CAPSTONE PROJECT

(BANKING PROJECT PROBABILITY OF DEFAULT)

shutterstock.com · 1900200964

BY – SAUMYA JAIN (BATCH: PGPDSBA.O.JUNE23.B)

Contents

Introduction - What did you wish to achieve while doing the project?

a) Defining problem statement

This business problem is a supervised learning example for a credit card company. The objective is to predict the probability of default (whether the customer will pay the credit card bill or not) based on the variables provided.

Basically it is needed to predict that the person who has credit card is defaulting or not.

b) Need of the study/project

The objective is to predict the probability of default (whether the customer will pay the credit card bill or not) based on the

variables provided. There are multiple variables on the credit card account, purchase and delinquency information which can be used in the modelling.

Last 2 years data of bank of the customers who has a credit card and is the person is defaulting or not in paying the bills.

c) Understanding business/social opportunity

- This helps in understanding the riskiness of the customers and how much credit is at stake in case the customer defaults. This is an extremely critical part in any organization that lends money [both secured and unsecured loans. Last 2 years data of bank of the customers who has a credit card and is the person is defaulting or not in paying the bills.

Some basic highlights according to 2024:

The credit card default rate in India has been rising significantly, reflecting the increased usage and associated financial stress among consumers. As of 2024, the outstanding credit card balance in India has reached Rs 2.4 lakh crore, with defaults on the rise.

EDA- Univariate / Bi-variate / Multi-variate analysis to understand relationship b/w variables.- Both visual and non-visual understanding of the data.

a) **Understanding how data was collected in terms of time, frequency, and methodology.**

- **Data was collected in terms of time:** The data set was of last 2 years i.e., 24 months.
- **Frequency:** In monthly basis for last 3 months and 6 months for the dataset that has been provided to us.
- **Methodology:** Last 2 years data of bank of the customers who has a credit card and is the person is defaulting or not in paying the bills.

b) **Visual inspection of data (rows, columns, descriptive details)**

The dataset named ‘PD_modelling_dataset’ has 36 columns in total and 99980 rows which has last 3 rows as blank and mostly data is not there so to proceed further we have removed the last 3 rows (Row Number – 99978,99979,99980)

So , after removing the last 3 rows we have 36 columns and 99977 as rows in the dataset.

We can see that there are many missing values in the dataset.

- **Head of the dataset :**

  Top 5 rows of the dataset

	userid	default	acct_amt_added_12_24m	acct_days_in_dc_12_24m	acct_days_in_rem_12_24m	acct_days_in_term_12_24m	acct_incoming_debt_vs_paid_0_24m	acct_status	acct_worst_status_0_3m
0	4567129	0.0	0	0.0	0.0	0.0	0.0	1.0	1.0
1	2635118	0.0	0	0.0	0.0	0.0	0.0	1.0	1.0
2	4804232	0.0	0	0.0	0.0	0.0	0.0	0.0	0.0
3	1442693	0.0	0	0.0	0.0	0.0	0.0	0.0	0.0
4	4575322	0.0	0	0.0	0.0	0.0	0.0	0.0	0.0

5 rows × 36 columns

- **Tail of the dataset:**

  Top last 5 rows of the dataset

	userid	default	acct_amt_added_12_24m	acct_days_in_dc_12_24m	acct_days_in_rem_12_24m	acct_days_in_term_12_24m	acct_incoming_debt_vs_paid_0_24m	acct_status	acct_worst_status_0_3
89971	1545432	0.0	0	0.0	0.0	0.0	0.0	0.0	0.
89972	3061692	0.0	0	0.0	0.0	0.0	0.0	0.0	0.
89973	1535658	0.0	0	0.0	0.0	0.0	0.0	1.0	1.
89974	2142946	0.0	0	0.0	0.0	0.0	0.0	0.0	0.
89975	3344745	0.0	0	0.0	0.0	0.0	0.0	0.0	0.

5 rows × 36 columns

Info of the dataset:

- Most of the dataset is of float datatype and some of the dataset is integer and object. There are many missing values present in the dataset.

```
[8] <class 'pandas.core.frame.DataFrame'>
RangeIndex: 99976 entries, 0 to 99975
Data columns (total 36 columns):
 #   Column                              Non-Null Count  Dtype
---  ------                              --------------  -----
 0   userid                              99976 non-null  int64
 1   default                             89976 non-null  float64
 2   acct_amt_added_12_24m               99976 non-null  int64
 3   acct_days_in_dc_12_24m              88140 non-null  float64
 4   acct_days_in_rem_12_24m             88140 non-null  float64
 5   acct_days_in_term_12_24m            88140 non-null  float64
 6   acct_incoming_debt_vs_paid_0_24m    40661 non-null  float64
 7   acct_status                         45603 non-null  float64
 8   acct_worst_status_0_3m              45603 non-null  float64
 9   acct_worst_status_12_24m            33215 non-null  float64
 10  acct_worst_status_3_6m              42274 non-null  float64
 11  acct_worst_status_6_12m             39626 non-null  float64
 12  age                                 99976 non-null  int64
 13  avg_payment_span_0_12m              76140 non-null  float64
 14  avg_payment_span_0_3m               50671 non-null  float64
 15  merchant_category                   99976 non-null  object
 16  merchant_group                      99967 non-null  object
 17  has_paid                            88942 non-null  float64
 18  max_paid_inv_0_12m                  88942 non-null  float64
 19  max_paid_inv_0_24m                  88942 non-null  float64
 20  name_in_email                       88942 non-null  object
 21  num_active_div_by_paid_inv_0_12m    70051 non-null  float64
 22  num_active_inv                      88942 non-null  float64
 23  num_arch_dc_0_12m                   88942 non-null  float64
 24  num_arch_dc_12_24m                  88942 non-null  float64
 25  num_arch_ok_0_12m                   88942 non-null  float64
 26  num_arch_ok_12_24m                  88942 non-null  float64
 27  num_arch_rem_0_12m                  88942 non-null  float64
 28  status_max_archived_0_6_months      88942 non-null  float64
 29  status_max_archived_0_12_months     88942 non-null  float64
 30  status_max_archived_0_24_months     88942 non-null  float64
```

- **Description of the dataset**

	userid	default	acct_amt_added_12_24m	acct_days_in_dc_12_24m	acct_days_in_rem_12_24m	acct_days_in_term_12_24m	acct_incoming_debt_vs_paid_0_24m	acct_status	acct_worst_s
count	8.997600e+04	89976.000000	8.997600e+04	89976.000000	89976.000000	89976.000000	89976.000000	89976.000000	8
mean	2.998571e+06	0.014315	1.227615e+04	0.191529	4.471415	0.253712	0.541510	0.475149	
std	1.154905e+06	0.118786	3.546356e+04	5.285650	21.614243	2.752792	17.189054	0.536425	
min	1.000053e+06	0.000000	0.000000e+00	0.000000	0.000000	0.000000	0.000000	0.000000	
25%	1.998872e+06	0.000000	0.000000e+00	0.000000	0.000000	0.000000	0.000000	0.000000	
50%	2.997714e+06	0.000000	0.000000e+00	0.000000	0.000000	0.000000	0.000000	0.000000	
75%	4.001145e+06	0.000000	4.984250e+03	0.000000	0.000000	0.000000	0.000952	1.000000	
max	4.999868e+06	1.000000	1.128775e+06	362.000000	365.000000	97.000000	3914.000000	4.000000	

- The mean , count , std , min , max has been calculated for every columns.

c) Understanding of attributes (variable info, renaming if required)

Info of the dataset:

- Most of the dataset is of float datatype and some of the dataset is integer and object. There are many missing values present in the dataset.

```
[8]  <class 'pandas.core.frame.DataFrame'>
     RangeIndex: 99976 entries, 0 to 99975
     Data columns (total 36 columns):
      #   Column                             Non-Null Count  Dtype
     ---  ------                             --------------  -----
      0   userid                             99976 non-null  int64
      1   default                            89976 non-null  float64
      2   acct_amt_added_12_24m              99976 non-null  int64
      3   acct_days_in_dc_12_24m             88140 non-null  float64
      4   acct_days_in_rem_12_24m            88140 non-null  float64
      5   acct_days_in_term_12_24m           88140 non-null  float64
      6   acct_incoming_debt_vs_paid_0_24m   40661 non-null  float64
      7   acct_status                        45603 non-null  float64
      8   acct_worst_status_0_3m             45603 non-null  float64
      9   acct_worst_status_12_24m           33215 non-null  float64
      10  acct_worst_status_3_6m             42274 non-null  float64
      11  acct_worst_status_6_12m            39626 non-null  float64
      12  age                                99976 non-null  int64
      13  avg_payment_span_0_12m             76140 non-null  float64
      14  avg_payment_span_0_3m              50671 non-null  float64
      15  merchant_category                  99976 non-null  object
      16  merchant_group                     99967 non-null  object
      17  has_paid                           88942 non-null  float64
      18  max_paid_inv_0_12m                 88942 non-null  float64
      19  max_paid_inv_0_24m                 88942 non-null  float64
      20  name_in_email                      88942 non-null  object
      21  num_active_div_by_paid_inv_0_12m   70051 non-null  float64
      22  num_active_inv                     88942 non-null  float64
      23  num_arch_dc_0_12m                  88942 non-null  float64
      24  num_arch_dc_12_24m                 88942 non-null  float64
      25  num_arch_ok_0_12m                  88942 non-null  float64
      26  num_arch_ok_12_24m                 88942 non-null  float64
      27  num_arch_rem_0_12m                 88942 non-null  float64
      28  status_max_archived_0_6_months     88942 non-null  float64
      29  status_max_archived_0_12_months    88942 non-null  float64
      30  status_max_archived_0_24_months    88942 non-null  float64
```

3. Exploratory Data Analysis

a) Univariate analysis (distribution and spread for every continuous attribute, distribution of data in categories for categorical ones)

- **Univariate analysis is done for all continuous values:**

I have separated all the numerical columns and defined all the columns as in one dataset as ‘num’.

Numerical columns that has been separated in one dataset as ‘num’ as below:

All integer, float values datatypes is there in num column.

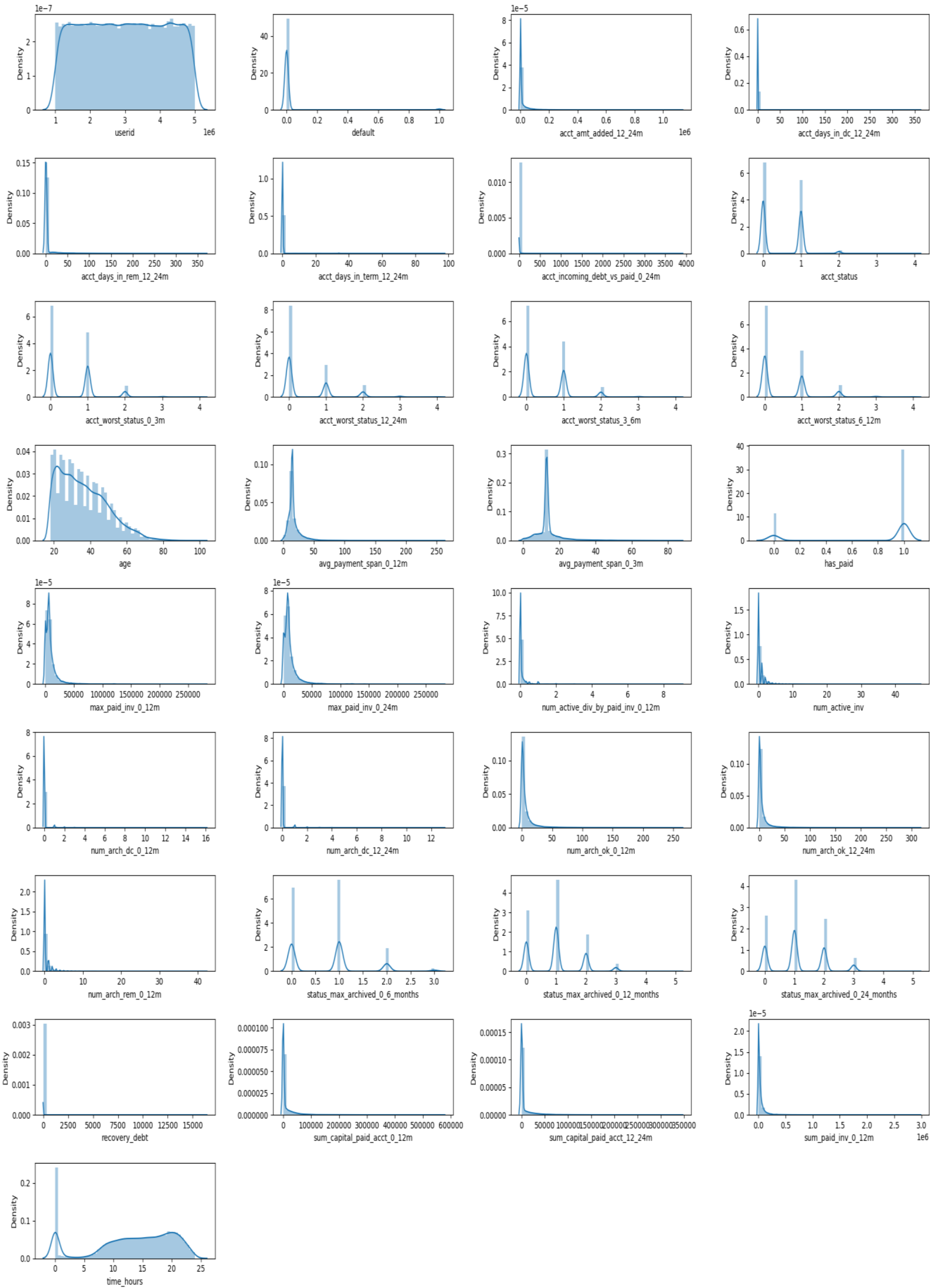
```
Index(['userid', 'default', 'acct_amt_added_12_24m', 'acct_days_in_dc_12_24m',
       'acct_days_in_rem_12_24m', 'acct_days_in_term_12_24m',
       'acct_incoming_debt_vs_paid_0_24m', 'acct_status',
       'acct_worst_status_0_3m', 'acct_worst_status_12_24m',
       'acct_worst_status_3_6m', 'acct_worst_status_6_12m', 'age',
       'avg_payment_span_0_12m', 'avg_payment_span_0_3m', 'has_paid',
       'max_paid_inv_0_12m', 'max_paid_inv_0_24m',
       'num_active_div_by_paid_inv_0_12m', 'num_active_inv',
       'num_arch_dc_0_12m', 'num_arch_dc_12_24m', 'num_arch_ok_0_12m',
       'num_arch_ok_12_24m', 'num_arch_rem_0_12m',
       'status_max_archived_0_6_months', 'status_max_archived_0_12_months',
       'status_max_archived_0_24_months', 'recovery_debt',
       'sum_capital_paid_acct_0_12m', 'sum_capital_paid_acct_12_24m',
       'sum_paid_inv_0_12m', 'time_hours'],
      dtype='object')
```

```
0   userid                           89976 non-null  int64
1   default                          89976 non-null  float64
2   acct_amt_added_12_24m            89976 non-null  int64
3   acct_days_in_dc_12_24m           89976 non-null  float64
4   acct_days_in_rem_12_24m          89976 non-null  float64
5   acct_days_in_term_12_24m         89976 non-null  float64
6   acct_incoming_debt_vs_paid_0_24m 89976 non-null  float64
7   acct_status                      89976 non-null  float64
8   acct_worst_status_0_3m           89976 non-null  float64
9   acct_worst_status_12_24m         89976 non-null  float64
10  acct_worst_status_3_6m           89976 non-null  float64
11  acct_worst_status_6_12m          89976 non-null  float64
12  age                              89976 non-null  int64
13  avg_payment_span_0_12m           89976 non-null  float64
14  avg_payment_span_0_3m            89976 non-null  float64
15  has_paid                         89976 non-null  float64
16  max_paid_inv_0_12m               89976 non-null  float64
17  max_paid_inv_0_24m               89976 non-null  float64
18  num_active_div_by_paid_inv_0_12m 89976 non-null  float64
19  num_active_inv                   89976 non-null  float64
20  num_arch_dc_0_12m                89976 non-null  float64
21  num_arch_dc_12_24m               89976 non-null  float64
22  num_arch_ok_0_12m                89976 non-null  float64
23  num_arch_ok_12_24m               89976 non-null  float64
24  num_arch_rem_0_12m               89976 non-null  float64
25  status_max_archived_0_6_months   89976 non-null  float64
26  status_max_archived_0_12_months  89976 non-null  float64
27  status_max_archived_0_24_months  89976 non-null  float64
28  recovery_debt                    89976 non-null  float64
29  sum_capital_paid_acct_0_12m      89976 non-null  float64
30  sum_capital_paid_acct_12_24m     89976 non-null  float64
31  sum_paid_inv_0_12m               89976 non-null  float64
```

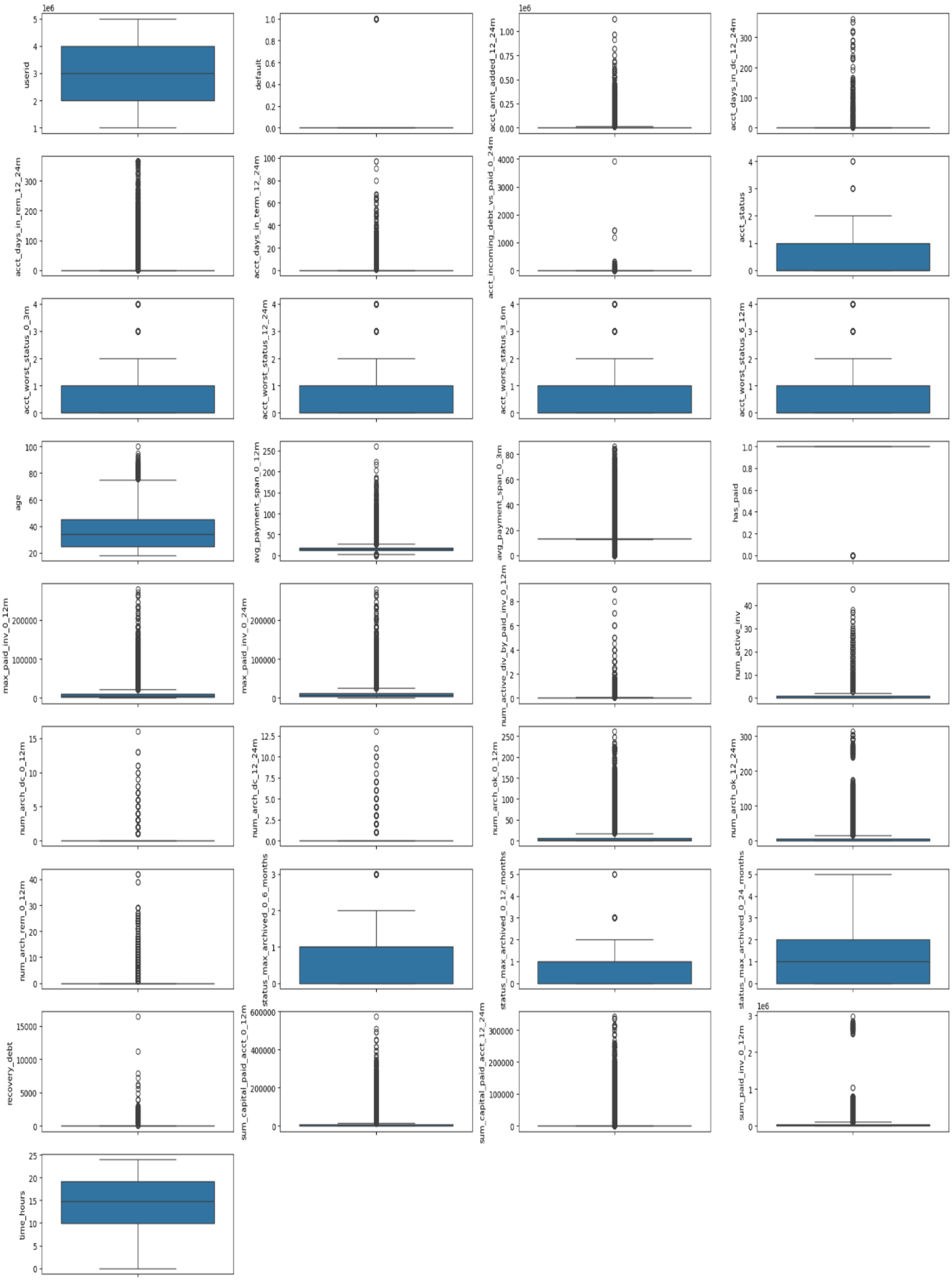
- We have plotted the histplot for all the continuous variable and for the categorical variable we have plotted the countplot .

Inference from the histplot:

- Mostly the values in the default colums are 0 i.e they had already paid the bill and is not defaulting as compared to non- defaulters.
- Most of the credit card user is of age range of 20 to 60 years.
- Maximum persons is taking average time while doing payment is 5 to 25 hrs which is pretty long.

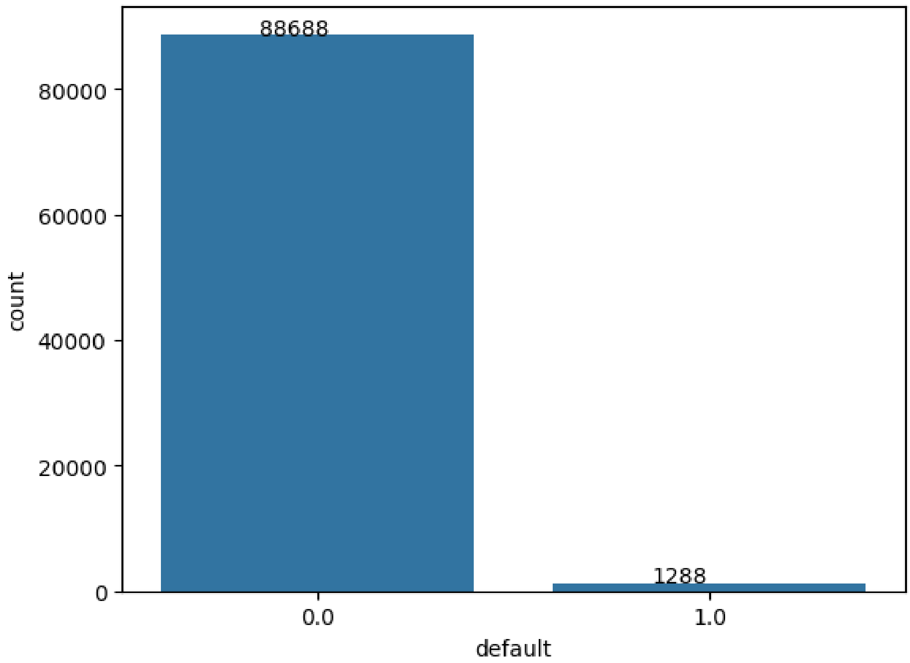

userid
default
acct_amt_added_12_24m
acct_days_in_dc_12_24m
acct_days_in_rem_12_24m
acct_days_in_term_12_24m
acct_incoming_debt_vs_paid_0_24m
acct_status
acct_worst_status_0_3m
acct_worst_status_12_24m
acct_worst_status_3_6m
acct_worst_status_6_12m
age
avg_payment_span_0_12m
avg_payment_span_0_3m
has_paid
max_paid_inv_0_12m
max_paid_inv_0_24m
num_active_div_by_paid_inv_0_12m
num_active_inv
num_arch_dc_0_12m
num_arch_dc_12_24m
num_arch_ok_0_12m
num_arch_ok_12_24m
num_arch_rem_0_12m
status_max_archived_0_6_months
status_max_archived_0_12_months
status_max_archived_0_24_months
recovery_debt
sum_capital_paid_acct_0_12m
sum_capital_paid_acct_12_24m
sum_paid_inv_0_12m
time_hours

- We can see that mostly all the category has outliers present and we are expecting the same also as it is a bank dataset and the values of that is extremely high .



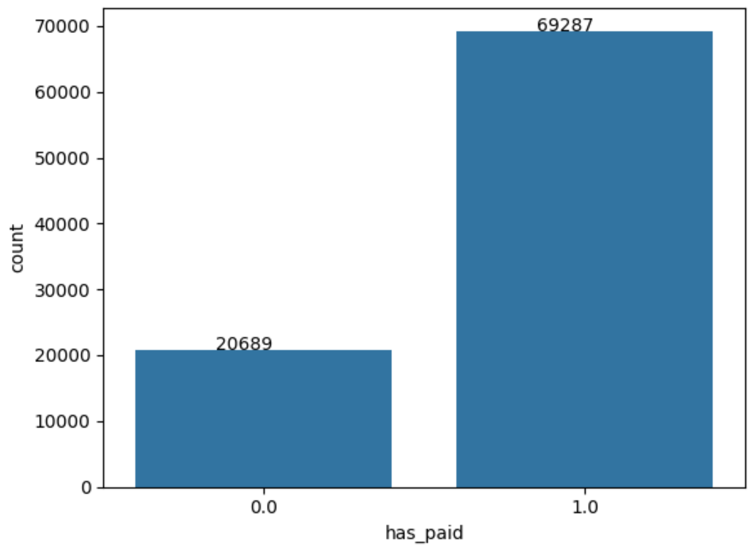
- **PLOT OF DEFAULT COLUMN**

- Most of the person count -88688 who has taken the credit card is not defaulting in paying the bill. However 1288 person is still defaulting.



- **BAR PLOT OF HAS PAID COLUMN**

Most of the customers 69287 person who owned a credit card has paid the bill and around 20689 has not paid the bill.



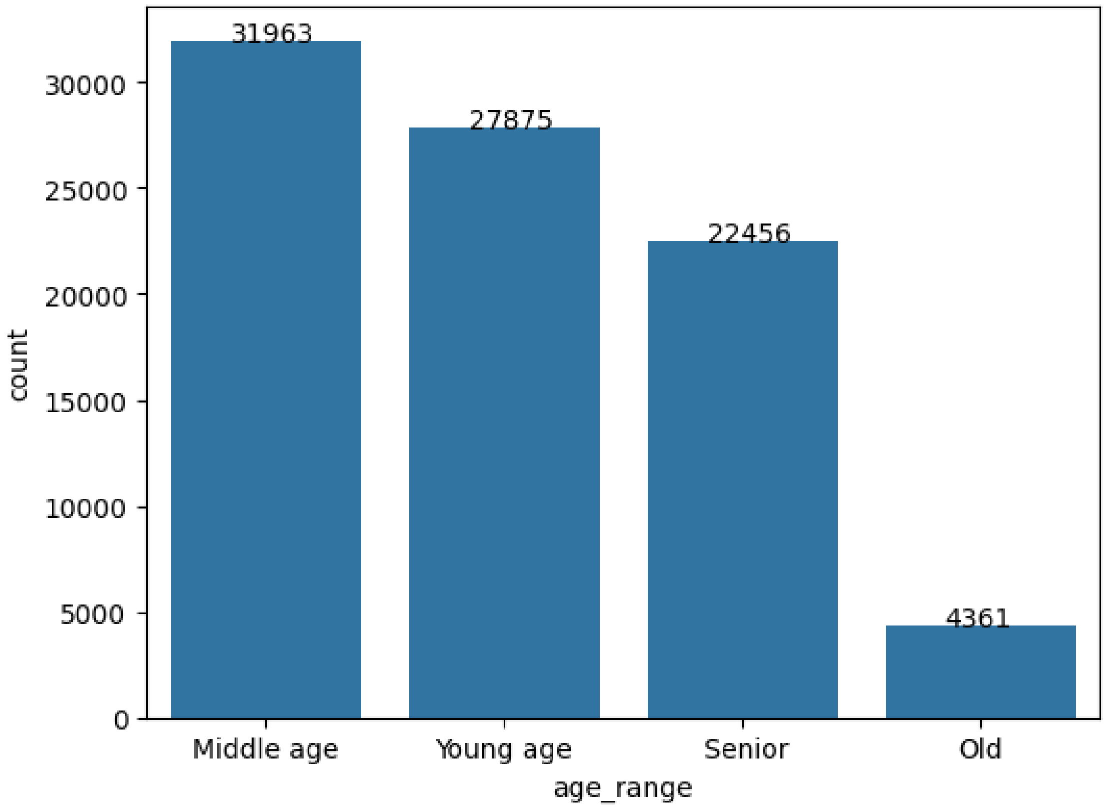
- **BAR PLOT OF DIFFERENT AGE GROUP**

  Maximum number of credit card holders are of middle age followed by young age.

  We have categorized the age into different groups.

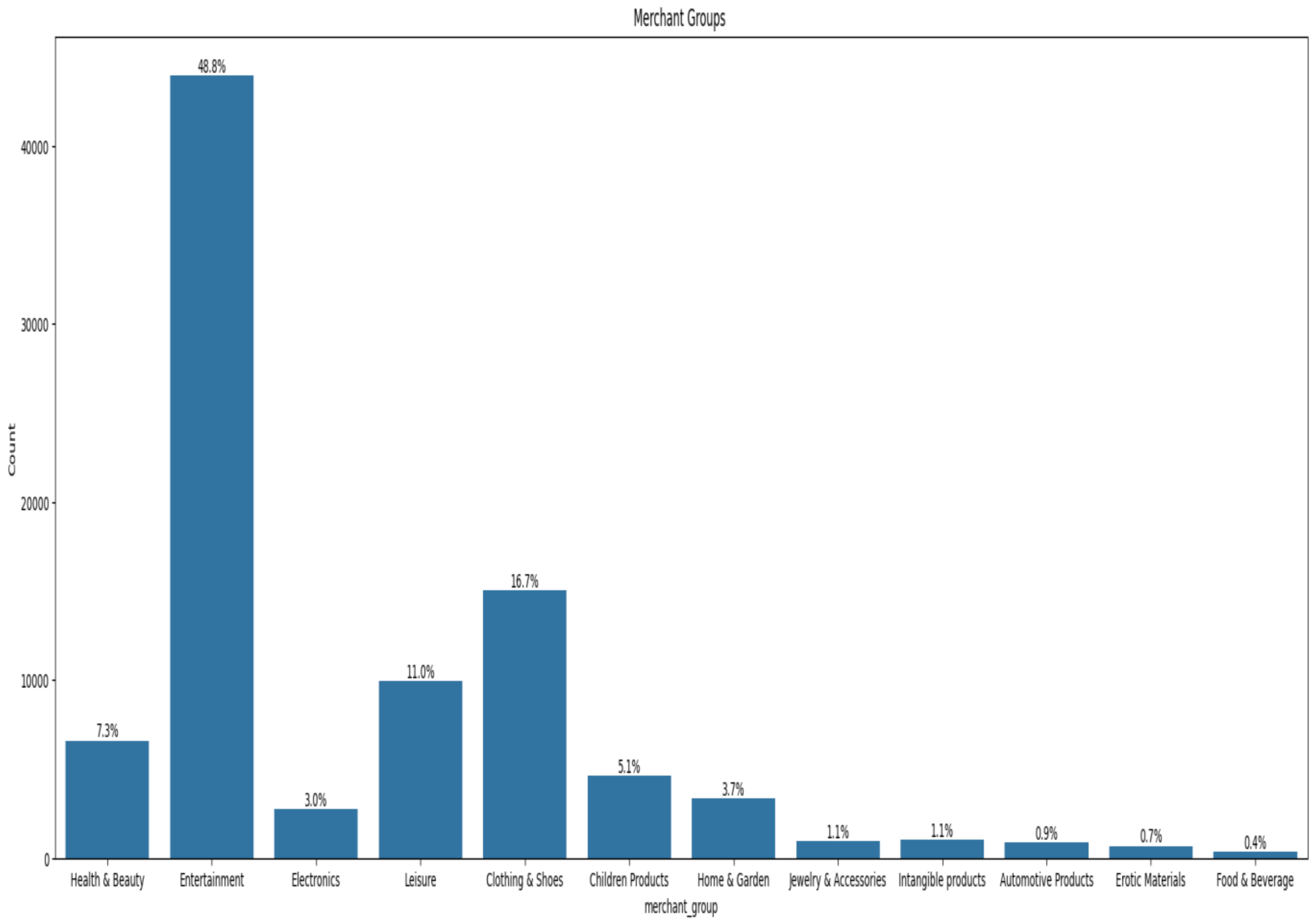
  Such as

  From18-28 --→ Young age
  From 28-42 -→Middle age
  From 42-60-→ Senior
  From 60-100-> old



- **BAR PLOT OF MERCHANT GROUP**

The maximum percentage i.e. 48.8% is of Entertainment category followed by Clothing and Shoes of 16.7% and least is of 0.4% of Food and Beverages category.



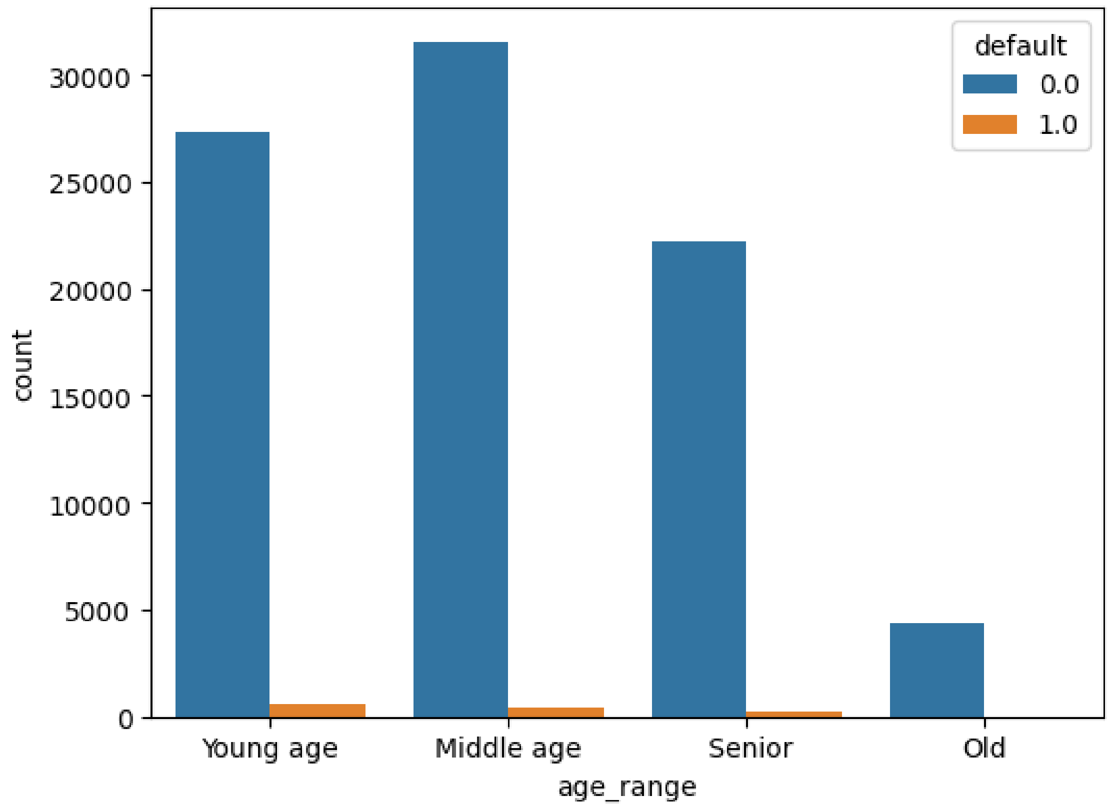
BI-VARIATE ANALYSIS:

- The graph is being plotted between default and the age group of the customer to find out which age group person is defaulting in paying credit card bill.
- Non default is denoted by 0 and of blue colour and 1 is of default – Orange colour.

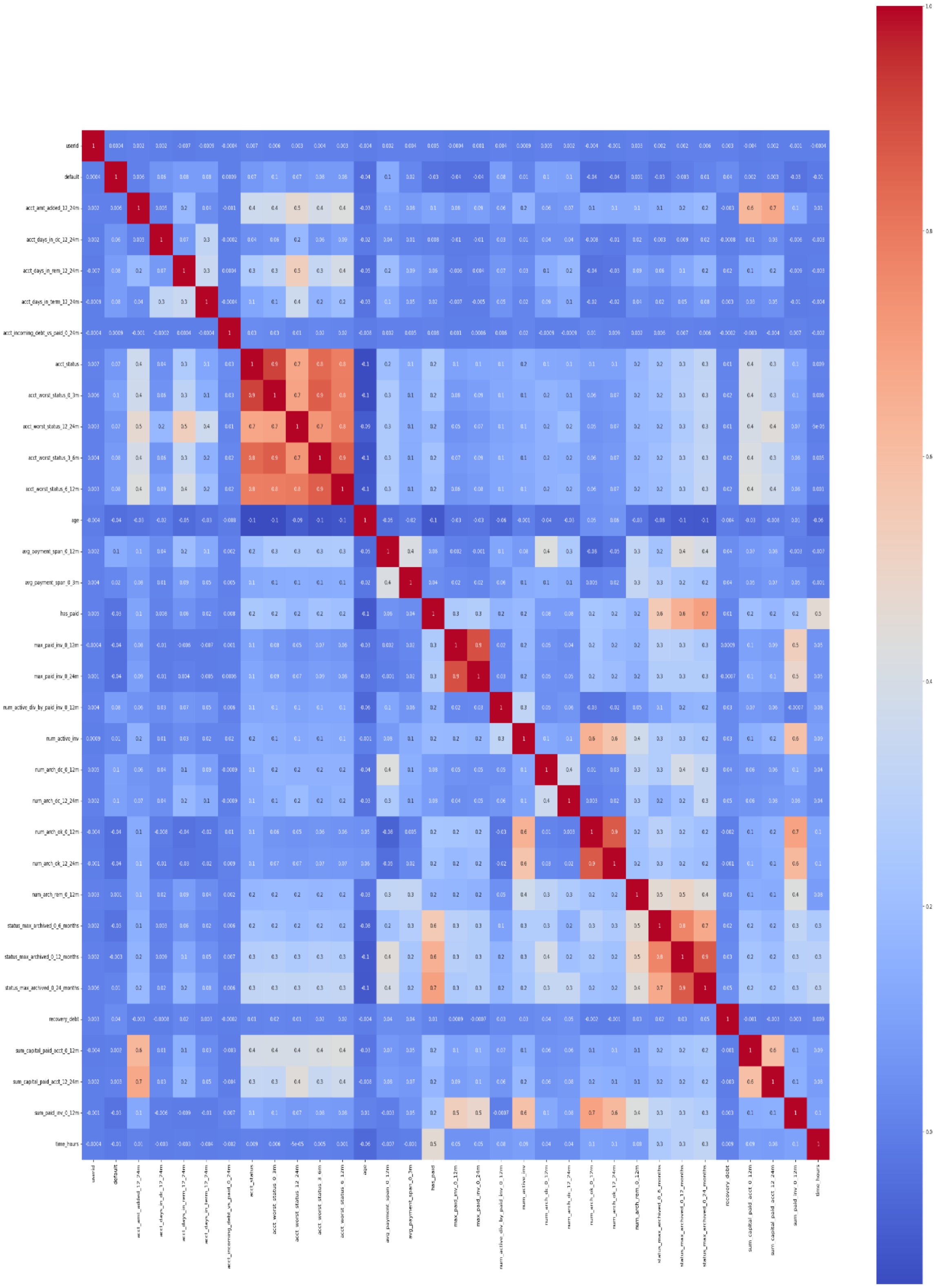
- We have found that mostly the person is non – defaulting and maximum number of the customer who hold the credit card is of Middle age.
- Approximately same number of people is defaulting in Young and Middle age and a smaller number of person is defaulting in Senior group.

- We have categorized the age into different groups.

  Such as
  From18-28 --→ Young age
  From 28-42 -→Middle age
  From 42-60-→ Senior
  From 60-100-> old



- **HEATMAP OF THE VARIABLES**



Data Cleaning and Pre-processing - Approach used for identifying and treating missing values and outlier treatment (and why) - Need for variable transformation (if any) - Variables removed or added and why (if any)

- **Removal of unwanted variables (if applicable)**

Yes , We have removed the column named ‘**name_in_email**’ as it is not adding any valuable insight to our EDA(Exploratory data Analysis).

- **Missing Value treatment (if applicable)**

Yes, there are a lot of missing values present in the dataset.

For imputing the missing values, we will first analyse each column and impute according to analyse of the data that it should be imputed by mean value or median value or 0 or KNN neighbour.

Firstly, the target column ‘**default’** we have taken. We found that the null that is present in the column ‘default’ is less than 10% so we have dropped that null from the column ‘default’ and there is now 89976. It is a good practice to remove the null values less than 10% from target column as if we impute it with some value the data is being mishandled and will not provide good insights or you can say pure insights.

It is always a good practice to have 90% of pure data rather than making it impure by adding 10% of impure data and making prediction of the target column.

We have done info to check the values present in the default column after dropping nulls.

```
Index: 89976 entries, 0 to 89975
Data columns (total 36 columns):
 #   Column                            Non-Null Count  Dtype  
---  ------                            --------------  -----  
 0   userid                            89976 non-null  int64  
 1   default                           89976 non-null  float64
 2   acct_amt_added_12_24m             89976 non-null  int64  
 3   acct_days_in_dc_12_24m            89976 non-null  float64
 4   acct_days_in_rem_12_24m           89976 non-null  float64
 5   acct_days_in_term_12_24m          89976 non-null  float64
 6   acct_incoming_debt_vs_paid_0_24m  89976 non-null  float64
 7   acct_status                       89976 non-null  float64
 8   acct_worst_status_0_3m            89976 non-null  float64
 9   acct_worst_status_12_24m          89976 non-null  float64
 10  acct_worst_status_3_6m            89976 non-null  float64
 11  acct_worst_status_6_12m           89976 non-null  float64
 12  age                               89976 non-null  int64  
 13  avg_payment_span_0_12m            89976 non-null  float64
 14  avg_payment_span_0_3m             89976 non-null  float64
 15  merchant_category                 89976 non-null  object 
 16  merchant_group                    89976 non-null  object 
 17  has_paid                          89976 non-null  float64
 18  max_paid_inv_0_12m                89976 non-null  float64
 19  max_paid_inv_0_24m                89976 non-null  float64
 20  num_active_div_by_paid_inv_0_12m  89976 non-null  float64
 21  num_active_inv                    89976 non-null  float64
 22  num_arch_dc_0_12m                 89976 non-null  float64
 23  num_arch_dc_12_24m                89976 non-null  float64
 24  num_arch_ok_0_12m                 89976 non-null  float64
 25  num_arch_ok_12_24m                89976 non-null  float64
 26  num_arch_rem_0_12m                89976 non-null  float64
 27  status_max_archived_0_6_months    89976 non-null  float64
 28  status_max_archived_0_12_months   89976 non-null  float64
 29  status_max_archived_0_24_months   89976 non-null  float64
 30  recovery_debt                     89976 non-null  float64
 31  sum_capital_paid_acct_0_12m       89976 non-null  float64
```

- The column named ‘acct_amt_added_12_24m ‘ don’t have any null so we are proceeding further.
- Column named ‘acct_days_in_dc_12_24m’ till column “acct_worst_status_6_12m” we have imputed it with 0 as we have seen that the all the values that is NA in these columns don’t been defaulted so the status of these columns will be NULL as they are not defaulting so we have imputed it with 0.

```
---  ------                           --------------  -----
 0   userid                           89976 non-null  int64
 1   default                          89976 non-null  float64
 2   acct_amt_added_12_24m            89976 non-null  int64
 3   acct_days_in_dc_12_24m           89976 non-null  float64
 4   acct_days_in_rem_12_24m          89976 non-null  float64
 5   acct_days_in_term_12_24m         89976 non-null  float64
 6   acct_incoming_debt_vs_paid_0_24m 89976 non-null  float64
 7   acct_status                      89976 non-null  float64
 8   acct_worst_status_0_3m           89976 non-null  float64
 9   acct_worst_status_12_24m         89976 non-null  float64
 10  acct_worst_status_3_6m           89976 non-null  float64
 11  acct_worst_status_6_12m          89976 non-null  float64
 12  age                              89976 non-null  int64
 13  avg_payment_span_0_12m           68508 non-null  float64
 14  avg_payment_span_0_3m            45594 non-null  float64
 15  merchant_category                89976 non-null  object
 16  merchant_group                   89967 non-null  object
 17  has_paid                         80032 non-null  float64
 18  max_paid_inv_0_12m               80032 non-null  float64
 19  max_paid_inv_0_24m               80032 non-null  float64
 20  name_in_email                    80032 non-null  object
 21  num_active_div_by_paid_inv_0_12m 63040 non-null  float64
 22  num_active_inv                   80032 non-null  float64
 23  num_arch_dc_0_12m                80032 non-null  float64
 24  num_arch_dc_12_24m               80032 non-null  float64
 25  num_arch_ok_0_12m                80032 non-null  float64
 26  num_arch_ok_12_24m               80032 non-null  float64
 27  num_arch_rem_0_12m               80032 non-null  float64
 28  status_max_archived_0_6_months   80032 non-null  float64
 29  status_max_archived_0_12_months  80032 non-null  float64
 30  status_max_archived_0_24_months  80032 non-null  float64
 31  recovery_debt                    80032 non-null  float64
 32  sum_capital_paid_acct_0_12m      80032 non-null  float64
 33  sum_capital_paid_acct_12_24m     80032 non-null  float64
 34  sum_paid_inv_0_12m               80032 non-null  float64
```

- Age don’t have any missing values so we are proceeding further
- Column named ‘ avg_payment_span_0_12m ‘ and ‘avg_payment_span_0_3m’ is being imputed with median as continuous values it’s a good practice to impute it with median.
- Column named ‘merchant_group’ is being imputed with ‘Food & Beverage’ as all the null are in category for Wine so we have imputed “merchant_group” with ‘Food & Beverage’.
- Column named ‘has_paid’ has null present in it so we have imputed the 0 and after that I have done value count to find out how many are 0 and 1 .

```
has_paid
1.0    77002
0.0    22974
Name: count, dtype: int64
```

- Column named ‘**max_paid_inv_0_12m’** till ‘**num_active_inv**’ is being imputed with median as it is the continuous variable and it is always a best option to impute it with median. After imputing with median we have checked the info and found that all the nulls have been imputed by medians and there is no null present till that column.

```
0   userid                             99976 non-null  int64
1   default                            89976 non-null  float64
2   acct_amt_added_12_24m              99976 non-null  int64
3   acct_days_in_dc_12_24m             99976 non-null  float64
4   acct_days_in_rem_12_24m            99976 non-null  float64
5   acct_days_in_term_12_24m           99976 non-null  float64
6   acct_incoming_debt_vs_paid_0_24m   99976 non-null  float64
7   acct_status                        99976 non-null  float64
8   acct_worst_status_0_3m             99976 non-null  float64
9   acct_worst_status_12_24m           99976 non-null  float64
10  acct_worst_status_3_6m             99976 non-null  float64
11  acct_worst_status_6_12m            99976 non-null  float64
12  age                                99976 non-null  int64
13  avg_payment_span_0_12m             99976 non-null  float64
14  avg_payment_span_0_3m              99976 non-null  float64
15  merchant_category                  99976 non-null  object
16  merchant_group                     99976 non-null  object
17  has_paid                           99976 non-null  float64
18  max_paid_inv_0_12m                 99976 non-null  float64
19  max_paid_inv_0_24m                 99976 non-null  float64
20  num_active_div_by_paid_inv_0_12m   99976 non-null  float64
21  num_active_inv                     99976 non-null  float64
22  num_arch_dc_0_12m                  88942 non-null  float64
23  num_arch_dc_12_24m                 88942 non-null  float64
24  num_arch_ok_0_12m                  88942 non-null  float64
25  num_arch_ok_12_24m                 88942 non-null  float64
26  num_arch_rem_0_12m                 88942 non-null  float64
27  status_max_archived_0_6_months     88942 non-null  float64
28  status_max_archived_0_12_months    88942 non-null  float64
29  status_max_archived_0_24_months    88942 non-null  float64
30  recovery_debt                      88942 non-null  float64
31  sum_capital_paid_acct_0_12m        88942 non-null  float64
32  sum_capital_paid_acct_12_24m       88942 non-null  float64
33  sum_paid_inv_0_12m                 88942 non-null  float64
34  time_hours                         88942 non-null  float64
35  age_range                          96272 non-null  category
```

- After that ‘**num_arch_dc_0_12m** ’ till the ‘**time_hours’** have null values so we have imputed it with 0 values . After imputing we have done info to find out if any missing values is still left. So there is no null values present in the dataset.

```
0   userid                              99976 non-null  int64
1   default                             89976 non-null  float64
2   acct_amt_added_12_24m               99976 non-null  int64
3   acct_days_in_dc_12_24m              99976 non-null  float64
4   acct_days_in_rem_12_24m             99976 non-null  float64
5   acct_days_in_term_12_24m            99976 non-null  float64
6   acct_incoming_debt_vs_paid_0_24m    99976 non-null  float64
7   acct_status                         99976 non-null  float64
8   acct_worst_status_0_3m              99976 non-null  float64
9   acct_worst_status_12_24m            99976 non-null  float64
10  acct_worst_status_3_6m              99976 non-null  float64
11  acct_worst_status_6_12m             99976 non-null  float64
12  age                                 99976 non-null  int64
13  avg_payment_span_0_12m              99976 non-null  float64
14  avg_payment_span_0_3m               99976 non-null  float64
15  has_paid                            99976 non-null  float64
16  max_paid_inv_0_12m                  99976 non-null  float64
17  max_paid_inv_0_24m                  99976 non-null  float64
18  num_active_div_by_paid_inv_0_12m    99976 non-null  float64
19  num_active_inv                      99976 non-null  float64
20  num_arch_dc_0_12m                   99976 non-null  float64
21  num_arch_dc_12_24m                  99976 non-null  float64
22  num_arch_ok_0_12m                   99976 non-null  float64
23  num_arch_ok_12_24m                  99976 non-null  float64
24  num_arch_rem_0_12m                  99976 non-null  float64
25  status_max_archived_0_6_months      99976 non-null  float64
26  status_max_archived_0_12_months     99976 non-null  float64
27  status_max_archived_0_24_months     99976 non-null  float64
28  recovery_debt                       99976 non-null  float64
29  sum_capital_paid_acct_0_12m         99976 non-null  float64
30  sum_capital_paid_acct_12_24m        99976 non-null  float64
31  sum_paid_inv_0_12m                  99976 non-null  float64
32  time_hours                          99976 non-null  float64
dtypes: float64(30), int64(3)
```

c) Outlier treatment (if required)

Linear Regression , Distance based algorithm is outlier treatment sensitive . So if we have to proceed with these algorithm outlier treatment is not needed and if we do Tree based algorithm , ensemble , KNN ,decision, Naïve Bayes based algorithm -outliers treatment needed . So it totally depend on machine learning technique we are going to use.

d) Variable transformation (if applicable)

There is no variable transformation as it is not needed in the given dataset.

e) Addition of new variables (if required)

Yes, We have created a new column named "age_range" as to group the ages into different groups.

- We have categorized the age into different groups.

  Such as

  From18-28 --→ Young age
  From 28-42 -→Middle age
  From 42-60-→ Senior
  From 60-100-> old

Model building - Clear on why was a particular model(s) chosen. - Effort to improve model performance.

Models	Train AUC-ROC score	Test AUC -ROC score	Recall Train	Recall Test	Accuracy Train	Accuracy Test	F1 Score Train	F1 Score Test	Model RanK
Logistic Tuned (Grid search CV)	0.512	0.522	0.04	0.02	0.99	0.99	0.07	0.04	4
Logistic regression (grid search CV) with threshold cutoff	0.77	0.77	0.86	0.77	0.71	0.77	0.08	0.09	3
LDA	0.851	0.852	0.2	0.2	0.97	0.98	0.18	0.19	5
Decision Tree	0.97	0.58	0.95	0.19	1	0.98	0.97	0.98	8
Decision Tree Tuned (SMOTE)	0.813	0.816	0.78	0.75	0.76	0.74	0.76	0.07	2
KNN	0.497	0.518	0.02	0.01	0.99	0.99	0.04	0.02	7
Boosting	0.5	0.5	0.03	0.02	0.99	0.99	0.02	0.01	6
Boosting Tuned	0.8	0.49	0.94	0.32	0.8	0.66	0.83	0.03	1

Note:

- Boosting tuned is our best model as we can see that the recall train and test recall is good for both and we need to have find that the person is default and model also stating that the person is default.
- Second best model is Decision tree Tuned (SMOTE) as the AUC-ROC score along with recall is good for this model.

- Third best model is Logistic Regression Grid search CC with threshold cutoff as the AUC-roc score recall is though less than the Decision tree tuned model but still it is 0.77 for AUC-ROC score and recall is also 0.86 for train and 0.77 for test dataset.
- Fourth best model of our is Logistic Regression Grid search CV as the train and test AUC – ROC is 0.512 and 0.522 respectively.
- Afterwards followed by LDA ,Boosting , KNN, Decision Tree.
- Logistic regression recall is very poor so it is definitely not a good model.
- We have taken priority as we have given priority for recall for train and test afterwards Train AUC -ROC score , Test AUC-ROC curve , Accuracy for train and test and lastly on F1 score.
- The main area to focus is True positive as we need to have find that the person is default and model also stating that the person is default.
- False Negative as it states that actually person is default, model said no default as it is a disadvantage, and it should be less to have the model to be good.
- Recall – As in recall both true positive and false negative is there, and both is an important factor so we need to consider recall also.
- F1 score- As in F1 score we take it in consideration when we have the problem to be imbalanced and both FN and TP are important and both also there in F1 score formula, so we need to consider the same also.

Model validation - How was the model validated ? Just accuracy, or anything else too ?

The dataset shared named “Probability of default dataset” is a Classification Problem dataset.

We have two type of machine learning problem either it is “Classification” or “Regression”.

This is classification problem in which the Predictive variable is “default” column.

In classification problem the Predictive variable is of categorical (Yes/ No) or (True/ False) and in Regression problem the Predictive variable is of continuous in nature (Number).

In Classification problem there is the below machine learning techniques that can be performed according to the dataset.

- Logistic Regression
- Linear Discriminant Analysis (LDA)
- Decision Tree
- Random Forest
- Ensemble Technique ---→ Bagging, Boosting
- Navie’s Bayes

Performance Measure for Classification are:

- AUC
- ROC
- Confusion matrix ---- Recall, Accuracy , F1 score , Mis-classification error , Precision.

Range of AUC/ROC should be closer to 1 for model tuning we need to increase AUC/ROC and inclined towards 1 .

Confusion Matrix consist of 4 parameters:

- True Negative(TN): When actually the person has no default , model also said no default.
- False Negative(FN): Actually person is default, model said no default.
- False Positive(FP): When actually person has no default , model said default.
- True Positive(TP): The person is default and model also stating that the person is default.

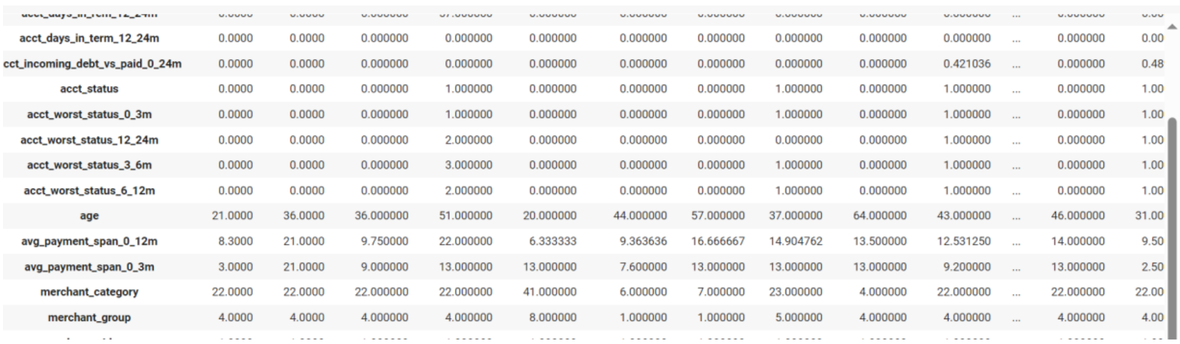
In the whole description and after understanding the dataset we learn the below points:

- The main area to focus is True positive as we need to have find that the person is default and model also stating that the person is default.
- False Negative as it states that actually person is default, model said no default as it is a disadvantage, and it should be less to have the model to be good.
- Recall – As in recall both true positive and false negative is there, and both is an important factor so we need to consider recall also.
- F1 score- As in F1 score we take it in consideration when we have the problem to be imbalanced and both FN and TP are important and both also there in F1 score formula, so we need to consider the same also.

LABEL ENCODING:

- We have seen that the dataset have the columns named "merchant_group" and "merchant_category" that is categorical columns so before proceeding to applying machine learning techniques.
- So we have done label encoding for the columns named "merchant_group" and "merchant_category" .

Dataset after Label encoding



acct_days_in_term_12_24m	0.0000	0.0000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...	0.000000	0.00
cct_incoming_debt_vs_paid_0_24m	0.0000	0.0000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.421036	...	0.000000	0.48
acct_status	0.0000	0.0000	0.000000	1.000000	0.000000	0.000000	0.000000	1.000000	0.000000	1.000000	...	0.000000	1.00
acct_worst_status_0_3m	0.0000	0.0000	0.000000	1.000000	0.000000	0.000000	0.000000	1.000000	0.000000	1.000000	...	0.000000	1.00
acct_worst_status_12_24m	0.0000	0.0000	0.000000	2.000000	0.000000	0.000000	0.000000	0.000000	0.000000	1.000000	...	0.000000	1.00
acct_worst_status_3_6m	0.0000	0.0000	0.000000	3.000000	0.000000	0.000000	0.000000	1.000000	0.000000	1.000000	...	0.000000	1.00
acct_worst_status_6_12m	0.0000	0.0000	0.000000	2.000000	0.000000	0.000000	0.000000	1.000000	0.000000	1.000000	...	0.000000	1.00
age	21.0000	36.0000	36.000000	51.000000	20.000000	44.000000	57.000000	37.000000	64.000000	43.000000	...	46.000000	31.00
avg_payment_span_0_12m	8.3000	21.0000	9.750000	22.000000	6.333333	9.363636	16.666667	14.904762	13.500000	12.531250	...	14.000000	9.50
avg_payment_span_0_3m	3.0000	21.0000	9.000000	13.000000	13.000000	7.600000	13.000000	13.000000	13.000000	9.200000	...	13.000000	2.50
merchant_category	22.0000	22.0000	22.000000	22.000000	41.000000	6.000000	7.000000	23.000000	4.000000	22.000000	...	22.000000	22.00
merchant_group	4.0000	4.0000	4.000000	4.000000	8.000000	1.000000	1.000000	5.000000	4.000000	4.000000	...	4.000000	4.00

Dropping of columns

- We have dropped columns named “age_range” , “userid” as it doesn’t been needed in proceeding machine learning as userid is being unique number so dropping of these columns is a better option.
- Scaling of dataset using Standscaler is also done so that all the values are scaled and will not influence.

LOGISTIC REGRESSION

- Logistic regression is a statistical method that is used for building machine learning models where the dependent variable is dichotomous: i.e. binary. Logistic regression is used to describe data and the relationship between one dependent variable and one or more independent variables
- The dataset after the above steps has been splitted into train and test dataset with 70% of train dataset and 30 % of test dataset.
- We have taken random state as 100.
- After splitting we got the below

```
Number of rows and columns of the training set for the independent variables: (62983, 33)
Number of rows and columns of the training set for the dependent variable: (62983,)
Number of rows and columns of the test set for the independent variables: (26993, 33)
Number of rows and columns of the test set for the dependent variable: (26993,)
```

- We have fitted the dataset into Logistic Regression model and after that we done the classification report for both train and test.

- Classification of TRAIN dataset (Logistic Regression)

	precision	recall	f1-score	support
0.0	0.99	1.00	0.99	62081
1.0	0.46	0.04	0.07	902
accuracy			0.99	62983
macro avg	0.73	0.52	0.53	62983
weighted avg	0.98	0.99	0.98	62983

We get after doing Logistic Regression for default 1 for test dataset:

- Recall – 0.04
- F1 score – 0.07
- Accuracy – 0.99
- Precision: 0.46

- Classification of TEST dataset (Logistic Regression)

	precision	recall	f1-score	support
0.0	0.99	1.00	0.99	26607
1.0	0.39	0.02	0.04	386
accuracy			0.99	26993
macro avg	0.69	0.51	0.52	26993
weighted avg	0.98	0.99	0.98	26993

We get after doing Logistic Regression for default 1 for test dataset:

- Recall – 0.02
- F1 score – 0.04
- Accuracy – 0.99
- Precision: 0.39

Checking and shifting threshold value of TEST AND TRAIN DATA

We saw that the recall for both the train and test dataset is very low so we have calculated optimum_threshold and getting the Predicted Classes and Probs.

- Classification of Train dataset (Logistic Regression) after taking optimum threshold as shown below

optimum_threshold for train is 0.013

	precision	recall	f1-score	support
0.0	1.00	0.71	0.83	62081
1.0	0.04	0.86	0.08	902
accuracy			0.71	62983
macro avg	0.52	0.78	0.45	62983
weighted avg	0.98	0.71	0.82	62983

- Recall – 0.86
- F1 score – 0.08
- Accuracy – 0.71
- Precision: 0.04

- Classification of Test dataset (Logistic Regression) after taking optimum threshold as shown below

optimum_threshold for test is 0.016

	precision	recall	f1-score	support
0.0	1.00	0.77	0.87	26607
1.0	0.05	0.77	0.09	386
accuracy			0.77	26993
macro avg	0.52	0.77	0.48	26993
weighted avg	0.98	0.77	0.86	26993

- Recall – 0.77
- F1 score – 0.09
- Accuracy – 0.77
- Precision: 0.05

After optimize threshold value we found that this has increased the recall for both test and train dataset as compared to earlier .

LOGISTIC REGRESSION TUNED:

- Applying GRID SEARCH CV on Logistic Regression
- Grid search best parameter and best estimator we get ,

```
{'penalty': 'l2', 'solver': 'lbfgs', 'tol': 0.001}

LogisticRegression(tol=0.001)
```

- Train dataset classification report after applying grid search

	precision	recall	f1-score	support
0.0	0.99	1.00	0.99	62081
1.0	0.46	0.04	0.07	902
accuracy			0.99	62983
macro avg	0.73	0.52	0.53	62983
weighted avg	0.98	0.99	0.98	62983

- Recall – 0.04
- F1 score – 0.07
- Accuracy – 0.99
- Precision: 0.46

- Test dataset classification report after applying grid search

	precision	recall	f1-score	support
0.0	0.99	1.00	0.99	26607
1.0	0.39	0.02	0.04	386
accuracy			0.99	26993
macro avg	0.69	0.51	0.52	26993
weighted avg	0.98	0.99	0.98	26993

- Recall – 0.02
- F1 score – 0.04
- Accuracy – 0.99
- Precision: 0.39

We saw that after applying Grid search CV also the recall value and F1 score are very less and that doesn't make them a good model so we have taken a decision to change the optimum threshold and then check the parameters to make our model perform better.

- Classification of Train dataset (Logistic Regression tuned(Grid search CV)) after taking optimum threshold as shown below

optimum_threshold for train is 0.013

	precision	recall	f1-score	support
0.0	1.00	0.71	0.83	62081
1.0	0.04	0.86	0.08	902
accuracy			0.71	62983
macro avg	0.52	0.78	0.45	62983
weighted avg	0.98	0.71	0.82	62983

- Recall – 0.86
- F1 score – 0.08
- Accuracy – 0.71
- Precision: 0.04

- Classification of Test dataset (Logistic Regression tuned(Grid search CV)) after taking optimum threshold as shown below

optimum_threshold for train is 0.016

	precision	recall	f1-score	support
0.0	1.00	0.77	0.87	26607
1.0	0.05	0.77	0.09	386
accuracy			0.77	26993
macro avg	0.52	0.77	0.48	26993
weighted avg	0.98	0.77	0.86	26993

- Recall – 0.77
- F1 score – 0.09
- Accuracy – 0.77
- Precision: 0.05

LINEAR DISCRIMINANT ANALYSIS

- Linear discriminant analysis (LDA) is an approach used in supervised machine learning to solve multi-class classification problems. LDA separates multiple classes with multiple features through data dimensionality reduction. This technique is important in data science as it helps optimize machine learning models.

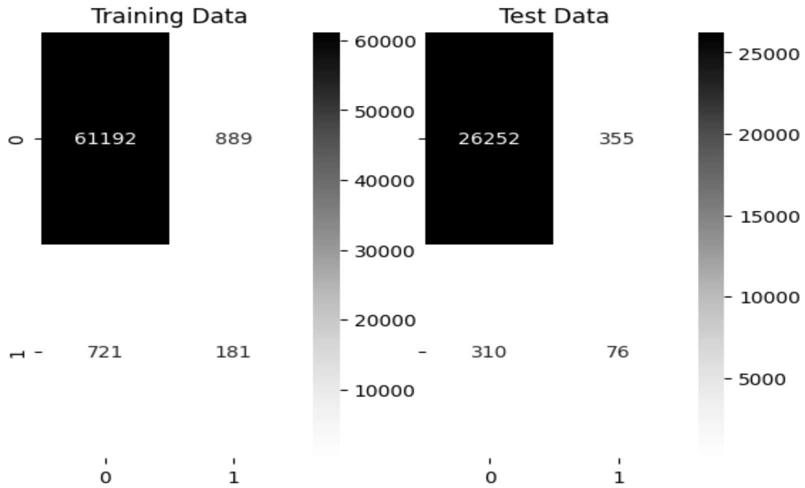
- We have fitted the dataset into LDA model and after that we done the classification report for both train and test.

```
Classification Report of the training data:

              precision    recall  f1-score   support

         0.0       0.99      0.99      0.99     62081
         1.0       0.17      0.20      0.18       902

    accuracy                           0.97     62983
   macro avg       0.58      0.59      0.59     62983
weighted avg       0.98      0.97      0.98     62983


Classification Report of the test data:

              precision    recall  f1-score   support

         0.0       0.99      0.99      0.99     26607
         1.0       0.18      0.20      0.19       386

    accuracy                           0.98     26993
   macro avg       0.58      0.59      0.59     26993
weighted avg       0.98      0.98      0.98     26993
```

- Train dataset parameters of LDA
    - Recall – 0.20
    - F1 score – 0.18
    - Accuracy – 0.97
    - Precision: 0.17

- TEST dataset parameters of LDA
    - Recall – 0.20
    - F1 score – 0.19
    - Accuracy – 0.98
    - Precision: 0.18

- Confusion matrix for train and test dataset of LDA model.



For {Customer who did not default (Label 0)}:

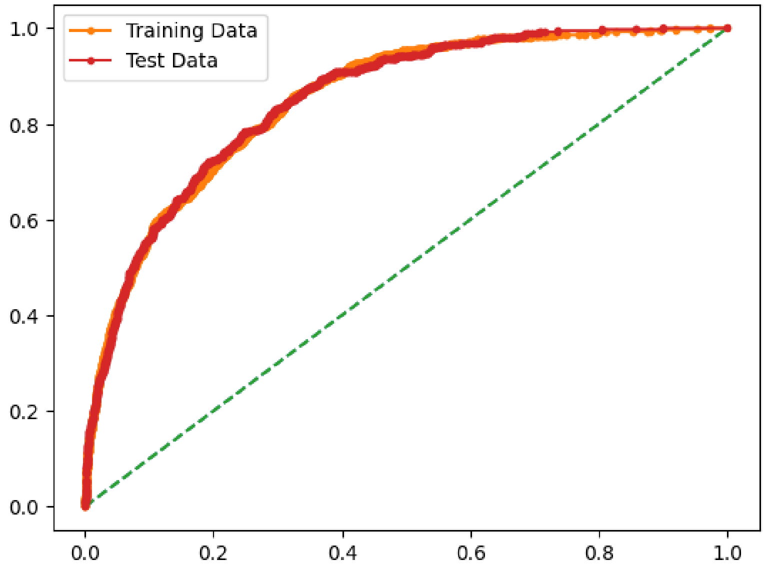
- Precision (99%) – 99% of Customers who did not default are correctly predicted, out of all Customers who did not default that are predicted .
- Recall (99%) – Out of all the Customers who actually did not default
- 99% of Customers who did not default have been predicted correctly.

For {Customer who did default (Label 1)}:

- Precision (18%) – 18% of Customers who did default are correctly predicted ,out of all Customers who did default that are predicted .
- Recall (20%) – Out of all the Customers who actually did default, 20% of Customers who did default have been predicted correctly.
- We have plotted AUC-ROC curve and then we get the below for training and test data:

```
AUC for the Training Data: 0.851
AUC for the Test Data: 0.852
```

Range of AUC/ROC should be closer to 1 for model tuning we need to increase AUC/ROC and inclined towards 1 .



AUC Value closer to 1 tells that there is good seperatibility between the predicted classes and thus the model is good for prediction

ROC Curve visually represents the above concept where the plot should be as far as possible from the diagnol.

Coefficient for Linear Discriminant Analysis we get,

```
array([[-0.06809739,  0.39474876,  0.26628739,  0.32313385, -0.00775995,
        -0.35275467,  1.32754767, -0.58380266, -0.27047406,  0.23411469,
        -0.25600232,  1.12057078, -0.28857387,  0.10196231, -0.03389687,
        -0.07477322,  0.06159821, -0.22762454,  0.64661046,  0.03994865,
         0.4229396 ,  0.54957393,  0.16664404, -0.17248501, -0.21098405,
        -0.00328953, -0.87849342, -0.01045442,  0.2978112 , -0.07663997,
        -0.01240708,  0.01491969,  0.10233758]])
```

Columns name:

```
Index(['acct_amt_added_12_24m', 'acct_days_in_dc_12_24m',
       'acct_days_in_rem_12_24m', 'acct_days_in_term_12_24m',
       'acct_incoming_debt_vs_paid_0_24m', 'acct_status',
       'acct_worst_status_0_3m', 'acct_worst_status_12_24m',
       'acct_worst_status_3_6m', 'acct_worst_status_6_12m', 'age',
       'avg_payment_span_0_12m', 'avg_payment_span_0_3m', 'merchant_category',
       'merchant_group', 'has_paid', 'max_paid_inv_0_12m',
       'max_paid_inv_0_24m', 'num_active_div_by_paid_inv_0_12m',
       'num_active_inv', 'num_arch_dc_0_12m', 'num_arch_dc_12_24m',
       'num_arch_ok_0_12m', 'num_arch_ok_12_24m', 'num_arch_rem_0_12m',
       'status_max_archived_0_6_months', 'status_max_archived_0_12_months',
       'status_max_archived_0_24_months', 'recovery_debt',
       'sum_capital_paid_acct_0_12m', 'sum_capital_paid_acct_12_24m',
       'sum_paid_inv_0_12m', 'time_hours'],
      dtype='object')
```

We have put the coefficient in an array with rounding off it two figures:

```
array([[-0.07,  0.39,  0.27,  0.32, -0.01, -0.35,  1.33, -0.58, -0.27,
         0.23, -0.26,  1.12, -0.29,  0.1 , -0.03, -0.07,  0.06, -0.23,
         0.65,  0.04,  0.42,  0.55,  0.17, -0.17, -0.21, -0.  , -0.88,
        -0.01,  0.3 , -0.08, -0.01,  0.01,  0.1 ]])
```

By the above equation and the coefficients it is clear that predictor 'acct_worst_status_0_3m' and 'avg_payment_span_0_12m' has the largest magnitude thus this helps in classifying the best predictor 'status_max_archived_0_12_months' has the smallest magnitude thus this helps in classifying the least .

DECISION TREE

- A decision tree is a supervised learning algorithm that is used for classification and regression modeling. Regression is a method used for predictive modeling, so these trees are used to either classify data or predict what will come next.

- The dataset has been splited into 70% of train data and 30% of test data.
- We have taken random state as 1.

- Importance of features in the tree building (The importance of a feature is computed as the
- (normalized) total reduction of the criterion brought by that feature. It is also known as the Gini importance)

	Imp
acct_amt_added_12_24m	0.024435
acct_days_in_dc_12_24m	0.005584
acct_days_in_rem_12_24m	0.023017
acct_days_in_term_12_24m	0.009337
acct_incoming_debt_vs_paid_0_24m	0.062770
acct_status	0.005662
acct_worst_status_0_3m	0.023435
acct_worst_status_12_24m	0.010389
acct_worst_status_3_6m	0.017898
acct_worst_status_6_12m	0.014533
age	0.096230
avg_payment_span_0_12m	0.077220
avg_payment_span_0_3m	0.029505
merchant_category	0.063362
merchant_group	0.032051
has_paid	0.005138
max_paid_inv_0_12m	0.031772
max_paid_inv_0_24m	0.051202
num_active_div_by_paid_inv_0_12m	0.022534
num_active_inv	0.017955
num_arch_dc_0_12m	0.011475
num_arch_dc_12_24m	0.019595
num_arch_ok_0_12m	0.012610
num_arch_ok_12_24m	0.016359
num_arch_rem_0_12m	0.009305
status_max_archived_0_6_months	0.004534
status_max_archived_0_12_months	0.003637
status_max_archived_0_24_months	0.003133
recovery_debt	0.004683
sum_capital_paid_acct_0_12m	0.027925
sum_capital_paid_acct_12_24m	0.020493
sum_paid_inv_0_12m	0.031971
time_hours	0.210252

- We have fitted the dataset into Decision Tree model and after that we done the classification report for both train and test.

- Classification Report of train Decision Tree

	precision	recall	f1-score	support
0.0	1.00	1.00	1.00	62076
1.0	1.00	0.95	0.97	907
accuracy			1.00	62983
macro avg	1.00	0.97	0.99	62983
weighted avg	1.00	1.00	1.00	62983

- Train dataset parameters of Decision Tree

- Recall – 0.95
- F1 score – 0.97
- Accuracy – 1
- Precision: 1

- Classification Report of test Decision Tree

	precision	recall	f1-score	support
0.0	0.99	0.99	0.99	26612
1.0	0.17	0.19	0.18	381
accuracy			0.98	26993
macro avg	0.58	0.59	0.59	26993
weighted avg	0.98	0.98	0.98	26993

- Recall – 0.19
- F1 score – 0.18
- Accuracy – 0.98
- Precision: 0.17

DECISION TREE TUNED

- We have applied Grid search CV on decision tree
- We have the best parameters for Grid search CV

```
{'ccp_alpha': 0.01,
 'criterion': 'gini',
 'max_depth': 10,
 'max_features': 'auto',
 'min_samples_leaf': 10}
```

- Classification report for train dataset after applying Grid search CV

	precision	recall	f1-score	support
0.0	0.99	1.00	0.99	62076
1.0	0.21	0.00	0.01	907
accuracy			0.99	62983
macro avg	0.60	0.50	0.50	62983
weighted avg	0.97	0.99	0.98	62983

- Recall – 0.0
- F1 score – 0.01
- Accuracy – 0.99
- Precision: 0.21

- Classification report for test dataset after applying Grid search CV

	precision	recall	f1-score	support
0.0	0.99	1.00	0.99	26612
1.0	0.43	0.01	0.02	381
accuracy			0.99	26993
macro avg	0.71	0.50	0.50	26993
weighted avg	0.98	0.99	0.98	26993

- Recall – 0.01
- F1 score – 0.02
- Accuracy – 0.99
- Precision: 0.43

Synthetic Minority Oversampling Technique (SMOTE) is a statistical technique for increasing the number of cases in the dataset in a balanced way. The component works by generating new instances from existing minority cases that we supply as input.

Synthetic Minority Oversampling Technique (SMOTE)

- Before Over sampling the counts of label 1 and 0 is 907 and 62076 that is unequal and a large difference is there between them so we will use SMOTE technique to move forward as SMOTE will help to generate new instance from existing minority cases that we supply as input.

```
Before OverSampling, counts of label '1': 907
Before OverSampling, counts of label '0': 62076
```

- After Oversampling we get both the counts of label 1 and 0 as 62076.

```
After OverSampling, counts of label '1': 62076
After OverSampling, counts of label '0': 62076
```

- We have fitted these data samples which has same equal counts of test and train dataset into the model
- Classification report of train dataset

	precision	recall	f1-score	support
0.0	0.77	0.74	0.75	62076
1.0	0.75	0.78	0.76	62076
accuracy			0.76	124152
macro avg	0.76	0.76	0.76	124152
weighted avg	0.76	0.76	0.76	124152

- Recall – 0.78
- F1 score – 0.76
- Accuracy – 0.76
- Precision: 0.75

- Classification report of test dataset.

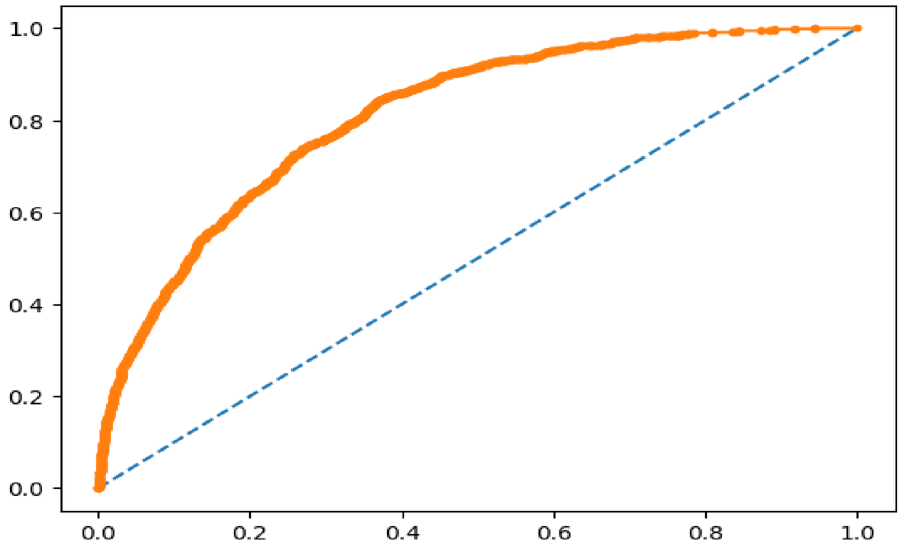
	precision	recall	f1-score	support
0.0	1.00	0.74	0.85	26612
1.0	0.04	0.75	0.07	381
accuracy			0.74	26993
macro avg	0.52	0.74	0.46	26993
weighted avg	0.98	0.74	0.84	26993

- Recall – 0.75
- F1 score – 0.07
- Accuracy – 0.74
- Precision: 0.04

- **AUC -ROC Curve for train**

AUC: 0.813

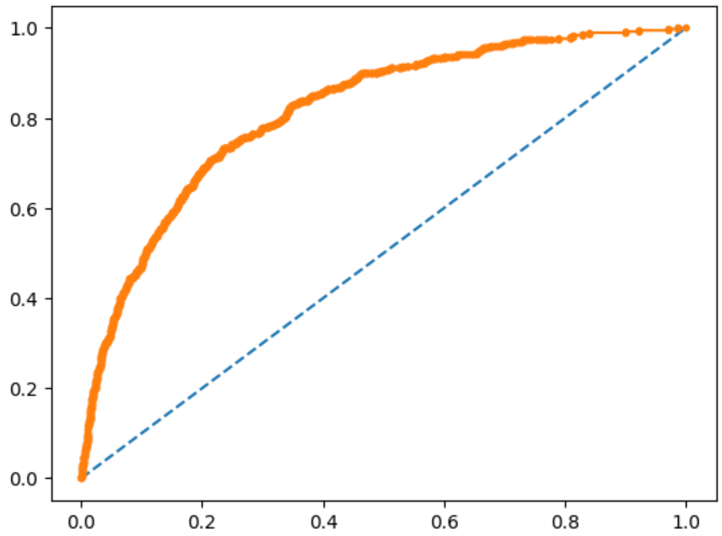
AUC VALUE we get for train is 0.813



- **AUC-ROC curve for test**

AUC VALUE we get for test is 0.816

AUC: 0.816



KNN (K-nearest Neighbours)

The k-nearest neighbour (KNN) algorithm is a non-parametric, supervised learning classifier, which uses proximity to make classifications or predictions about the grouping of an individual data point. It is one of the popular and simplest classification and regression classifiers used in machine learning today.

(K-NN) algorithm is a versatile and widely used machine learning algorithm that is primarily used for its simplicity and ease of implementation. It does not require any assumptions about the underlying data distribution. It can also handle both numerical and categorical data, making it a flexible choice for various types of datasets in classification and regression tasks. It is a non-parametric method that makes predictions based on the similarity of data points in a given dataset. K-NN is less sensitive to outliers compared to other algorithms.

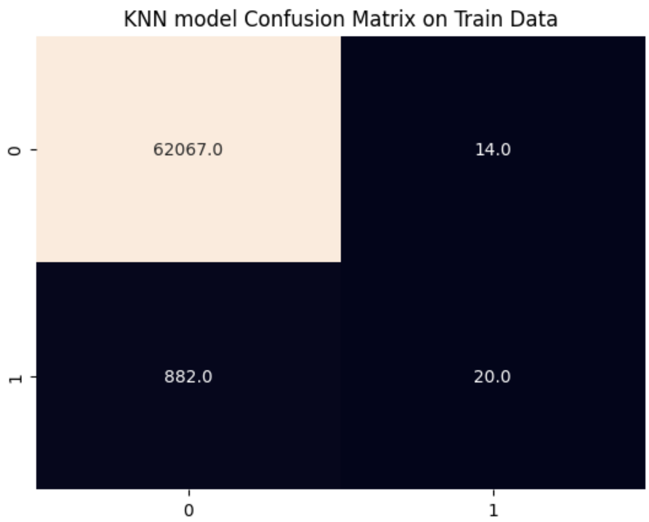
The K-NN algorithm works by finding the K nearest neighbors to a given data point based on a distance metric, such as Euclidean distance. The class or value of the data point is then determined by the majority vote or average of the K neighbors. This approach allows the algorithm to adapt to different patterns and make predictions based on the local structure of the data.

- We have fitted the dataset into KNN model
- Classification report of train dataset

KNN model Classification Report on Train Data

	precision	recall	f1-score	support
0.0	0.99	1.00	0.99	62081
1.0	0.59	0.02	0.04	902
accuracy			0.99	62983
macro avg	0.79	0.51	0.52	62983
weighted avg	0.98	0.99	0.98	62983

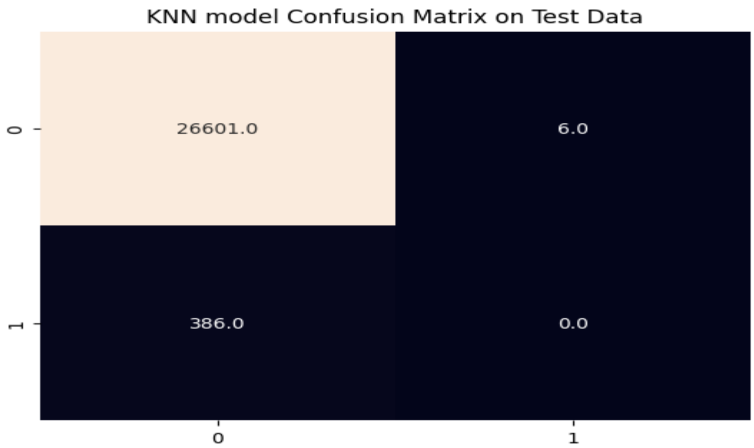
- Recall – 0.02
- F1 score – 0.04
- Accuracy – 0.99
- Precision: 0.59



- Classification report of test dataset

```
KNN model Classification Report on Test Data
              precision    recall  f1-score   support

         0.0       0.99      1.00      0.99     26607
         1.0       0.00      0.00      0.00       386

    accuracy                           0.99     26993
   macro avg       0.49      0.50      0.50     26993
weighted avg       0.97      0.99      0.98     26993
```

- Recall – 0.0
- F1 score – 0.0
- Accuracy – 0.99



- AUC-ROC score for training data:

```
AUC for the Training Data: 0.497
```

- AUC-ROC score for test data:

```
AUC for the test Data: 0.518
```

BAGGING

- The model has been fitted by Bagging classifier.
- Train model score , Confusion matrix , Classification Report of train model.

```
0.9994442944921645
[[62081     0]
 [   35   867]]
              precision    recall  f1-score   support

         0.0       1.00      1.00      1.00     62081
         1.0       1.00      0.96      0.98       902

    accuracy                           1.00     62983
   macro avg       1.00      0.98      0.99     62983
weighted avg       1.00      1.00      1.00     62983
```

- Train model score – 0.99
- Recall – 0.96
- F1 score – 0.98
- Accuracy – 1

- Test model score , Confusion matrix , Classification Report of test model.

```
0.9856629496536139
[[26606     1]
 [  386     0]]
              precision    recall  f1-score   support

         0.0       0.99      1.00      0.99     26607
         1.0       0.00      0.00      0.00       386

    accuracy                           0.99     26993
   macro avg       0.49      0.50      0.50     26993
weighted avg       0.97      0.99      0.98     26993
```

- Test model score 0.98
- Recall – 1
- F1 score – 0
- Accuracy – 0.99

BOOSTING

- The data has been fitted by Boosting classifier.
- Train model score, Confusion matrix , Classification Report of train model

```
Adaptive Boosting Model Classification Report on Train Data
              precision    recall  f1-score   support

         0.0       0.99      1.00      0.99     62081
         1.0       0.00      0.00      0.00       902

    accuracy                           0.99     62983
   macro avg       0.49      0.50      0.50     62983
weighted avg       0.97      0.99      0.98     62983

Adaptive Boosting Model CROC AUC Score on Train Data
 0.5
```

- AUC-ROC score 0.5
- Recall – 0.03
- F1 score – 0.04
- Accuracy – 0.99

- Train model score, Confusion matrix , Classification Report of train model

Adaptive Boosting Model Classification Report on Test Data

	precision	recall	f1-score	support
0.0	0.99	1.00	0.99	26607
1.0	0.00	0.00	0.00	386
accuracy			0.99	26993
macro avg	0.49	0.50	0.50	26993
weighted avg	0.97	0.99	0.98	26993

Adaptive Boosting Model ROC AUC Score on Test Data
0.5

- AUC-ROC score 0.5
- Recall – 0.02
- F1 score – 0.02
- Accuracy – 0.99

BOOSTING TUNED

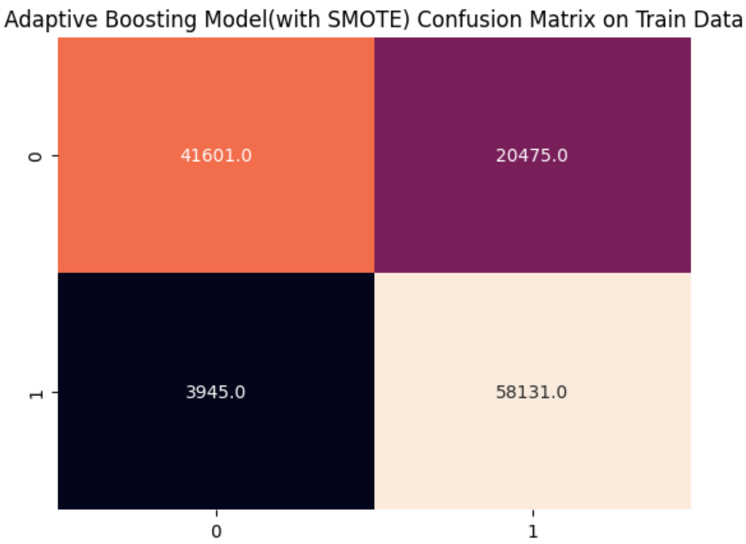
- We have to tuned the boosting to have the better results using SMOTE.
- TRAIN classification report , AUC-ROC SCORE , CONFUSION MATRX

Adaptive Boosting Model (with SMOTE) Classification Report on Train Data

	precision	recall	f1-score	support
0.0	0.91	0.67	0.77	62076
1.0	0.74	0.94	0.83	62076
accuracy			0.80	124152
macro avg	0.83	0.80	0.80	124152
weighted avg	0.83	0.80	0.80	124152

Adaptive Boosting Model (with SMOTE) ROC AUC Score on Train Data
0.8033056253624589

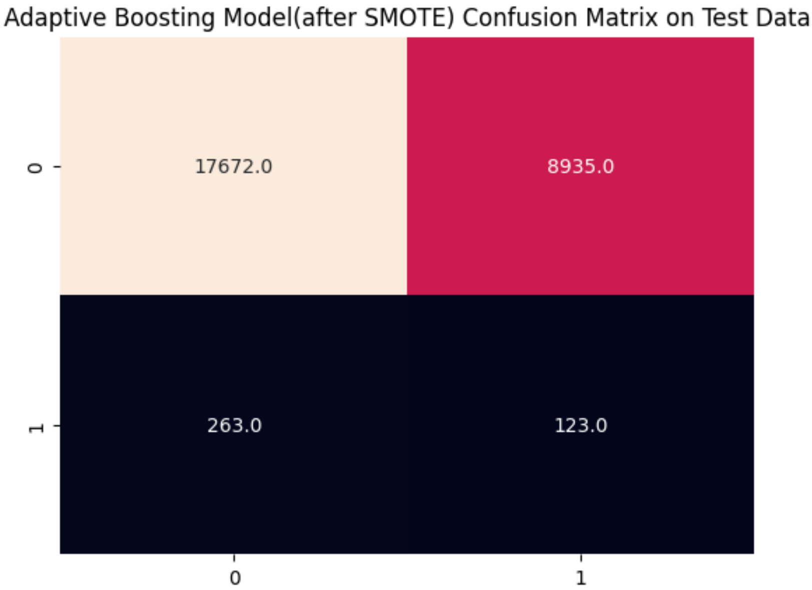
- AUC-ROC score 0.8
- Recall – 0.94
- F1 score – 0.83
- Accuracy – 0.80



- **TEST classification report , AUC-ROC SCORE , CONFUSION MATRX**

```
Adaptive Boosting Model (after SMOTE) Classification Report on Test Data
              precision    recall  f1-score   support

         0.0       0.99      0.66      0.79     26607
         1.0       0.01      0.32      0.03       386

    accuracy                           0.66     26993
   macro avg       0.50      0.49      0.41     26993
weighted avg       0.97      0.66      0.78     26993

Adaptive Boosting Model (after SMOTE) ROC AUC SCore on Test Data
 0.4914194830882286
```

- AUC-ROC score 0.49
- Recall – 0.32
- F1 score – 0.03
- Accuracy – 0.66



INFERENCE

In Classification problem there is the below machine learning techniques that can be performed according to the dataset.

- Logistic Regression
- Linear Discriminant Analysis (LDA)
- Decision Tree
- Random Forest
- Ensemble Technique ---→ Bagging, Boosting
- Navie's Bayes

Performance Measure for Classification are:

- AUC
- ROC

- Confusion matrix ---- Recall, Accuracy , F1 score , Mis-classification error , Precision.

Range of AUC/ROC should be closer to 1 for model tuning we need to increase AUC/ROC and inclined towards 1 .

Confusion Matrix consist of 4 parameters:

- True Negative(TN): When actually the person has no default , model also said no default.
- False Negative(FN): Actually person is default, model said no default.
- False Positive(FP): When actually person has no default , model said default.
- True Positive(TP): The person is default and model also stating that the person is default.

In the whole description and after understanding the dataset we learn the below points:

- The main area to focus is True positive as we need to have find that the person is default and model also stating that the person is default.

Confusion Matrix consist of 4 parameters:

- True Negative (TN): When actually the person has no default , model also said no default.
- False Negative (FN): Actually person is default, model said no default.
- False Positive (FP): When actually person has no default , model said default.
- True Positive (TP): The person is default and model also stating that the person is default.

Final interpretation / recommendation - Very clear and crisp on what recommendations do you want to give to the management / client.

- Most of the person count -approx **88 thousand** who has taken the credit card is not defaulting in paying the bill. However, **approx. 1300** person is still defaulting
- We have found that mostly the person is non – defaulting and maximum number of the customer who hold the credit card is of **Middle age**.
- Approximately same number of people is defaulting in **Young and Middle age** and a **smaller number** of person is defaulting in **Senior group** and this is observed that after paying the bill also portal takes time to react and change the status.
- **Parameters which has highest Impact**: -
- **avg_payment_span_0_12m** has the direct impact on predicting whether the customer is going to default or not. As more the time taken to pay the bill higher is the chances to default
- **acct_incoming_debt_vs_paid_0_24m** as the most the debit to paid ratio higher the due amount to pay which might result in chances to default
- **max_paid_inv_0_24m**: - Higher is the maximum paid invoice means customer is using the credit card very frequently which might result in chances to default
- **sum_capital_paid_acct_0_12m**: - sum of principal balance paid on account in the last one year. Higher is the amount more is the chances to default

RECOMMENDATION:

- This is observed that after paying the bill also portal takes **time to react** and change the status as we can see from the column named 'num_active' that after paying also the person still is in the list of default however there is no pavement due from there end. So

technical challenges is there while updating the status. This need to check further and action should be taken to solve this issue

- Maximum persons is **taking average time while doing payment** is 5 to 25 hrs which is long. So might be there is less offers that is provided to the customer and that’s the reason they are taking more time in doing payments as they visit other cards payment offer and which ever taking best offer, they are proceeding with that.
- We need to **launch new offers to reduce the time** taken by the customer for doing the payment and will refer others also to take this card . This will help us more in terms of having more customers to join