Introduction

A Microfinance Institution (MFI) is an organization that offers financial services to low income populations. MFS becomes very useful when targeting especially the unbanked poor families living in remote areas with not much sources of income. The Microfinance services (MFS) provided by MFI are Group Loans, Agricultural Loans, Individual Business Loans and so on. Many microfinance institutions (MFI), experts and donorsare supporting the idea of using mobile financial services (MFS) which they feel are more convenient and efficient, and cost saving, than the traditional high-touch model used since long for the purpose of delivering microfinance services. Though, the MFI industry is primarily focusing on low income families and are very useful in such areas, the implementation of MFShas been uneven with both significant challenges and successes. Today, microfinance is widely accepted as a poverty-reduction tool, representing $70 billion in outstanding loans and a global outreach of 200 million clients. We are working with one such client that is in Telecom Industry. They are a fixed wireless telecommunications network provider. They have launched various products and have developed its business and organization based on the budget operator model, offering better products at Lower Prices to all value conscious customers through a strategy of disruptive innovation that focuses on the subscriber. They understand the importance of communication and how it affects a person's life, thus, focusing on providing their services and products to low income families and poor customers that can help them in the need of hour. They arecollaborating with an MFI to provide micro-credit on mobile balances to be paid back in 5 days. The Consumer is believed to be defaulter if he deviates from the path of paying back the loaned amount within the time duration of 5 days. For the loan amount of 5 (in Indonesian Rupiah), payback amount should be6(in Indonesian Rupiah), while, for the loan amount of 10(in Indonesian Rupiah), the payback amount should be 12(in Indonesian Rupiah). The sample data is provided to us from our client database. It is hereby given to you for this exercise. In order to improve the selection of customers for the credit, the client wants some predictions that could help them in further investment and improvement in selection of customers.

Feature Description label :Flag indicating whether the user paid back the credit amount within 5 days of issuing the loan{1:success, 0:failure} msisdn :mobile number of user aon :age on cellular network in days daily_decr30 :Daily amount spent from main account, averaged over last 30 days (in Indonesian Rupiah) daily_decr90 :Daily amount spent from main account, averaged over last 90 days (in Indonesian Rupiah) rental30 :Average main account balance over last 30 days rental90 :Average main account balance over last 90 days last_rech_date_ma :Number of days till last recharge of main account last_rech_date_da : Number of days till last recharge of data account last_rech_amt_ma : Amount of last recharge of main account (in Indonesian Rupiah) cnt_ma_rech30 : Number of times main account got recharged in last 30 days fr_ma_rech30 : Frequency of main account recharged in last 30 days sumamnt_ma_rech30 : Total amount of recharge in main account over last 30 days (in Indonesian Rupiah) medianamnt_ma_rech30 : Median of amount of recharges done in main account over last 30 days at user level (in Indonesian Rupiah) medianmarechprebal30 : Median of main account balance just before recharge in last 30 days at user level (in Indonesian Rupiah) cnt_ma_rech90 : Number of times main account got recharged in last 90 days fr_ma_rech90 : Frequency of main account recharged in last 90 days sumamnt_ma_rech90: Total amount of recharge in main account over last 90 days (in Indian Rupee) medianamnt_ma_rech90 :Median of amount of recharges done in main account over last 90 days at user level (in Indian Rupee) medianmarechprebal90 : Median of main account balance just before recharge in last 90 days at user level (in Indian Rupee) cnt_da_rech30 : Number of times data account got recharged in last 30 days fr_da_rech30 : Frequency of data account recharged in last 30 days cnt_da_rech90 : Number of times data account got recharged in last 90 days fr_da_rech90 : Frequency of data account recharged in last 90 days cnt_loans30 : Number of loans taken by user in last 30 days amnt_loans30 : Total amount of loans taken by user in last 30 days maxamnt_loans30 : maximum amount of loan taken by the user in last 30 days medianamnt_loans30 : Median of amounts of loan taken by the user in last 30 days cnt_loans90: Number of loans taken by user in last 90 days amnt_loans90 :Total amount of loans taken by user in last 90 days maxamnt_loans90 : maximum amount of loan taken by the user in last 90 days medianamnt_loans90: Median of amounts of loan taken by the user in last 90 days payback30 :Average payback time in days over last 30 days payback90: Average payback time in days over last 90 days pcircle: telecom circle pdate :date

```
#Data Reading and Analysis

#importing libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

#Reading data
df=pd.read_csv('Data file.csv')
df.head()
```

```
   Unnamed: 0  label       msisdn    aon  daily_decr30  
daily_decr90  \
0           1      0  21408I70789  272.0   3055.050000   3065.150000

1           2      1  76462I70374  712.0  12122.000000  12124.750000

2           3      1  17943I70372  535.0   1398.000000   1398.000000

3           4      1  55773I70781  241.0     21.228000     21.228000

4           5      1  03813I82730  947.0    150.619333    150.619333


   rental30  rental90  last_rech_date_ma  last_rech_date_da  ...  \
0    220.13    260.13                2.0                0.0  ...   
1   3691.26   3691.26               20.0                0.0  ...   
2    900.13    900.13                3.0                0.0  ...   
```

```
3    159.42    159.42               41.0               0.0  ...
4   1098.90   1098.90                4.0               0.0  ...

   maxamnt_loans30  medianamnt_loans30  cnt_loans90  amnt_loans90  \
0              6.0                 0.0          2.0            12
1             12.0                 0.0          1.0            12
2              6.0                 0.0          1.0             6
3              6.0                 0.0          2.0            12
4              6.0                 0.0          7.0            42

   maxamnt_loans90  medianamnt_loans90  payback30  payback90  pcircle
\
0                6                 0.0  29.000000  29.000000      UPW

1               12                 0.0   0.000000   0.000000      UPW

2                6                 0.0   0.000000   0.000000      UPW

3                6                 0.0   0.000000   0.000000      UPW

4                6                 0.0   2.333333   2.333333      UPW


        pdate
0  2016-07-20
1  2016-08-10
2  2016-08-19
3  2016-06-06
4  2016-06-22

[5 rows x 37 columns]
```

```
df.drop('Unnamed: 0',axis=1,inplace=True)
```

```
df.head()
```

```
   label       msisdn    aon  daily_decr30  daily_decr90  rental30
rental90  \
0      0  21408I70789  272.0   3055.050000   3065.150000    220.13
260.13
1      1  76462I70374  712.0  12122.000000  12124.750000   3691.26
3691.26
2      1  17943I70372  535.0   1398.000000   1398.000000    900.13
900.13
3      1  55773I70781  241.0     21.228000     21.228000    159.42
159.42
4      1  03813I82730  947.0    150.619333    150.619333   1098.90
1098.90

   last_rech_date_ma  last_rech_date_da  last_rech_amt_ma  ...  \
```

```
0               2.0                  0.0              1539  ...
1              20.0                  0.0              5787  ...
2               3.0                  0.0              1539  ...
3              41.0                  0.0               947  ...
4               4.0                  0.0              2309  ...

   maxamnt_loans30  medianamnt_loans30  cnt_loans90  amnt_loans90  \
0              6.0                 0.0          2.0            12
1             12.0                 0.0          1.0            12
2              6.0                 0.0          1.0             6
3              6.0                 0.0          2.0            12
4              6.0                 0.0          7.0            42

   maxamnt_loans90  medianamnt_loans90  payback30  payback90  pcircle
\
0                6                 0.0  29.000000  29.000000      UPW

1               12                 0.0   0.000000   0.000000      UPW

2                6                 0.0   0.000000   0.000000      UPW

3                6                 0.0   0.000000   0.000000      UPW

4                6                 0.0   2.333333   2.333333      UPW


        pdate
0  2016-07-20
1  2016-08-10
2  2016-08-19
3  2016-06-06
4  2016-06-22

[5 rows x 36 columns]
```

```
#let's dive into depth
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 209593 entries, 0 to 209592
Data columns (total 36 columns):
 #   Column                Non-Null Count   Dtype
---  ------                --------------   -----
 0   label                 209593 non-null  int64
 1   msisdn                209593 non-null  object
 2   aon                   209593 non-null  float64
 3   daily_decr30          209593 non-null  float64
 4   daily_decr90          209593 non-null  float64
 5   rental30              209593 non-null  float64
 6   rental90              209593 non-null  float64
```

```
 7   last_rech_date_ma     209593 non-null  float64
 8   last_rech_date_da     209593 non-null  float64
 9   last_rech_amt_ma      209593 non-null  int64
 10  cnt_ma_rech30         209593 non-null  int64
 11  fr_ma_rech30          209593 non-null  float64
 12  sumamnt_ma_rech30     209593 non-null  float64
 13  medianamnt_ma_rech30  209593 non-null  float64
 14  medianmarechprebal30  209593 non-null  float64
 15  cnt_ma_rech90         209593 non-null  int64
 16  fr_ma_rech90          209593 non-null  int64
 17  sumamnt_ma_rech90     209593 non-null  int64
 18  medianamnt_ma_rech90  209593 non-null  float64
 19  medianmarechprebal90  209593 non-null  float64
 20  cnt_da_rech30         209593 non-null  float64
 21  fr_da_rech30          209593 non-null  float64
 22  cnt_da_rech90         209593 non-null  int64
 23  fr_da_rech90          209593 non-null  int64
 24  cnt_loans30           209593 non-null  int64
 25  amnt_loans30          209593 non-null  int64
 26  maxamnt_loans30       209593 non-null  float64
 27  medianamnt_loans30    209593 non-null  float64
 28  cnt_loans90           209593 non-null  float64
 29  amnt_loans90          209593 non-null  int64
 30  maxamnt_loans90       209593 non-null  int64
 31  medianamnt_loans90    209593 non-null  float64
 32  payback30             209593 non-null  float64
 33  payback90             209593 non-null  float64
 34  pcircle               209593 non-null  object
 35  pdate                 209593 non-null  object
dtypes: float64(21), int64(12), object(3)
memory usage: 57.6+ MB
```

```
# let's check null values
df.isnull().sum()
```

```
label                   0
msisdn                  0
aon                     0
daily_decr30            0
daily_decr90            0
rental30                0
rental90                0
last_rech_date_ma       0
last_rech_date_da       0
last_rech_amt_ma        0
cnt_ma_rech30           0
fr_ma_rech30            0
sumamnt_ma_rech30       0
medianamnt_ma_rech30    0
medianmarechprebal30    0
cnt_ma_rech90           0
```

```
fr_ma_rech90            0
sumamnt_ma_rech90       0
medianamnt_ma_rech90    0
medianmarechprebal90    0
cnt_da_rech30           0
fr_da_rech30            0
cnt_da_rech90           0
fr_da_rech90            0
cnt_loans30             0
amnt_loans30            0
maxamnt_loans30         0
medianamnt_loans30      0
cnt_loans90             0
amnt_loans90            0
maxamnt_loans90         0
medianamnt_loans90      0
payback30               0
payback90               0
pcircle                 0
pdate                   0
dtype: int64
```

```
print("shape of data set is ",df.shape)
```

```
shape of data set is  (209593, 36)
```

Data Preprocessing

#1' Remove columns where number of unique value is only 1. Let's look at no of unique values for each column.We will remove all columns where number of unique value is only 1 because that will not make any sense in the analysis

```
unique = df.nunique()
unique = unique[unique.values == 1]
```

```
df.drop(labels = list(unique.index), axis =1, inplace=True)
print("So now we are left with",df.shape ,"rows & columns.")
```

```
So now we are left with (209593, 35) rows & columns.
```

```
df.head()
```

```
   label       msisdn    aon  daily_decr30  daily_decr90  rental30  
rental90  \
0      0  21408I70789  272.0   3055.050000   3065.150000    220.13   
260.13
1      1  76462I70374  712.0  12122.000000  12124.750000   3691.26   
3691.26
2      1  17943I70372  535.0   1398.000000   1398.000000    900.13   
900.13
```

```
3      1  55773I70781  241.0     21.228000     21.228000    159.42
159.42
4      1  03813I82730  947.0    150.619333    150.619333   1098.90
1098.90

   last_rech_date_ma  last_rech_date_da  last_rech_amt_ma  ...
amnt_loans30  \
0                2.0                0.0              1539  ...
12
1               20.0                0.0              5787  ...
12
2                3.0                0.0              1539  ...
6
3               41.0                0.0               947  ...
12
4                4.0                0.0              2309  ...
42

   maxamnt_loans30  medianamnt_loans30  cnt_loans90  amnt_loans90  \
0              6.0                 0.0          2.0            12
1             12.0                 0.0          1.0            12
2              6.0                 0.0          1.0             6
3              6.0                 0.0          2.0            12
4              6.0                 0.0          7.0            42

   maxamnt_loans90  medianamnt_loans90  payback30  payback90
pdate
0                6                 0.0  29.000000  29.000000  2016-07-
20
1               12                 0.0   0.000000   0.000000  2016-08-
10
2                6                 0.0   0.000000   0.000000  2016-08-
19
3                6                 0.0   0.000000   0.000000  2016-06-
06
4                6                 0.0   2.333333   2.333333  2016-06-
22

[5 rows x 35 columns]
```

```
df.describe().transpose()
```

```
                         count          mean           std
min  \
label                 209593.0      0.875177      0.330519
0.000000
aon                   209593.0   8112.343445  75696.082531    -
48.000000
daily_decr30          209593.0   5381.402289   9220.623400    -
93.012667
```

```
daily_decr90          209593.0   6082.515068  10918.812767    -
93.012667
rental30              209593.0   2692.581910   4308.586781 -
23737.140000
rental90              209593.0   3483.406534   5770.461279 -
24720.580000
last_rech_date_ma     209593.0   3755.847800  53905.892230    -
29.000000
last_rech_date_da     209593.0   3712.202921  53374.833430    -
29.000000
last_rech_amt_ma      209593.0   2064.452797   2370.786034
0.000000
cnt_ma_rech30         209593.0      3.978057      4.256090
0.000000
fr_ma_rech30          209593.0   3737.355121  53643.625172
0.000000
sumamnt_ma_rech30     209593.0   7704.501157  10139.621714
0.000000
medianamnt_ma_rech30  209593.0   1812.817952   2070.864620
0.000000
medianmarechprebal30  209593.0   3851.927942  54006.374433   -
200.000000
cnt_ma_rech90         209593.0      6.315430      7.193470
0.000000
fr_ma_rech90          209593.0      7.716780     12.590251
0.000000
sumamnt_ma_rech90     209593.0  12396.218352  16857.793882
0.000000
medianamnt_ma_rech90  209593.0   1864.595821   2081.680664
0.000000
medianmarechprebal90  209593.0     92.025541    369.215658   -
200.000000
cnt_da_rech30         209593.0    262.578110   4183.897978
0.000000
fr_da_rech30          209593.0   3749.494447  53885.414979
0.000000
cnt_da_rech90         209593.0      0.041495      0.397556
0.000000
fr_da_rech90          209593.0      0.045712      0.951386
0.000000
cnt_loans30           209593.0      2.758981      2.554502
0.000000
amnt_loans30          209593.0     17.952021     17.379741
0.000000
maxamnt_loans30       209593.0    274.658747   4245.264648
0.000000
medianamnt_loans30    209593.0      0.054029      0.218039
0.000000
cnt_loans90           209593.0     18.520919    224.797423
0.000000
```

```
amnt_loans90          209593.0     23.645398     26.469861
0.000000
maxamnt_loans90       209593.0      6.703134      2.103864
0.000000
medianamnt_loans90    209593.0      0.046077      0.200692
0.000000
payback30             209593.0      3.398826      8.813729
0.000000
payback90             209593.0      4.321485     10.308108
0.000000

                           25%          50%       75%          max
label                    1.000     1.000000      1.00     1.000000
aon                    246.000   527.000000    982.00 999860.755168
daily_decr30            42.440  1469.175667   7244.00 265926.000000
daily_decr90            42.692  1500.000000   7802.79 320630.000000
rental30               280.420  1083.570000   3356.94 198926.110000
rental90               300.260  1334.000000   4201.79 200148.110000
last_rech_date_ma        1.000     3.000000      7.00 998650.377733
last_rech_date_da        0.000     0.000000      0.00 999171.809410
last_rech_amt_ma       770.000  1539.000000   2309.00  55000.000000
cnt_ma_rech30            1.000     3.000000      5.00    203.000000
fr_ma_rech30             0.000     2.000000      6.00 999606.368132
sumamnt_ma_rech30     1540.000  4628.000000  10010.00 810096.000000
medianamnt_ma_rech30   770.000  1539.000000   1924.00  55000.000000
medianmarechprebal30    11.000    33.900000     83.00 999479.419319
cnt_ma_rech90            2.000     4.000000      8.00    336.000000
fr_ma_rech90             0.000     2.000000      8.00     88.000000
sumamnt_ma_rech90     2317.000  7226.000000  16000.00 953036.000000
medianamnt_ma_rech90   773.000  1539.000000   1924.00  55000.000000
medianmarechprebal90    14.600    36.000000     79.31  41456.500000
cnt_da_rech30            0.000     0.000000      0.00  99914.441420
fr_da_rech30             0.000     0.000000      0.00 999809.240107
cnt_da_rech90            0.000     0.000000      0.00     38.000000
fr_da_rech90             0.000     0.000000      0.00     64.000000
cnt_loans30              1.000     2.000000      4.00     50.000000
amnt_loans30             6.000    12.000000     24.00    306.000000
maxamnt_loans30          6.000     6.000000      6.00  99864.560864
medianamnt_loans30       0.000     0.000000      0.00      3.000000
cnt_loans90              1.000     2.000000      5.00   4997.517944
amnt_loans90             6.000    12.000000     30.00    438.000000
maxamnt_loans90          6.000     6.000000      6.00     12.000000
medianamnt_loans90       0.000     0.000000      0.00      3.000000
payback30                0.000     0.000000      3.75    171.500000
payback90                0.000     1.666667      4.50    171.500000
```

```
#Here we check the summary of object and datetime columns
df.describe(include=['object','datetime']).transpose()
```

```
        count  unique          top  freq
msisdn  209593  186243  04581I85330     7
pdate   209593      82   2016-07-04  3150
```

Observation:

Summary statistics shows all the statistics of our dataset i.e. mean, median and other calculation. Mean is greater than median in all the columns so aur data is right skewed. The difference between 75% and maximum is higher that's why outliers are removed which needs to be removed. The pdate column tells the date when the data is collect. It contains only three month data. msidn is a mobile number of user and mobile number is unique for every customers. There are only 186243 unique number out of 209593 so rest of the data is duplicates entry so we have to remove those entry.

```
df1=df.copy()
```

```
#Deleting the duplicates entry in msidn column
df = df.drop_duplicates(subset = 'msisdn',keep='first')
df.shape
```

```
(186243, 35)
```

```
#Data Exploration
```

```
#Printing the object datatypes and their unique values.

for column in df.columns:
    if df[column].dtypes == object:
        print(str(column) + ' : ' + str(df[column].unique()))

print('*****************************************************************
*******************************************')
        print('\n')
```

```
msisdn : ['21408I70789' '76462I70374' '17943I70372' ... '22758I85348'
'59712I82733'
 '65061I85339']
*****************************************************************
************************************


pdate : ['2016-07-20' '2016-08-10' '2016-08-19' '2016-06-06' '2016-06-
22'
 '2016-07-02' '2016-07-05' '2016-08-05' '2016-06-15' '2016-06-08'
 '2016-06-12' '2016-06-20' '2016-06-29' '2016-06-16' '2016-08-03'
 '2016-06-24' '2016-07-04' '2016-07-03' '2016-07-01' '2016-08-08'
 '2016-06-26' '2016-06-23' '2016-07-06' '2016-07-09' '2016-06-10'
 '2016-06-07' '2016-06-27' '2016-08-11' '2016-06-30' '2016-06-19'
 '2016-07-26' '2016-08-14' '2016-06-14' '2016-06-21' '2016-06-25'
```

```
 '2016-06-28' '2016-06-11' '2016-07-27' '2016-07-23' '2016-08-16'
 '2016-08-15' '2016-06-02' '2016-06-05' '2016-08-02' '2016-07-28'
 '2016-07-18' '2016-08-18' '2016-07-16' '2016-07-29' '2016-07-21'
 '2016-06-03' '2016-06-13' '2016-08-01' '2016-07-13' '2016-07-10'
 '2016-06-09' '2016-07-15' '2016-07-11' '2016-08-09' '2016-08-12'
 '2016-07-22' '2016-06-04' '2016-07-24' '2016-06-18' '2016-08-13'
 '2016-06-17' '2016-08-07' '2016-07-12' '2016-08-06' '2016-07-19'
 '2016-08-21' '2016-08-04' '2016-07-25' '2016-07-30' '2016-08-17'
 '2016-07-08' '2016-07-14' '2016-06-01' '2016-07-07' '2016-07-17'
 '2016-07-31' '2016-08-20']
***********************************************************************
************************************
```

Observation:

contains only one circle area data. So it have not any impact in our model if we drop this feature.

```
#Printing the float datatype columns and number of unique values in
the particular columns.

for column in df.columns:
    if df[column].dtype==np.number:
        print(str(column) + ' : ' + str(df[column].nunique()))
        print(df[column].nunique())

print('//////**********************************************************
**********************///////')
```

```
aon : 4282
4282
//////*****************************************************************
***************///////
daily_decr30 : 130323
130323
//////*****************************************************************
***************///////

C:\Users\hamsa\AppData\Local\Temp/ipykernel_25188/319834993.py:4:
DeprecationWarning: Converting `np.inexact` or `np.floating` to a
dtype is deprecated. The current result is `float64` which is not
strictly correct.
  if df[column].dtype==np.number:

daily_decr90 : 139842
139842
//////*****************************************************************
```

***************///////
rental30 : 117881
117881
//////**
***************///////
rental90 : 125595
125595
//////**
***************///////
last_rech_date_ma : 1061
1061
//////**
***************///////
last_rech_date_da : 1061
1061
//////**
***************///////
fr_ma_rech30 : 961
961
//////**
***************///////
sumamnt_ma_rech30 : 13130
13130
//////**
***************///////
medianamnt_ma_rech30 : 501
501
//////**
***************///////
medianmarechprebal30 : 28486
28486
//////**
***************///////
medianamnt_ma_rech90 : 602
602
//////**
***************///////
medianmarechprebal90 : 28064
28064
//////**
***************///////
cnt_da_rech30 : 949
949
//////**
***************///////
fr_da_rech30 : 960
960
//////**
***************///////
maxamnt_loans30 : 924

```
924
//////****************************************************************
***************///////
medianamnt_loans30 : 6
6
//////****************************************************************
***************///////
cnt_loans90 : 968
968
//////****************************************************************
***************///////
medianamnt_loans90 : 6
6
//////****************************************************************
***************///////
payback30 : 1249
1249
//////****************************************************************
***************///////
payback90 : 2128
2128
//////****************************************************************
***************///////
```

```
#Checking the number of number of defaulter and non defaulter customers.
df['label'].value_counts()
```

```
1    160383
0     25860
Name: label, dtype: int64
```

```
#Checking the defaulter customers percentage wise.
df['label'].value_counts(normalize=True) *100
```

```
1    86.114914
0    13.885086
Name: label, dtype: float64
```

Observation:

After seeing the label column which is also our target feature for this dataset it is clearly shown that 86.11% of data is label 1 and only 13.8% of data is label 0 so our dataset is implanced. So before making the ML model first we have to do sampling to get rid off imblance dataset.

```
#check cor-relation
df_cor = df.corr()
df_cor
```

	label	aon	daily_decr30	daily_decr90 \
label	1.000000	-0.004035	0.174901	0.173016
aon	-0.004035	1.000000	0.000630	0.000052
daily_decr30	0.174901	0.000630	1.000000	0.977659
daily_decr90	0.173016	0.000052	0.977659	1.000000
rental30	0.057207	-0.002930	0.427503	0.420561
rental90	0.075869	-0.002618	0.444932	0.457443
last_rech_date_ma	0.004113	0.001853	-0.000171	0.000058
last_rech_date_da	0.001814	-0.001796	-0.001311	-0.001484
last_rech_amt_ma	0.139969	0.004102	0.287181	0.275195
cnt_ma_rech30	0.244728	-0.004315	0.444365	0.419650
fr_ma_rech30	0.001129	-0.000436	0.000766	0.001091
sumamnt_ma_rech30	0.207727	-0.000397	0.630202	0.597542
medianamnt_ma_rech30	0.149780	0.004446	0.307440	0.294838
medianmarechprebal30	-0.004835	0.004221	-0.000854	-0.000688
cnt_ma_rech90	0.245941	-0.003957	0.576787	0.582115
fr_ma_rech90	0.094709	0.005517	-0.061858	-0.063740
sumamnt_ma_rech90	0.212666	0.000160	0.754042	0.759865
medianamnt_ma_rech90	0.129527	0.005022	0.269721	0.262627
medianmarechprebal90	0.041728	-0.001128	0.042276	0.041210
cnt_da_rech30	0.004184	0.002445	0.000312	-0.000128
fr_da_rech30	-0.000137	0.000806	-0.002442	-0.002189
cnt_da_rech90	0.003601	0.000868	0.038944	0.031408
fr_da_rech90	-0.005779	0.006379	0.019874	0.015944

cnt_loans30	0.197565	-0.003157	0.346504	0.321006
amnt_loans30	0.199916	-0.003302	0.454169	0.430940
maxamnt_loans30	-0.000274	-0.003096	0.001569	0.001283
medianamnt_loans30	0.050067	0.004679	-0.005629	0.000012
cnt_loans90	0.004305	0.000192	0.008865	0.009220
amnt_loans90	0.205065	-0.003336	0.542179	0.544854
maxamnt_loans90	0.086033	-0.000975	0.396803	0.394487
medianamnt_loans90	0.041265	0.002346	-0.031485	-0.029046
payback30	0.050892	0.002246	0.033669	0.025432
payback90	0.053776	0.002549	0.056822	0.050147

	rental30	rental90	last_rech_date_ma	\
label	0.057207	0.075869	0.004113	
aon	-0.002930	-0.002618	0.001853	
daily_decr30	0.427503	0.444932	-0.000171	
daily_decr90	0.420561	0.457443	0.000058	
rental30	1.000000	0.955233	-0.000949	
rental90	0.955233	1.000000	-0.001758	
last_rech_date_ma	-0.000949	-0.001758	1.000000	
last_rech_date_da	0.003294	0.002643	0.002629	
last_rech_amt_ma	0.128773	0.123436	-0.000754	
cnt_ma_rech30	0.220472	0.218618	0.006491	
fr_ma_rech30	0.000272	0.001057	-0.001165	
sumamnt_ma_rech30	0.258656	0.246626	0.002544	
medianamnt_ma_rech30	0.132083	0.122747	-0.002716	
medianmarechprebal30	-0.001112	-0.001047	0.004216	
cnt_ma_rech90	0.295746	0.329330	0.006131	
fr_ma_rech90	-0.022353	-0.024882	0.000881	
sumamnt_ma_rech90	0.324302	0.342772	0.002345	
medianamnt_ma_rech90	0.113115	0.106832	-0.001947	
medianmarechprebal90	0.029945	0.032886	-0.001506	
cnt_da_rech30	-0.001286	-0.001307	-0.003344	
fr_da_rech30	-0.001917	-0.001997	-0.003469	
cnt_da_rech90	0.073169	0.057332	-0.003700	
fr_da_rech90	0.047579	0.037829	-0.002232	
cnt_loans30	0.162833	0.154900	0.002308	
amnt_loans30	0.217586	0.216641	0.001031	
maxamnt_loans30	-0.001525	-0.002189	0.001681	
medianamnt_loans30	-0.013746	-0.006703	0.002430	

```
cnt_loans90           0.003026  0.004301          -0.000216
amnt_loans90          0.280233  0.307920           0.000664
maxamnt_loans90       0.225449  0.241772          -0.003097
medianamnt_loans90   -0.032555 -0.031045           0.003261
payback30             0.075530  0.069847          -0.002857
payback90             0.099533  0.104731          -0.001787

                      last_rech_date_da  last_rech_amt_ma
cnt_ma_rech30  ...  \
label                          0.001814          0.139969
0.244728  ...
aon                           -0.001796          0.004102      -
0.004315  ...
daily_decr30                  -0.001311          0.287181
0.444365  ...
daily_decr90                  -0.001484          0.275195
0.419650  ...
rental30                       0.003294          0.128773
0.220472  ...
rental90                       0.002643          0.123436
0.218618  ...
last_rech_date_ma              0.002629         -0.000754
0.006491  ...
last_rech_date_da              1.000000         -0.000699
0.002690  ...
last_rech_amt_ma              -0.000699          1.000000
0.008012  ...
cnt_ma_rech30                  0.002690          0.008012
1.000000  ...
fr_ma_rech30                   0.000958          0.002998
0.002295  ...
sumamnt_ma_rech30              0.000080          0.456707
0.646356  ...
medianamnt_ma_rech30           0.000184          0.796969
0.002987  ...
medianmarechprebal30           0.003673         -0.002597
0.000556  ...
cnt_ma_rech90                  0.001924          0.028202
0.884131  ...
fr_ma_rech90                   0.001071          0.109126      -
0.130383  ...
sumamnt_ma_rech90             -0.000296          0.436776
0.572447  ...
medianamnt_ma_rech90          -0.000321          0.824654      -
0.039974  ...
medianmarechprebal90           0.004731          0.125195
0.018759  ...
cnt_da_rech30                 -0.003807         -0.002644
0.003369  ...
fr_da_rech30                   0.000455         -0.003196      -
```

```
0.000292  ...
cnt_da_rech90                -0.001229          0.015274
0.011810  ...
fr_da_rech90                  0.000210          0.016371
0.005453  ...
cnt_loans30                   0.001722         -0.019892
0.733577  ...
amnt_loans30                  0.001443          0.017706
0.723759  ...
maxamnt_loans30               0.001135          0.000558      -
0.001186  ...
medianamnt_loans30            0.000009          0.029945      -
0.058580  ...
cnt_loans90                  -0.002355          0.000444
0.012307  ...
amnt_loans90                  0.001179          0.024067
0.658939  ...
maxamnt_loans90               0.002294          0.148656
0.180305  ...
medianamnt_loans90           -0.002258          0.022939      -
0.063378  ...
payback30                    -0.000020         -0.026037
0.057166  ...
payback90                     0.000699         -0.013236
0.031696  ...

                      cnt_loans30  amnt_loans30  maxamnt_loans30  \
label                    0.197565      0.199916        -0.000274
aon                     -0.003157     -0.003302        -0.003096
daily_decr30             0.346504      0.454169         0.001569
daily_decr90             0.321006      0.430940         0.001283
rental30                 0.162833      0.217586        -0.001525
rental90                 0.154900      0.216641        -0.002189
last_rech_date_ma        0.002308      0.001031         0.001681
last_rech_date_da        0.001722      0.001443         0.001135
last_rech_amt_ma        -0.019892      0.017706         0.000558
cnt_ma_rech30            0.733577      0.723759        -0.001186
fr_ma_rech30             0.003172      0.002860        -0.002389
sumamnt_ma_rech30        0.461596      0.503819         0.001551
medianamnt_ma_rech30    -0.019427      0.018799         0.002353
medianmarechprebal30     0.000720      0.000802        -0.001916
cnt_ma_rech90            0.657093      0.679860        -0.000981
fr_ma_rech90            -0.095577     -0.101195         0.001455
sumamnt_ma_rech90        0.412481      0.481843         0.001867
medianamnt_ma_rech90    -0.052248     -0.015216         0.001287
medianmarechprebal90    -0.040414     -0.030404         0.002022
cnt_da_rech30            0.003343      0.002567        -0.002259
fr_da_rech30            -0.000826     -0.001507        -0.000645
cnt_da_rech90            0.018352      0.026531        -0.001474
fr_da_rech90             0.007443      0.012066        -0.000409
```

cnt_loans30	1.000000	0.956576	-0.000302
amnt_loans30	0.956576	1.000000	-0.000240
maxamnt_loans30	-0.000302	-0.000240	1.000000
medianamnt_loans30	-0.087408	-0.071821	0.008336
cnt_loans90	0.013875	0.013252	0.003364
amnt_loans90	0.850820	0.896534	-0.001706
maxamnt_loans90	0.150610	0.342611	0.000132
medianamnt_loans90	-0.092829	-0.082189	0.009825
payback30	0.086703	0.078025	-0.000741
payback90	0.051928	0.048781	-0.000159

	medianamnt_loans30	cnt_loans90	amnt_loans90 \
label	0.050067	0.004305	0.205065
aon	0.004679	0.000192	-0.003336
daily_decr30	-0.005629	0.008865	0.542179
daily_decr90	0.000012	0.009220	0.544854
rental30	-0.013746	0.003026	0.280233
rental90	-0.006703	0.004301	0.307920
last_rech_date_ma	0.002430	-0.000216	0.000664
last_rech_date_da	0.000009	-0.002355	0.001179
last_rech_amt_ma	0.029945	0.000444	0.024067
cnt_ma_rech30	-0.058580	0.012307	0.658939
fr_ma_rech30	-0.000416	0.003980	0.002794
sumamnt_ma_rech30	-0.026794	0.008296	0.460148
medianamnt_ma_rech30	0.033407	-0.000636	0.026795
medianmarechprebal30	-0.002140	0.002095	0.001382
cnt_ma_rech90	-0.045644	0.013583	0.753304
fr_ma_rech90	0.019313	-0.002041	-0.109715
sumamnt_ma_rech90	-0.014902	0.010786	0.533540
medianamnt_ma_rech90	0.037270	0.000538	-0.013136
medianmarechprebal90	0.030114	-0.000650	-0.033457
cnt_da_rech30	-0.000995	-0.001694	0.003354
fr_da_rech30	-0.001739	-0.003855	0.000602
cnt_da_rech90	-0.003062	0.001208	0.021544
fr_da_rech90	-0.002319	0.002682	0.008969
cnt_loans30	-0.087408	0.013875	0.850820
amnt_loans30	-0.071821	0.013252	0.896534
maxamnt_loans30	0.008336	0.003364	-0.001706
medianamnt_loans30	1.000000	-0.002246	-0.061690
cnt_loans90	-0.002246	1.000000	0.015576
amnt_loans90	-0.061690	0.015576	1.000000
maxamnt_loans90	0.062114	0.001653	0.332799
medianamnt_loans90	0.915271	-0.001256	-0.090813
payback30	-0.005404	0.000008	0.068102
payback90	0.003336	-0.000901	0.048437

	maxamnt_loans90	medianamnt_loans90	payback30 \
label	0.086033	0.041265	0.050892

aon	-0.000975	0.002346	0.002246
daily_decr30	0.396803	-0.031485	0.033669
daily_decr90	0.394487	-0.029046	0.025432
rental30	0.225449	-0.032555	0.075530
rental90	0.241772	-0.031045	0.069847
last_rech_date_ma	-0.003097	0.003261	-0.002857
last_rech_date_da	0.002294	-0.002258	-0.000020
last_rech_amt_ma	0.148656	0.022939	-0.026037
cnt_ma_rech30	0.180305	-0.063378	0.057166
fr_ma_rech30	-0.001276	-0.001572	0.002473
sumamnt_ma_rech30	0.260943	-0.034854	0.008018
medianamnt_ma_rech30	0.160716	0.023826	-0.016644
medianmarechprebal30	-0.002558	-0.002058	0.001593
cnt_ma_rech90	0.248826	-0.064568	0.021222
fr_ma_rech90	-0.038461	0.015097	0.037328
sumamnt_ma_rech90	0.323032	-0.034848	-0.021271
medianamnt_ma_rech90	0.135380	0.032597	-0.034525
medianmarechprebal90	0.028135	0.032048	-0.030236
cnt_da_rech30	-0.001141	-0.001589	-0.000655
fr_da_rech30	-0.000945	-0.001692	0.001505
cnt_da_rech90	0.036098	-0.002749	0.014265
fr_da_rech90	0.021811	-0.001325	0.000673
cnt_loans30	0.150610	-0.092829	0.086703
amnt_loans30	0.342611	-0.082189	0.078025

maxamnt_loans30	0.000132	0.009825	-0.000741
medianamnt_loans30	0.062114	0.915271	-0.005404
cnt_loans90	0.001653	-0.001256	0.000008
amnt_loans90	0.332799	-0.090813	0.068102
maxamnt_loans90	1.000000	0.036631	0.015477
medianamnt_loans90	0.036631	1.000000	-0.012944
payback30	0.015477	-0.012944	1.000000
payback90	0.033080	-0.010585	0.827801

	payback90
label	0.053776
aon	0.002549
daily_decr30	0.056822
daily_decr90	0.050147
rental30	0.099533
rental90	0.104731
last_rech_date_ma	-0.001787
last_rech_date_da	0.000699
last_rech_amt_ma	-0.013236
cnt_ma_rech30	0.031696
fr_ma_rech30	0.002102
sumamnt_ma_rech30	-0.003634
medianamnt_ma_rech30	-0.001416
medianmarechprebal30	0.001695
cnt_ma_rech90	0.009319
fr_ma_rech90	0.079096
sumamnt_ma_rech90	-0.022272
medianamnt_ma_rech90	-0.022386
medianmarechprebal90	-0.029950
cnt_da_rech30	0.000488
fr_da_rech30	-0.001122
cnt_da_rech90	0.024713
fr_da_rech90	0.001550
cnt_loans30	0.051928
amnt_loans30	0.048781
maxamnt_loans30	-0.000159
medianamnt_loans30	0.003336
cnt_loans90	-0.000901
amnt_loans90	0.048437
maxamnt_loans90	0.033080
medianamnt_loans90	-0.010585

```
payback30               0.827801
payback90               1.000000

[33 rows x 33 columns]
```

Observation:

daily_decr30 and daily_decr90 features are highly correlated with each otheer. rental30 and rental90 features are highly correlated with each other. cnt_loans30 and amount_loans30 columns are highly correlated with each other. amount_loans30 is also highly correlated with amount_loans90 column. medianamnt_loans30 and medianamnt_loans90 is highly correlated with each other. We have to drop one of the features which are highly correlated with other feayures. And if we dont do this then our model will face multicolinearity problem.

```
#Dropping the columns which is highly correlated with each other do
avoid multicolinearity problem.
df.drop(columns=['daily_decr30','rental30','amnt_loans30','medianamnt_
loans30'], axis=1 , inplace = True)

#Now checking the shape
print(df.shape)
#Checking the unique value in pdate column.
df['pdate'].nunique()
```

```
(186243, 31)

82
```

```
#Making the new column Day, Month and year from pdate column
df['pDay']=pd.to_datetime(df['pdate'],format='%Y/%m/%d').dt.day
df['pMonth']=pd.to_datetime(df['pdate'],format='%Y/%m/%d').dt.month
df['pYear']=pd.to_datetime(df['pdate'],format='%Y/%m/%d').dt.year

df.head()
```

```
   label       msisdn    aon  daily_decr90  rental90
last_rech_date_ma  \
0      0  21408I70789  272.0   3065.150000    260.13
2.0
1      1  76462I70374  712.0  12124.750000   3691.26
20.0
2      1  17943I70372  535.0   1398.000000    900.13
3.0
3      1  55773I70781  241.0     21.228000    159.42
41.0
4      1  03813I82730  947.0    150.619333   1098.90
4.0

   last_rech_date_da  last_rech_amt_ma  cnt_ma_rech30
```

```
fr_ma_rech30  ...  \
0                0.0              1539             2
21.0  ...
1                0.0              5787             1
0.0  ...
2                0.0              1539             1
0.0  ...
3                0.0               947             0
0.0  ...
4                0.0              2309             7
2.0  ...

   cnt_loans90  amnt_loans90  maxamnt_loans90  medianamnt_loans90
payback30  \
0          2.0            12                6                 0.0
29.000000
1          1.0            12               12                 0.0
0.000000
2          1.0             6                6                 0.0
0.000000
3          2.0            12                6                 0.0
0.000000
4          7.0            42                6                 0.0
2.333333

   payback90       pdate  pDay  pMonth  pYear
0  29.000000  2016-07-20    20       7   2016
1   0.000000  2016-08-10    10       8   2016
2   0.000000  2016-08-19    19       8   2016
3   0.000000  2016-06-06     6       6   2016
4   2.333333  2016-06-22    22       6   2016

[5 rows x 34 columns]
```

```
#Checking the number of months
df['pMonth'].unique()
```

```
array([7, 8, 6], dtype=int64)
```

```
#After fetching the data from pdate column now we are going to drop it
because it has not any significant role.
df.drop(columns=['pdate'],axis=1, inplace = True)

#Seprate the categorical columns and Numerical columns
cat_df,num_df=[],[]

for i in df.columns:
    if df[i].dtype==object:
        cat_df.append(i)
    elif (df[i].dtypes=='int64') | (df[i].dtypes=='float64') |
(df[i].dtypes=='int32'):
```

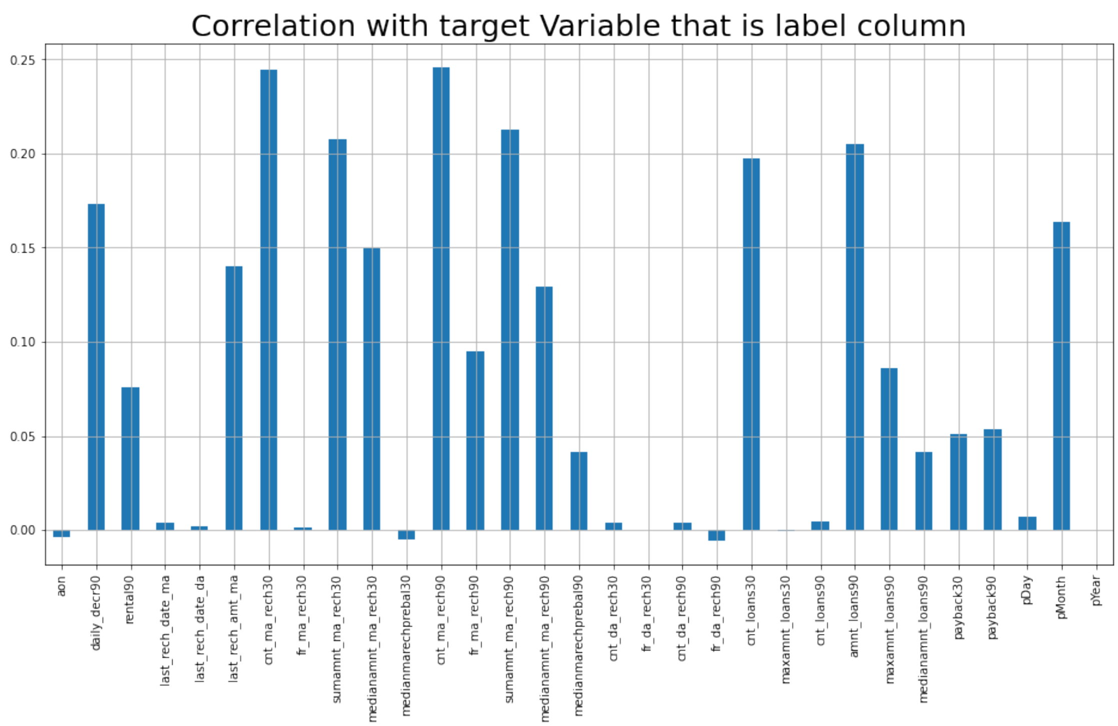
```
        num_df.append(i)
    else: continue

print('>>> Total Number of Feature::', df.shape[1])
print('>>> Number of categorical features::', len(cat_df))
print('>>> Number of Numerical Feature::', len(num_df))
```

```
>>> Total Number of Feature:: 33
>>> Number of categorical features:: 1
>>> Number of Numerical Feature:: 32
```

Data Visualization

```
#Checking the correlation with target variable
plt.figure(figsize=(16,8))
df.drop('label',
axis=1).corrwith(df['label']).plot(kind='bar',grid=True)
plt.xticks(rotation='vertical')
plt.title("Correlation with target Variable that is label
column",fontsize=25)
```

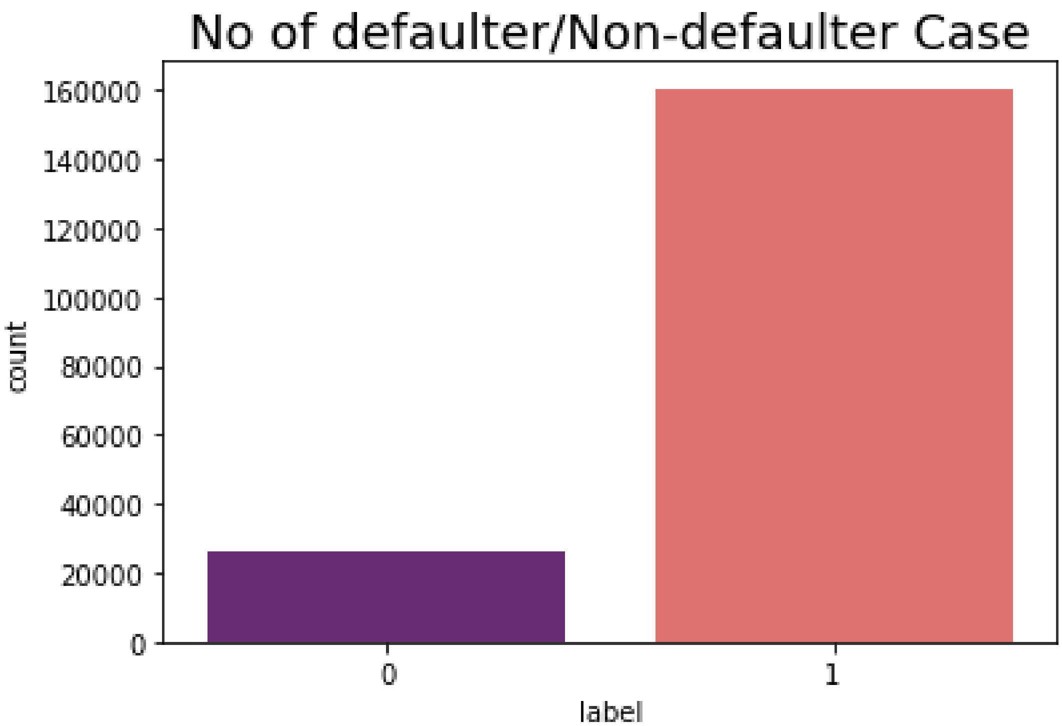
```
Text(0.5, 1.0, 'Correlation with target Variable that is label
column')
```



Observation:

Here we see the correlation of the columns with respect to the target column that is label.

```
#Checking the number of Fraud cases.
sns.countplot(x='label', data=df, palette='magma')
plt.title('No of defaulter/Non-defaulter Case',fontsize=18)
plt.show()

print(df['label'].value_counts())
```




```
1    160383
0     25860
Name: label, dtype: int64
```

Observation:

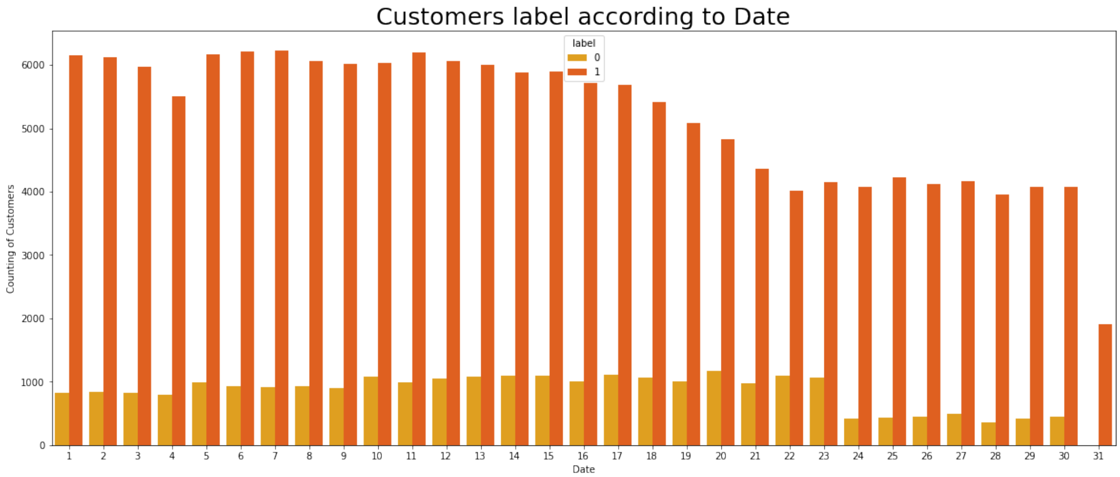
Label 1 indicates loan has been payed i.e Non-Defaulter and label 0 indicates indicates that the loan has not beenpayed i.e. defaulter.

```
#Plotting the Histogram
df.hist(figsize=(20,20),color='r')
plt.show()
```

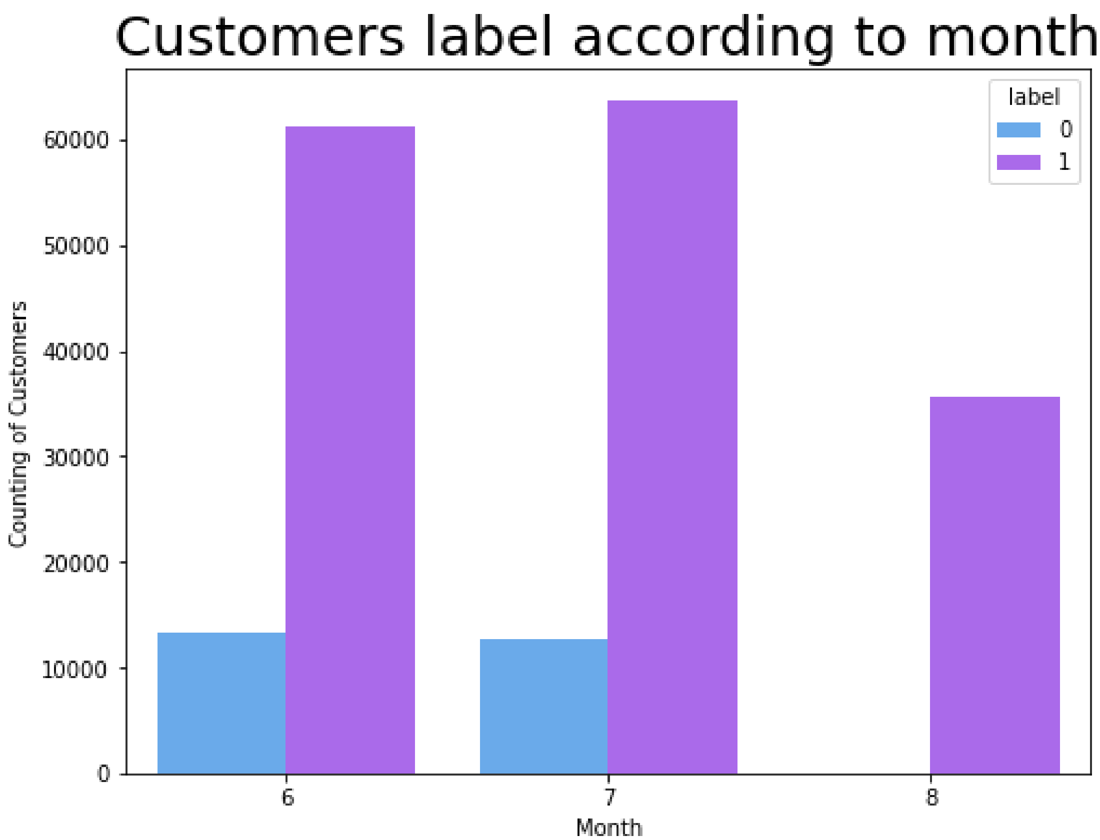
Observation:

We plot the histogram to display the shape and spread of continuous sample data.In a histogram, each bar groups numbers into ranges. Taller bars show that more data falls in that range

```
#Customer label according to Date
plt.figure(figsize=(20,8))
sns.countplot(x="pDay", hue='label', data=df, palette='autumn_r')
plt.title("Customers label according to Date", fontsize=25)
plt.xlabel('Date')
plt.ylabel('Counting of Customers')
plt.show()
```




```python
#Customer label according to Month
plt.figure(figsize=(8,6))
sns.countplot(x="pMonth", hue='label', data=df, palette='cool')
plt.title("Customers label according to month", fontsize=25)
plt.xlabel('Month')
plt.ylabel('Counting of Customers')
plt.show()
```

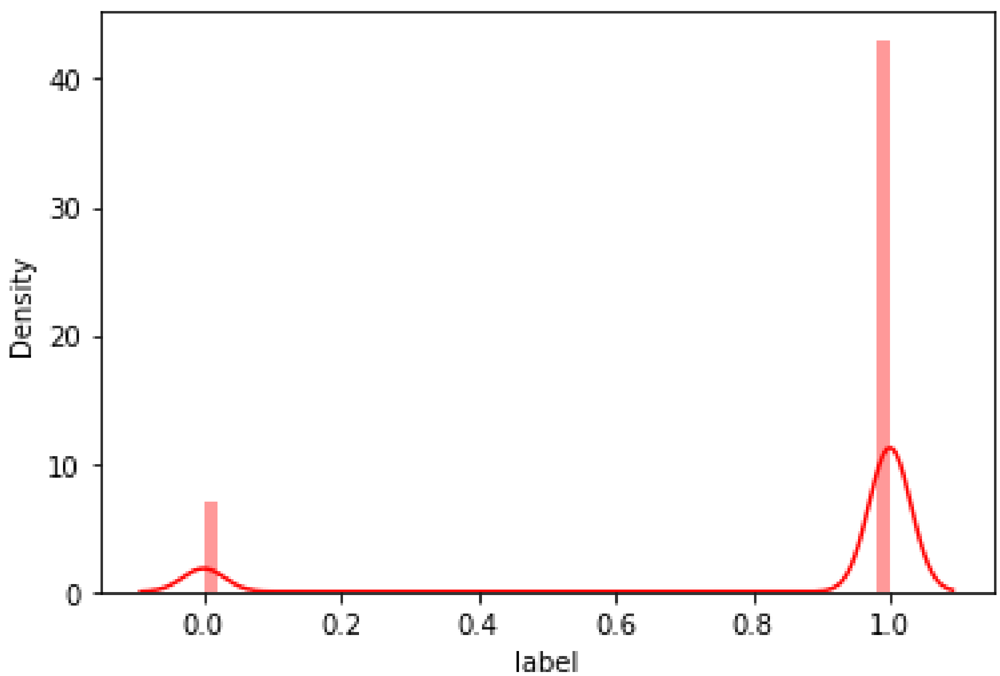


Observation:

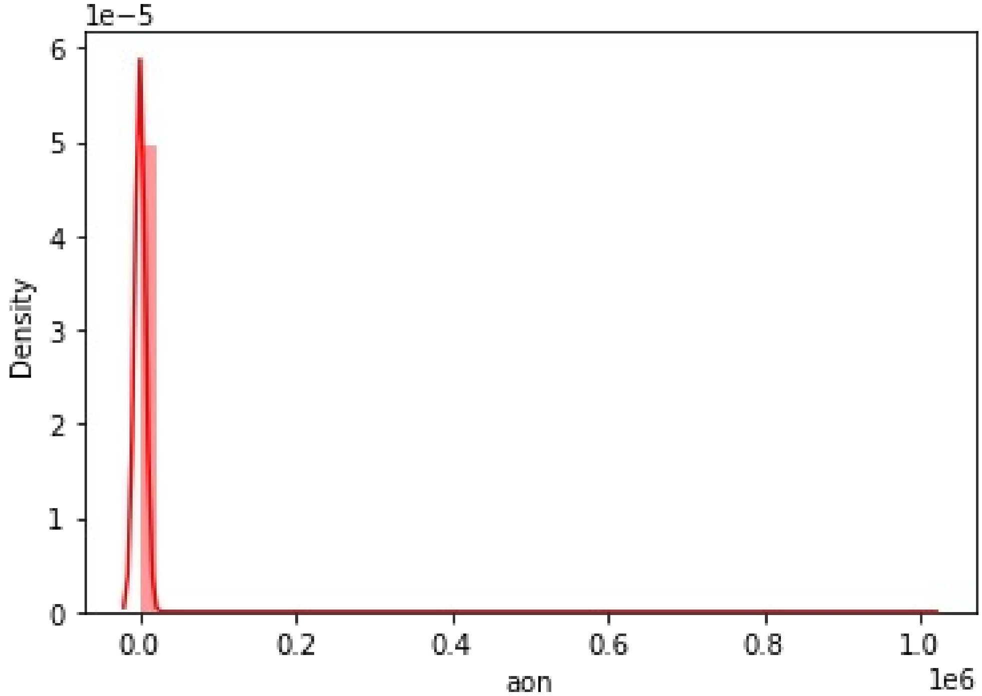
The first figure which is date vs label shows that the customers who did not pay their loans are from date 10 to 23. There are severals customers at June and July month who did not pay their loan.

```
#checking skewness

for col in df.describe().columns:
    sns.distplot(df[col],color='r')
    plt.show()
```

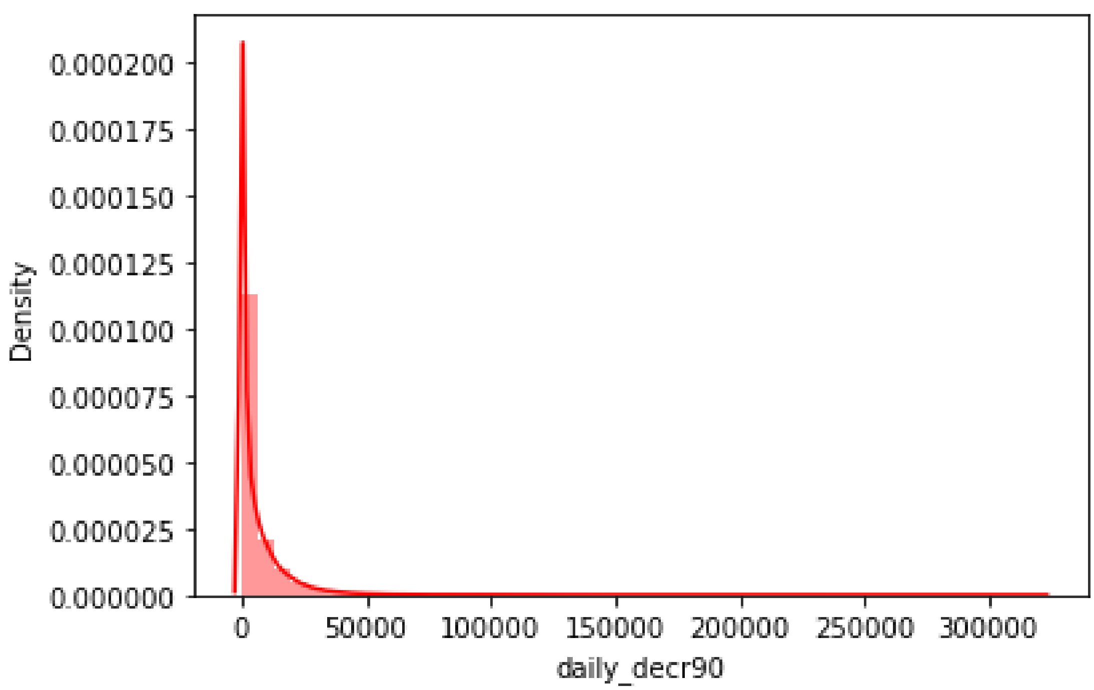
```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```



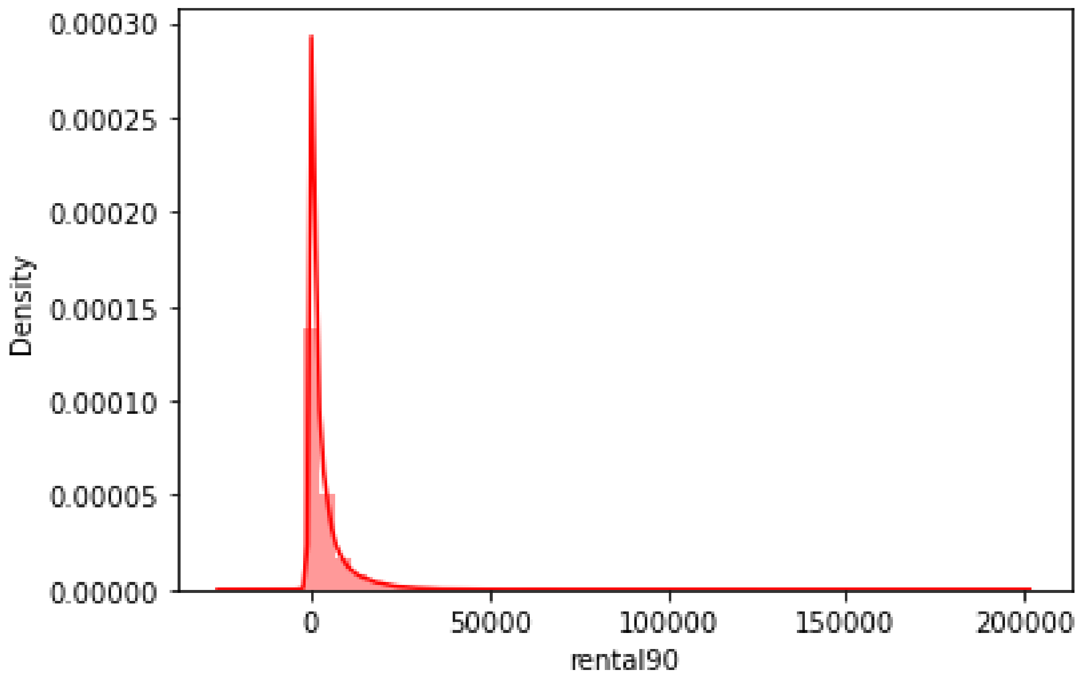

```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```



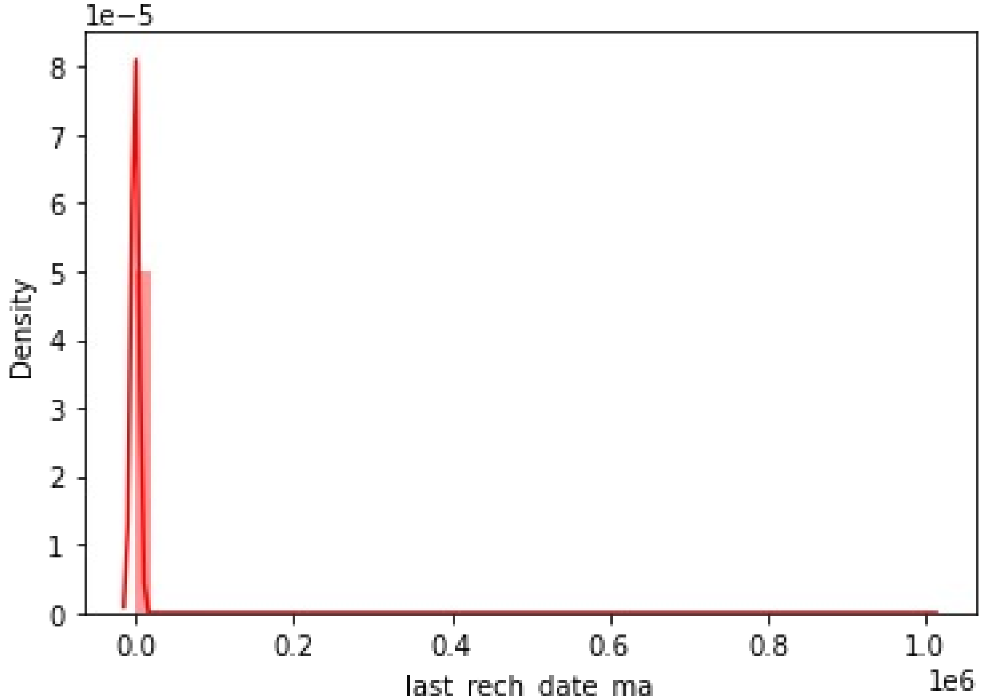

```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\
distributions.py:2619: FutureWarning: `distplot` is a deprecated
function and will be removed in a future version. Please adapt your
code to use either `displot` (a figure-level function with similar
flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```



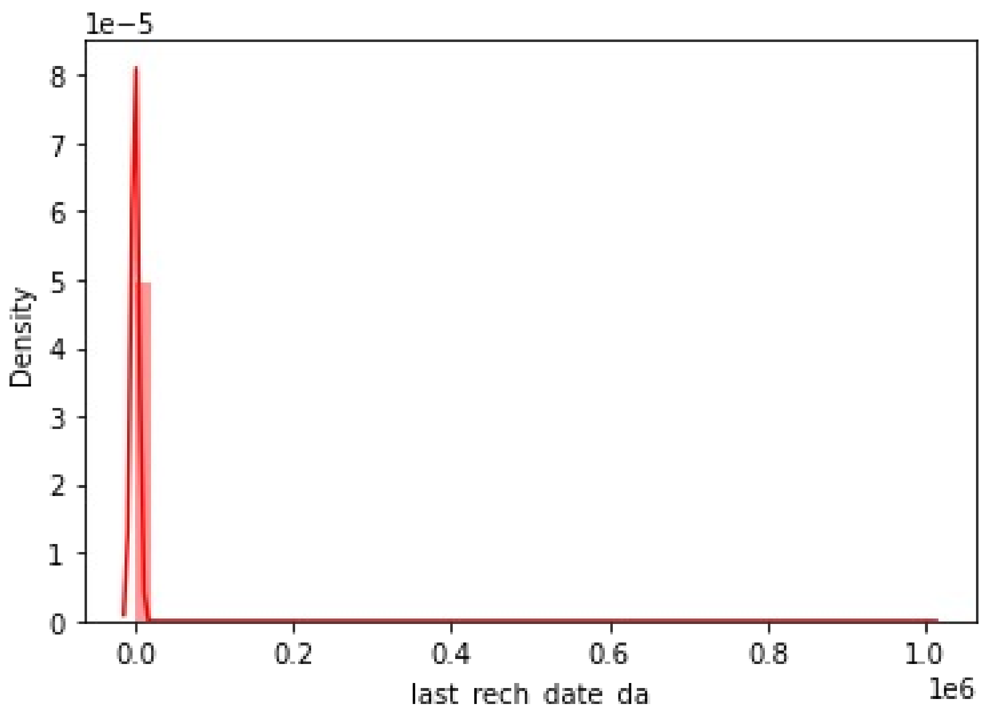
```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\
distributions.py:2619: FutureWarning: `distplot` is a deprecated
function and will be removed in a future version. Please adapt your
code to use either `displot` (a figure-level function with similar
flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```



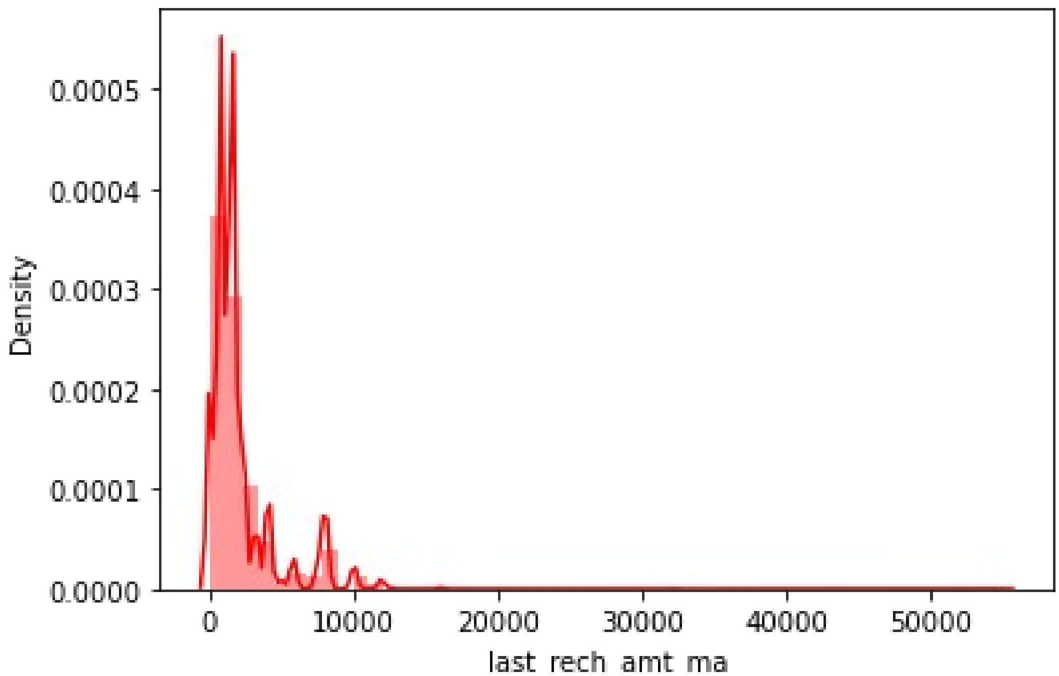

```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\
distributions.py:2619: FutureWarning: `distplot` is a deprecated
function and will be removed in a future version. Please adapt your
code to use either `displot` (a figure-level function with similar
flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```



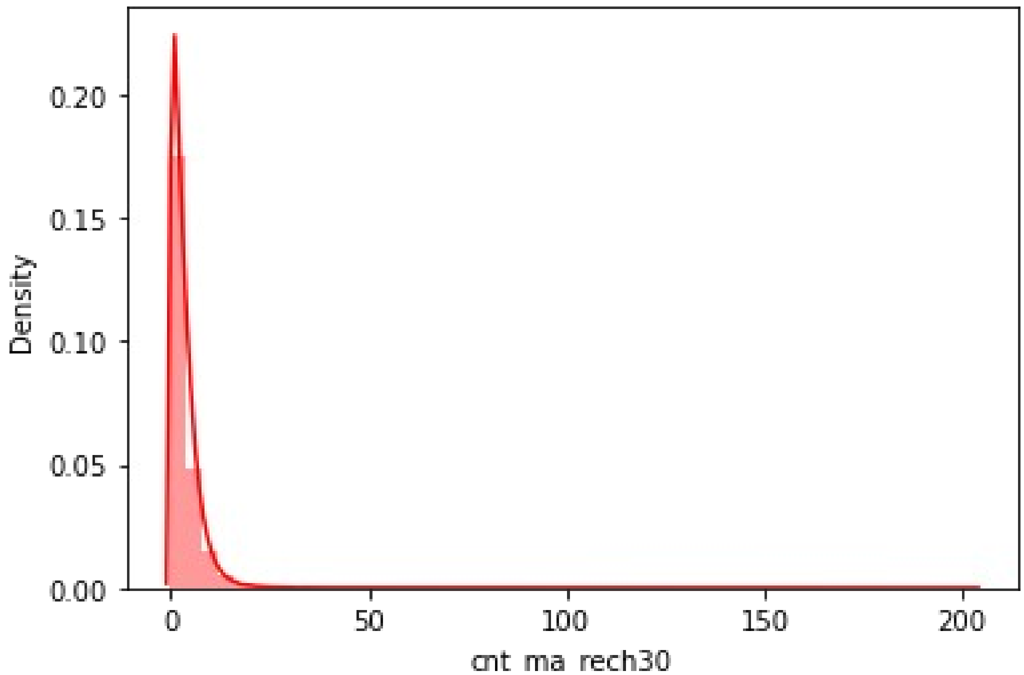

```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\
distributions.py:2619: FutureWarning: `distplot` is a deprecated
function and will be removed in a future version. Please adapt your
code to use either `displot` (a figure-level function with similar
flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```



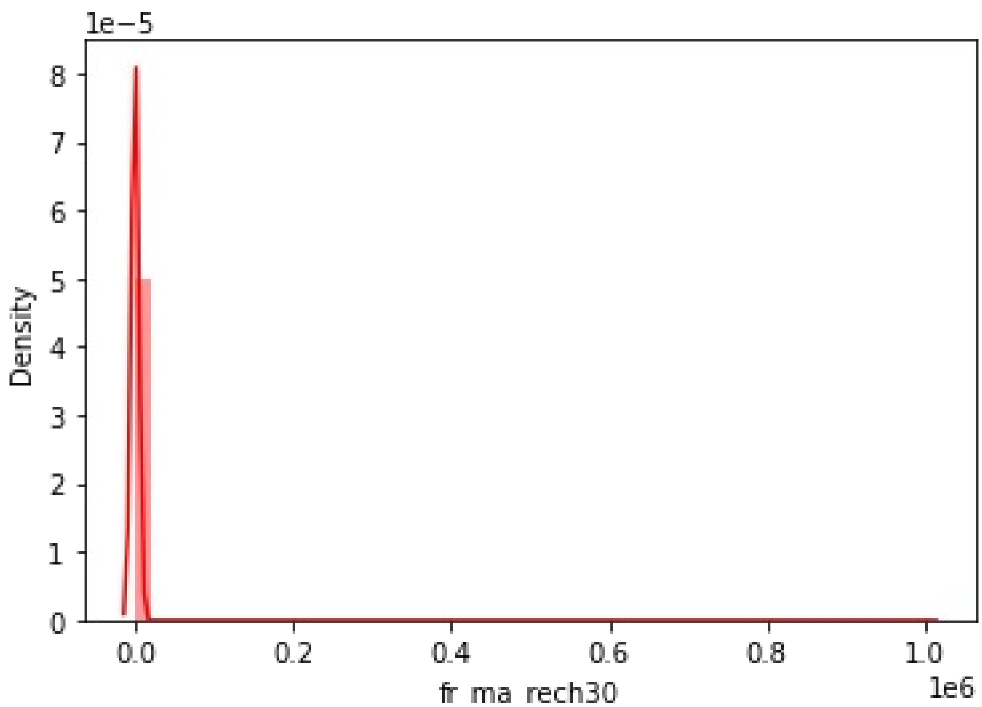
```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\
distributions.py:2619: FutureWarning: `distplot` is a deprecated
function and will be removed in a future version. Please adapt your
code to use either `displot` (a figure-level function with similar
flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```



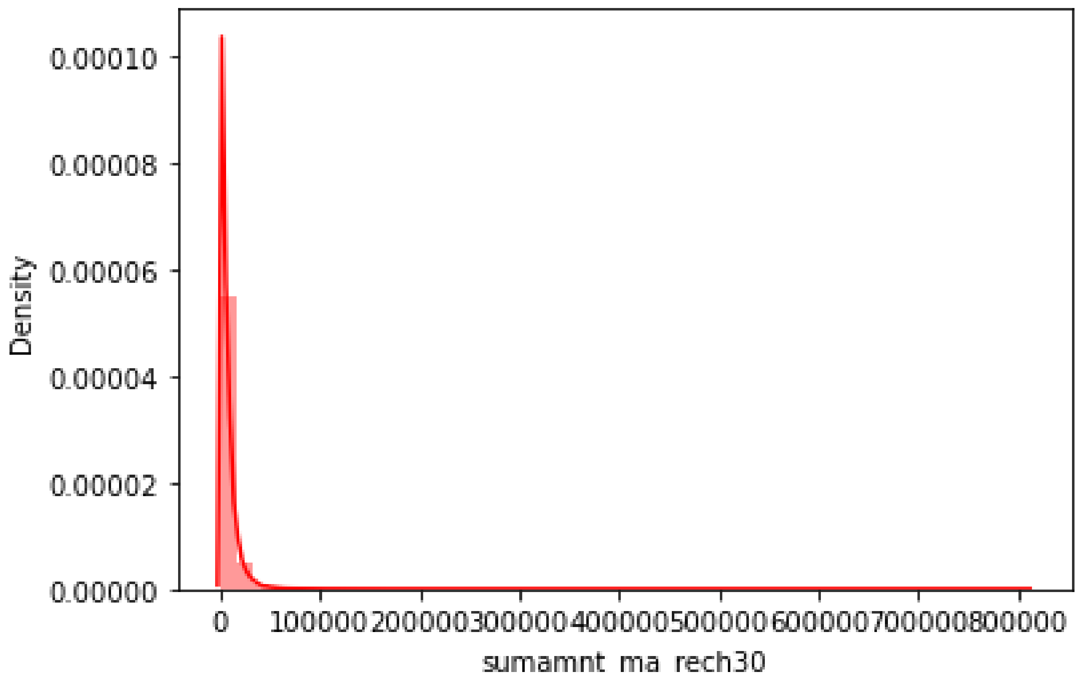

```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\
distributions.py:2619: FutureWarning: `distplot` is a deprecated
function and will be removed in a future version. Please adapt your
code to use either `displot` (a figure-level function with similar
flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```



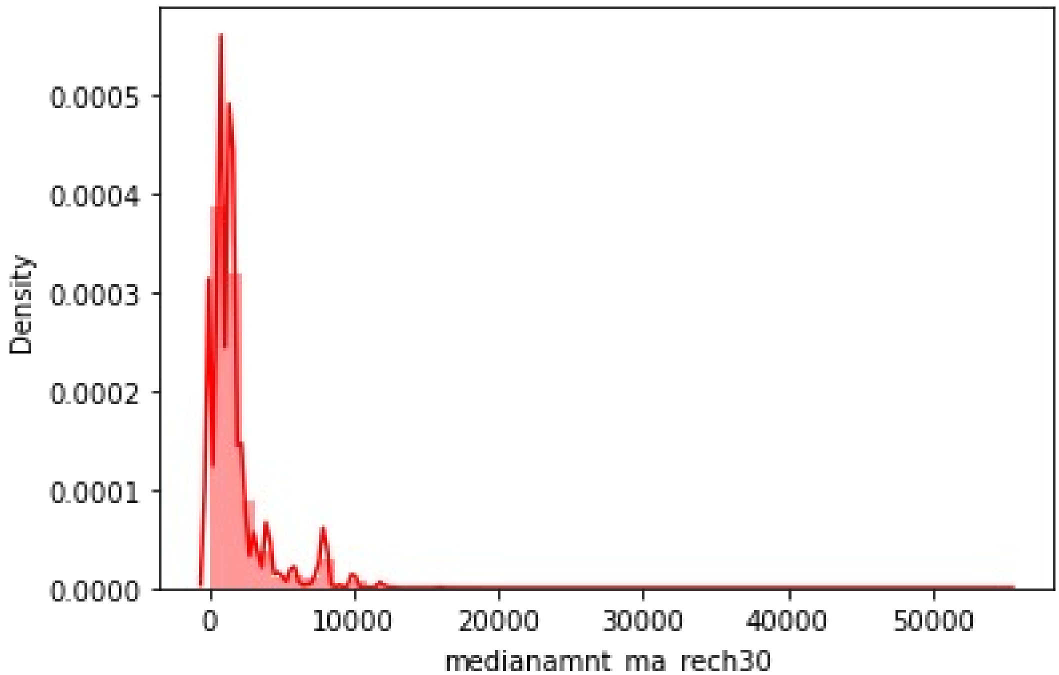

```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\
distributions.py:2619: FutureWarning: `distplot` is a deprecated
function and will be removed in a future version. Please adapt your
code to use either `displot` (a figure-level function with similar
flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```



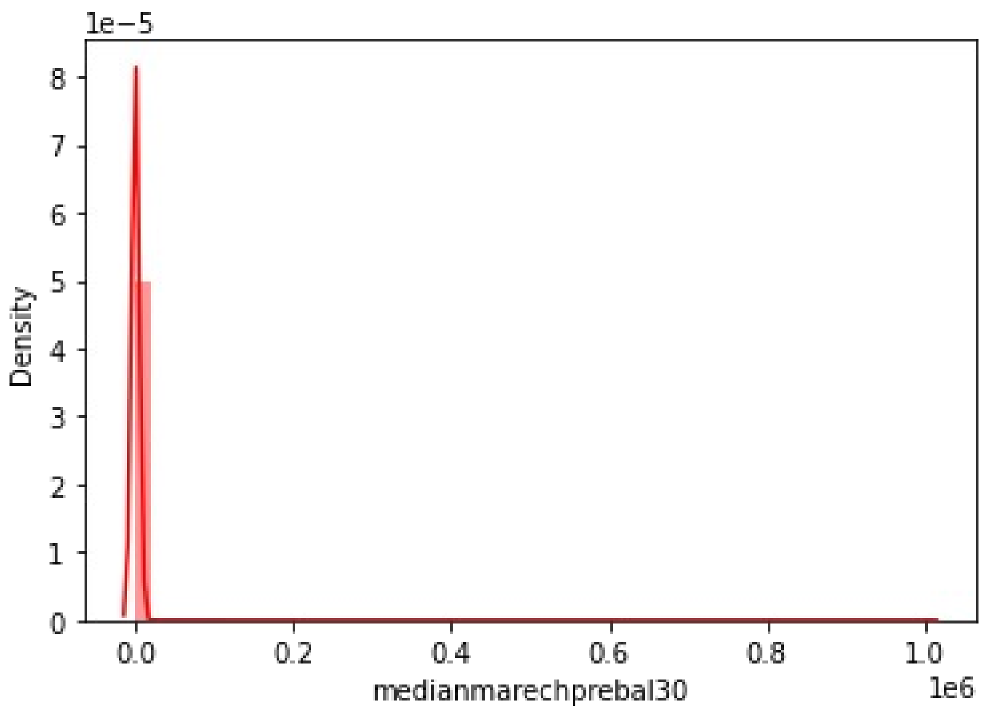
```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\
distributions.py:2619: FutureWarning: `distplot` is a deprecated
function and will be removed in a future version. Please adapt your
code to use either `displot` (a figure-level function with similar
flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```



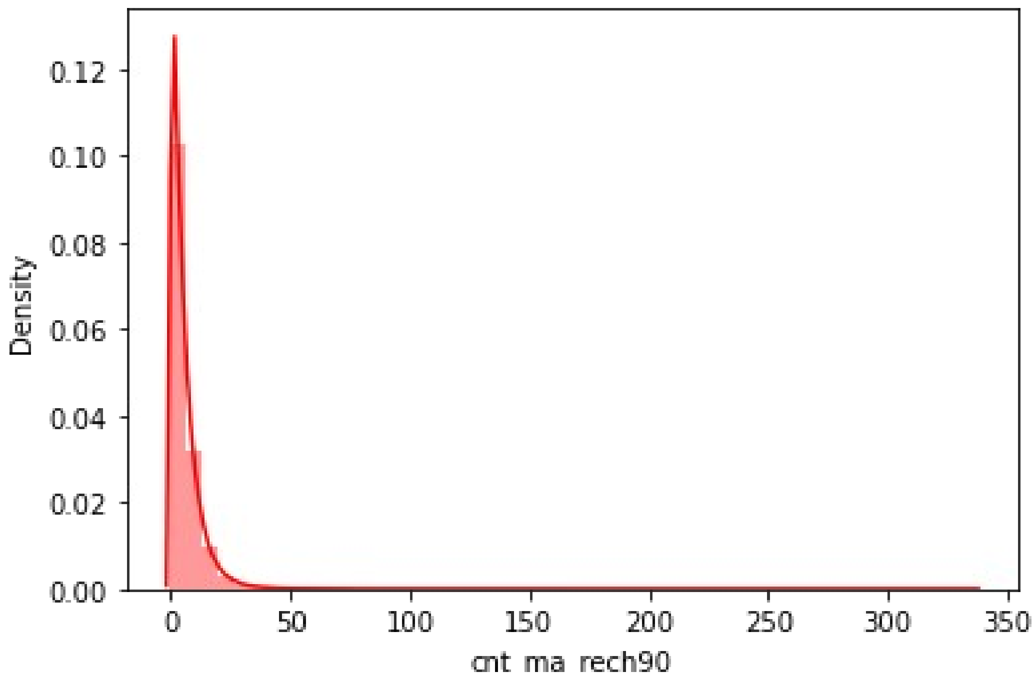

```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\
distributions.py:2619: FutureWarning: `distplot` is a deprecated
function and will be removed in a future version. Please adapt your
code to use either `displot` (a figure-level function with similar
flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```



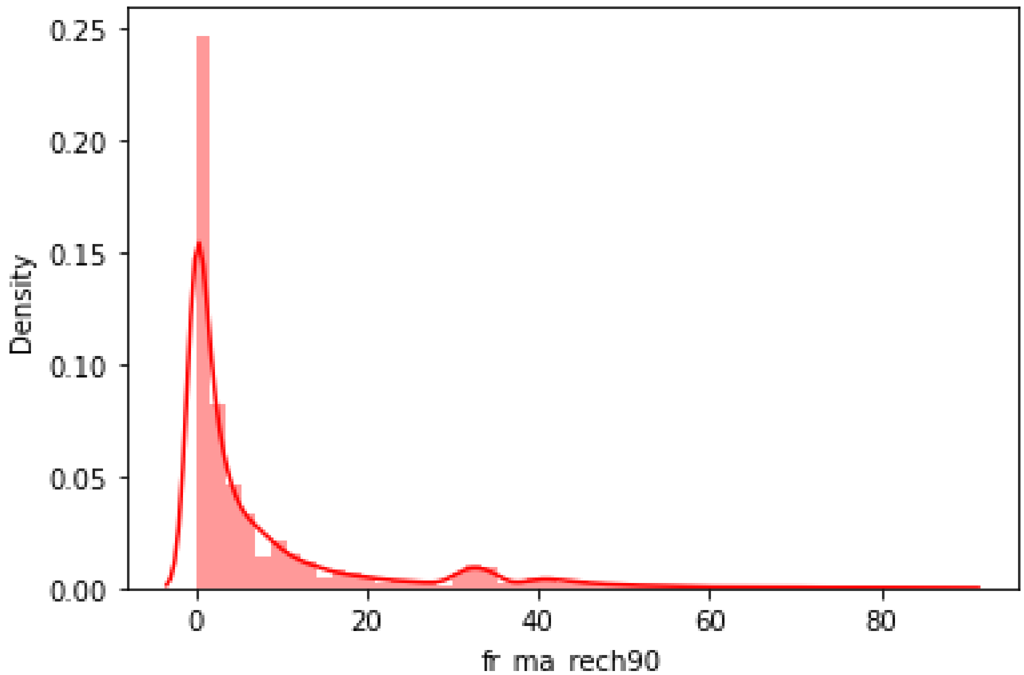

```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\
distributions.py:2619: FutureWarning: `distplot` is a deprecated
function and will be removed in a future version. Please adapt your
code to use either `displot` (a figure-level function with similar
flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```



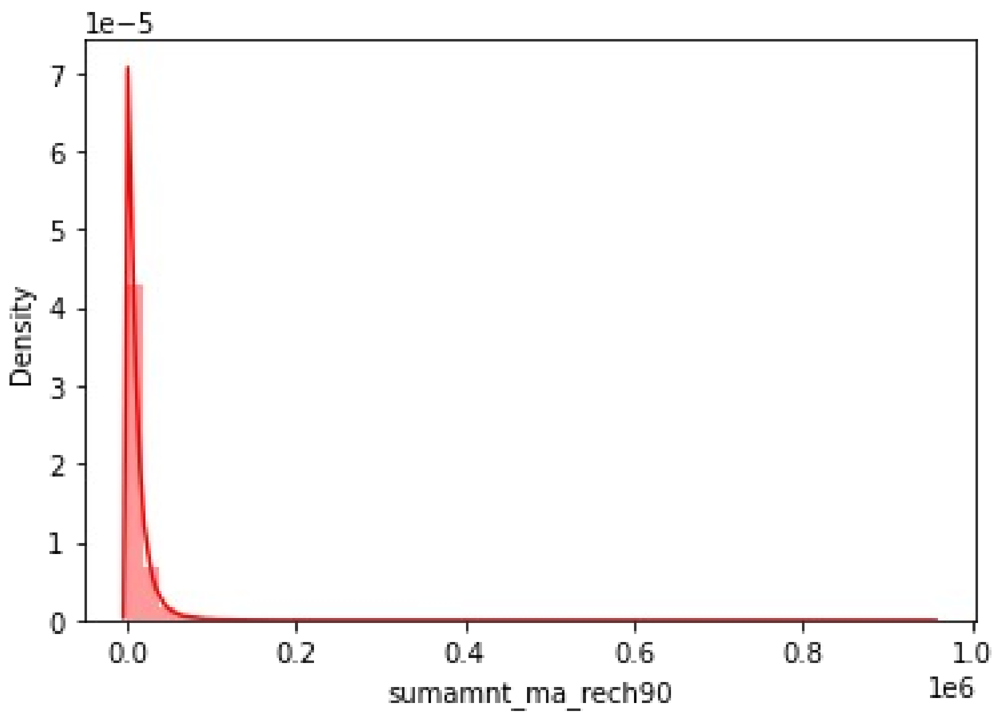
```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\
distributions.py:2619: FutureWarning: `distplot` is a deprecated
function and will be removed in a future version. Please adapt your
code to use either `displot` (a figure-level function with similar
flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```



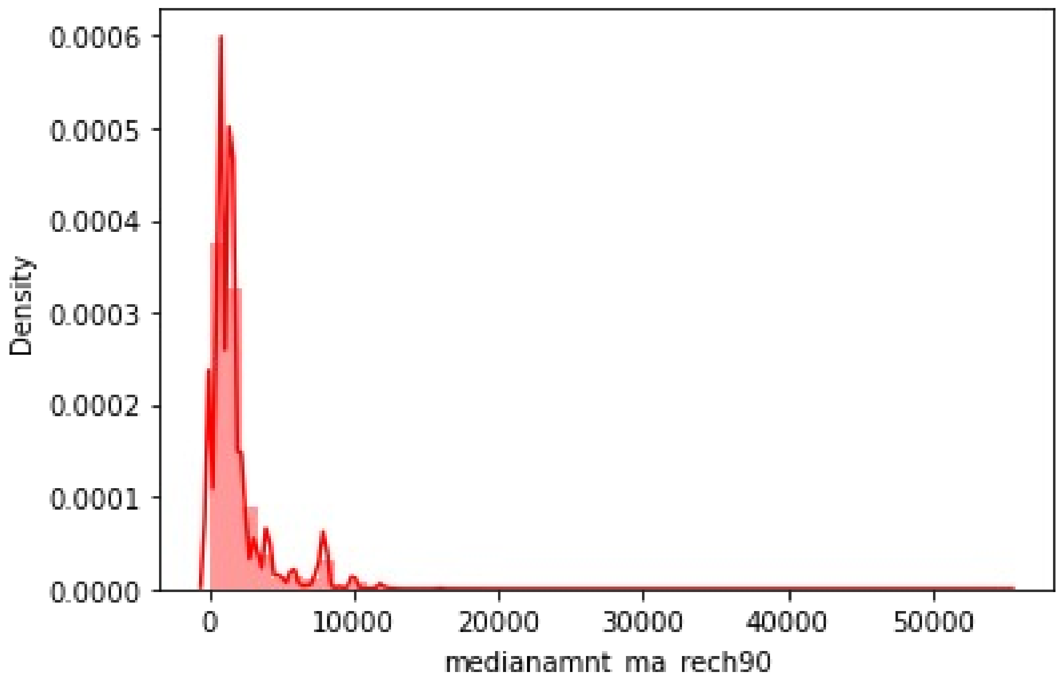

```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\
distributions.py:2619: FutureWarning: `distplot` is a deprecated
function and will be removed in a future version. Please adapt your
code to use either `displot` (a figure-level function with similar
flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```



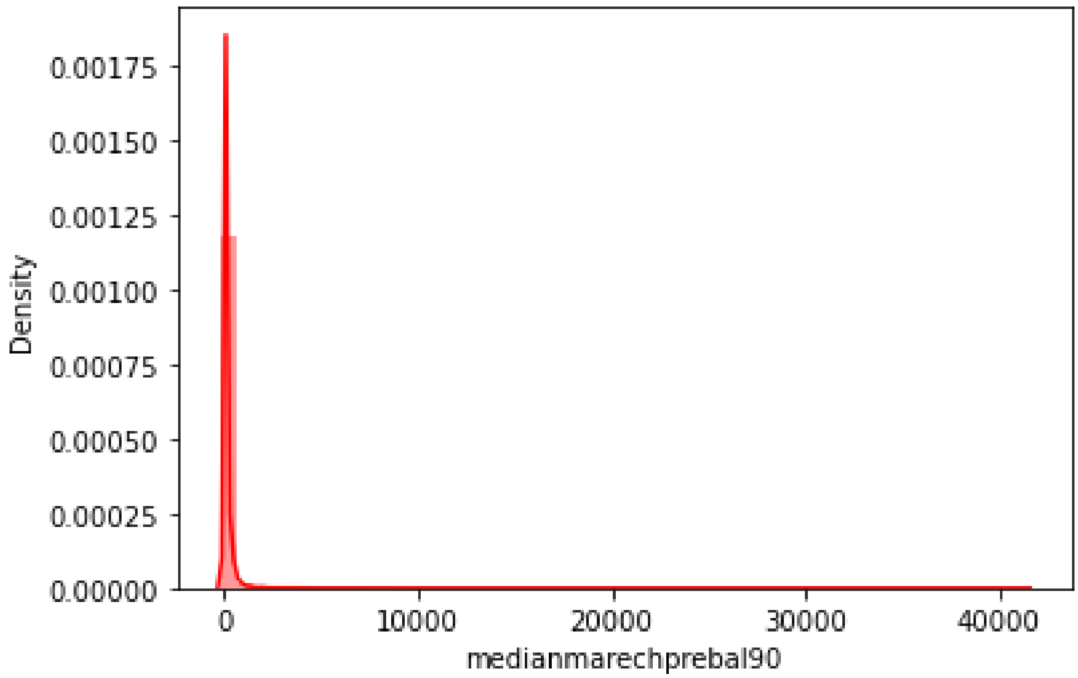

```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\
distributions.py:2619: FutureWarning: `distplot` is a deprecated
function and will be removed in a future version. Please adapt your
code to use either `displot` (a figure-level function with similar
flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```



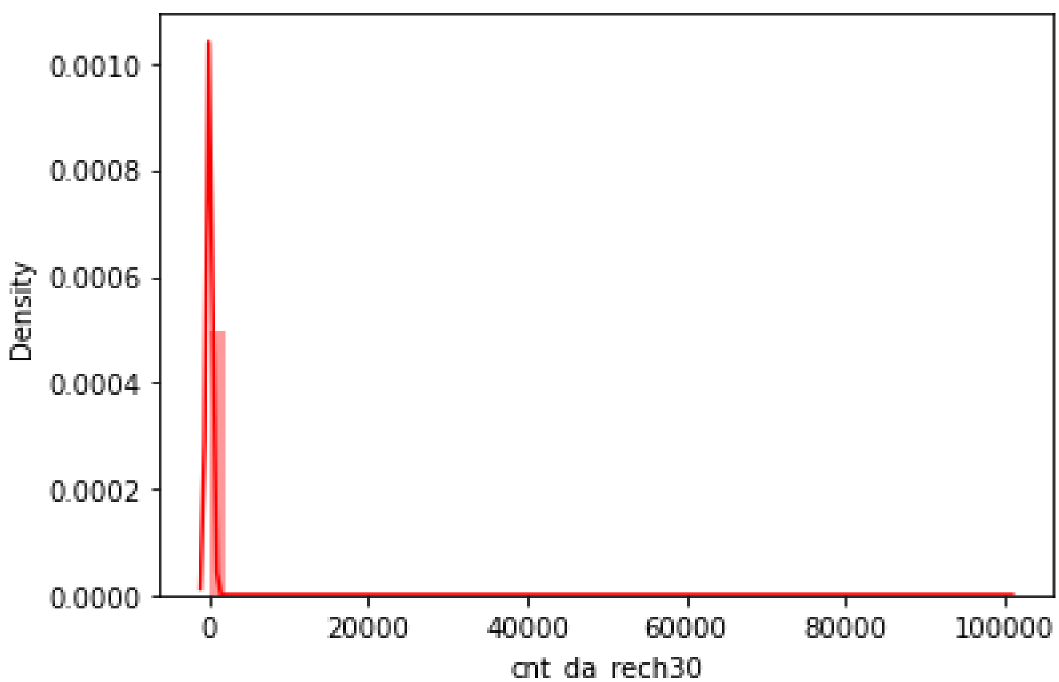
```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\
distributions.py:2619: FutureWarning: `distplot` is a deprecated
function and will be removed in a future version. Please adapt your
code to use either `displot` (a figure-level function with similar
flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```



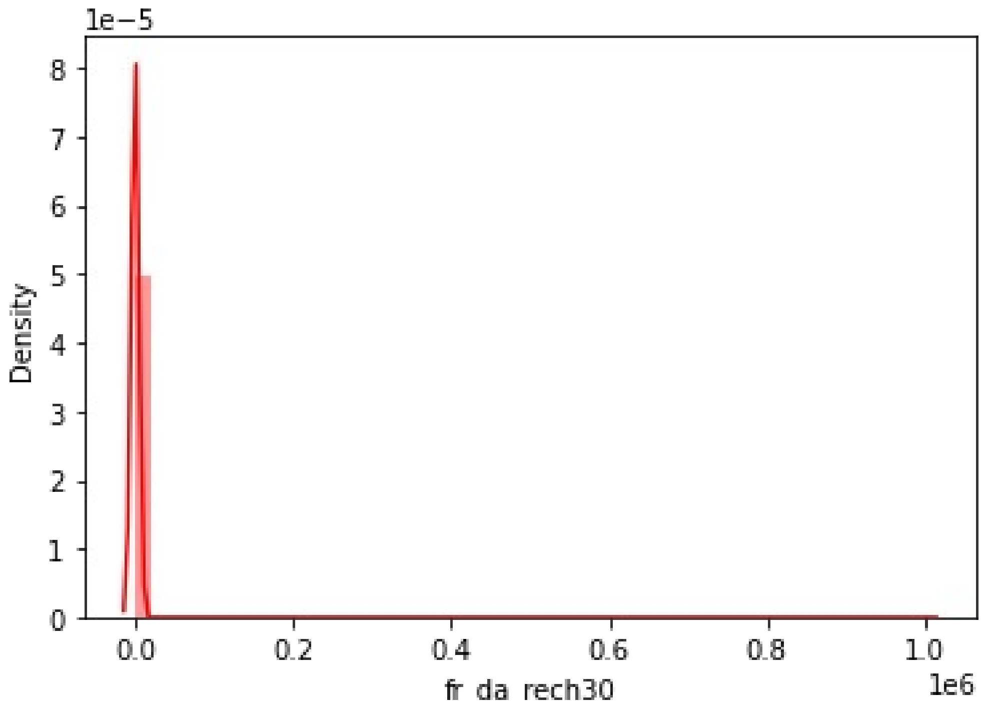

```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\
distributions.py:2619: FutureWarning: `distplot` is a deprecated
function and will be removed in a future version. Please adapt your
code to use either `displot` (a figure-level function with similar
flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```



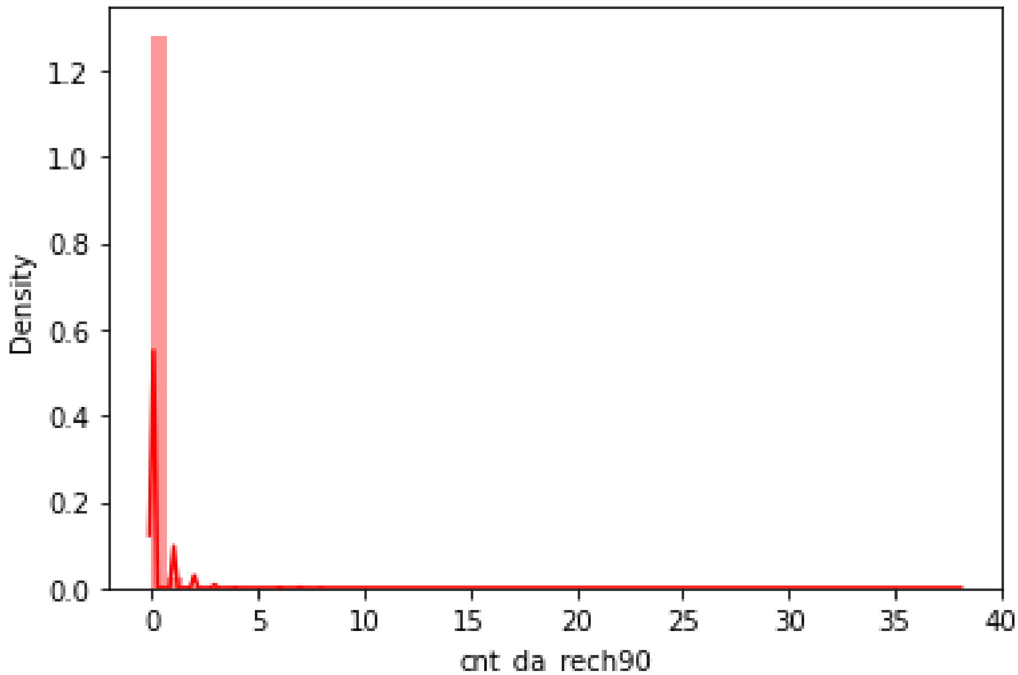

```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\
distributions.py:2619: FutureWarning: `distplot` is a deprecated
function and will be removed in a future version. Please adapt your
code to use either `displot` (a figure-level function with similar
flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```



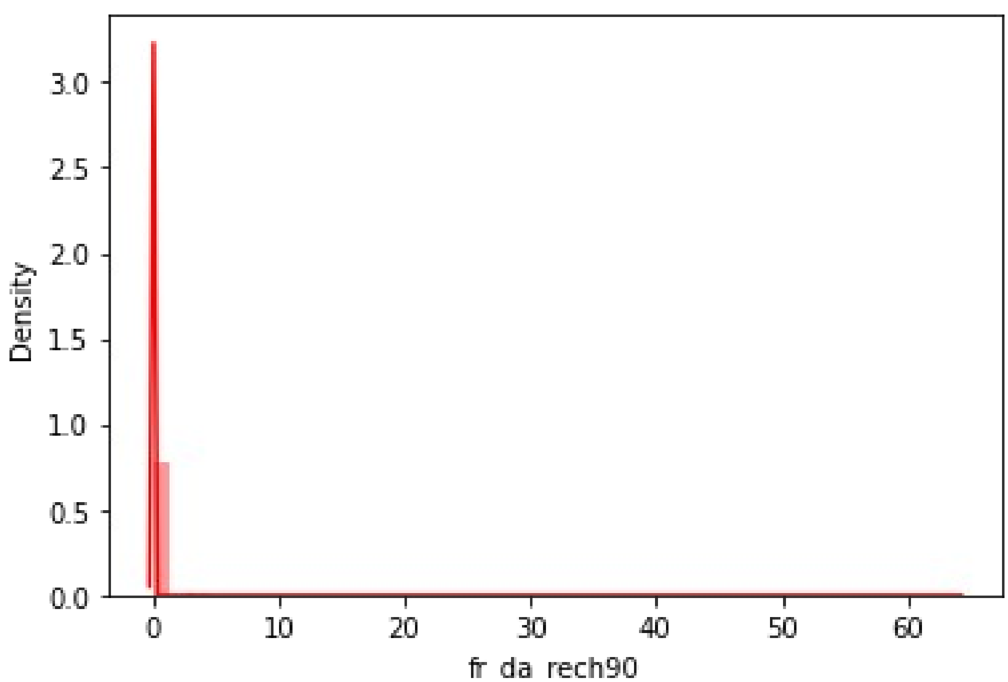
```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\
distributions.py:2619: FutureWarning: `distplot` is a deprecated
function and will be removed in a future version. Please adapt your
code to use either `displot` (a figure-level function with similar
flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```



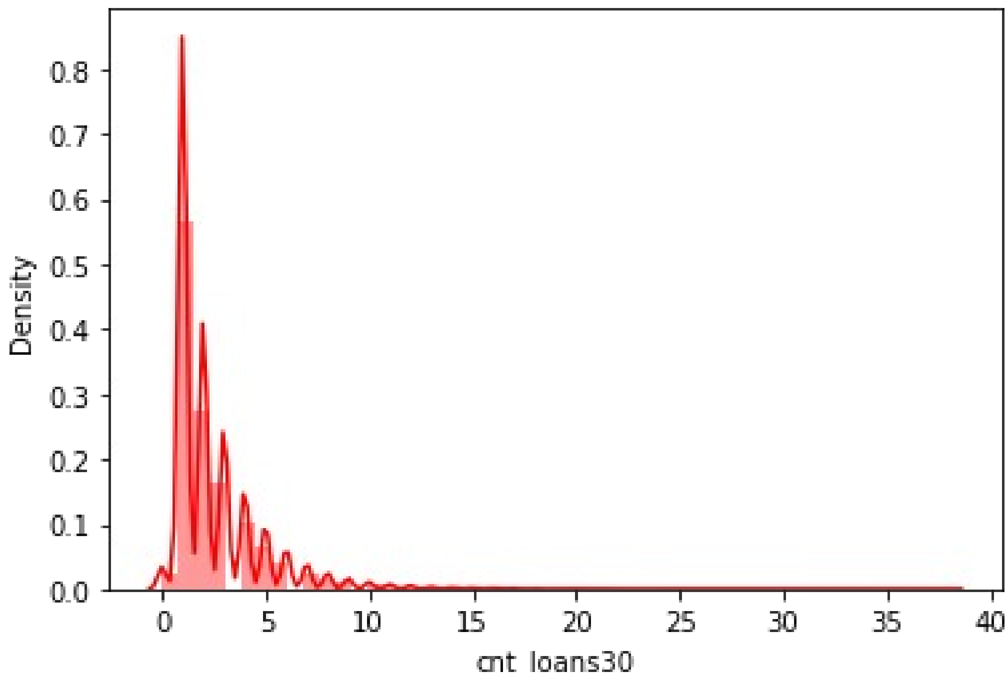

```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\
distributions.py:2619: FutureWarning: `distplot` is a deprecated
function and will be removed in a future version. Please adapt your
code to use either `displot` (a figure-level function with similar
flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```



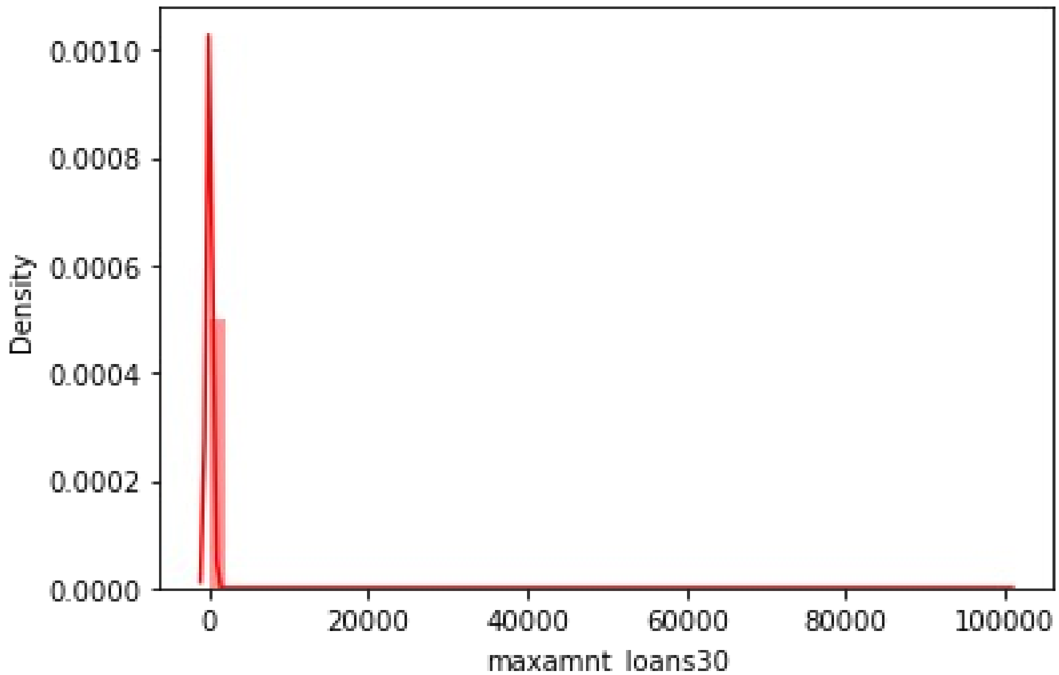

```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\
distributions.py:2619: FutureWarning: `distplot` is a deprecated
function and will be removed in a future version. Please adapt your
code to use either `displot` (a figure-level function with similar
flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```



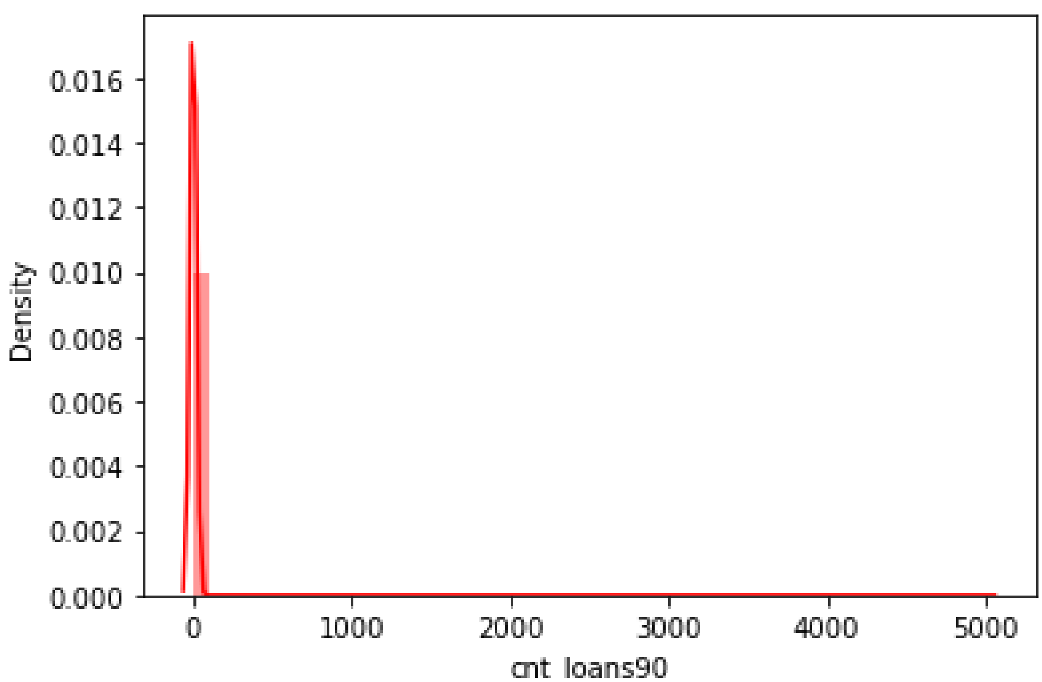
```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\
distributions.py:2619: FutureWarning: `distplot` is a deprecated
function and will be removed in a future version. Please adapt your
code to use either `displot` (a figure-level function with similar
flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```



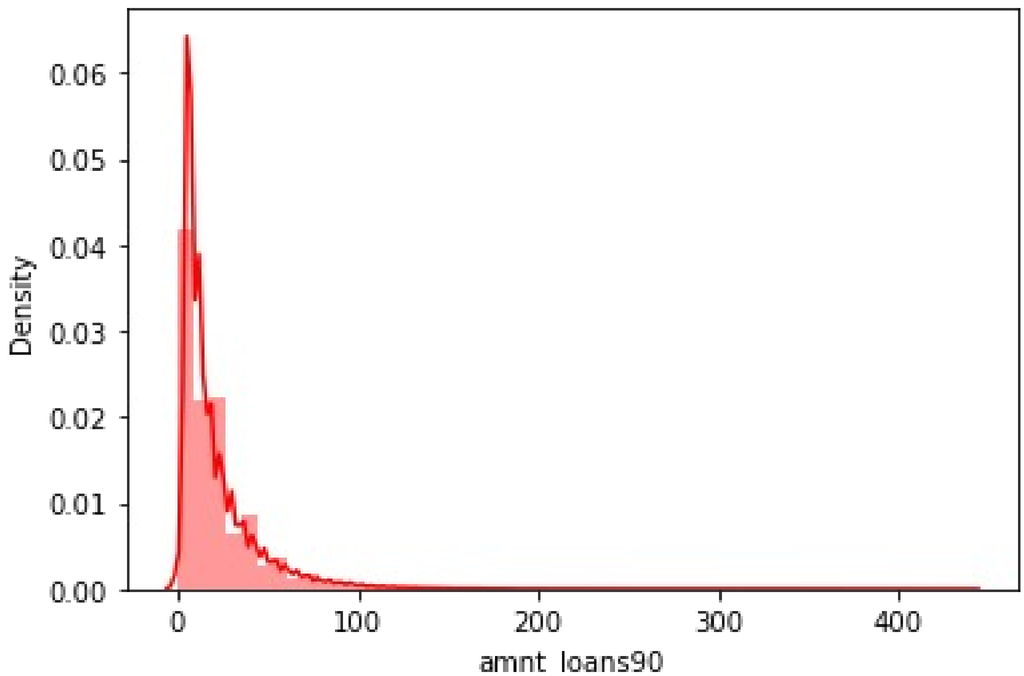

```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\
distributions.py:2619: FutureWarning: `distplot` is a deprecated
function and will be removed in a future version. Please adapt your
code to use either `displot` (a figure-level function with similar
flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```



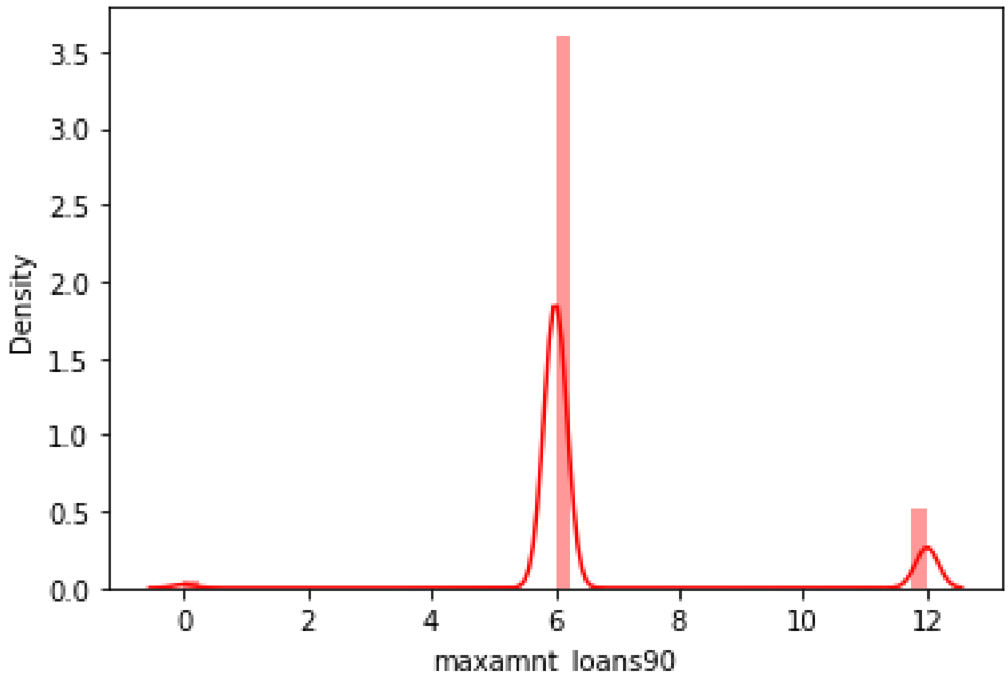

```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\
distributions.py:2619: FutureWarning: `distplot` is a deprecated
function and will be removed in a future version. Please adapt your
code to use either `displot` (a figure-level function with similar
flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```



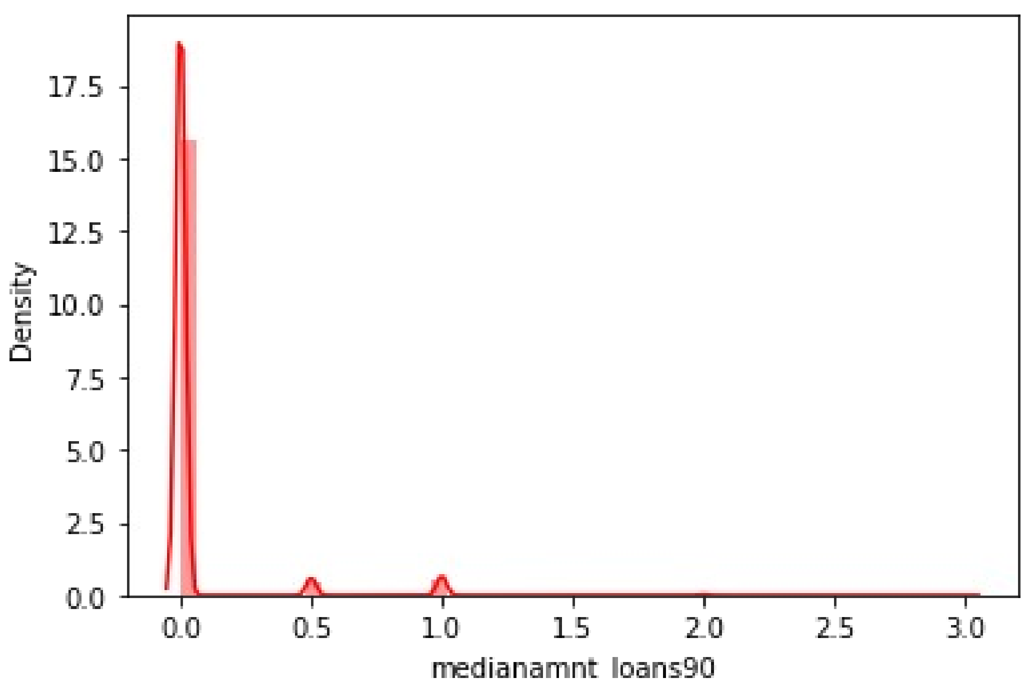
```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\
distributions.py:2619: FutureWarning: `distplot` is a deprecated
function and will be removed in a future version. Please adapt your
code to use either `displot` (a figure-level function with similar
flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```



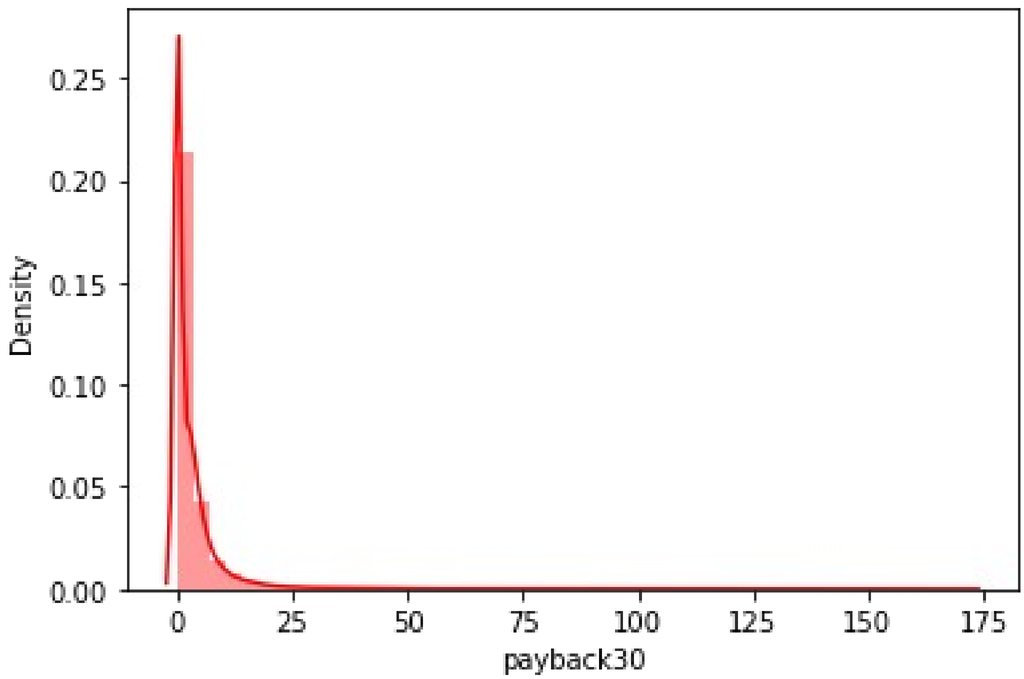

```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\
distributions.py:2619: FutureWarning: `distplot` is a deprecated
function and will be removed in a future version. Please adapt your
code to use either `displot` (a figure-level function with similar
flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```



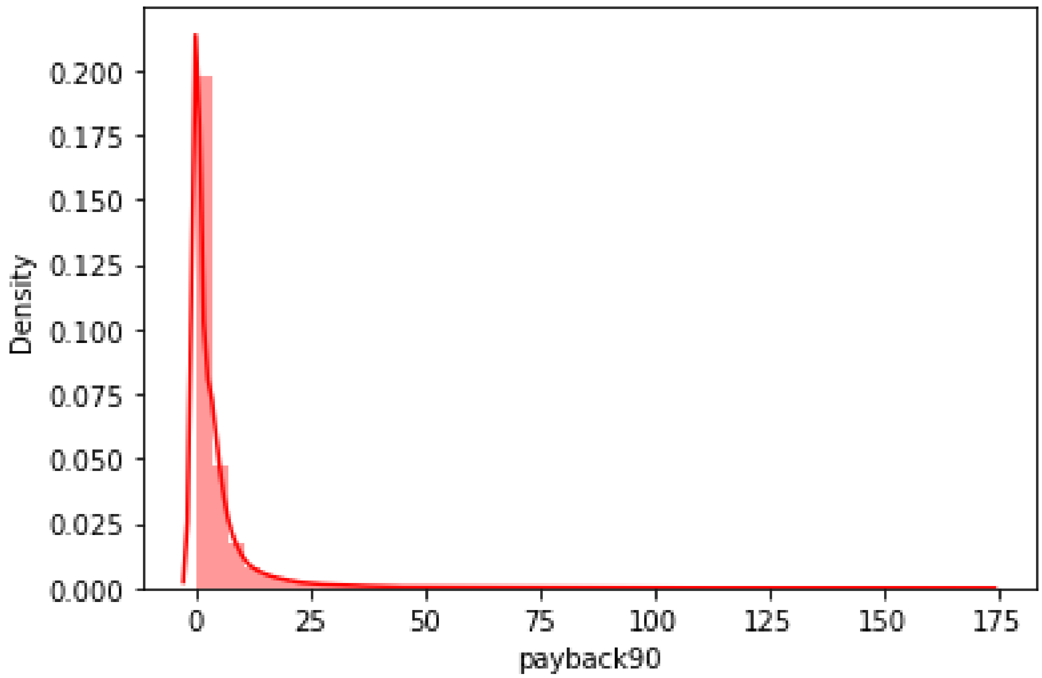

```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\
distributions.py:2619: FutureWarning: `distplot` is a deprecated
function and will be removed in a future version. Please adapt your
code to use either `displot` (a figure-level function with similar
flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```



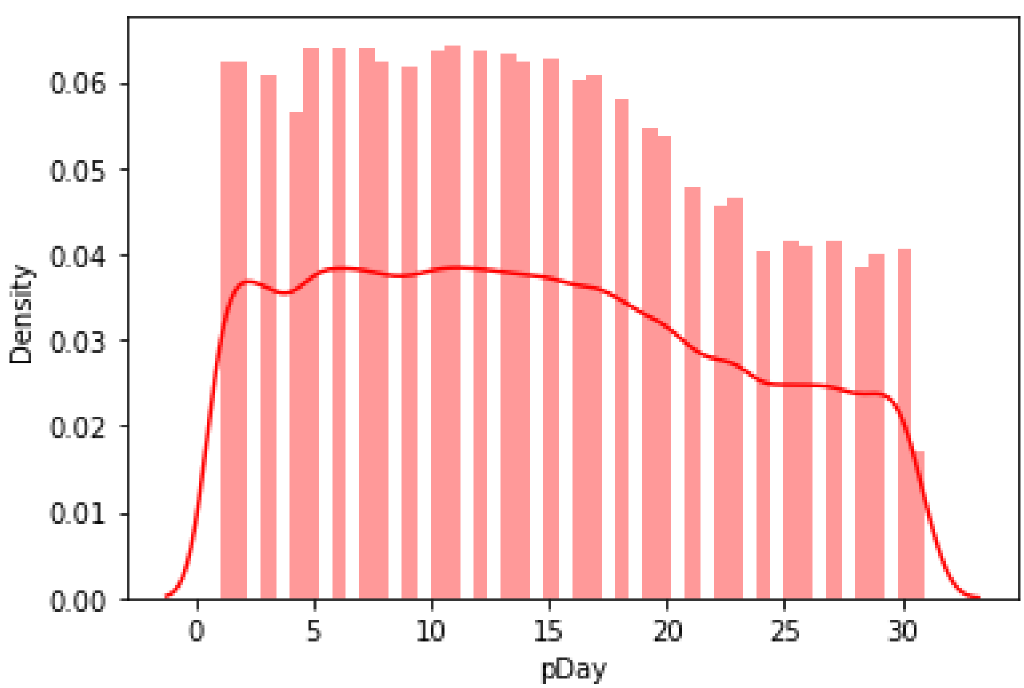
```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\
distributions.py:2619: FutureWarning: `distplot` is a deprecated
function and will be removed in a future version. Please adapt your
code to use either `displot` (a figure-level function with similar
flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```



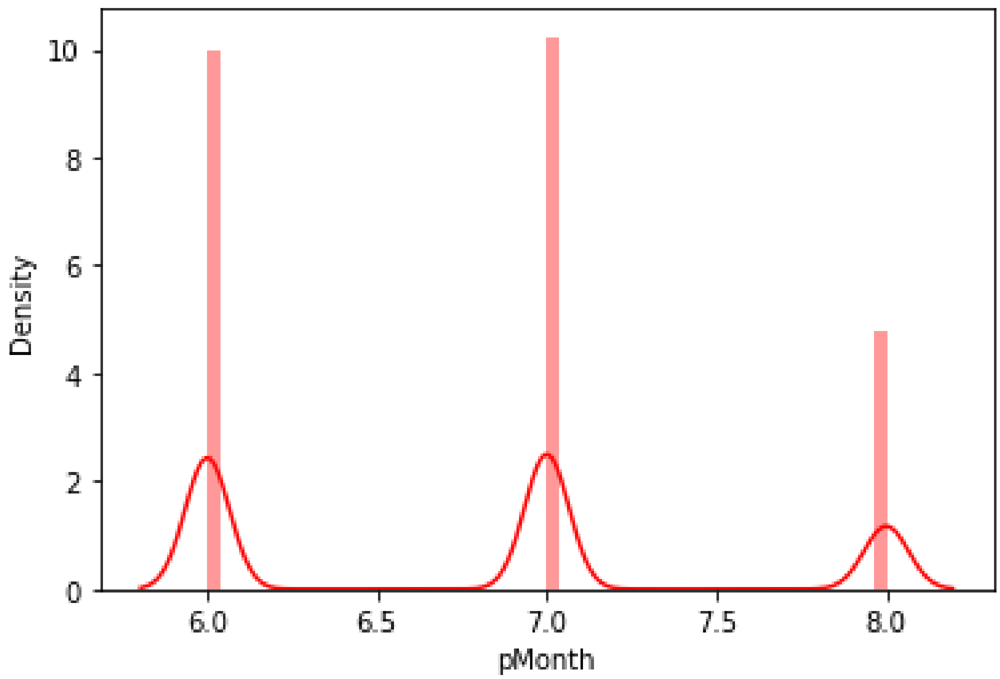

```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\
distributions.py:2619: FutureWarning: `distplot` is a deprecated
function and will be removed in a future version. Please adapt your
code to use either `displot` (a figure-level function with similar
flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```



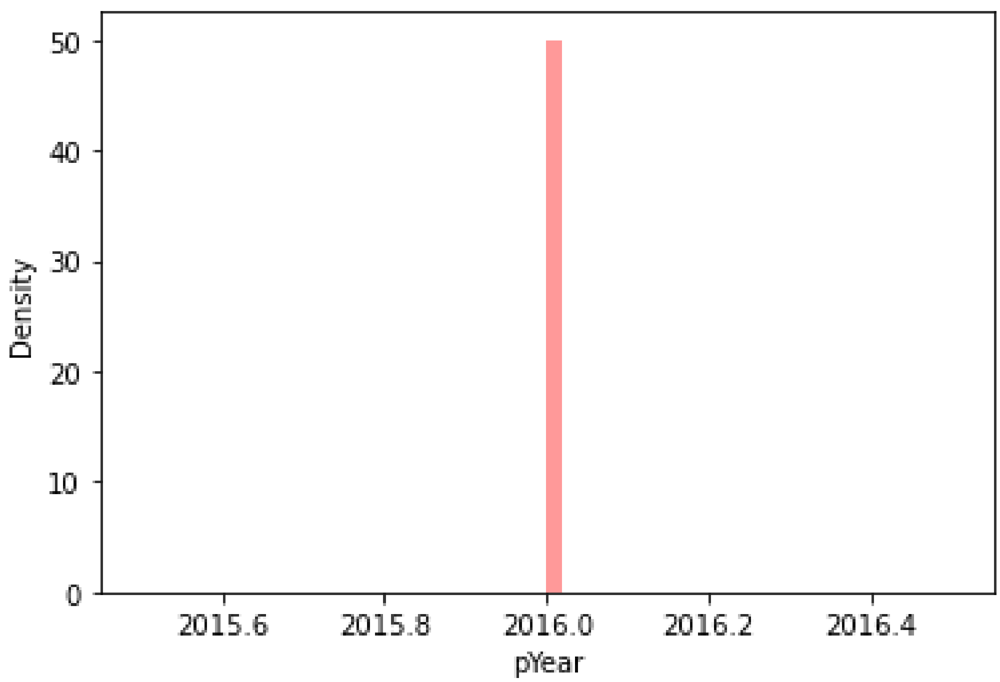

```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\
distributions.py:2619: FutureWarning: `distplot` is a deprecated
function and will be removed in a future version. Please adapt your
code to use either `displot` (a figure-level function with similar
flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```



```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\
distributions.py:2619: FutureWarning: `distplot` is a deprecated
function and will be removed in a future version. Please adapt your
code to use either `displot` (a figure-level function with similar
flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```




```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\
distributions.py:2619: FutureWarning: `distplot` is a deprecated
function and will be removed in a future version. Please adapt your
code to use either `displot` (a figure-level function with similar
flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\
distributions.py:316: UserWarning: Dataset has 0 variance; skipping
density estimate. Pass `warn_singular=False` to disable this warning.
  warnings.warn(msg, UserWarning)
```




```
df.skew()
```

```
C:\Users\hamsa\AppData\Local\Temp/ipykernel_25188/1665899112.py:1:
FutureWarning: Dropping of nuisance columns in DataFrame reductions
(with 'numeric_only=None') is deprecated; in a future version this
will raise TypeError.  Select only valid columns before calling the
reduction.
  df.skew()
```

```
label                   -2.088847
aon                     10.365026
daily_decr90             4.301490
rental90                 4.530925
last_rech_date_ma       14.852116
last_rech_date_da       14.781824
last_rech_amt_ma         3.830612
cnt_ma_rech30            3.471313
fr_ma_rech30            14.822224
sumamnt_ma_rech30        7.134012
medianamnt_ma_rech30     3.519213
medianmarechprebal30    14.677544
cnt_ma_rech90            3.558616
fr_ma_rech90             2.250443
sumamnt_ma_rech90        5.231693
medianamnt_ma_rech90     3.753115
medianmarechprebal90    43.576364
cnt_da_rech30           17.749485
fr_da_rech30            14.728609
cnt_da_rech90           28.396293
```

```
fr_da_rech90            28.959851
cnt_loans30              2.737584
maxamnt_loans30         17.718074
cnt_loans90             16.717192
amnt_loans90             3.165962
maxamnt_loans90          1.650198
medianamnt_loans90       4.774958
payback30                8.193009
payback90                6.763241
pDay                     0.200706
pMonth                   0.351293
pYear                    0.000000
dtype: float64
```

```
#Treating Skewness via square root method.
#df.skew()
#for col in df.skew().index:
    #if col in df.describe().columns:
        #if df[col].skew()>0.55:
            #df[col]=np.sqrt(df[col])
```

```
df.skew()
```

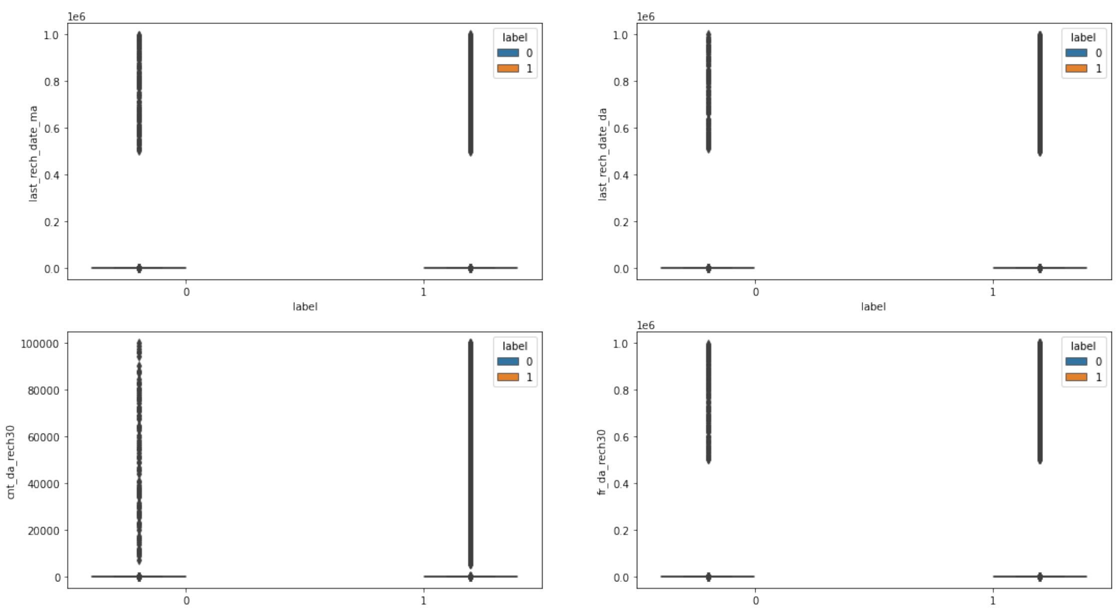
```
C:\Users\hamsa\AppData\Local\Temp/ipykernel_25188/547062910.py:1:
FutureWarning: Dropping of nuisance columns in DataFrame reductions
(with 'numeric_only=None') is deprecated; in a future version this
will raise TypeError.  Select only valid columns before calling the
reduction.
  df.skew()
```

```
label                   -2.088847
aon                     10.365026
daily_decr90             4.301490
rental90                 4.530925
last_rech_date_ma       14.852116
last_rech_date_da       14.781824
last_rech_amt_ma         3.830612
cnt_ma_rech30            3.471313
fr_ma_rech30            14.822224
sumamnt_ma_rech30        7.134012
medianamnt_ma_rech30     3.519213
medianmarechprebal30    14.677544
cnt_ma_rech90            3.558616
fr_ma_rech90             2.250443
sumamnt_ma_rech90        5.231693
medianamnt_ma_rech90     3.753115
medianmarechprebal90    43.576364
cnt_da_rech30           17.749485
fr_da_rech30            14.728609
cnt_da_rech90           28.396293
fr_da_rech90            28.959851
```

```
cnt_loans30              2.737584
maxamnt_loans30         17.718074
cnt_loans90             16.717192
amnt_loans90             3.165962
maxamnt_loans90          1.650198
medianamnt_loans90       4.774958
payback30                8.193009
payback90                6.763241
pDay                     0.200706
pMonth                   0.351293
pYear                    0.000000
dtype: float64
```

```
#plotting outliers

fig, ((ax1, ax2), (ax3, ax4)) = plt.subplots(nrows=2, ncols=2, figsize
= (18, 10))
sns.boxplot(ax=ax1, x = 'label', y = 'last_rech_date_ma', hue =
'label', data = df)
sns.boxplot(ax=ax2, x = 'label', y = 'last_rech_date_da', hue =
'label', data = df)
sns.boxplot(ax=ax3, x = 'label', y = 'cnt_da_rech30', hue = 'label',
data = df)
sns.boxplot(ax=ax4, x = 'label', y = 'fr_da_rech30', hue = 'label',
data = df)
```

```
<AxesSubplot:xlabel='label', ylabel='fr_da_rech30'>
```



Observation:

There are too many outliers present in our dataset.So we need to remove it. But before removing please check that only 8 to 10% of data removed.

```
#Creating a copy of our dataset
df2=df1.copy()
#Dropping the object columns
df1.drop(columns=['msisdn','pdate'],axis=1,inplace=True)
```

```
df1.columns
```

```
Index(['label', 'aon', 'daily_decr30', 'daily_decr90', 'rental30',
'rental90',
       'last_rech_date_ma', 'last_rech_date_da', 'last_rech_amt_ma',
       'cnt_ma_rech30', 'fr_ma_rech30', 'sumamnt_ma_rech30',
       'medianamnt_ma_rech30', 'medianmarechprebal30',
'cnt_ma_rech90',
       'fr_ma_rech90', 'sumamnt_ma_rech90', 'medianamnt_ma_rech90',
       'medianmarechprebal90', 'cnt_da_rech30', 'fr_da_rech30',
       'cnt_da_rech90', 'fr_da_rech90', 'cnt_loans30', 'amnt_loans30',
       'maxamnt_loans30', 'medianamnt_loans30', 'cnt_loans90',
'amnt_loans90',
       'maxamnt_loans90', 'medianamnt_loans90', 'payback30',
'payback90'],
      dtype='object')
```

```
from scipy.stats import zscore
z=np.abs(zscore(df1))
z
```

```
           label       aon  daily_decr30  daily_decr90  rental30
rental90  \
0       2.647896  0.103577      0.252299      0.276346  0.573844
0.558583
1       0.377658  0.097764      0.731037      0.553380  0.231788
0.036020
2       0.377658  0.100102      0.432011      0.429033  0.416020
0.447674
3       0.377658  0.103986      0.581326      0.555125  0.587935
0.576036
4       0.377658  0.094660      0.567293      0.543274  0.369886
0.413227
...          ...       ...           ...           ...       ...
...
209588  0.377658  0.101833      0.567157      0.543159  0.372140
0.414910
209589  0.377658  0.092969      0.579622      0.553686  0.223791
0.304144
209590  0.377658  0.093788      0.700790      0.533194  0.735567
0.937500
```

```
209591  0.377658  0.084289      0.770755      0.594558  0.529352
0.433039
209592  0.377658  0.086284      0.096744      0.141746  0.512620
0.494278

        last_rech_date_ma  last_rech_date_da  last_rech_amt_ma
cnt_ma_rech30  \
0                0.069637           0.069550          0.221637
0.464760
1                0.069303           0.069550          1.570178
0.699718
2                0.069619           0.069550          0.221637
0.699718
3                0.068914           0.069550          0.471344
0.934677
4                0.069600           0.069550          0.103151
0.710030
...                   ...                ...               ...
...
209588           0.069656           0.069550          0.836664
0.229802
209589           0.069600           0.069550          0.544737
0.005156
209590           0.069619           0.069550          0.221637
0.240114
209591           0.069637           0.068838          0.544737
0.240114
209592           0.069433           0.069550          2.303692
0.464760

        ...  cnt_loans30  amnt_loans30  maxamnt_loans30
medianamnt_loans30  \
0       ...     0.297116      0.342470         0.063284
0.247794
1       ...     0.688582      0.342470         0.061871
0.247794
2       ...     0.688582      0.687700         0.063284
0.247794
3       ...     0.297116      0.342470         0.063284
0.247794
4       ...     1.660218      1.383682         0.063284
0.247794
...     ...          ...           ...              ...
...
209588  ...     0.297116      0.342470         0.063284
0.247794
209589  ...     0.094351      0.002761         0.063284
0.247794
209590  ...     0.485818      1.383682         0.061871
0.247794
```

```
209591  ...     0.297116      0.002761        0.061871
0.247794
209592  ...     0.297116      0.002761        0.061871
0.247794

        cnt_loans90  amnt_loans90  maxamnt_loans90  medianamnt_loans90
\
0          0.073493      0.439950         0.334212            0.229594

1          0.077941      0.439950         2.517690            0.229594

2          0.077941      0.666624         0.334212            0.229594

3          0.073493      0.439950         0.334212            0.229594

4          0.051250      0.693417         0.334212            0.229594

...             ...           ...              ...                 ...

209588     0.073493      0.439950         0.334212            0.229594

209589     0.069044      0.213277         0.334212            0.229594

209590     0.055699      1.146764         2.517690            0.229594

209591     0.069044      0.013396         2.517690            0.229594

209592     0.073493      0.213277         2.517690            0.229594


        payback30  payback90
0        2.904700   2.394093
1        0.385630   0.419233
2        0.385630   0.419233
3        0.385630   0.419233
4        0.120890   0.192873
...           ...        ...
209588   0.272170   0.322221
209589   0.272170   0.322221
209590   0.068209   0.047356
209591   0.385630   0.599385
209592   0.385630   0.419233

[209593 rows x 33 columns]
```

```
threshold=3
print(np.where(z>3))
```

```
(array([    21,     22,     22, ..., 209586, 209587, 209587],
dtype=int64), array([15, 15, 32, ..., 28, 26, 30], dtype=int64))
```

```
df1_new=df1[(z<3).all(axis=1)]
```

```
#Checking the shape
print(df1.shape,'\t\t',df1_new.shape)
```

```
(209593, 33)           (161465, 33)
```

```
#Converting the categorical data into numeric variables
# Transform Non numeric columns into Numeric columns

from sklearn.preprocessing import LabelEncoder

le=LabelEncoder()

for column in df.columns:
    if df[column].dtype==np.number:
        continue
    df[column]=le.fit_transform(df[column])
```

```
C:\Users\hamsa\AppData\Local\Temp/ipykernel_25188/2334802243.py:9:
DeprecationWarning: Converting `np.inexact` or `np.floating` to a
dtype is deprecated. The current result is `float64` which is not
strictly correct.
  if df[column].dtype==np.number:
```

```
df.head()
```

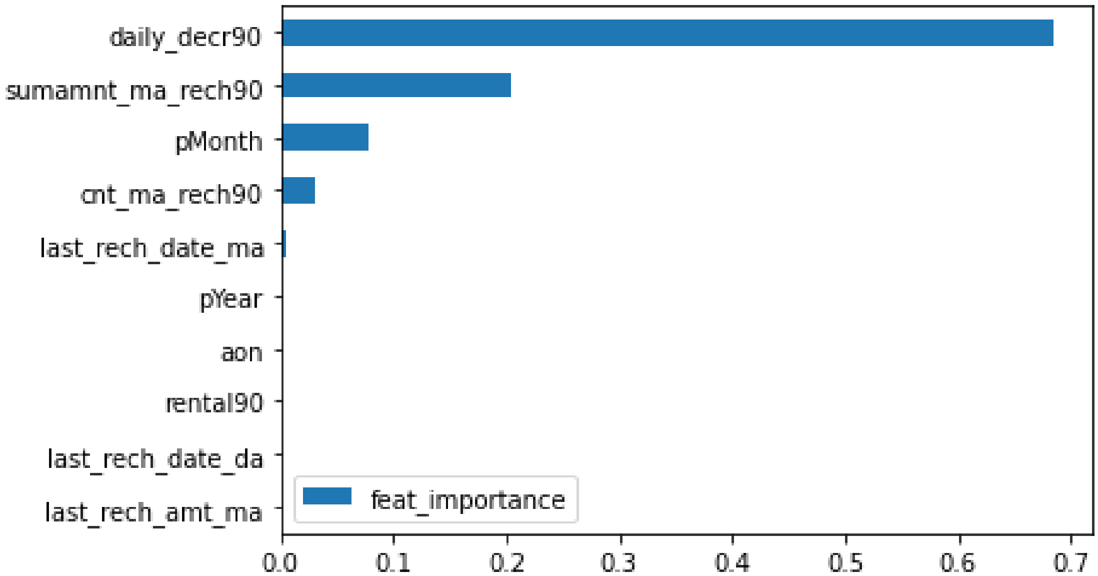
```
   label  msisdn    aon  daily_decr90  rental90  last_rech_date_ma  \
0      0   40191  272.0   3065.150000    260.13                2.0
1      1  142291  712.0  12124.750000   3691.26               20.0
2      1   33594  535.0   1398.000000    900.13                3.0
3      1  104157  241.0     21.228000    159.42               41.0
4      1    6910  947.0    150.619333   1098.90                4.0

   last_rech_date_da  last_rech_amt_ma  cnt_ma_rech30
fr_ma_rech30  ...  \
0                0.0                14              2
21.0  ...
1                0.0                38              1
0.0  ...
2                0.0                14              1
0.0  ...
3                0.0                10              0
0.0  ...
4                0.0                23              7
2.0  ...

   maxamnt_loans30  cnt_loans90  amnt_loans90  maxamnt_loans90  \
```

```
0              6.0          2.0              2                 1
1             12.0          1.0              2                 2
2              6.0          1.0              1                 1
3              6.0          2.0              2                 1
4              6.0          7.0              7                 1

   medianamnt_loans90  payback30  payback90  pDay  pMonth  pYear
0                 0.0  29.000000  29.000000    19       1      0
1                 0.0   0.000000   0.000000     9       2      0
2                 0.0   0.000000   0.000000    18       2      0
3                 0.0   0.000000   0.000000     5       0      0
4                 0.0   2.333333   2.333333    21       0      0

[5 rows x 33 columns]
```

```
#feature importance

#Splitting the data into x and y

x = df.drop(['label'], axis=1)

y = df['label']

from sklearn.tree import DecisionTreeClassifier
dt = DecisionTreeClassifier(max_depth=3)
dt.fit(x, y)
```

```
DecisionTreeClassifier(max_depth=3)
```

```
dt_features = pd.DataFrame(dt.feature_importances_, index=x.columns,
columns=['feat_importance'])
dt_features.sort_values('feat_importance').tail(10).plot.barh()
plt.show()
```



By looking at the daily_decr90 which is Daily amount spent from main account, averaged over last 90 days (in Indonesian Rupiah), it seems that this feature helps to discriminate the data indeed. This feature can bring insights for company when analyzing a customers.

Model Training

```
#Scaling in input variables
from sklearn.preprocessing import StandardScaler
ss=StandardScaler()
x=ss.fit_transform(x)

#Splitting the data into training and testing data

from sklearn.model_selection import train_test_split,cross_val_score
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=.20,rando
m_state=42,stratify=y)

from sklearn.neighbors import KNeighborsClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.naive_bayes import GaussianNB
from sklearn.ensemble import RandomForestClassifier

KNN=KNeighborsClassifier(n_neighbors=10)
LR=LogisticRegression()
DT=DecisionTreeClassifier(random_state=20)
GNB=GaussianNB()
RF=RandomForestClassifier()

models = []
models.append(('KNeighborsClassifier', KNN))
models.append(('LogisticRegression', LR))
```

```
models.append(('DecisionTreeClassifier',DT))
models.append(('GaussianNB', GNB))
models.append(('RandomForestClassifier', RF))

from sklearn.metrics import
classification_report,confusion_matrix,accuracy_score,roc_curve,auc

Model=[]
score=[]
cvs=[]
rocscore=[]
for name,model in models:

print('*****************************',name,'**************************
*****')
    print('\n')
    Model.append(name)
    model.fit(x_train,y_train.values.ravel())
    print(model)
    pre=model.predict(x_test)
    print('\n')
    AS=accuracy_score(y_test,pre)
    print('Accuracy_score = ', AS)
    score.append(AS*100)
    print('\n')
    sc=cross_val_score(model,x,y,cv=10,scoring='accuracy').mean()
    print('Cross_val_Score = ', sc)
    cvs.append(sc*100)
    print('\n')
    false_positive_rate, true_positive_rate, thresholds =
roc_curve(y_test,pre)
    roc_auc= auc(false_positive_rate, true_positive_rate)
    print('roc_auc_score = ',roc_auc)
    rocscore.append(roc_auc*100)
    print('\n')
    print('classification_report\n',classification_report(y_test,pre))
    print('\n')
    cm=confusion_matrix(y_test,pre)
    print(cm)
    print('\n')
    plt.figure(figsize=(10,40))
    plt.subplot(911)
    plt.title(name)
    print(sns.heatmap(cm,annot=True))
    plt.subplot(912)
    plt.title(name)
    plt.plot(false_positive_rate, true_positive_rate, label = 'AUC=
%0.2f'%roc_auc)
    plt.legend(loc='lower right')
    plt.ylabel('True Positive Rate')
```

```
    plt.xlabel('False Positive Rate')
    print('\n\n')
```

```
**************************** KNeighborsClassifier
********************************



KNeighborsClassifier(n_neighbors=10)



Accuracy_score =  0.8699025477194019
```

```
result=pd.DataFrame({'Model': Model, 'Accuracy_score': score, 
'Cross_val_score':cvs, 'Roc_auc_curve':rocscore})
result
```