the prediction will be done and after clicking on show output button, the prediction result will be displayed in the main panel.




http://127.0.0.1:7056
Open in Browser
Publish
Enter BILL_AMT5:
0
927,171
-81,334 19,517 120,368 221,219 322,070 422,921 523,772 624,623 725,474 826,325 927,171
Enter BILL_AMT6:
-339,603
0
961,664
-339,603 -79,349 50,778 180,905 311,032 441,159 571,286 701,413 831,540 961,664
Enter Pay_Amt1:
0
873,552
0 87,356 174,712 262,068 349,424 436,780 524,136 611,492 698,848 786,204 873,552
Enter Pay_Amt2:
0
1,684,259
0 168,426 336,852 505,278 673,704 842,130 1,010,556 1,347,408 1,684,259
Enter Pay_Amt3:
0
896,040
0 89,604 179,208 268,812 358,416 448,020 537,624 627,228 716,832 806,436 896,040
Enter Pay_Amt4:
0
621,000
0 62,100 124,200 186,300 248,400 310,500 372,600 434,700 496,800 558,900 621,000
Enter Pay_Amt5:
0
426,529
0 42,653 85,306 127,959 170,612 213,265 255,918 298,571 341,224 383,877 426,529
Enter Pay_Amt6:
0
528,666
0 52,867 105,734 158,601 211,468 264,335 317,202 370,069 422,936 475,803 528,666
show output


http://127.0.0.1:7056
Open in Browser
Publish
Default of Credit Card Clients
Enter LIMIT_BAL:
10,000
1,000,000
10,000 109,000 208,000 307,000 406,000 505,000 604,000 703,000 802,000 901,000 1,000,000
Please select Gender:
Male
Please select Education:
0
Please select Marital Status:
0
Enter Age:
21
79
21 27 33 39 45 51 57 63 69 75 79
Enter Pay_1:
-2
0
8
-2 -1 0 1 2 3 4 5 6 7 8
Enter Pay_2:
-2
0
8
-2 -1 0 1 2 3 4 5 6 7 8
Enter Pay_3:
-2
0
8
-2 -1 0 1 2 3 4 5 6 7 8
Enter Pay_4:
-2
0
8
DatasetExploration
Plots
Models
Models Accuracy Comparison
Prediction
[1] 1