GREAT LAKES
INSTITUTE OF MANAGEMENT. CHENNAI

greatlearning
Learning for Life

PREDICTION OF CREDIT CARD DEFAULT PAYMENTS

Post Graduate Program in Data Science Engineering
Location: Bangalore Batch: PGPDSE-FT NOV, 2021

Submitted by

Ahsan Ali

Naga Bhavani

Roshan Singh Hari

Supriya Raj

Visisht NM

Mentored by

Mr. Jatin

DEFAULT OF CREDIT CARD CLIENTS

INTRODUCTION

The domain chosen for the capstone project is of finance (Banking) sector. A credit card is a financial instrument that allows the card holder to make transactions without paying for it instantly. The credit card issuing company pays for the goods or the services on the card holder’s behalf and issues a credit card statement to the card holder at the end of the billing cycle. It mentions the amount owed, which needs to be repaid on or before the due date.

When you fail to repay the spent amount in time (referred to as the credit card billing cycle), which comes to you as your credit card bill, you become a credit card defaulter. If used responsibly, a Credit Card can help you build a good credit history, also allowing you to get loans at favorable Interest Rates.

DATASET INFORMATION

This dataset contains information on default payments, demographic factors, credit data, history of payment and bill statements of credit card clients in Taiwan from April 2005 to September 2005.

PROBLEM STATEMENT

Preserving the financial health of customers is of great relevance for companies involved with the financial system. But you may wonder how to preserve the financial health of each client? well, the solution for this problem consists of calculating the probability of payment of each client according to some variables and doing some strategies to anticipate customer needs

In Taiwan, in February 2006, debt from credit cards and cash cards reached $268 billion USD. More than half a million people were not able to repay their loans which caused crises. In order to avoid such problems, bank need to know which customer will default or not.

BUSINESS OBJECTIVE

By using more data and analyzing customer default probability, the credit scoring systems are able to predict behavior, thereby helping lenders come to a more conclusive decision based on data. ML allows innovative work on data analysis wherein a bespoke solution is being offered to consumers.

Financial sectors and social lending platforms are actively investing on lending. But financial institutions might face huge capital loss if they approved the loan/Credit Cards without having any prior assessment of default risk of particular person. So, to avoid such risk we can use data science, Machine learning algorithms.

Objective of this project is to predict the probability of default on a given obligation, in this case, credit cards. This will allow the generation of strategies that minimizes the risk of deterioration of the client's financial health. Additionally, to facilitate the development of collection strategies, it is proposed to use Bagging and Boosting algorithms to find homogeneous segments within the population and thus provide differential treatment to each customer.

The bank risk every time it issues credit card/loan. Problem here is that can it reliably predict who is likely to default? If so, the bank may be able to prevent the loss by providing the customer with alternative options (such as forbearance or debt consolidation, etc.).

VARIABLES DESCRIPTION

Variables	Data Type (Previous)	Data Type (Changed	Description
ID	Object	Double	ID of the client
LIMIT_BAL	Object	Object	Amount of given credit in NT dollars (includes individual and family/supplementary credit).
SEX	Object	Object	Gender of the consumer. (1=male, 2=female).
EDUCATION	Object	Object	Education level of the consumer (1= graduate school, 2= university, 3= high school, 4= others, 5= unknown, 6=unknown).
MARRIAGE	Object	Double	Marital Status (1=married, 2=single, 3=others).
AGE	Object	Object	Age in years.
History of past payment. The repayment status in September, 2005*	Object	Object	Repayment status in September 2005, (-2 = No consumption, -1 = Pay Duly, 0= Revolving Credit, 1= Payment delay for one month, 2= payment delay for two months,…, 8 = payment delay for 8 months).
History of past payment. The repayment status in August, 2005*	Object	Object	Repayment status in August, 2005.
History of past payment. The repayment status in July, 2005*	Object	Object	Repayment status in July, 2005.
History of past payment. The repayment status in June, 2005*	Object	Object	Repayment status in June, 2005.
History of past payment. The repayment status in May, 2005*	Object	Object	Repayment status in May, 2005.
History of past payment. The repayment status in April, 2005*	Object	Object	Repayment status in April, 2005.
Amount of bill statement in September, 2005 (NT dollar)	Object	Double	Amount of bill statement in September, 2005 (NT dollar)

Amount of bill statement in August , 2005 (NT dollar)	Object	Double	Amount of bill statement in August, 2005 (NT dollar)
Amount of bill statement in July, 2005 (NT dollar)	Object	Double	Amount of bill statement in July, 2005 (NT dollar)
Amount of bill statement in June, 2005 (NT dollar)	Object	Double	Amount of bill statement in June, 2005 (NT dollar)
Amount of bill statement in May, 2005 (NT dollar)	Object	Double	Amount of bill statement in May, 2005 (NT dollar)
Amount of bill statement in April, 2005 (NT dollar)	Object	Double	Amount of bill statement in April, 2005 (NT dollar)
Amount of previous payment. Paid in September, 2005 (NT dollar)	Object	Double	Amount of previous payment. Paid in September, 2005 (NT dollar)
Amount of previous payment. Paid in August , 2005 (NT dollar)	Object	Double	Amount of previous payment. Paid in August , 2005 (NT dollar)
Amount of previous payment. Paid in July, 2005 (NT dollar)	Object	Double	Amount of previous payment. Paid in July , 2005 (NT dollar)
Amount of previous payment. Paid in June, 2005 (NT dollar)	Object	Double	Amount of previous payment. Paid in June, 2005 (NT dollar)
Amount of previous payment. Paid in May, 2005 (NT dollar)	Object	Double	Amount of previous payment. Paid in May, 2005 (NT dollar)
Amount of previous payment. Paid in April, 2005 (NT dollar)	Object	Double	Amount of previous payment. Paid in April, 2005 (NT dollar)
default payment next month	Object	Int	default payment next month (TARGET VARIABLE).

The dataset which we are using has data of six months (from April 2005 to September 2005). It includes the bill amount, paid amount, status of delay in payment etc.

CRITICAL ASSESSMENT OF TOPIC SURVEY

Find the key area, gaps identified in the topic survey where the project can add value to the customers and business.

This project can add value to banking and finance sector. It will help bank and other finance sector which issue credit/loan/credit amounts without taking much risk of losing money or getting into loss. As we know, when banks go under huge loss, it impacts every common people and entire country economically. Therefore, this project aims to bridge this gap of uncertainty by utilizing a data-driven approach by using past data of credit card customers in conjunction with machine learning to predict whether or not a consumer will default on their credit cards**.**

What key gaps are you trying to solve?

Currently the banks are considering only salary for issuing the credit card. This increases the probability of occurrence of loss to the banks. This project aims to fill this gap, by considering other key variables and applying various machine Learning algorithms to predict, whether the consumer is going to be a defaulter or not.

DATA PRE-PROCESSING

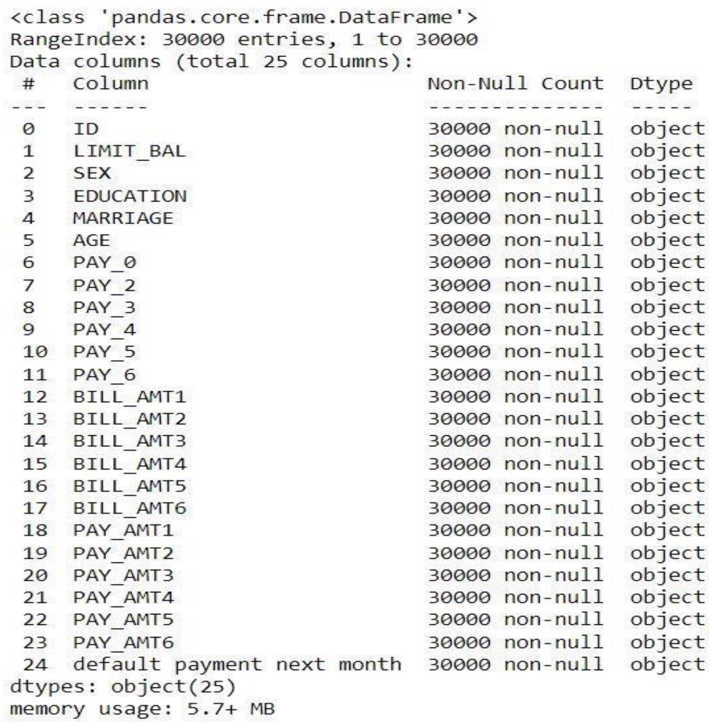
Data pre-processing is a step in the data analysis process that takes raw data and transforms it into a format that can be understood and analyzed by computers during machine learning process.

Raw data in real-world contains more noises, missing values and overall, it is messy. Not only may it contain errors and inconsistencies, but it is often is incomplete, uniform design and doesn’t have a regular form.

Machines like to process nice and tidy information, they read data as 1s and 0s. So, data preprocessing is required task for cleaning the data and making it suitable for machine learning models, it also increasing efficiency and accuracy of machine learning models. The data set contains 30000 rows and 25 columns

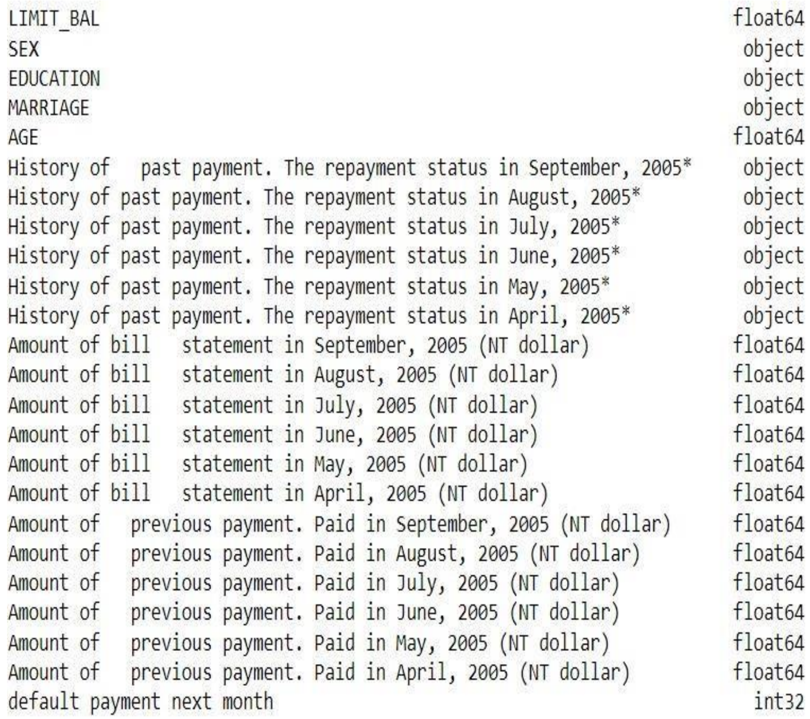
DATA TYPES

Before data type conversion



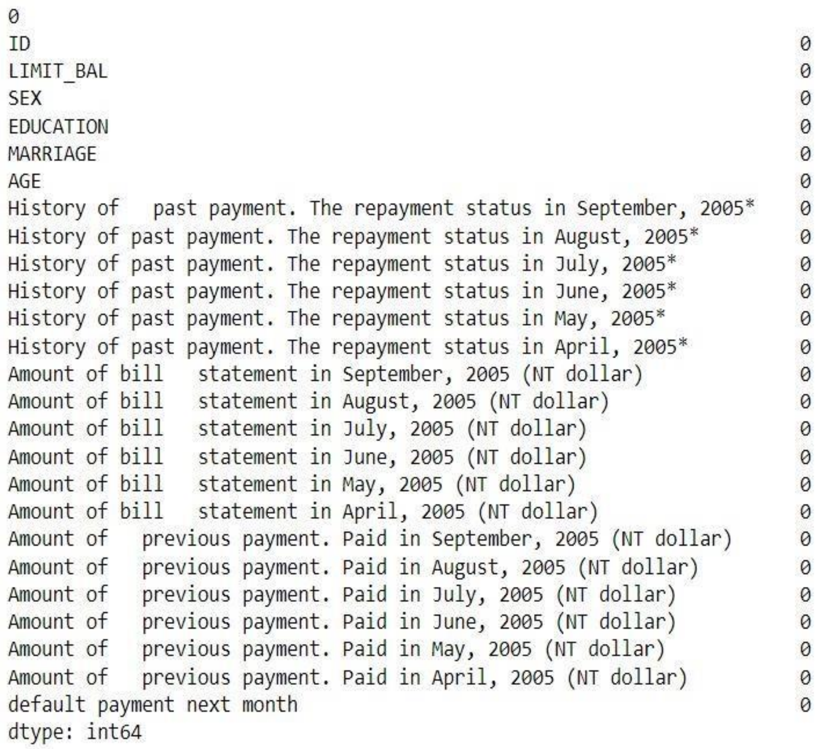
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 30000 entries, 1 to 30000
Data columns (total 25 columns):
 #   Column                      Non-Null Count  Dtype 
---  ------                      --------------  ----- 
 0   ID                          30000 non-null  object
 1   LIMIT_BAL                   30000 non-null  object
 2   SEX                         30000 non-null  object
 3   EDUCATION                   30000 non-null  object
 4   MARRIAGE                    30000 non-null  object
 5   AGE                         30000 non-null  object
 6   PAY_0                       30000 non-null  object
 7   PAY_2                       30000 non-null  object
 8   PAY_3                       30000 non-null  object
 9   PAY_4                       30000 non-null  object
 10  PAY_5                       30000 non-null  object
 11  PAY_6                       30000 non-null  object
 12  BILL_AMT1                   30000 non-null  object
 13  BILL_AMT2                   30000 non-null  object
 14  BILL_AMT3                   30000 non-null  object
 15  BILL_AMT4                   30000 non-null  object
 16  BILL_AMT5                   30000 non-null  object
 17  BILL_AMT6                   30000 non-null  object
 18  PAY_AMT1                    30000 non-null  object
 19  PAY_AMT2                    30000 non-null  object
 20  PAY_AMT3                    30000 non-null  object
 21  PAY_AMT4                    30000 non-null  object
 22  PAY_AMT5                    30000 non-null  object
 23  PAY_AMT6                    30000 non-null  object
 24  default payment next month  30000 non-null  object
dtypes: object(25)
memory usage: 5.7+ MB
```

After Data type conversion, according to the data definition



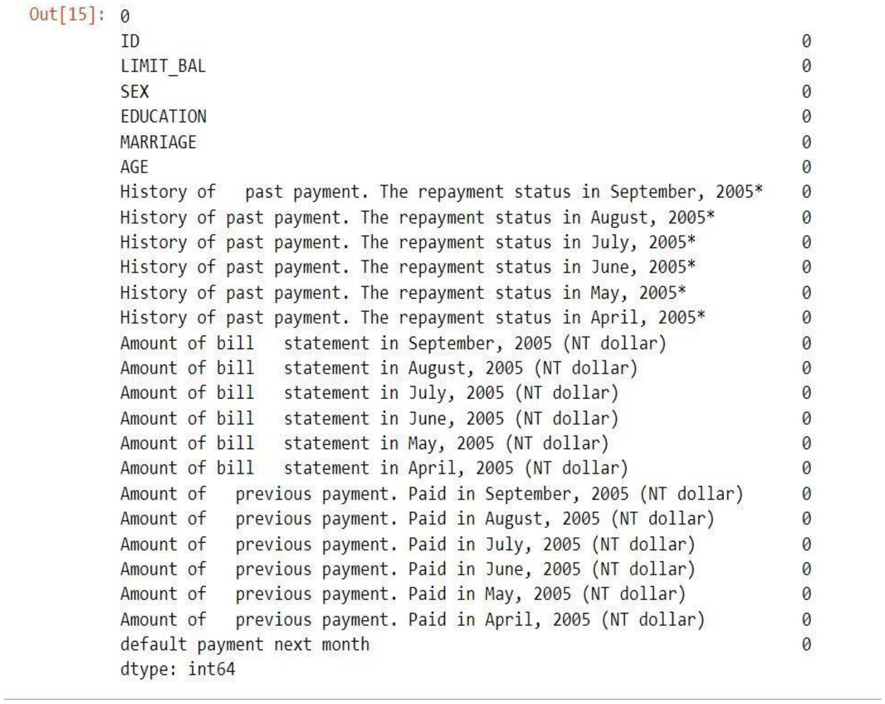
```
LIMIT_BAL                                                                float64
SEX                                                                       object
EDUCATION                                                                 object
MARRIAGE                                                                  object
AGE                                                                      float64
History of   past payment. The repayment status in September, 2005*       object
History of past payment. The repayment status in August, 2005*            object
History of past payment. The repayment status in July, 2005*              object
History of past payment. The repayment status in June, 2005*              object
History of past payment. The repayment status in May, 2005*               object
History of past payment. The repayment status in April, 2005*             object
Amount of bill   statement in September, 2005 (NT dollar)                float64
Amount of bill   statement in August, 2005 (NT dollar)                   float64
Amount of bill   statement in July, 2005 (NT dollar)                     float64
Amount of bill   statement in June, 2005 (NT dollar)                     float64
Amount of bill   statement in May, 2005 (NT dollar)                      float64
Amount of bill   statement in April, 2005 (NT dollar)                    float64
Amount of   previous payment. Paid in September, 2005 (NT dollar)        float64
Amount of   previous payment. Paid in August, 2005 (NT dollar)           float64
Amount of   previous payment. Paid in July, 2005 (NT dollar)             float64
Amount of   previous payment. Paid in June, 2005 (NT dollar)             float64
Amount of   previous payment. Paid in May, 2005 (NT dollar)              float64
Amount of   previous payment. Paid in April, 2005 (NT dollar)            float64
default payment next month                                                 int32
```

Checking for Missing values and Duplicates



```
0
ID                                                                     0
LIMIT_BAL                                                              0
SEX                                                                    0
EDUCATION                                                              0
MARRIAGE                                                               0
AGE                                                                    0
History of   past payment. The repayment status in September, 2005*    0
History of past payment. The repayment status in August, 2005*         0
History of past payment. The repayment status in July, 2005*           0
History of past payment. The repayment status in June, 2005*           0
History of past payment. The repayment status in May, 2005*            0
History of past payment. The repayment status in April, 2005*          0
Amount of bill   statement in September, 2005 (NT dollar)              0
Amount of bill   statement in August, 2005 (NT dollar)                 0
Amount of bill   statement in July, 2005 (NT dollar)                   0
Amount of bill   statement in June, 2005 (NT dollar)                   0
Amount of bill   statement in May, 2005 (NT dollar)                    0
Amount of bill   statement in April, 2005 (NT dollar)                  0
Amount of   previous payment. Paid in September, 2005 (NT dollar)      0
Amount of   previous payment. Paid in August, 2005 (NT dollar)         0
Amount of   previous payment. Paid in July, 2005 (NT dollar)           0
Amount of   previous payment. Paid in June, 2005 (NT dollar)           0
Amount of   previous payment. Paid in May, 2005 (NT dollar)            0
Amount of   previous payment. Paid in April, 2005 (NT dollar)          0
default payment next month                                             0
dtype: int64
```

- There are no missing values in the data set

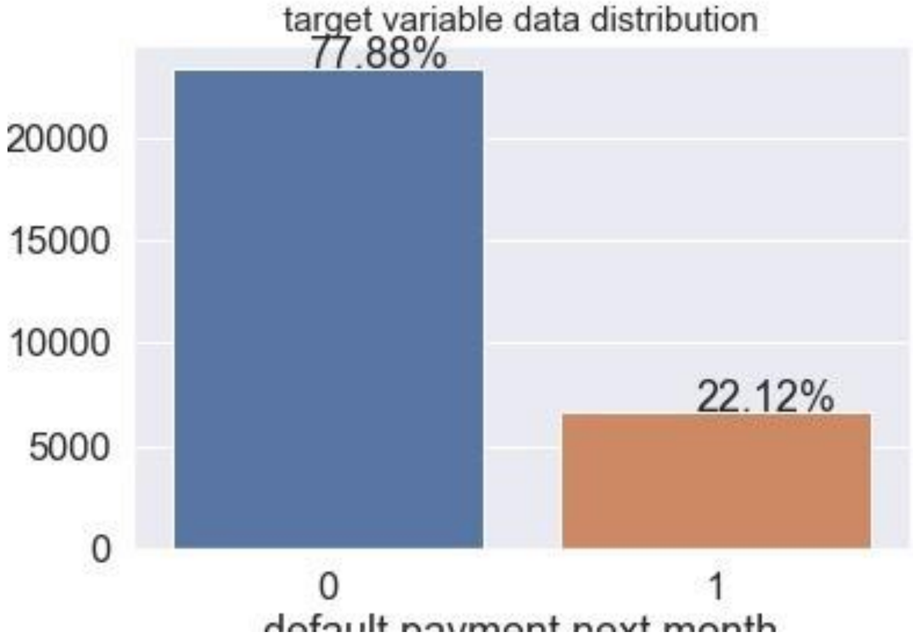


```
Out[15]: 0
         ID                                                                     0
         LIMIT_BAL                                                              0
         SEX                                                                    0
         EDUCATION                                                              0
         MARRIAGE                                                               0
         AGE                                                                    0
         History of   past payment. The repayment status in September, 2005*    0
         History of past payment. The repayment status in August, 2005*         0
         History of past payment. The repayment status in July, 2005*           0
         History of past payment. The repayment status in June, 2005*           0
         History of past payment. The repayment status in May, 2005*            0
         History of past payment. The repayment status in April, 2005*          0
         Amount of bill   statement in September, 2005 (NT dollar)              0
         Amount of bill   statement in August, 2005 (NT dollar)                 0
         Amount of bill   statement in July, 2005 (NT dollar)                   0
         Amount of bill   statement in June, 2005 (NT dollar)                   0
         Amount of bill   statement in May, 2005 (NT dollar)                    0
         Amount of bill   statement in April, 2005 (NT dollar)                  0
         Amount of   previous payment. Paid in September, 2005 (NT dollar)      0
         Amount of   previous payment. Paid in August, 2005 (NT dollar)         0
         Amount of   previous payment. Paid in July, 2005 (NT dollar)           0
         Amount of   previous payment. Paid in June, 2005 (NT dollar)           0
         Amount of   previous payment. Paid in May, 2005 (NT dollar)            0
         Amount of   previous payment. Paid in April, 2005 (NT dollar)          0
         default payment next month                                             0
         dtype: int64
```

- There are no duplicated values in the above dataset.

TARGET VARIABLE

- The target variable of above dataset is ‘ default payment next month ’. We have to predict whether the person will be a defaulter or not.
- The below chart shows the distribution of classes in target variable.(0 : Non-Defaulter, 1 : Defaulter.

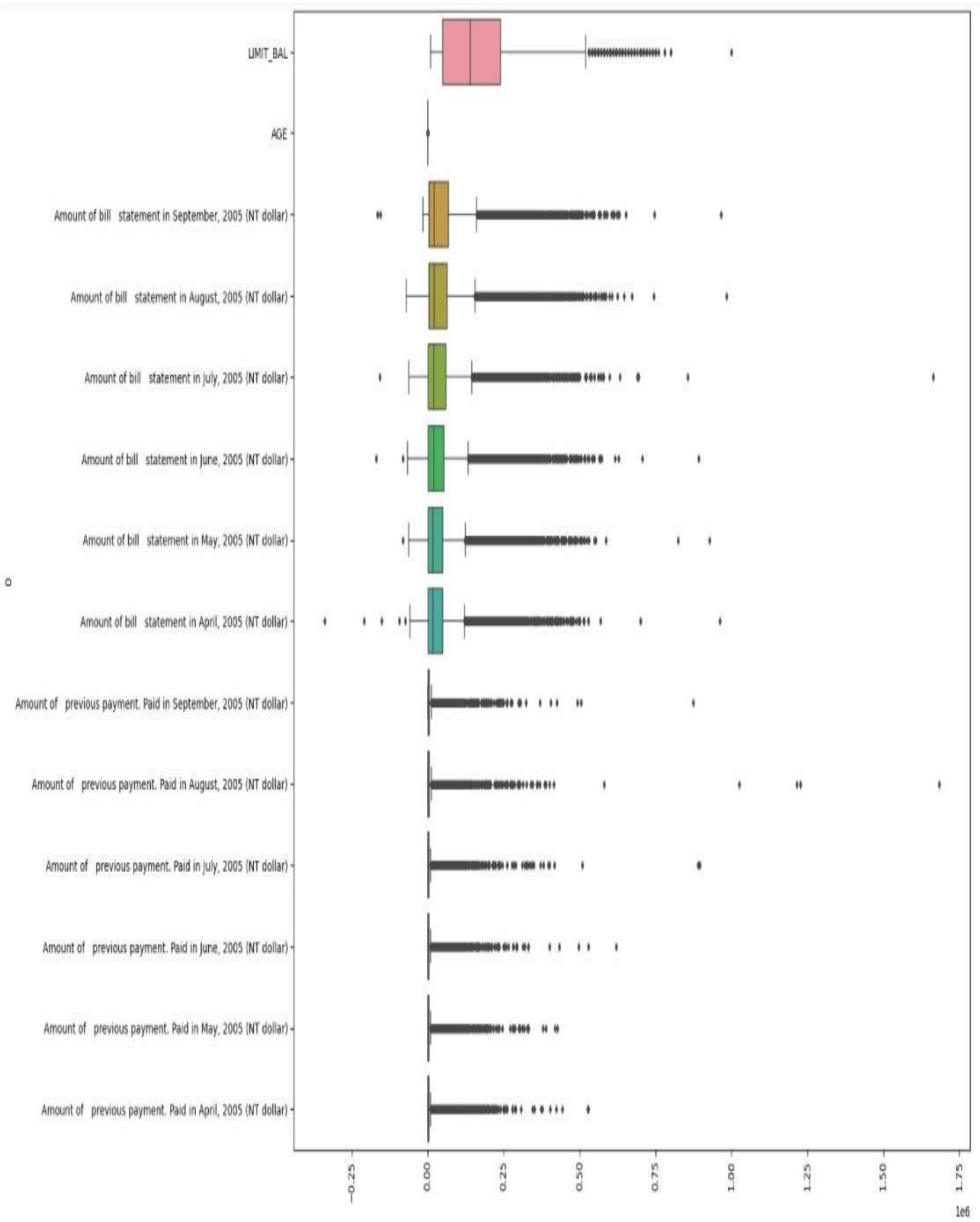


Default payment next month

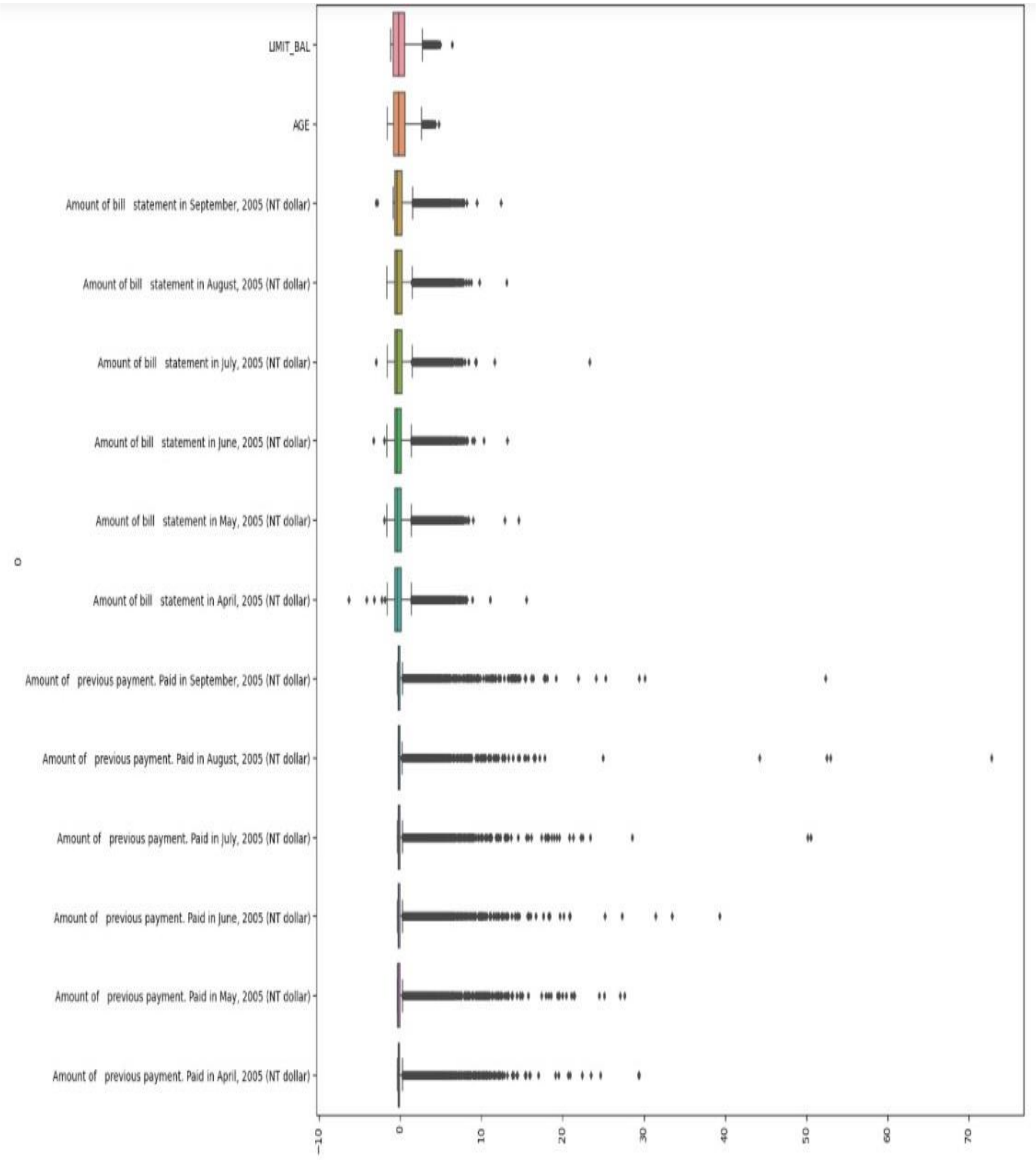
- From the above graph we can infer that there are 22.12% defaulters out of 30000 card holders. And we can also observe that the distribution of target variable is imbalanced.

OUTLIERS

- An OUTLIER is a datapoint which does not follow the same pattern as other datapoints. OUTLIERS can influence our model accuracy.
- Treating OUTLIERS is one of the most important step in data analysis and machine learning problems. We should properly check all the outliers and treat them according to our problem statement and business domain.
- There are several ways to treat the OUTLIERS, some of them are :
- Inter Quartile Range Method (IQR)
- Capping the outliers with upper and lower limits.
- Eliminating the extreme values.



The above box plot showcases outliers in the data set. Since all the features are not having the same range it is difficult to visualize some t=of the features like ‘AGE’ and ‘Amount of previous payments’ . Hence we scaled the data so that we can bring all the features on the same scale

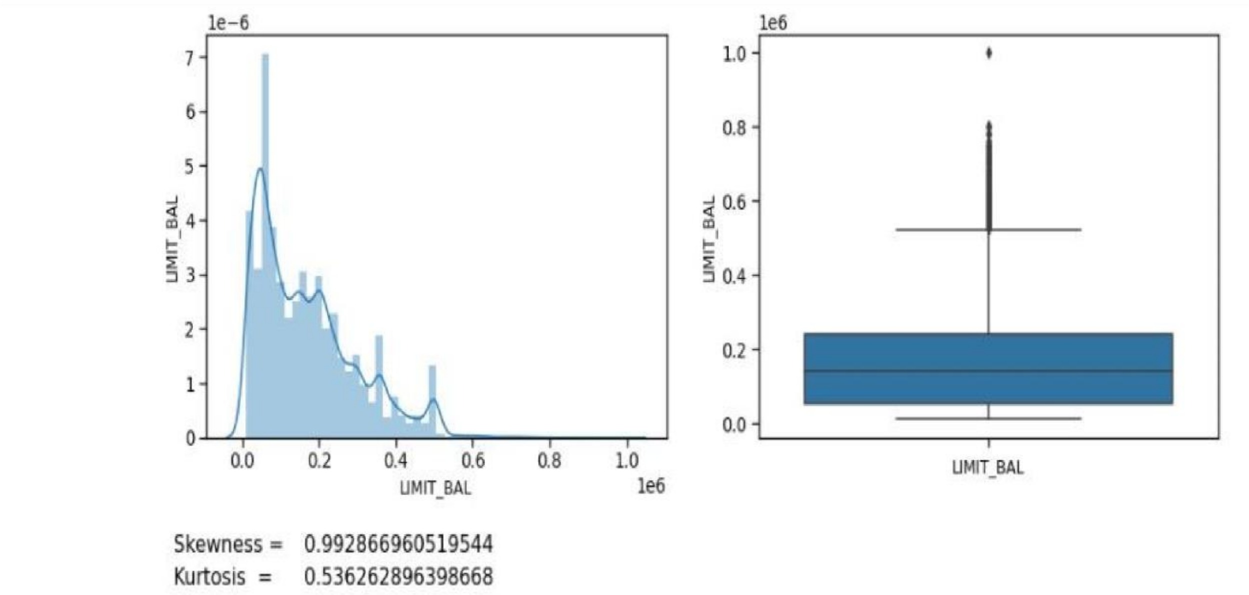


The diagram above shows the outliers after scaling the data. Now we can observe outliers more precisely and we can also infer that most of the numerical features are highly positively skewed.

EXPLORATORY DATA ANALYSIS

UNIVARIATE ANALYSIS

1. Limit Balance

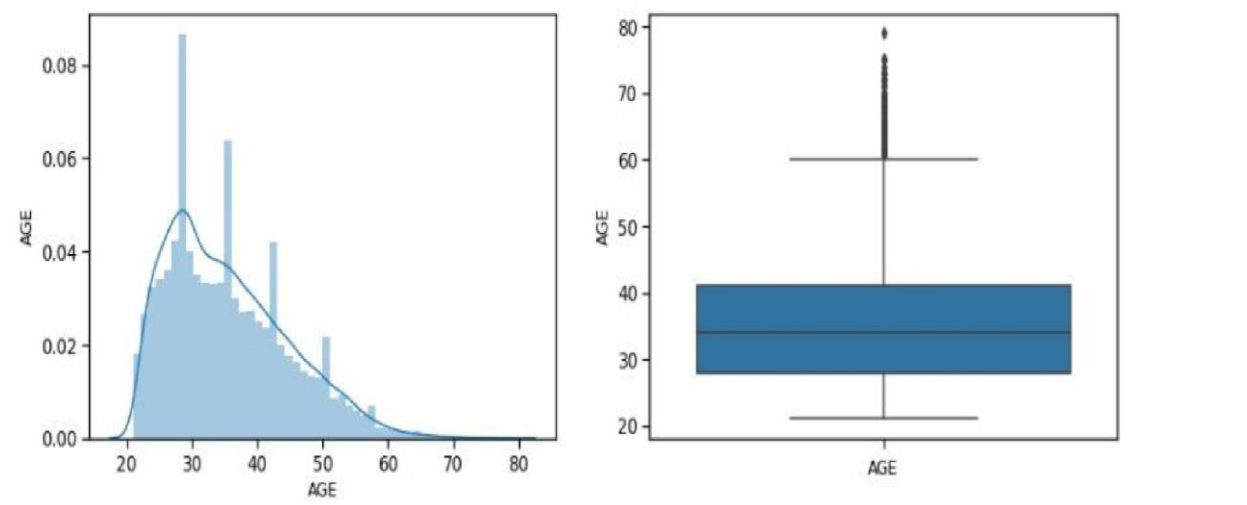


```
Skewness =    0.992866960519544
Kurtosis  =   0.536262896398668
```

For Numerical Variables :- We plot the distribution curve and box plot to study the variation of the numerical data.

- Limit Balance in right skewed
- It is mesokurtic and has a elongated tail on the right.
- There are outliers present in the LIMIT BALANCE.

2. Age



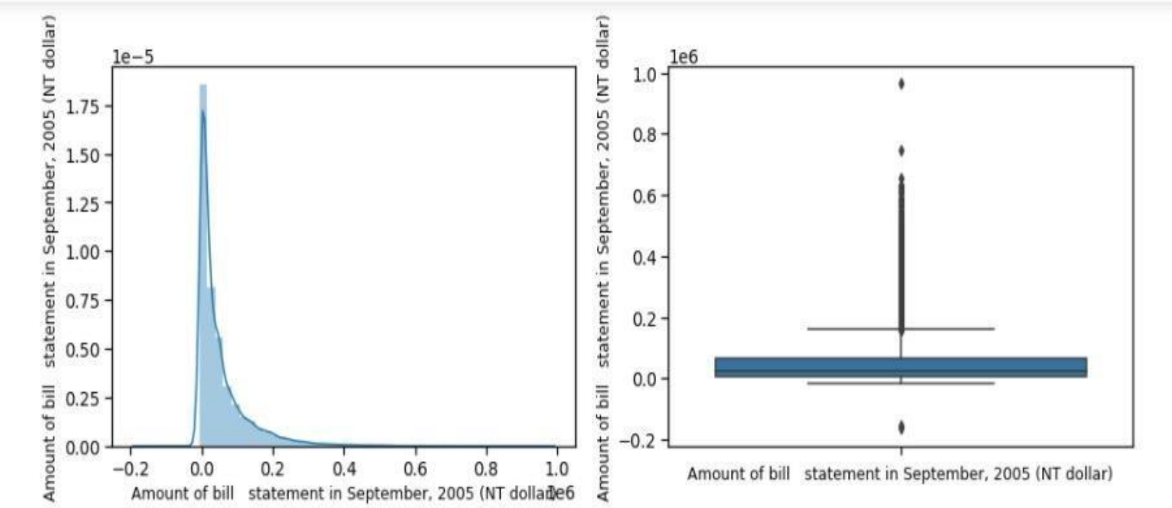
```
Skewness =    0.7322458687830562
Kurtosis  =    0.04430337823580954
```

Most of the costumers are around 25-40 years of age.

- Age is right skewed.
- It is mesokurtic.

The IQR of Age lies from 20 to 60, hence we can clearly see outliers are present

3. Amount of bill in September,2005



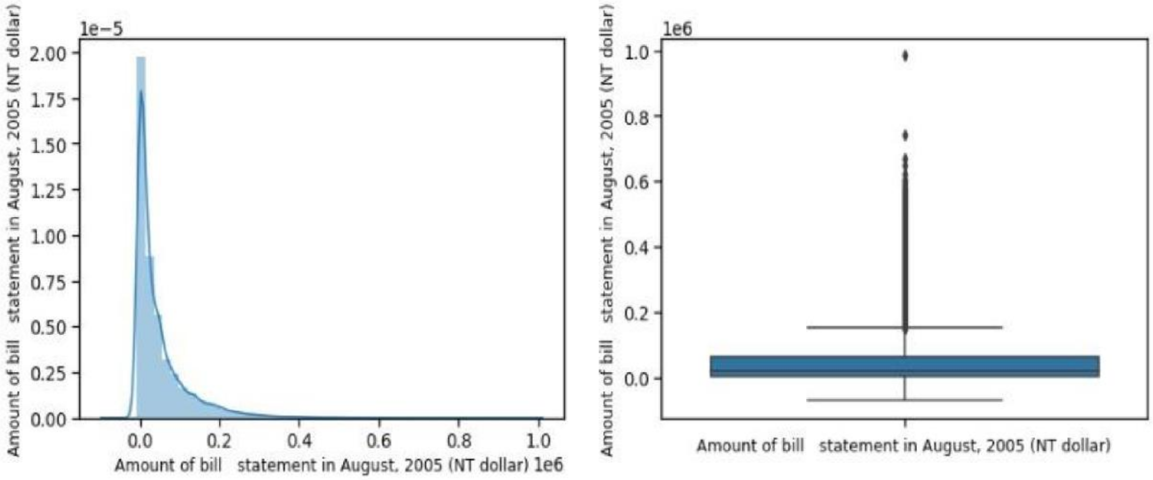
```
Skewness =    2.6638610220232612
Kurtosis  =    9.806289341330837
```

- It is highly skewed.

- It is leptokurtic and has a wide tail.

- There are large number of outliers, that means there some users who make big purchases using their credit cards.

4.Amount of bill in August, 2005



```
Skewness =   2.7052208534082856
Kurtosis  =   10.302945922629279
```

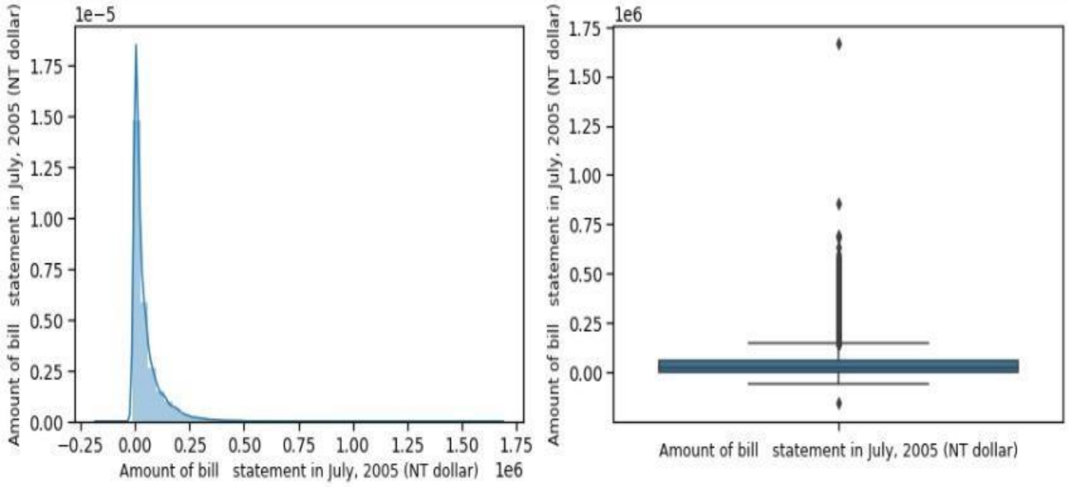
- Most of the bill amount ranges from 0-20000 USD.

- It is highly skewed , and having outliers.

- It is leptokurtic, with a wide tail on the right.

- We can also see a big jump in the outliers after 0.75 in box plot above.

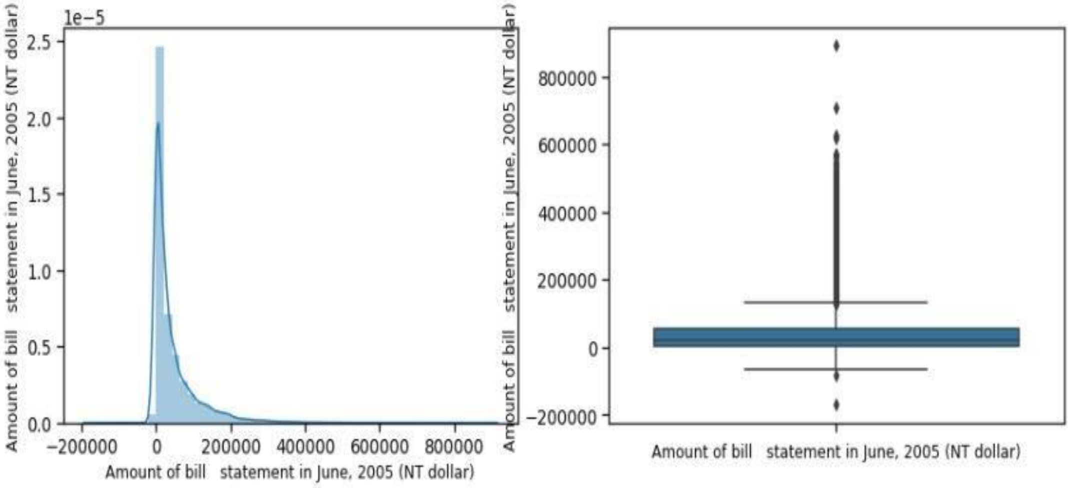
5. Amount of bill in July,2005



```
Skewness =   3.0878300462007244
Kurtosis =   19.783255144801103
```

- Most of the bill amount ranges from 0-20000 USD
- It is highly skewed , and having outliers.
- It is leptokurtic, with a wide tail on the right.
- We can also see a big jump in the outliers after 0.75 in box plot above

6. Amount of bill in June,2005



```
Skewness =   2.8219652908028117
Kurtosis =   11.309324826831903
```

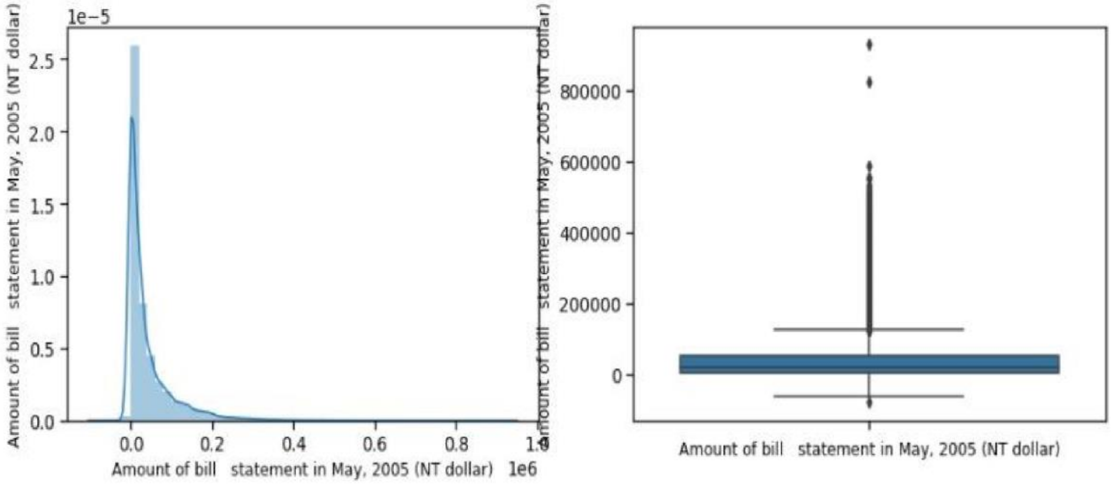
- The bill amount for June,2005 is ranging from -170000 to 891586.

- We can see some of the bill amounts are less than 0 that means these are either they have revolving credits or the bank owes the money to customer.

- It is highly skewed, and having outliers on both sides.

- It is leptokurtic, with a wide tail.

7. Amount of bill in May, 2005



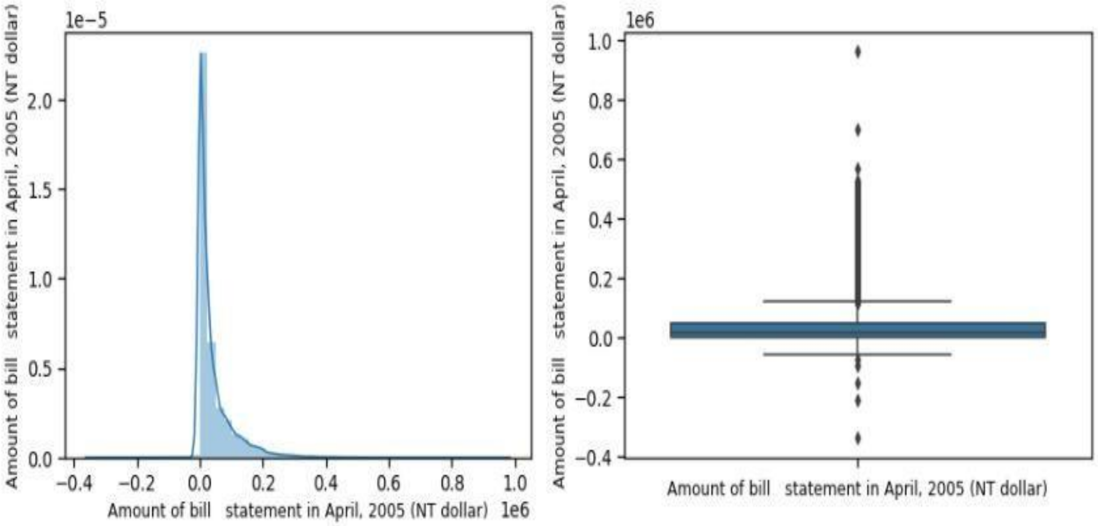
```
Skewness =   2.8763798667028633
Kurtosis =   12.30588128593057
```

- We can see some of the bill amount are less than 0 that means these are either they have revolving credits or the bank owes the money to customer.

- It is highly skewed , and having outliers on both sides.

- It is leptokurtic, with a wide tail on the right.

8. Amount of bill in April,2005

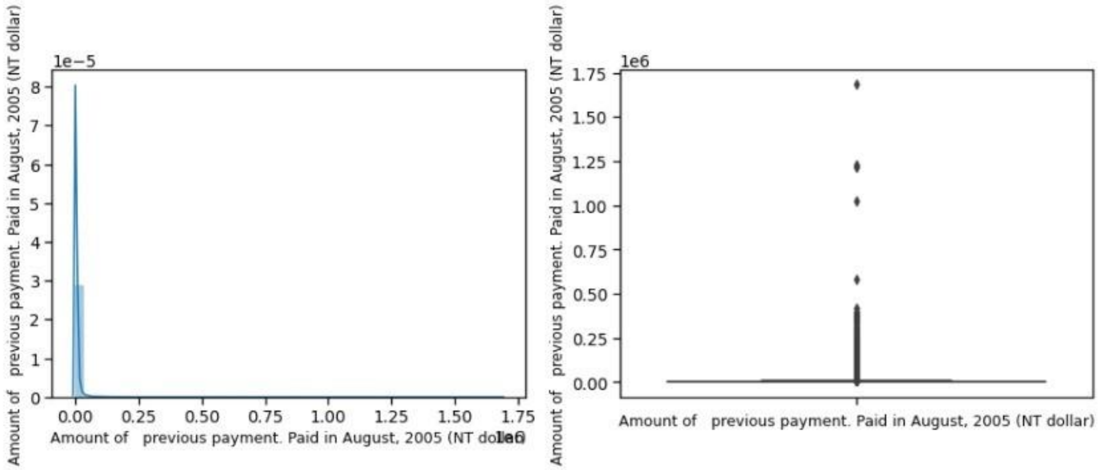


Skewness = 2.8466445756603678
Kurtosis = 12.270705286713094

- It is highly skewed , and having outliers on both sides.

- It is leptokurtic, with a wide tail.

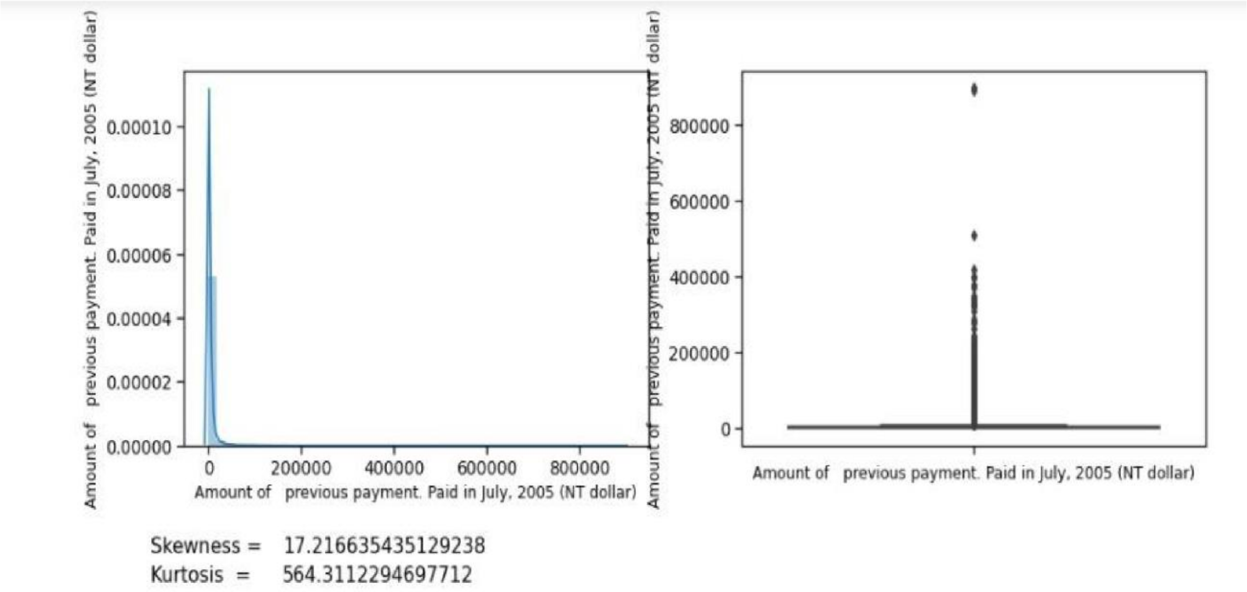
- 9. Amount of previous payment, paid in August, 2005.



Skewness = 30.45381745016943
Kurtosis = 1641.6319110097434

- It is very highly skewed
- having a leptokurtic curve, with wide right tail.

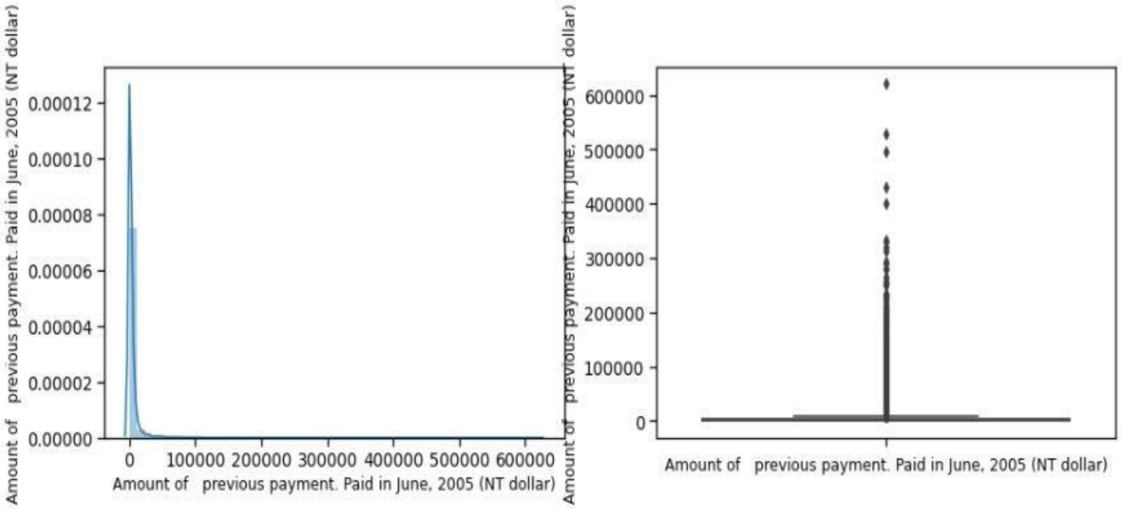
10. Amount of previous payment, paid in July, 2005.


Amount of previous payment. Paid in July, 2005 (NT dollar)
0.00010
0.00008
0.00006
0.00004
0.00002
0.00000
0
200000
400000
600000
800000
Amount of previous payment. Paid in July, 2005 (NT dollar)
Amount of previous payment. Paid in July, 2005 (NT dollar)
800000
600000
400000
200000
0
Amount of previous payment. Paid in July, 2005 (NT dollar)

```
Skewness =   17.216635435129238
Kurtosis =   564.3112294697712
```

- It is having very high extreme values.
- It is highly skewed,
- And it is having a leptokurtic curve, with wide right tail.

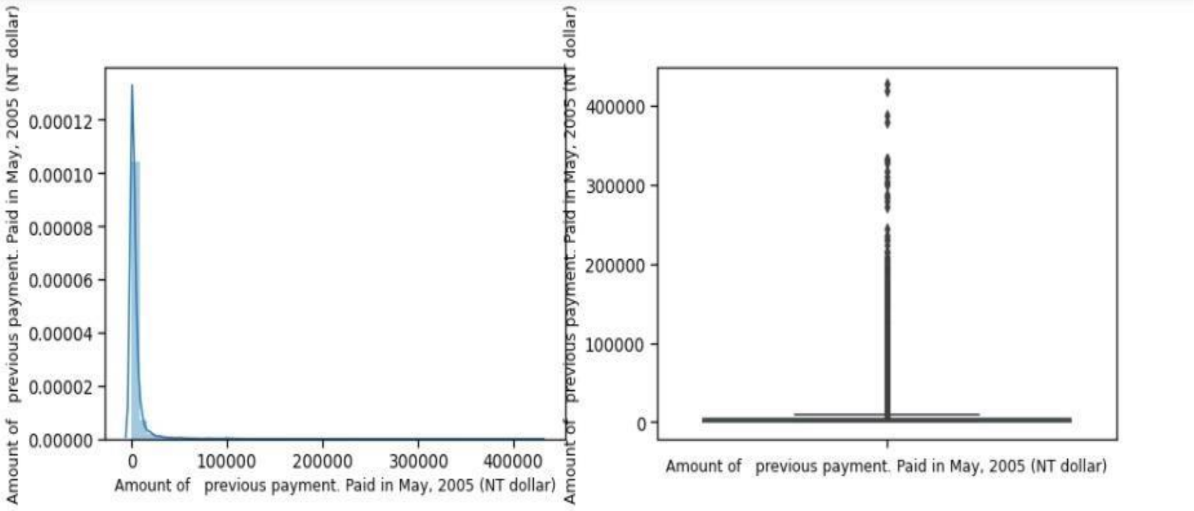
11.Amount of previous payment, paid in June, 2005.



Skewness = 12.904984823542545
Kurtosis = 277.3337677160758

- It is having very high extreme values, it ranges from 0 up to more than 600000.
- It is highly skewed,
- And it is having a leptokurtic curve, with wide right tail.

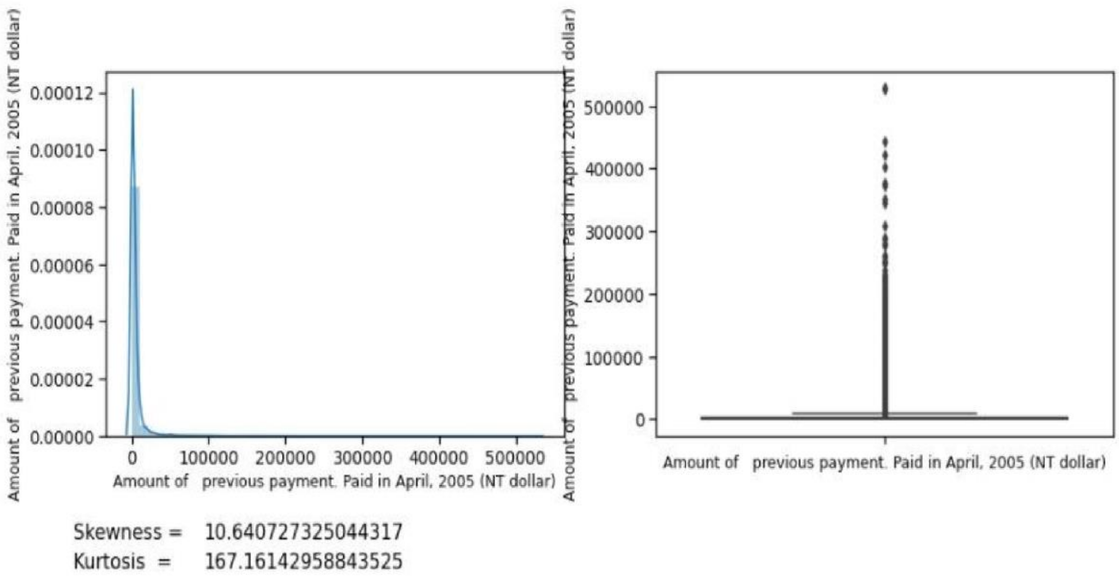
12.Amount of previous payment, paid in May, 2005.



Skewness = 11.127417052173817
Kurtosis = 180.06394016204655

- It is having very high extreme values, it ranges from 0 up to more than 400000
- It is highly skewed
- And it is having a leptokurtic curve, with wide right tail.

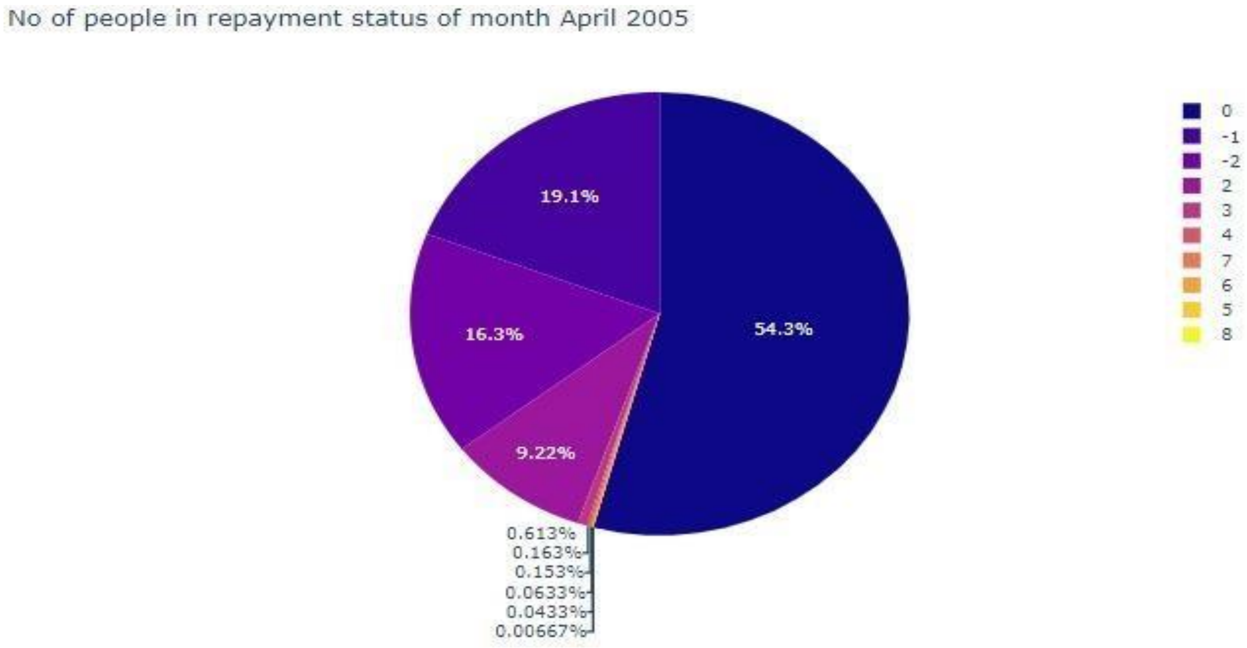
13 .Amount of previous payment, paid in April, 2005.

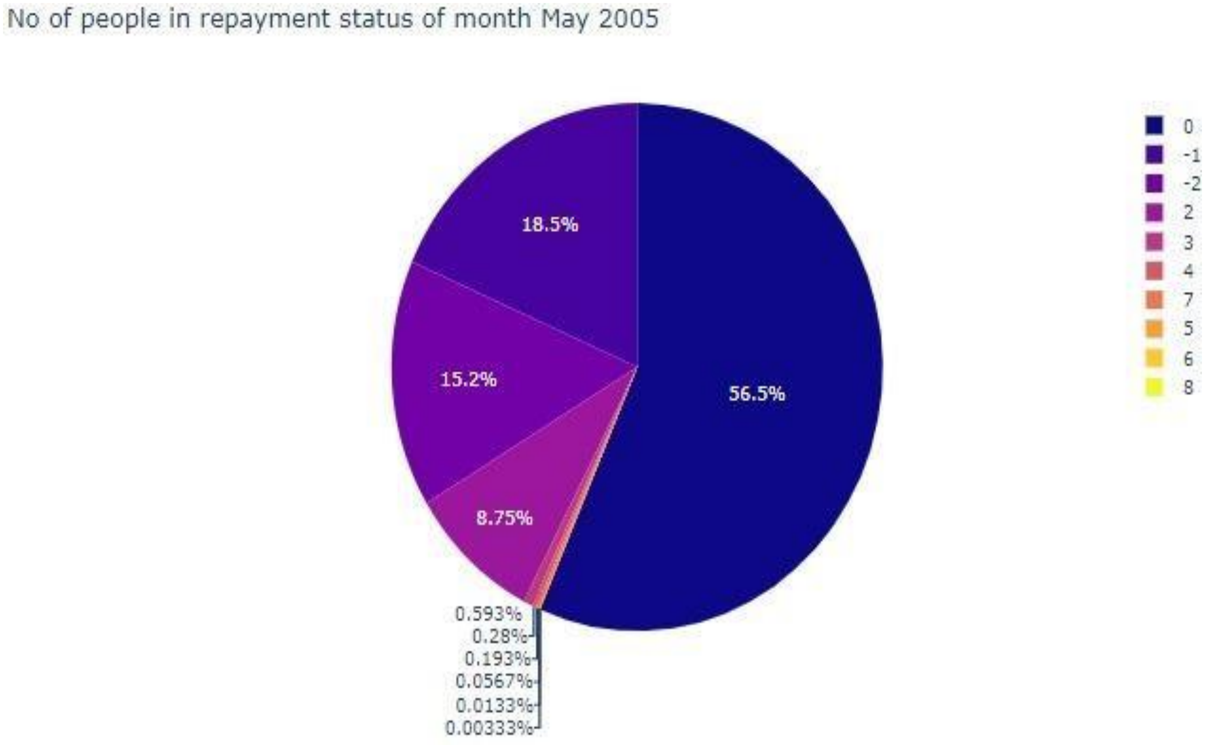


```
Skewness =   10.640727325044317
Kurtosis  =   167.16142958843525
```

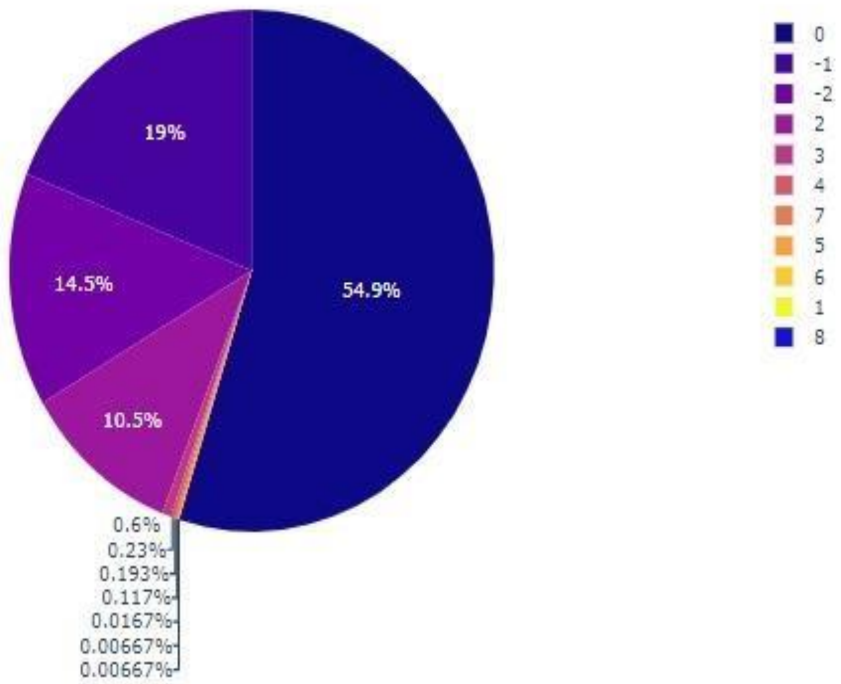
- It is having very high extreme values, it ranges from 0 up to more than 500000.
- It is highly skewed,
- And it is having a leptokurtic curve, with wide right tail.

Pie Charts for History of Payment Status

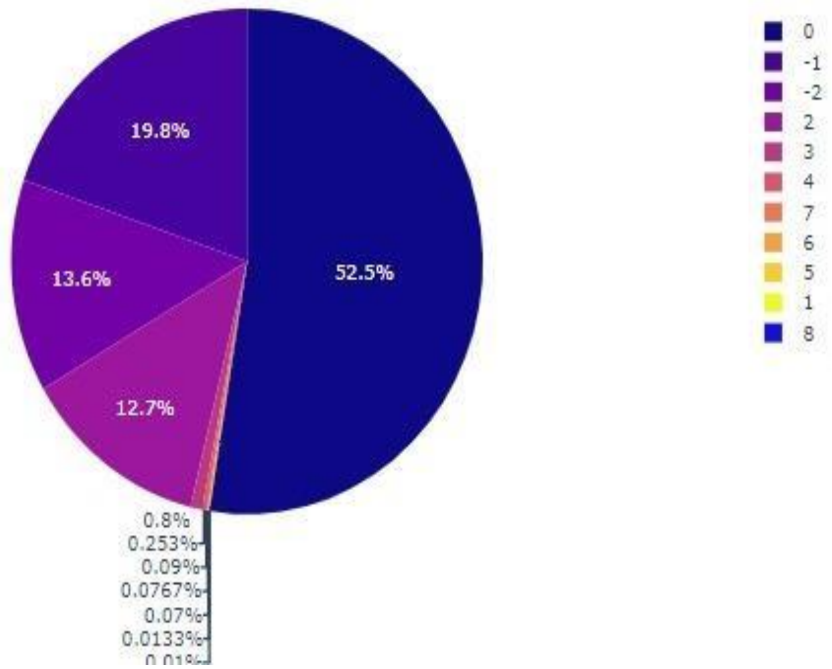




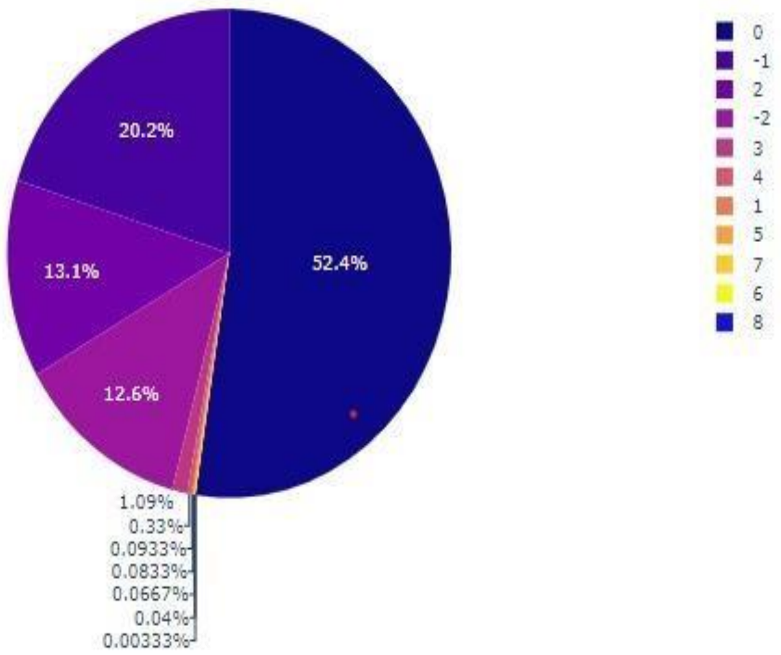
No of people in repayment status of month June 2005



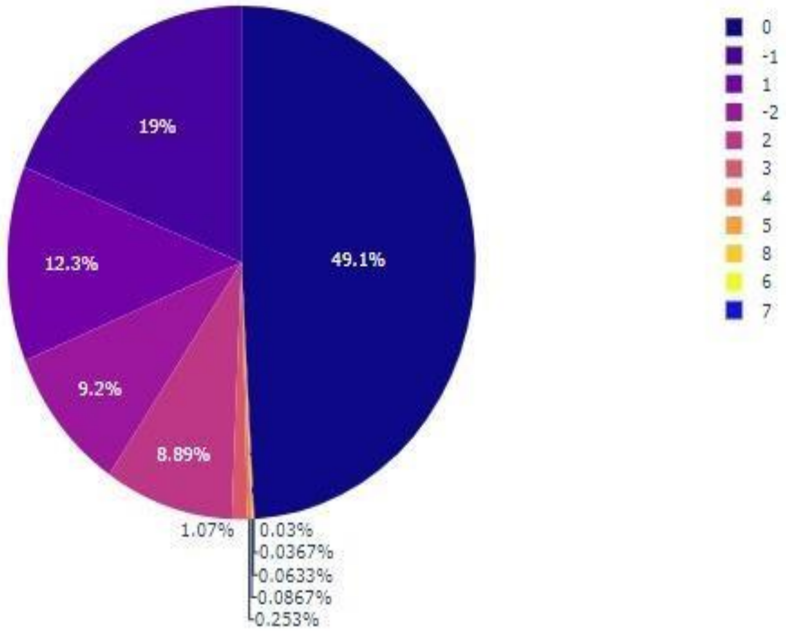
No of people in repayment status of month July 2005



No of people in repayment status of month August 2005



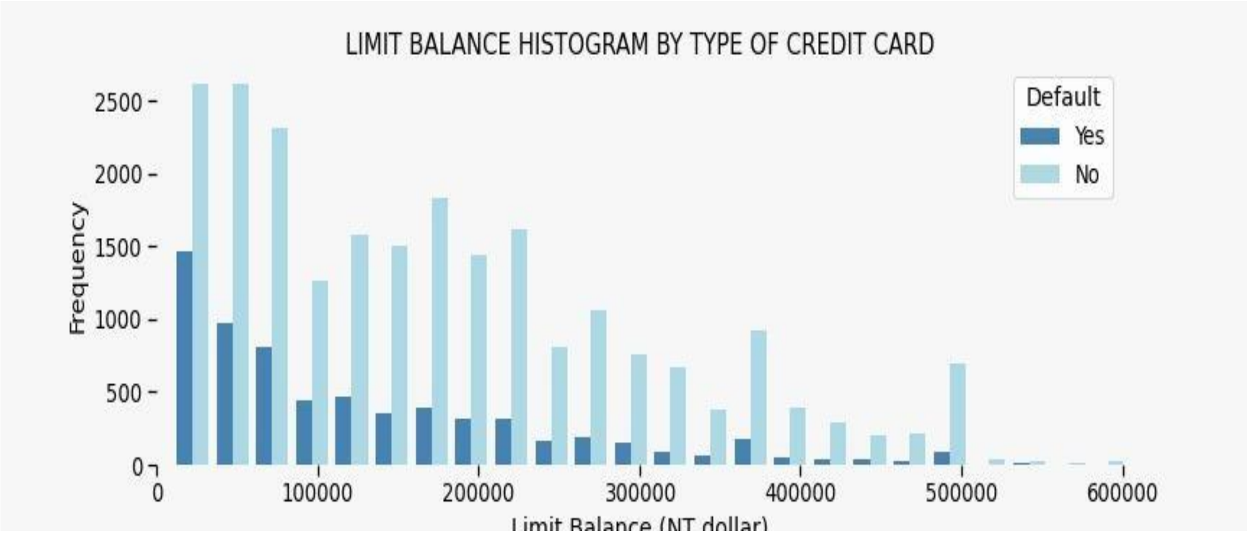
No of people in repayment status of month september 2005



- From the above graph, we observe that majority of people fall under category of either revolving credit(0),paid duly(-1) or no consumption(-2)
- Also, most of the people are not delaying the payment for more than 2 months

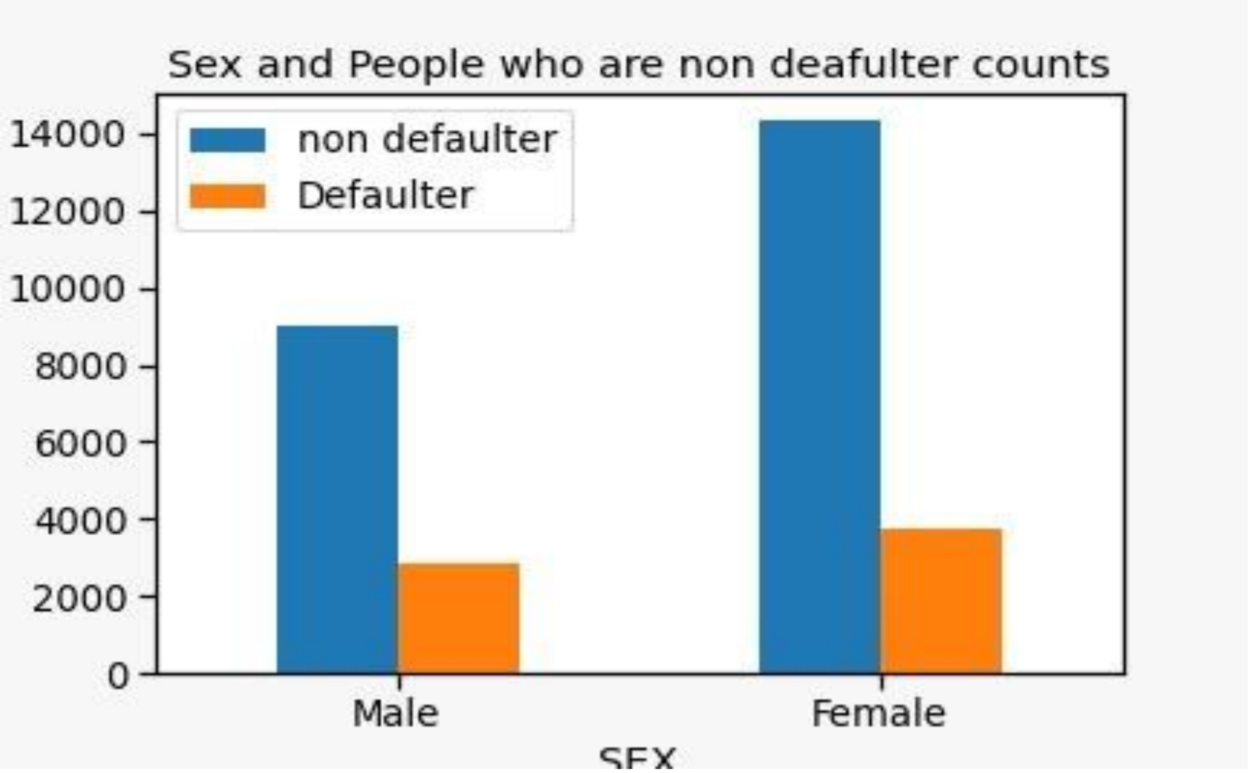
BIVARIATE ANALYSIS

1. LIMIT BALANCE V/S DEFAULTER



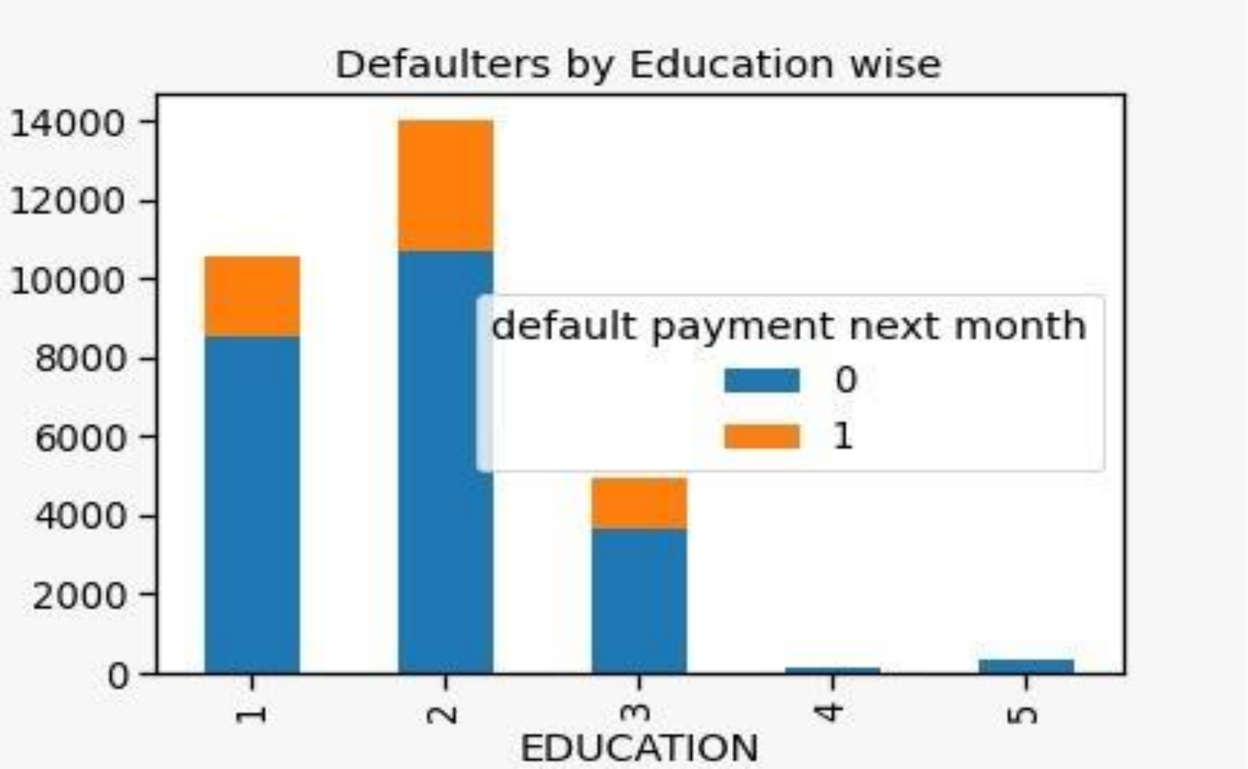
- The above histogram shows the count of defaulters and non-defaulter having different credit limits,
- We can see as the credit limit increases the number of defaulters decreases.
- Category of holders with Limit Balance of 10000 is having about 1500 defaulters out of 4000 i.e., about 37.5%.
- Which leads us to conclude that customers having lower credit limit are more likely to default may be due to lesser income, higher expenses, inflation and several other factors.

2.GENDER V/S DEFAULTER



- From the above count plot, can infer that number of female card holders is higher than male card holders.
- But if we look at the defaulters, we would come to know that the percentage of default is higher in male users than that of female users.
- Among all the card holders there are approximately 20% defaulters I.e.,7000 of 30000, out of which :
- Male users amounts to 42.85% of 20%, I.e., 3000 out of 7000.
- Female users amounts to 57.15% of 20%, I.e., 4000 out of 7000.

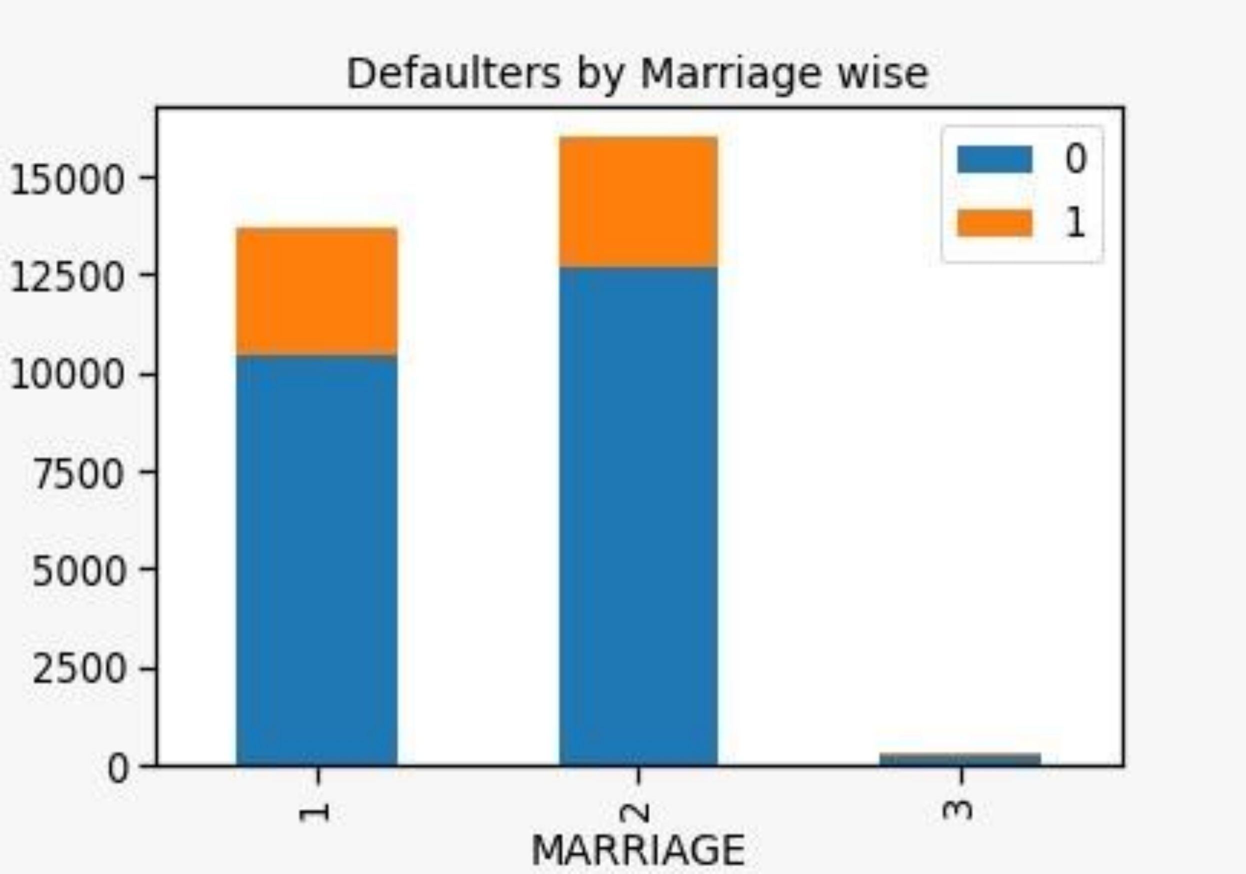
3.EDUCATION LEVEL V/S DEFAULTERS



The above chart shows the distribution of number of total card users by the level of the education.

We can see that people with education level 2 i.e., University are the ones who defaults the most, followed by education 1 and 3 respectively i.e., graduate school and high school.

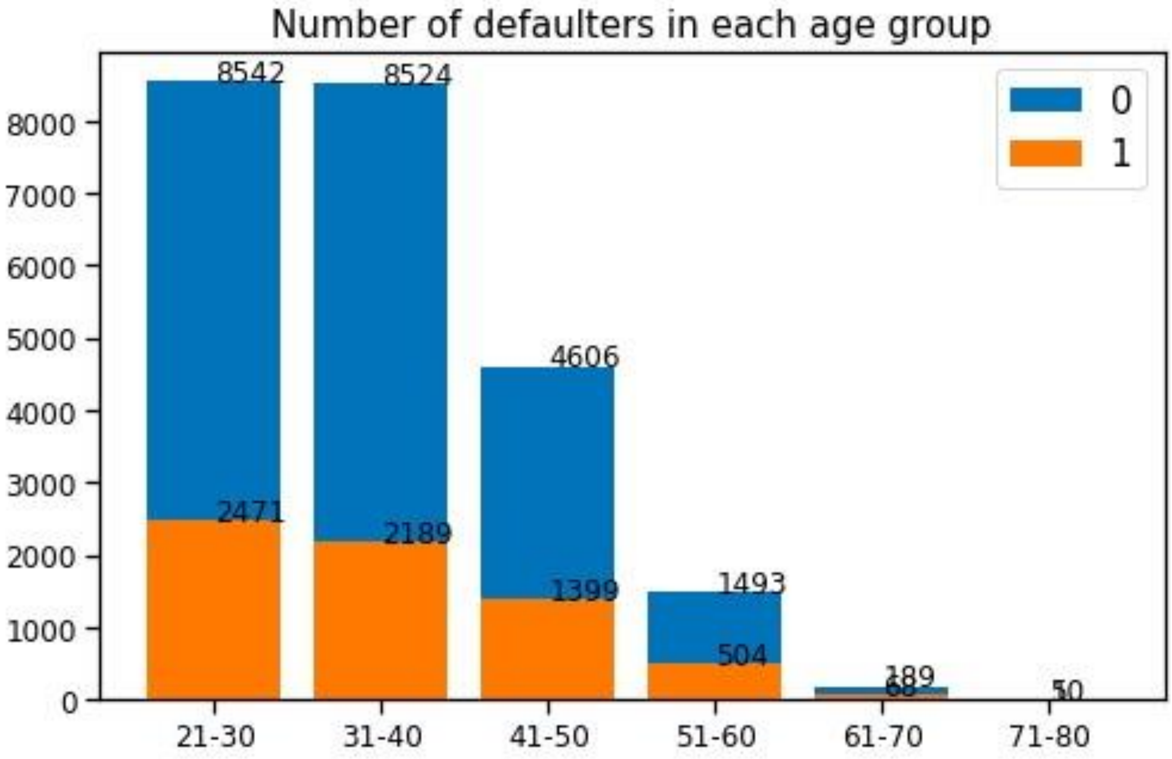
4.MARITAL STATUS V/S DEFAULTER



The above count plot showcases the number of defaulters and non-defaulters based on their marital status.

We can see that number of defaulters in both the married and unmarried category similar.

5. AGE GROUP V/S DEFAULTERS



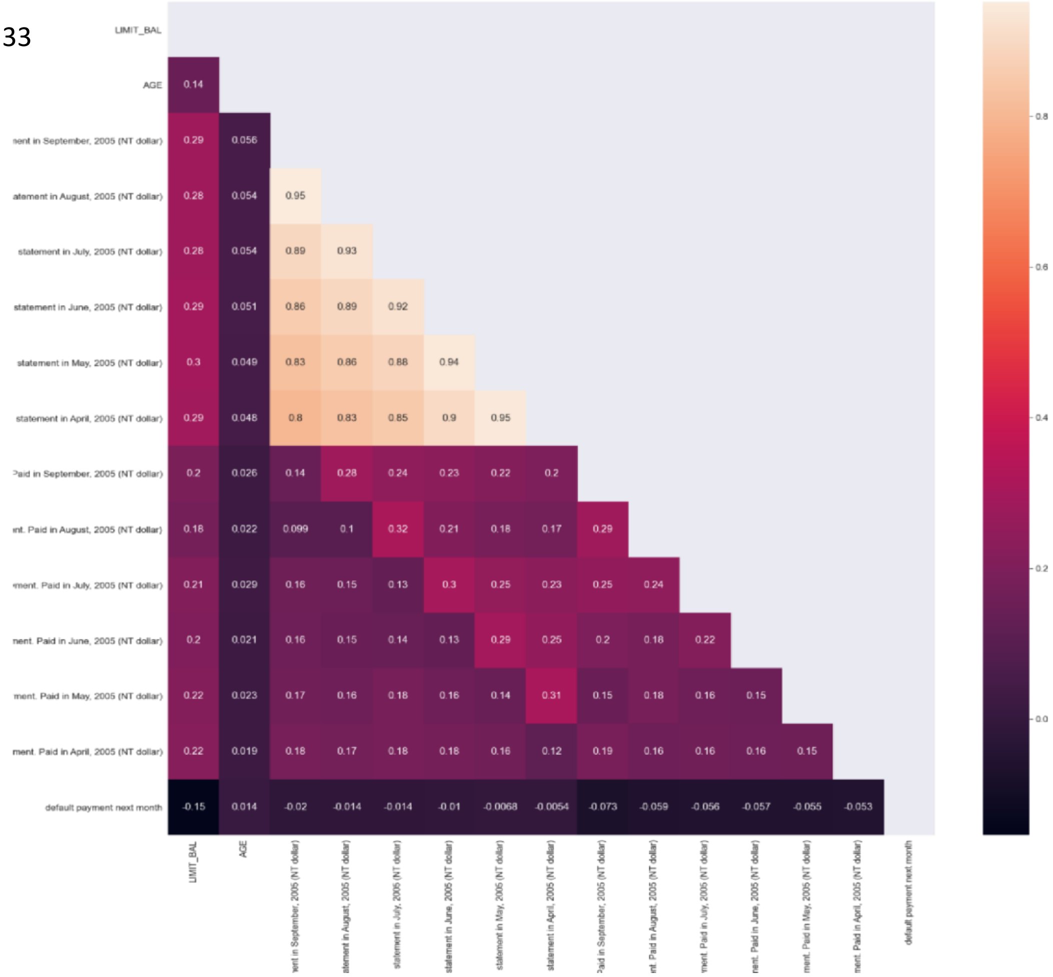
From the above plot we can see that most of the defaulters belong to the age group of 21-30 and 31-40 and the number of defaulters decreases with the increase in age.

It is more like for the users with lower age to have lower salary as well due to which they often default in payment.

On the other hand people with more age are more likely to be more financially stable hat is why the old age groups have lesser number of defaulters as compared to lower age groups.

6.CORRELATION MATRIX

33



Heat-Map - Pearson Correlation Matrix (Assumption : For the Pearson correlation, both variables should be normally distributed. Other assumptions include linearity and homoscedasticity)

It gives a measure of how much two numeric variables are linearly correlated. It tries to obtain a best fit line between two numeric variables and how close the points are to a fitted line.

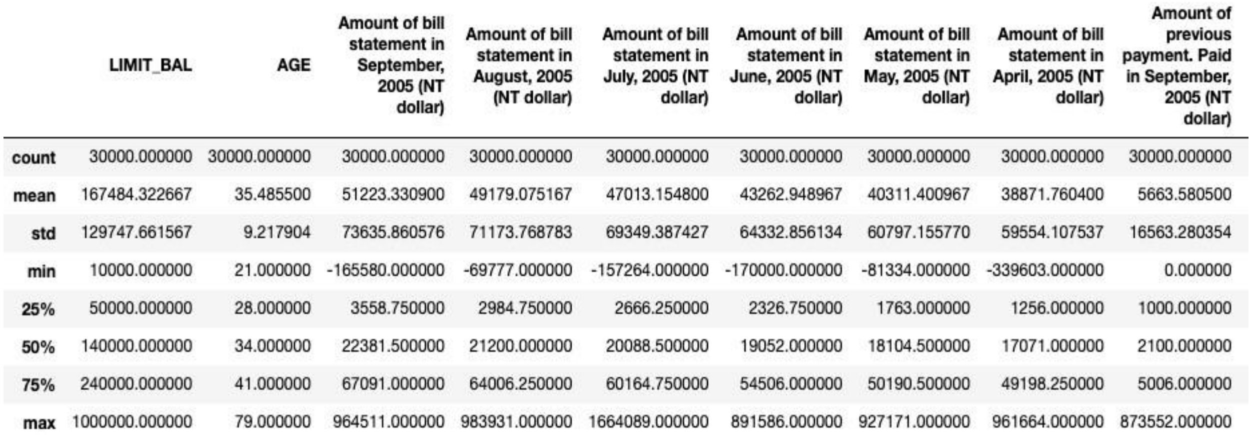
1. From the graph we can see that high level of collinearity is not present in our data.

2. But we can see some high to moderate level of correlations between:

3. Bill statement of current month and the previous months.

STATISTICAL TESTS

5-Stats Summary



	LIMIT_BAL	AGE	Amount of bill statement in September, 2005 (NT dollar)	Amount of bill statement in August, 2005 (NT dollar)	Amount of bill statement in July, 2005 (NT dollar)	Amount of bill statement in June, 2005 (NT dollar)	Amount of bill statement in May, 2005 (NT dollar)	Amount of bill statement in April, 2005 (NT dollar)	Amount of previous payment. Paid in September, 2005 (NT dollar)
count	30000.000000	30000.000000	30000.000000	30000.000000	30000.000000	30000.000000	30000.000000	30000.000000	30000.000000
mean	167484.322667	35.485500	51223.330900	49179.075167	47013.154800	43262.948967	40311.400967	38871.760400	5663.580500
std	129747.661567	9.217904	73635.860576	71173.768783	69349.387427	64332.856134	60797.155770	59554.107537	16563.280354
min	10000.000000	21.000000	-165580.000000	-69777.000000	-157264.000000	-170000.000000	-81334.000000	-339603.000000	0.000000
25%	50000.000000	28.000000	3558.750000	2984.750000	2666.250000	2326.750000	1763.000000	1256.000000	1000.000000
50%	140000.000000	34.000000	22381.500000	21200.000000	20088.500000	19052.000000	18104.500000	17071.000000	2100.000000
75%	240000.000000	41.000000	67091.000000	64006.250000	60164.750000	54506.000000	50190.500000	49198.250000	5006.000000
max	1000000.000000	79.000000	964511.000000	983931.000000	1664089.000000	891586.000000	927171.000000	961664.000000	873552.000000

From the above 5-stats summary,

minimum limit balance is 10,000, 75% of the balances is 24,000 and max is 10,00,000 that also represents there are outliers.
Minimum age is 21 and max age is 79
And, if we observe the min balance in the amount of bill categories, the amount is in negative which means that bank owes the amount to the customer. And, the maximum amount range is also high which also gives us an idea that there are outliers in the data.

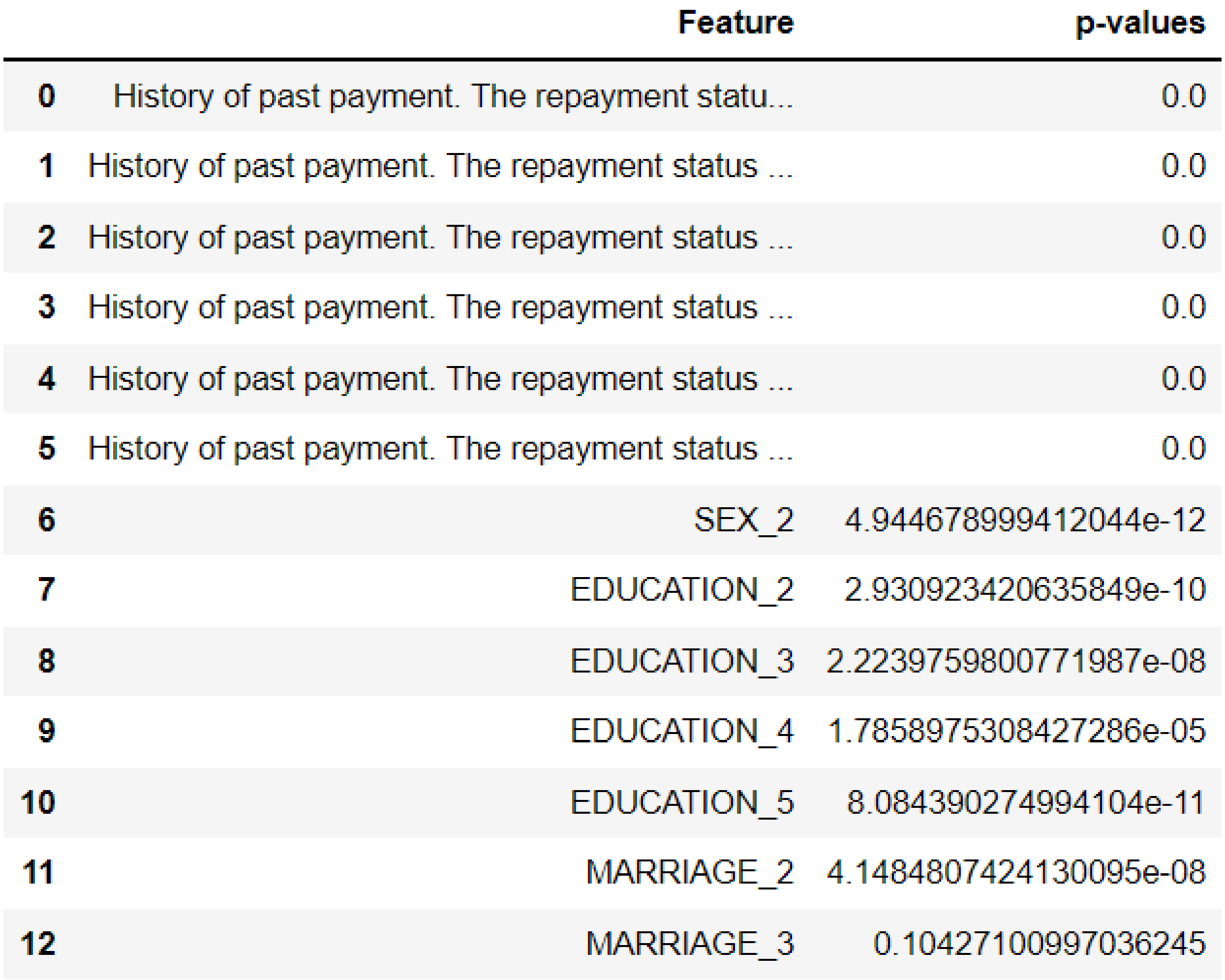
Chi-Square Test

Categorical columns – For categorical columns we perform chi-square test to check for the significance of the categorical column with respect to default payment next month column.

Statistics

Hypothesis of Chi-square test

H0 : Attributes are independent



	Feature	p-values
0	History of past payment. The repayment statu...	0.0
1	History of past payment. The repayment status ...	0.0
2	History of past payment. The repayment status ...	0.0
3	History of past payment. The repayment status ...	0.0
4	History of past payment. The repayment status ...	0.0
5	History of past payment. The repayment status ...	0.0
6	SEX_2	4.944678999412044e-12
7	EDUCATION_2	2.930923420635849e-10
8	EDUCATION_3	2.2239759800771987e-08
9	EDUCATION_4	1.7858975308427286e-05
10	EDUCATION_5	8.084390274994104e-11
11	MARRIAGE_2	4.1484807424130095e-08
12	MARRIAGE_3	0.10427100997036245

Attributes are dependent
#Since the p-values is less than 0.05, we reject the null hypothesis. Hence all the features are significant.

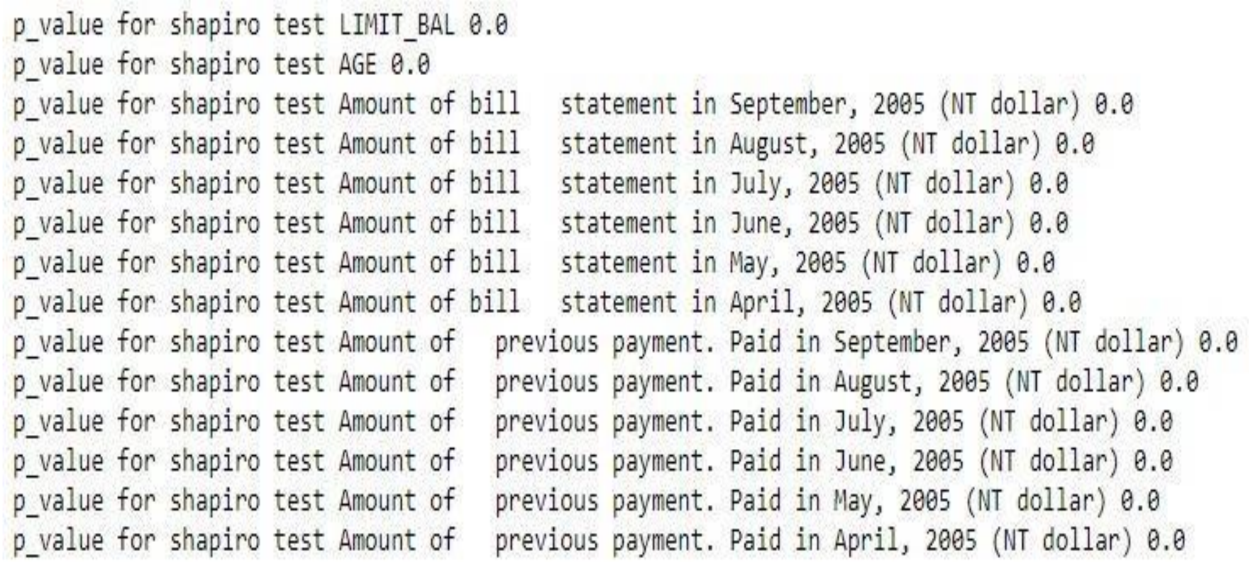
Shapiro-Walk Test

We perform Shapiro test to check if the numerical features are normally distributed or not.

Hypothesis for Shapiro Test

H0: Data is normally distributed

H1: Data is not normally distributed



```
p_value for shapiro test LIMIT_BAL 0.0
p_value for shapiro test AGE 0.0
p_value for shapiro test Amount of bill   statement in September, 2005 (NT dollar) 0.0
p_value for shapiro test Amount of bill   statement in August, 2005 (NT dollar) 0.0
p_value for shapiro test Amount of bill   statement in July, 2005 (NT dollar) 0.0
p_value for shapiro test Amount of bill   statement in June, 2005 (NT dollar) 0.0
p_value for shapiro test Amount of bill   statement in May, 2005 (NT dollar) 0.0
p_value for shapiro test Amount of bill   statement in April, 2005 (NT dollar) 0.0
p_value for shapiro test Amount of   previous payment. Paid in September, 2005 (NT dollar) 0.0
p_value for shapiro test Amount of   previous payment. Paid in August, 2005 (NT dollar) 0.0
p_value for shapiro test Amount of   previous payment. Paid in July, 2005 (NT dollar) 0.0
p_value for shapiro test Amount of   previous payment. Paid in June, 2005 (NT dollar) 0.0
p_value for shapiro test Amount of   previous payment. Paid in May, 2005 (NT dollar) 0.0
p_value for shapiro test Amount of   previous payment. Paid in April, 2005 (NT dollar) 0.0
```

#Since p-value is less than 0.05, we reject the null hypothesis. Hence the data is not normally distributed and we perform non parametric tests.

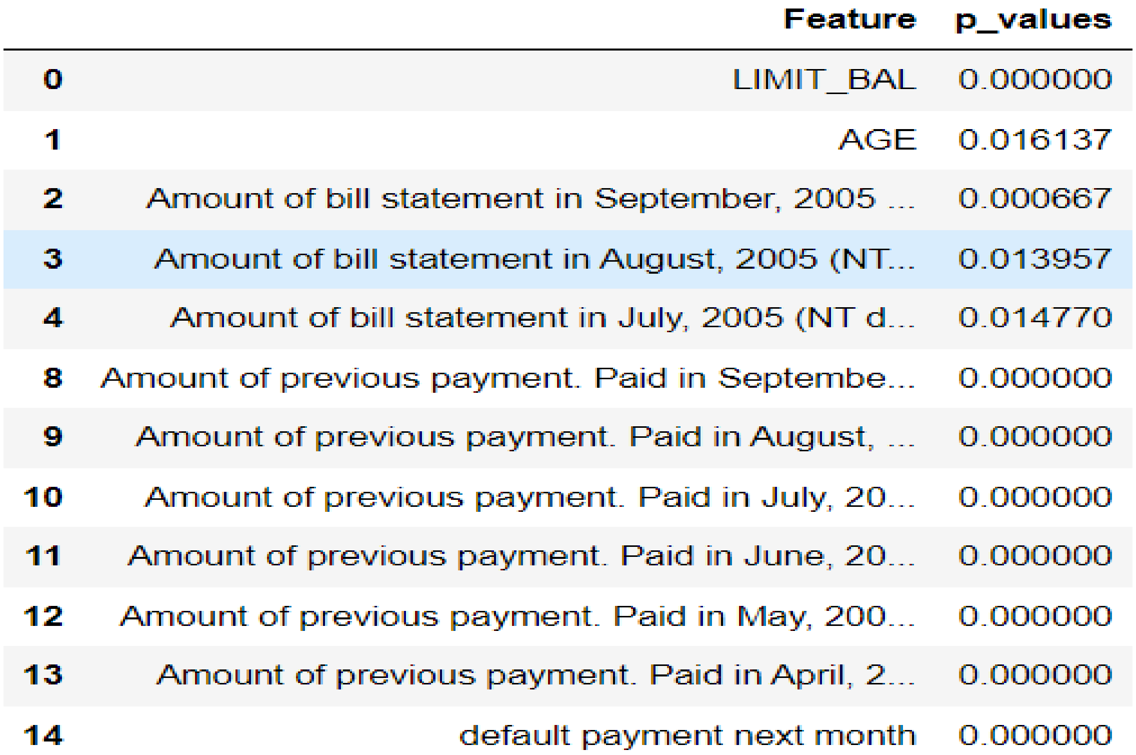
ANOVA Test

For numerical columns we perform parametric and non-parametric tests for the numerical columns. Under parametric test we perform ANOVA and under non-parametric test we perform Mann Whitney U test.

Hypothesis for numerical tests

H0 : Two samples have the same mean (i.e insignificant)

H1 : Two samples have different mean (i.e significant)



	Feature	p_values
0	LIMIT_BAL	0.000000
1	AGE	0.016137
2	Amount of bill statement in September, 2005 ...	0.000667
3	Amount of bill statement in August, 2005 (NT...	0.013957
4	Amount of bill statement in July, 2005 (NT d...	0.014770
8	Amount of previous payment. Paid in Septembe...	0.000000
9	Amount of previous payment. Paid in August, ...	0.000000
10	Amount of previous payment. Paid in July, 20...	0.000000
11	Amount of previous payment. Paid in June, 20...	0.000000
12	Amount of previous payment. Paid in May, 200...	0.000000
13	Amount of previous payment. Paid in April, 2...	0.000000
14	default payment next month	0.000000

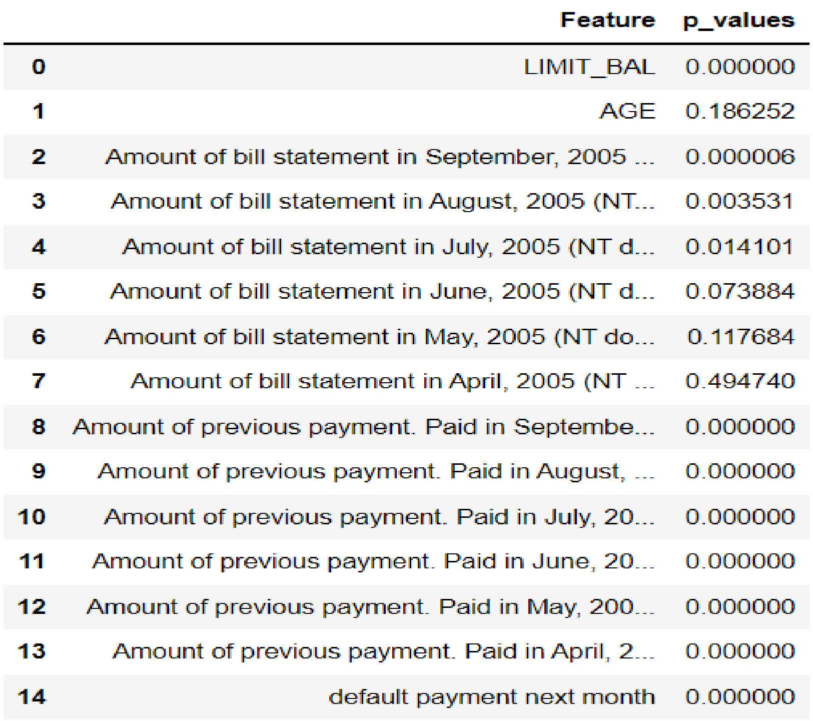
Since, we observe that the
p-values for some
variables are greater than 0.05. Hence these variables are insignificant.

SO, ANOVA tells
us all the variables are significant to our target.

Whitney U Test

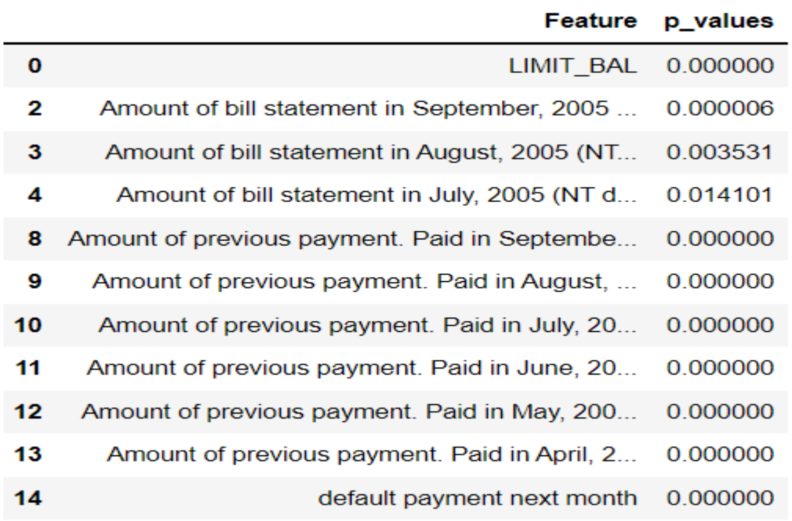
Hypothesis of Mann-Whitney U Test

H0 : Two samples have the same mean (i.e insignificant)
H1 : Two samples have different mean (i.e significant)



	Feature	p_values
0	LIMIT_BAL	0.000000
1	AGE	0.186252
2	Amount of bill statement in September, 2005 ...	0.000006
3	Amount of bill statement in August, 2005 (NT...	0.003531
4	Amount of bill statement in July, 2005 (NT d...	0.014101
5	Amount of bill statement in June, 2005 (NT d...	0.073884
6	Amount of bill statement in May, 2005 (NT do...	0.117684
7	Amount of bill statement in April, 2005 (NT ...	0.494740
8	Amount of previous payment. Paid in Septembe...	0.000000
9	Amount of previous payment. Paid in August, ...	0.000000
10	Amount of previous payment. Paid in July, 20...	0.000000
11	Amount of previous payment. Paid in June, 20...	0.000000
12	Amount of previous payment. Paid in May, 200...	0.000000
13	Amount of previous payment. Paid in April, 2...	0.000000
14	default payment next month	0.000000

From the above table we will only consider the variables having p-value less than 0.05.



	Feature	p_values
0	LIMIT_BAL	0.000000
2	Amount of bill statement in September, 2005 ...	0.000006
3	Amount of bill statement in August, 2005 (NT...	0.003531
4	Amount of bill statement in July, 2005 (NT d...	0.014101
8	Amount of previous payment. Paid in Septembe...	0.000000
9	Amount of previous payment. Paid in August, ...	0.000000
10	Amount of previous payment. Paid in July, 20...	0.000000
11	Amount of previous payment. Paid in June, 20...	0.000000
12	Amount of previous payment. Paid in May, 200...	0.000000
13	Amount of previous payment. Paid in April, 2...	0.000000
14	default payment next month	0.000000

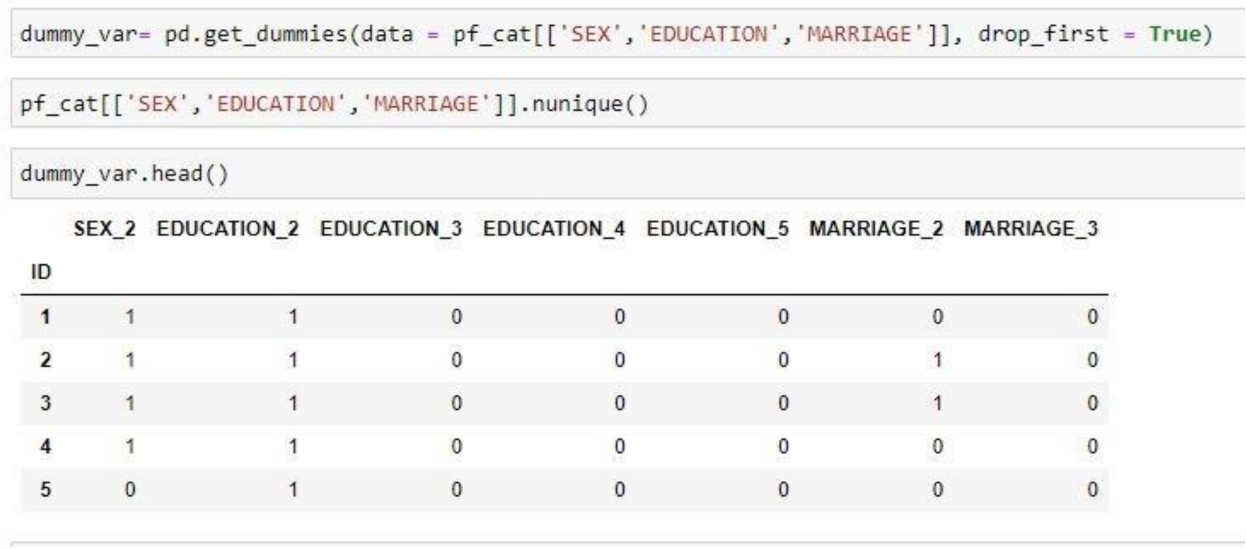
The above variable are the most significant variables.

BASE MODEL

Logistic regression model

We have selected Logistic Regression as our base model. For this we have encoded Education, Marriage and Sex using get dummies () and rest all the categorical variables using Label Encoder and we have scaled the numerical columns using Standard Scaler ()

Encoding:



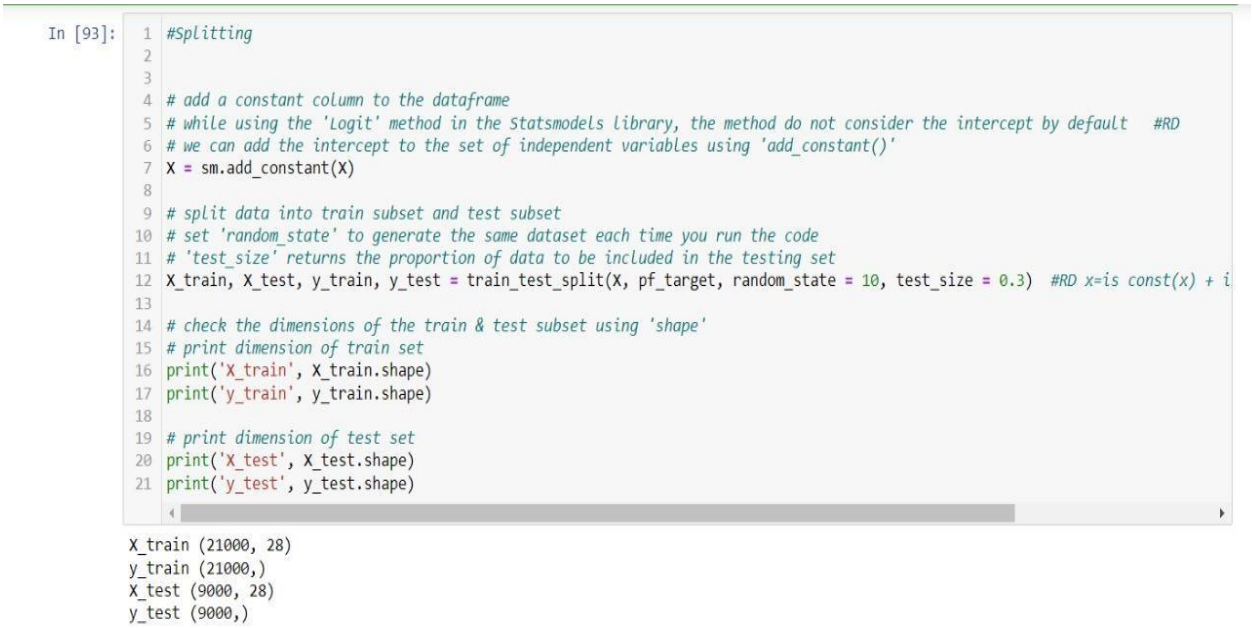
```
dummy_var= pd.get_dummies(data = pf_cat[['SEX','EDUCATION','MARRIAGE']], drop_first = True)
```

```
pf_cat[['SEX','EDUCATION','MARRIAGE']].nunique()
```

```
dummy_var.head()
```

	SEX_2	EDUCATION_2	EDUCATION_3	EDUCATION_4	EDUCATION_5	MARRIAGE_2	MARRIAGE_3
ID							
1	1	1	0	0	0	0	0
2	1	1	0	0	0	1	0
3	1	1	0	0	0	1	0
4	1	1	0	0	0	0	0
5	0	1	0	0	0	0	0

Splitting:

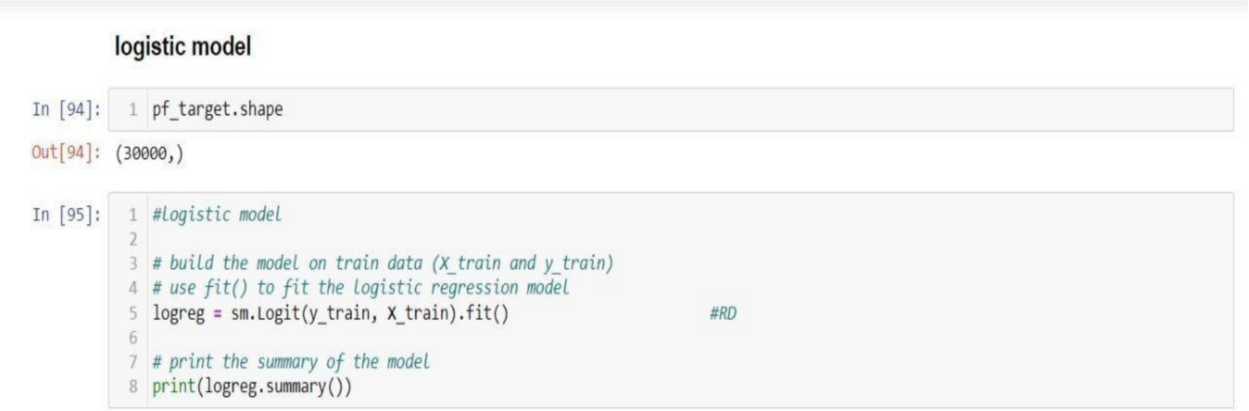


In [93]:

```
#Splitting


# add a constant column to the dataframe
# while using the 'Logit' method in the Statsmodels library, the method do not consider the intercept by default   #RD
# we can add the intercept to the set of independent variables using 'add_constant()'
X = sm.add_constant(X)

# split data into train subset and test subset
# set 'random_state' to generate the same dataset each time you run the code
# 'test_size' returns the proportion of data to be included in the testing set
X_train, X_test, y_train, y_test = train_test_split(X, pf_target, random_state = 10, test_size = 0.3)  #RD x=is const(x) + i

# check the dimensions of the train & test subset using 'shape'
# print dimension of train set
print('X_train', X_train.shape)
print('y_train', y_train.shape)

# print dimension of test set
print('X_test', X_test.shape)
print('y_test', y_test.shape)
```

```
X_train (21000, 28)
y_train (21000,)
X_test (9000, 28)
y_test (9000,)
```

Model Training:



logistic model

In [94]:

```
pf_target.shape
```

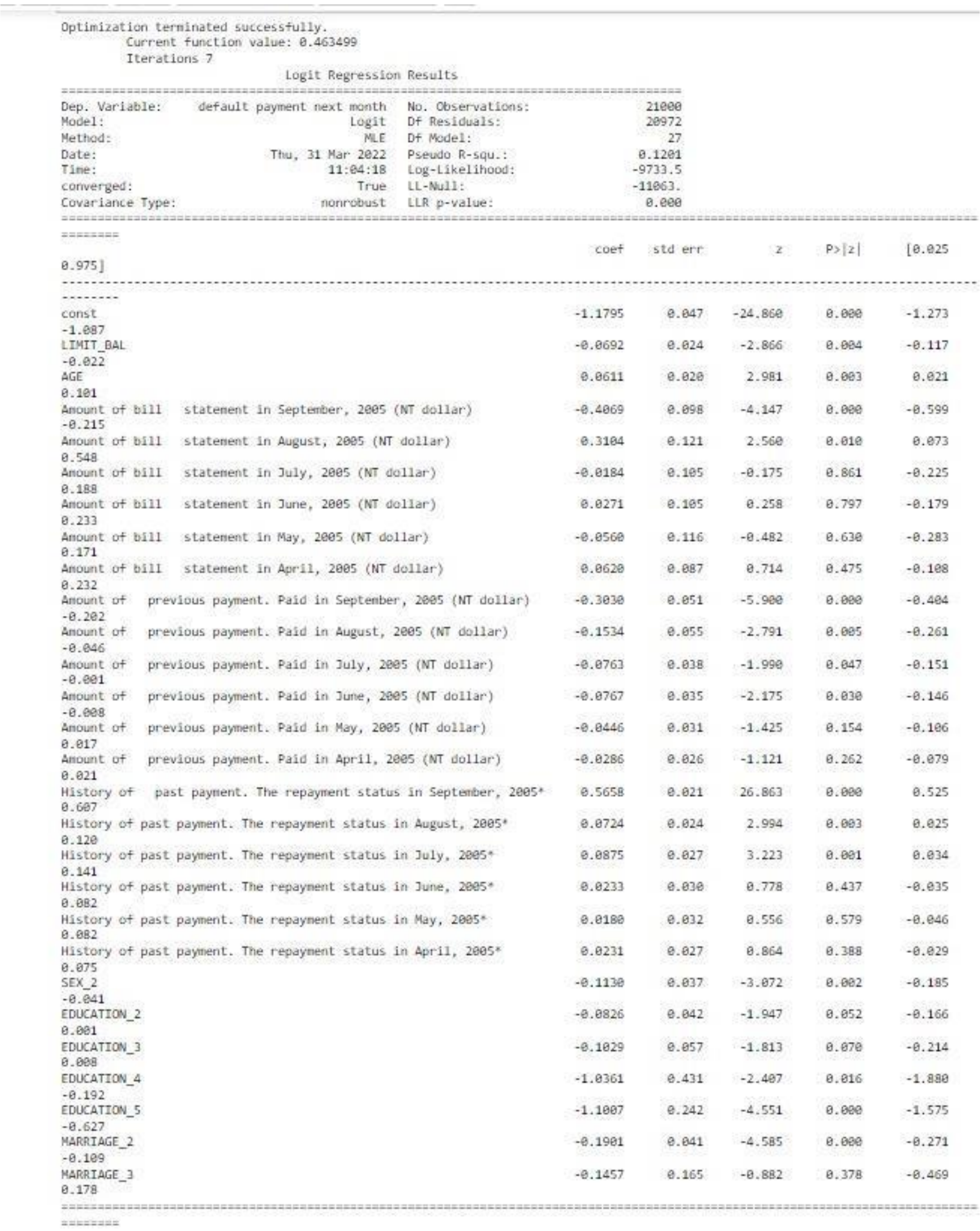
Out[94]:

```
(30000,)
```

In [95]:

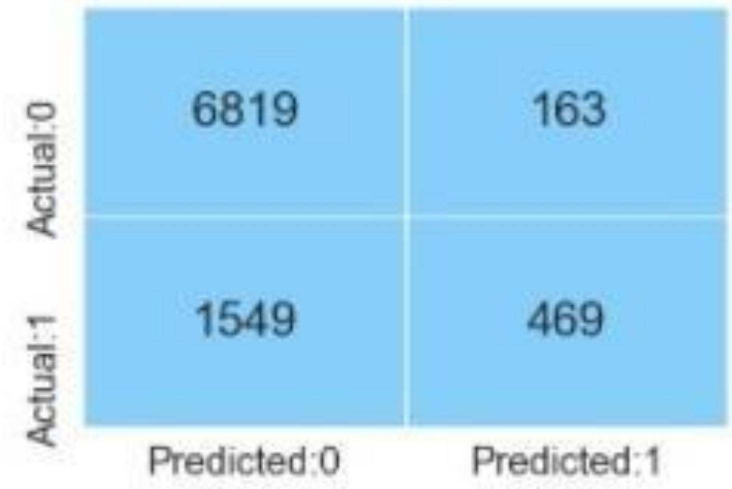
```
#logistic model

# build the model on train data (X_train and y_train)
# use fit() to fit the logistic regression model
logreg = sm.Logit(y_train, X_train).fit()                    #RD

# print the summary of the model
print(logreg.summary())
```

Model Summary:



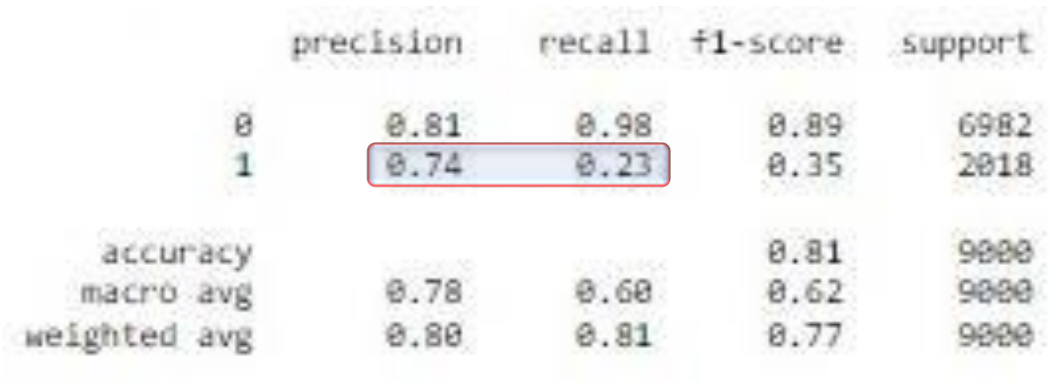
```
Optimization terminated successfully.
         Current function value: 0.463499
         Iterations 7
                               Logit Regression Results
======================================================================================
Dep. Variable:     default payment next month   No. Observations:                21000
Model:                                  Logit   Df Residuals:                    20972
Method:                                   MLE   Df Model:                           27
Date:                        Thu, 31 Mar 2022   Pseudo R-squ.:                  0.1201
Time:                                11:04:18   Log-Likelihood:                -9733.5
converged:                               True   LL-Null:                       -11063.
Covariance Type:                    nonrobust   LLR p-value:                     0.000
================================================================================================================================
========
                                                                          coef    std err          z      P>|z|      [0.025
0.975]
--------------------------------------------------------------------------------------------------------------------------------
--------
const                                                                  -1.1795      0.047    -24.860      0.000      -1.273
-1.087
LIMIT_BAL                                                              -0.0692      0.024     -2.866      0.004      -0.117
-0.022
AGE                                                                     0.0611      0.020      2.981      0.003       0.021
0.101
Amount of bill   statement in September, 2005 (NT dollar)              -0.4069      0.098     -4.147      0.000      -0.599
-0.215
Amount of bill   statement in August, 2005 (NT dollar)                  0.3104      0.121      2.560      0.010       0.073
0.548
Amount of bill   statement in July, 2005 (NT dollar)                   -0.0184      0.105     -0.175      0.861      -0.225
0.188
Amount of bill   statement in June, 2005 (NT dollar)                    0.0271      0.105      0.258      0.797      -0.179
0.233
Amount of bill   statement in May, 2005 (NT dollar)                    -0.0560      0.116     -0.482      0.630      -0.283
0.171
Amount of bill   statement in April, 2005 (NT dollar)                   0.0620      0.087      0.714      0.475      -0.108
0.232
Amount of   previous payment. Paid in September, 2005 (NT dollar)      -0.3030      0.051     -5.900      0.000      -0.404
-0.202
Amount of   previous payment. Paid in August, 2005 (NT dollar)         -0.1534      0.055     -2.791      0.005      -0.261
-0.046
Amount of   previous payment. Paid in July, 2005 (NT dollar)           -0.0763      0.038     -1.990      0.047      -0.151
-0.001
Amount of   previous payment. Paid in June, 2005 (NT dollar)           -0.0767      0.035     -2.175      0.030      -0.146
-0.008
Amount of   previous payment. Paid in May, 2005 (NT dollar)            -0.0446      0.031     -1.425      0.154      -0.106
0.017
Amount of   previous payment. Paid in April, 2005 (NT dollar)          -0.0286      0.026     -1.121      0.262      -0.079
0.021
History of   past payment. The repayment status in September, 2005*     0.5658      0.021     26.863      0.000       0.525
0.607
History of past payment. The repayment status in August, 2005*          0.0724      0.024      2.994      0.003       0.025
0.120
History of past payment. The repayment status in July, 2005*            0.0875      0.027      3.223      0.001       0.034
0.141
History of past payment. The repayment status in June, 2005*            0.0233      0.030      0.778      0.437      -0.035
0.082
History of past payment. The repayment status in May, 2005*             0.0180      0.032      0.556      0.579      -0.046
0.082
History of past payment. The repayment status in April, 2005*           0.0231      0.027      0.864      0.388      -0.029
0.075
SEX_2                                                                  -0.1130      0.037     -3.072      0.002      -0.185
-0.041
EDUCATION_2                                                            -0.0826      0.042     -1.947      0.052      -0.166
0.001
EDUCATION_3                                                            -0.1029      0.057     -1.813      0.070      -0.214
0.008
EDUCATION_4                                                            -1.0361      0.431     -2.407      0.016      -1.880
-0.192
EDUCATION_5                                                            -1.1007      0.242     -4.551      0.000      -1.575
-0.627
MARRIAGE_2                                                             -0.1901      0.041     -4.585      0.000      -0.271
-0.109
MARRIAGE_3                                                             -0.1457      0.165     -0.882      0.378      -0.469
0.178
================================================================================================================================
========
```

Confusion Matrix:



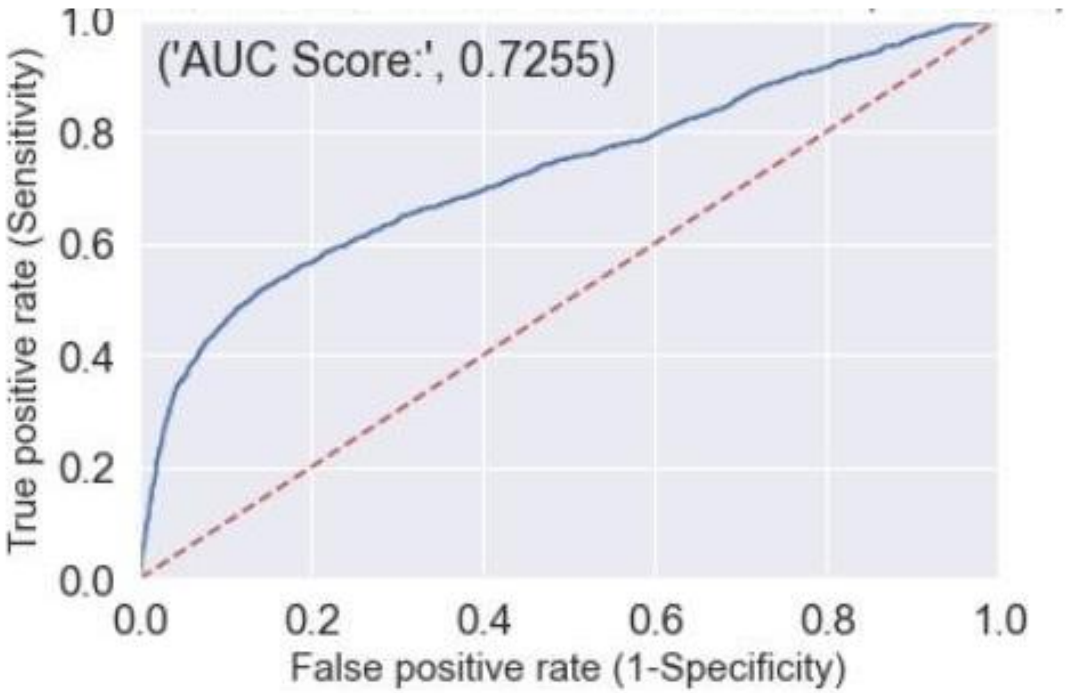
	Predicted:0	Predicted:1
Actual:0	6819	163
Actual:1	1549	469

Classification Report on Test Data:



```
              precision    recall  f1-score   support

           0       0.81      0.98      0.89      6982
           1       0.74      0.23      0.35      2018

    accuracy                           0.81      9000
   macro avg       0.78      0.60      0.62      9000
weighted avg       0.80      0.81      0.77      9000
```

ROC Curve:


('AUC Score:', 0.7255)
True positive rate (Sensitivity)
False positive rate (1-Specificity)
0.0
0.2
0.4
0.6
0.8
1.0
0.0
0.2
0.4
0.6
0.8
1.0

MODEL BUILDING AND METHODS

From EDA, we have observed that presence of cardinality in certain categorical variables. In order to build models, we need to use appropriate encoding techniques to address this issue.

Encoding of Categorical Variables and Numerical Values Treatment and Feature Engineering

Under feature engineering we perform the following-

1. We binned the age columns into 3 classes namely ‘young’, ‘middle’, ‘old’
2. For the columns of ‘history of past pavement’, we have assumed that every customer who had defaulted even for one month in the last 6 months is considered as ‘defaulter’, else non-defaulter and we have replaced it with 1 and 0 respectively and the same is stored in new column called ‘Outstanding’
3. We separated the numerical and categorical variables and performed the following
    - Numerical-We scaled the numerical data using ‘Robust Scaling method’ and then applied PCA(Principle Component Analysis) to reduce dimensions.
    - Categorical- Encoding using Get-Dummies

4.Before PCA we had 13 numerical variables, after PCA the number of dimensions got reduced to 7.

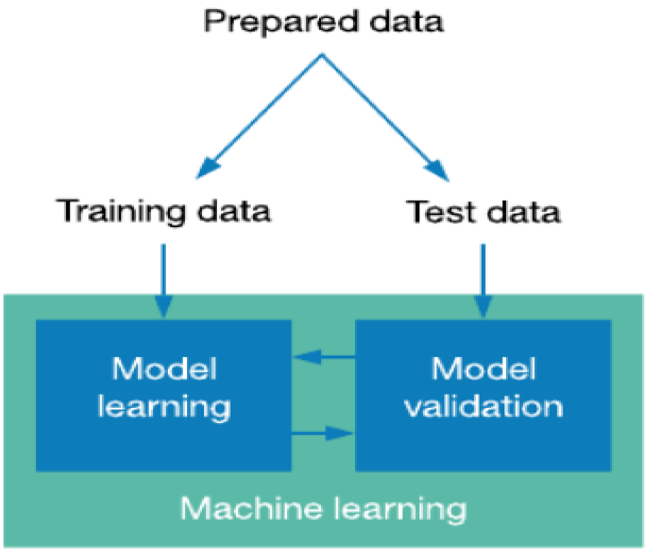
5. We selected 7 PCA’s because it was explaining 98.9570% variation of the data.Then we concatenated the 7 PCA Components with the encoded data and formed a new dataframe.

.

Model Building

Step by step approach for model building: -

1. After performing encoding for the categorical features and transforming the numerical variables, we split the data into train data and test data. Model data uses train data to learn whereas test data is used to evaluate or validate the trained model.



2. For some of the categorical variables we used dummy encoder and our initial models which we built were Logistic Regression and Decision Tree.

3.From these models, we did not achieve desired amount of accuracy, precision and recall Even though we achieve moderate level of accuracy for the model, we get low precision and recall value. since there is presence of high amount of class imbalance.

4.We performed Smote to balance the target variable. And we build non-linear models such as Decision Tree, Random Forest, KNN and XG Boost Classifier. For these models, we performed hyper parameter tuning. In addition ,we used Gaussian Naïve Bayes, Gradient Boost, Ada Boost algorithms.

Over-Sampling and Under-Sampling

An imbalanced classification problem is an example of a classification problem where the distribution of examples across the known classes is biased or skewed. The distribution can vary from a slight bias to a severe imbalance where there is one example in the minority class for hundreds, thousands, or millions of examples in the majority class or classes.

Imbalanced classifications pose a challenge for predictive modelling as most of the machine learning algorithms used for classification were designed around the assumption of an equal number of examples for each class. This results in models that have poor predictive performance, specifically for the minority class. This is a problem because typically, the minority class is more important and therefore the problem is more sensitive to classification errors for the minority class than the majority class.

Imbalanced classification refers to a classification predictive modelling problem where the number of examples in the training dataset for each class label is not balanced. That is, where the class distribution is not equal or close to equal, and is instead biased or skewed.

One approach to addressing the problem of class imbalance is to randomly resample the training dataset. The two main approaches to randomly resampling an imbalanced dataset are to delete examples from the majority class, called under-sampling, and to duplicate examples from the minority class, called over-sampling.

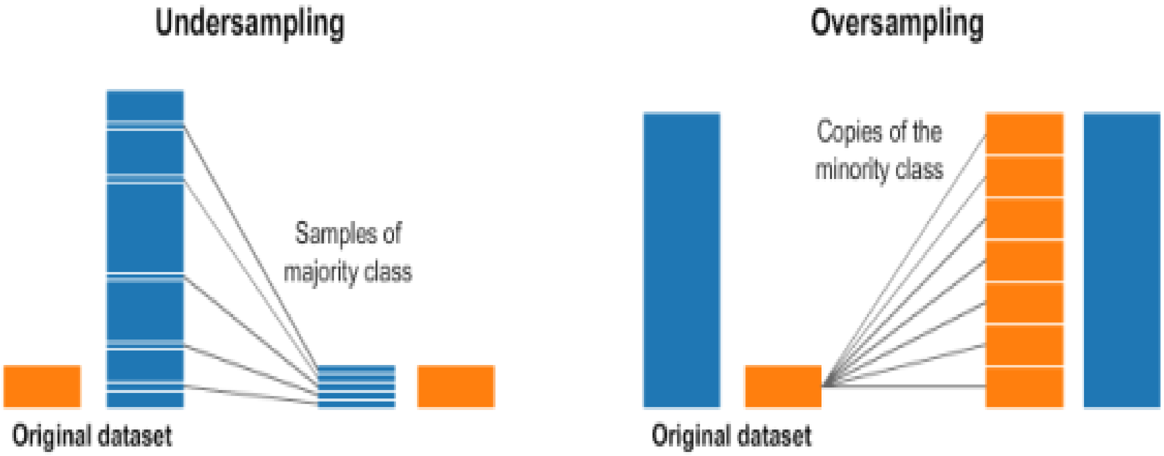
Random over-sampling involves randomly duplicating examples from the minority class and adding them to the training dataset.

Examples from the training dataset are selected randomly with replacement. This means that examples from the minority class can be chosen and added to the new “more balanced” training dataset multiple times; they are selected from the original training dataset, added to the new training dataset, and then returned or “replaced” in the original dataset, allowing them to be selected again.

In some cases, seeking a balanced distribution for a severely imbalanced dataset can cause affected algorithms to overfit the minority class, leading to increased generalization error. The effect can be better performance on the training dataset, but worse performance on the holdout or test dataset.

Random under-sampling involves randomly selecting examples from the majority class to delete from the training dataset.

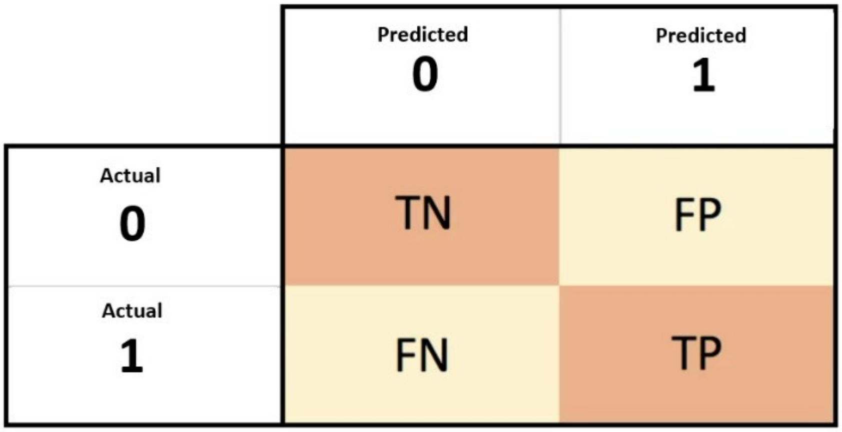
This has the effect of reducing the number of examples in the majority class in the transformed version of the training dataset. This process can be repeated until the desired class distribution is achieved, such as an equal number of examples for each class.



A limitation of under-sampling is that examples from the majority class are deleted that may be useful, important, or perhaps critical to fitting a robust decision boundary. Given that examples are deleted randomly, there is no way to detect or preserve “good” or more information-rich examples from the majority class.

Confusion Matrix :

In the field of machine learning and specifically the problem of statistical classification, a confusion matrix, also known as an error matrix, is a specific table layout that allows visualization of the performance of an algorithm, typically a supervised learning one.



	Predicted 0	Predicted 1
Actual 0	TN	FP
Actual 1	FN	TP

Precision and Recall Trade-Off

Precision: It is the accuracy of positive predictions.

$$Precision = \frac{True\ Positive}{True\ Postive + False\ Positive}$$

That mean, when the model predicts that a customer will be a defaulter, it is correct around %precision times.

Recall: It is the ratio of positive instance that are correctly detected. It is also called sensitivity.

$$Recall = \frac{True\ Positive}{True\ Postive + False\ Negative}$$

Hence, for all the customers that who were actually defaulters, recall tells us how many the model correctly identified as defaulters.

Accuracy: Accuracy is the ratio of the total number of correct predictions and the total number of predictions.

$$Accuracy = \frac{True\ Positive}{True\ Postive + False\ Negative + False\ Positive + False\ Negative}$$

Using accuracy as a defining metric for our model does make sense intuitively, but more often than not, it is always advisable to use Precision and Recall too. There might be other situations where our accuracy is very high, but our precision or recall is low.

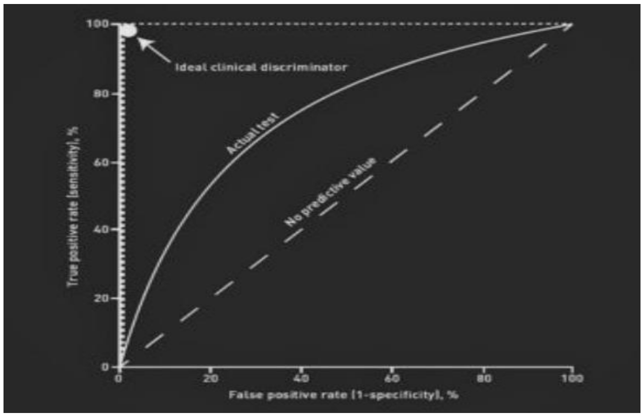
F1-score: F1-score is the Harmonic mean of the Precision and Recall.

$$F1 - Score = \frac{2 * Precision * Recall}{Precision + Recall}$$

Roc Curve:

A receiver operating characteristic curve, or ROC curve, is a graphical plot that illustrates the diagnostic ability of a binary classifier system as its discrimination threshold is varied.

AUC stands for "Area under the ROC Curve." That is, AUC measures the entire two-dimensional area underneath the entire ROC curve (like integral calculus) from (0,0) to (1,1).



Ensemble Methods

Ensemble learning helps improve machine learning results by combining several models. This approach allows the production of better predictive performance compared to a single model. Basic idea is to learn a set of classifiers (experts) and to allow them to vote. Bagging and Boosting are two types of Ensemble Learning. These two decrease the variance of a single estimate as they combine several estimates from different models. So, the result may be a model with higher stability. Let's understand these two terms in a glimpse.

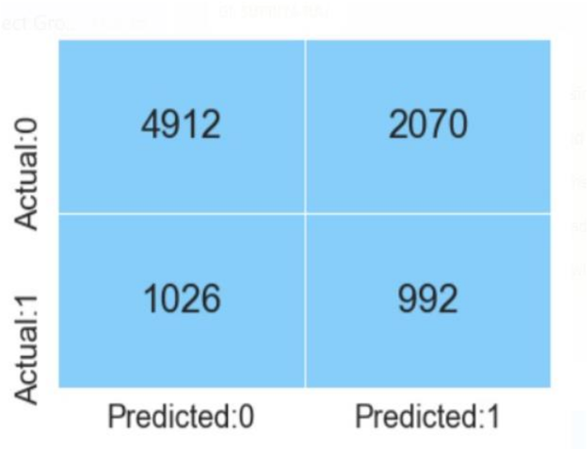
Bagging: It is a homogeneous weak learners' model that learns from each other independently in parallel and combines them for determining the model average. Some of the examples of bagging Decision tree random forest, knn, gnb

Boosting: It is also a homogeneous weak learners' model but works differently from Bagging. In this model, learners learn sequentially and adaptively to improve model predictions of a learning algorithm. Some of the examples of boosting Adaptive Boosting, Gradient boosting, Extreme Gradient Boosting

Models building

1.Decision Tree Model:

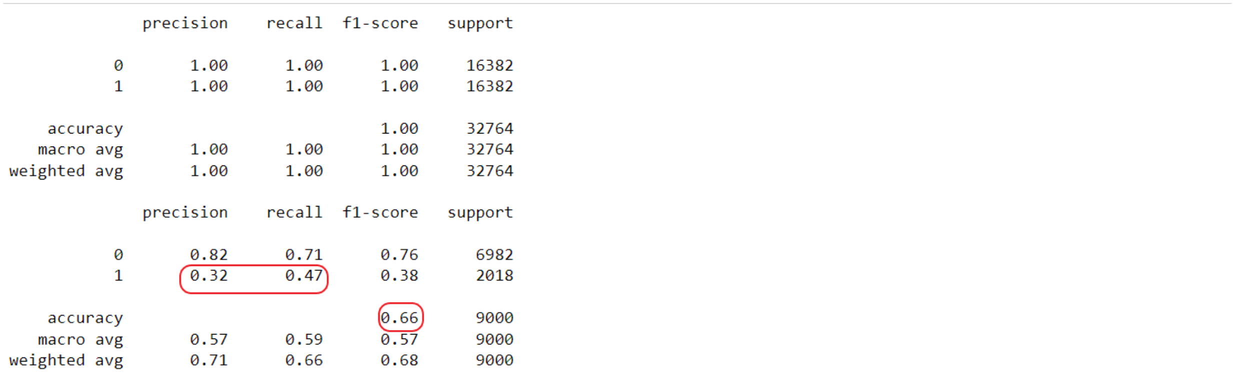
a)Confusion Matrix: As we can see from below total of 5904 data points are predicted correctly and around 3096 data points are wrongly predicted.



	Predicted:0	Predicted:1
Actual:0	4912	2070
Actual:1	1026	992

b) Train and Test Classification Report:

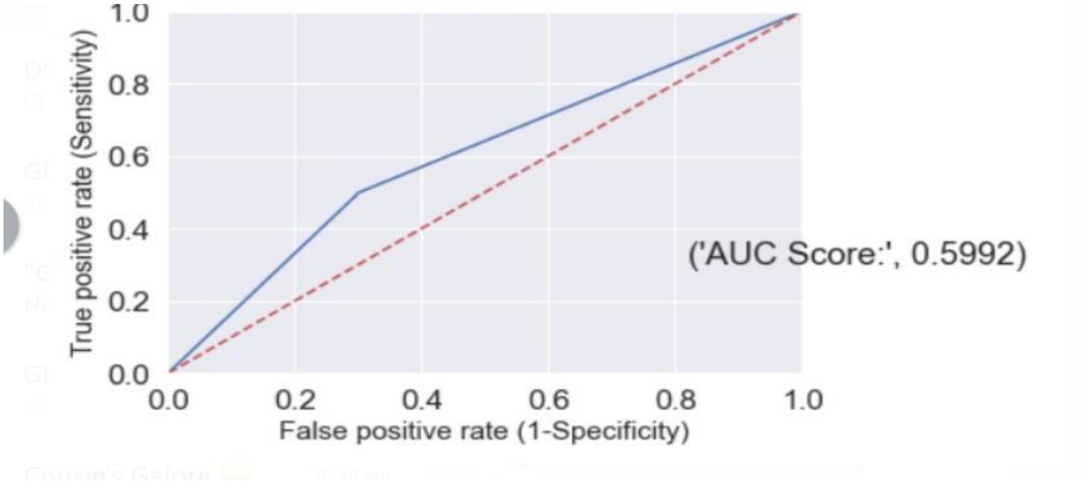
As we can see from below report the train results are perfect ,i.e 1, but the test results shows only 0.66 overall accuracy and very low precision and recall(0.32 and 0.47 respectively) which is no where near good and we can say model is overfitting on train data.



```
              precision    recall  f1-score   support

           0       1.00      1.00      1.00     16382
           1       1.00      1.00      1.00     16382

    accuracy                           1.00     32764
   macro avg       1.00      1.00      1.00     32764
weighted avg       1.00      1.00      1.00     32764

              precision    recall  f1-score   support

           0       0.82      0.71      0.76      6982
           1       0.32      0.47      0.38      2018

    accuracy                           0.66      9000
   macro avg       0.57      0.59      0.57      9000
weighted avg       0.71      0.66      0.68      9000
```

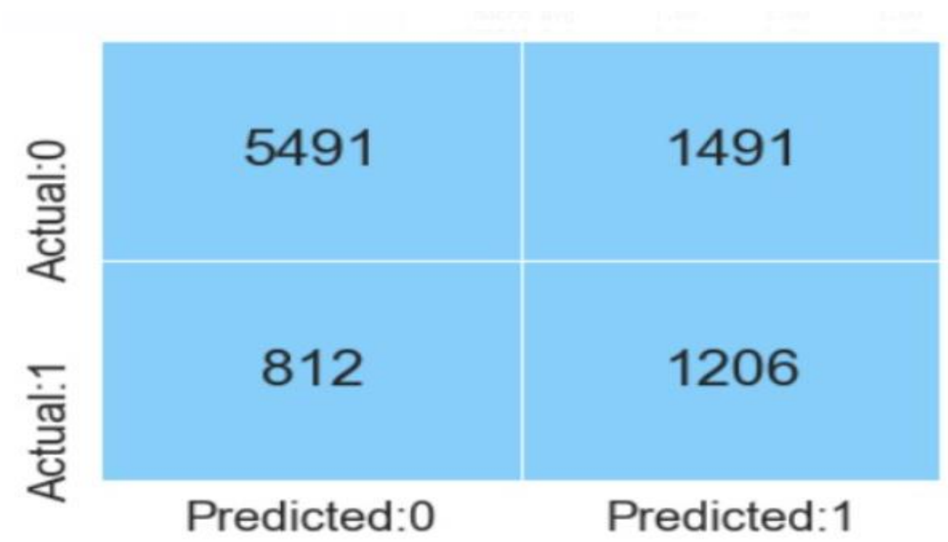
c)Roc Curve :

From above graph , we can see that the blue line (which represents our model) is very close to red line. Ideally, the blue line should be as far away from red line as possible to call the model as good model. And the AUC score has a limit of range between 0 and 1, the closer to 1 the better the model is . But here we can see that AUC score is just 0.59.



2. Decision Tree Model with Grid Search :

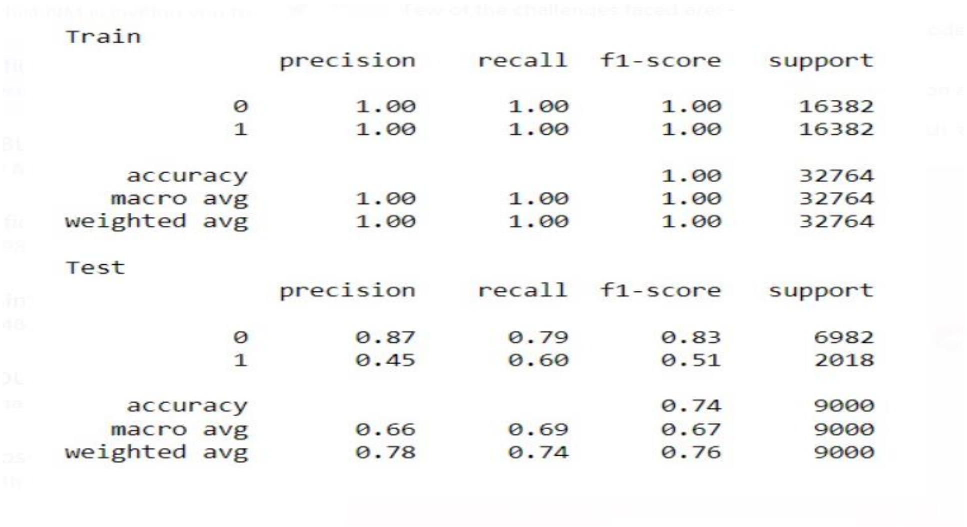
a)Confusion Matrix: As we can see from below total of 6697 data points are predicted correctly and around 2303 data points are wrongly predicted.



	Predicted:0	Predicted:1
Actual:0	5491	1491
Actual:1	812	1206

b) Train and Test Classification Report:

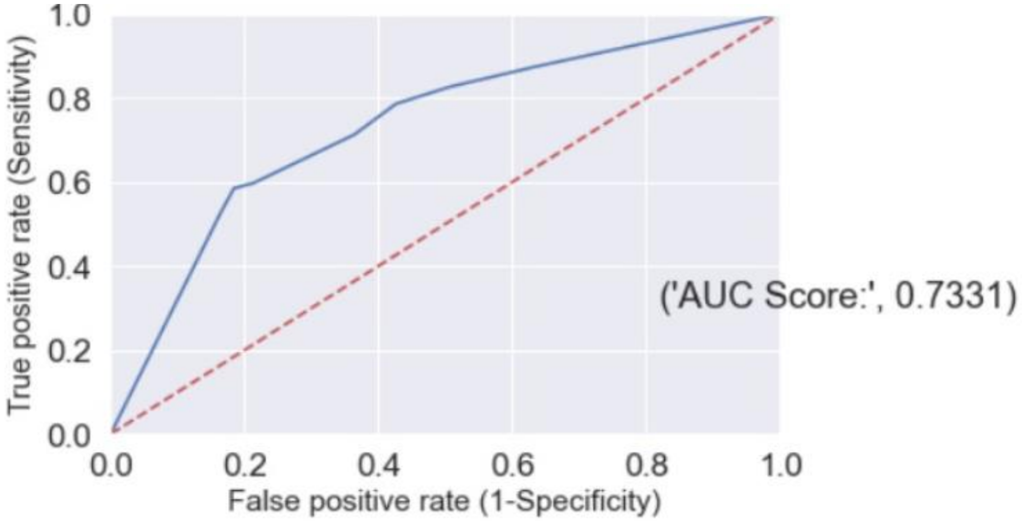
As we can see from below report the train results are perfect ,i.e 1, but the test results shows only 0.74 overall accuracy, and precision and recall(0.60 and 0.79 respectively) which is moderately good. and we can say model is overfitting on train data.



```
Train
              precision    recall  f1-score   support

           0       1.00      1.00      1.00     16382
           1       1.00      1.00      1.00     16382

    accuracy                           1.00     32764
   macro avg       1.00      1.00      1.00     32764
weighted avg       1.00      1.00      1.00     32764

Test
              precision    recall  f1-score   support

           0       0.87      0.79      0.83      6982
           1       0.45      0.60      0.51      2018

    accuracy                           0.74      9000
   macro avg       0.66      0.69      0.67      9000
weighted avg       0.78      0.74      0.76      9000
```

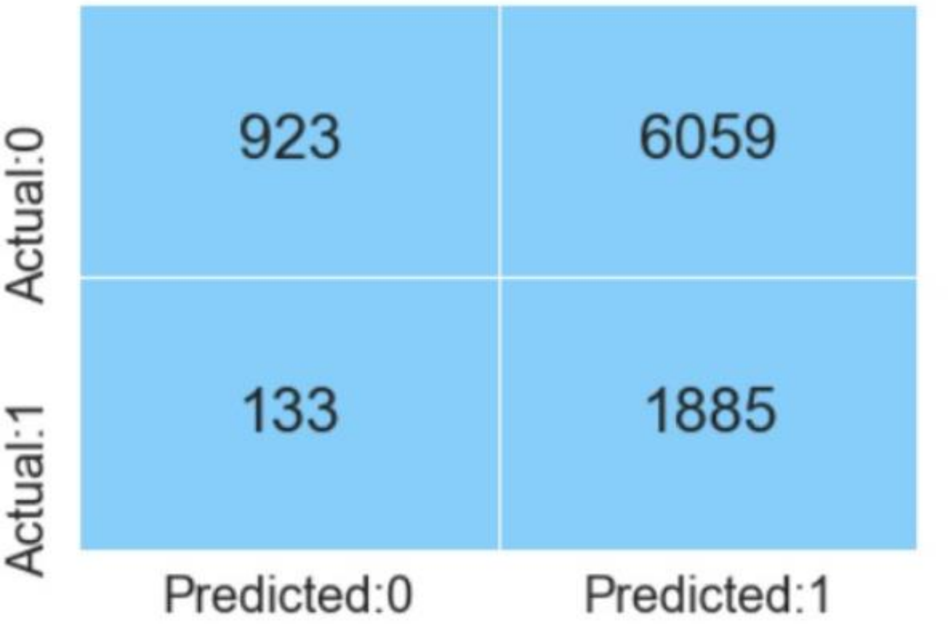
c)ROC Curve:

From below graph, we can see that the blue line (Actual Test) is farther away from no predictive value line i.e. (red line) compared to last model, and the AUC score is 0.7331 which is also greater than last model. Hence this model is better than previous model.



Random Forest:

a) Confusion Matrix: As we can see from below total of 2808 data points are predicted correctly and around 6192 data points are wrongly predicted.

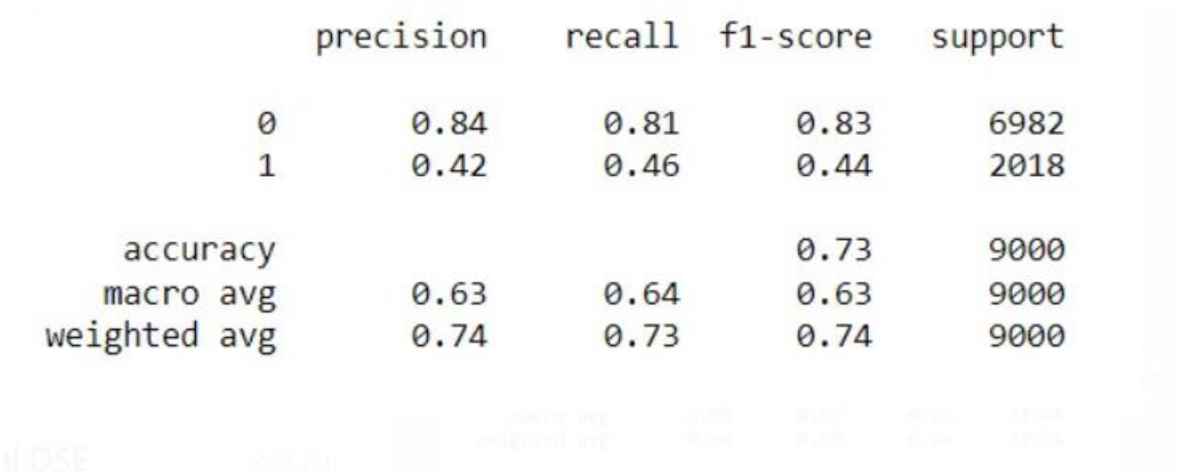


	Predicted:0	Predicted:1
Actual:0	923	6059
Actual:1	133	1885

b) Test Classification Report:

As we can see from below report the test results, and precision and recall(0.42 and 0.46 respectively) But the accuracy is 0.73.

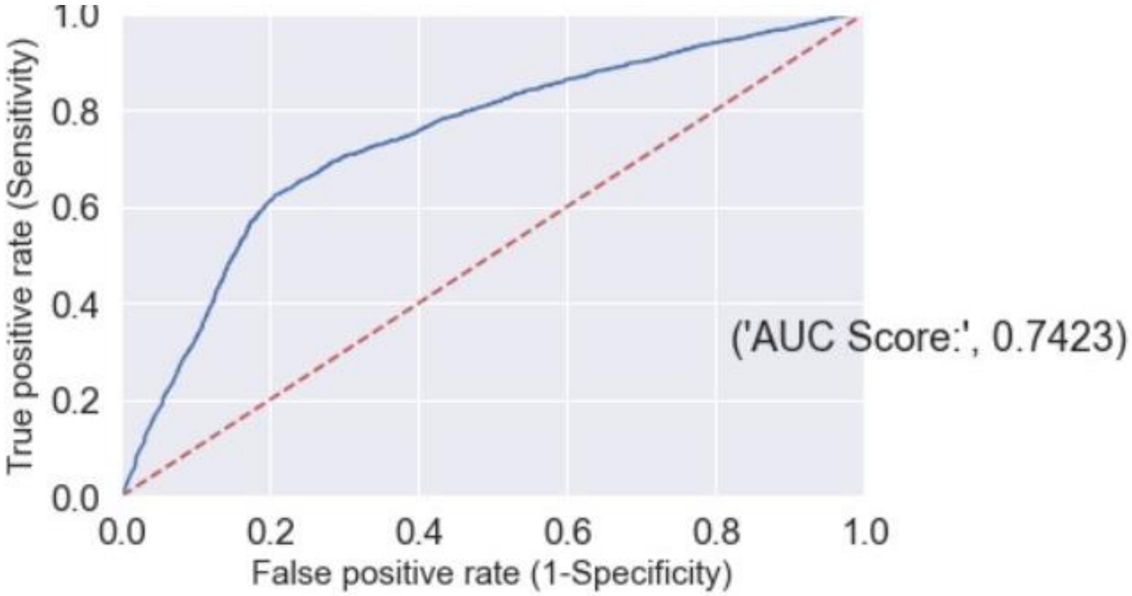
Test-score



```
              precision    recall  f1-score   support

           0       0.84      0.81      0.83      6982
           1       0.42      0.46      0.44      2018

    accuracy                           0.73      9000
   macro avg       0.63      0.64      0.63      9000
weighted avg       0.74      0.73      0.74      9000
```

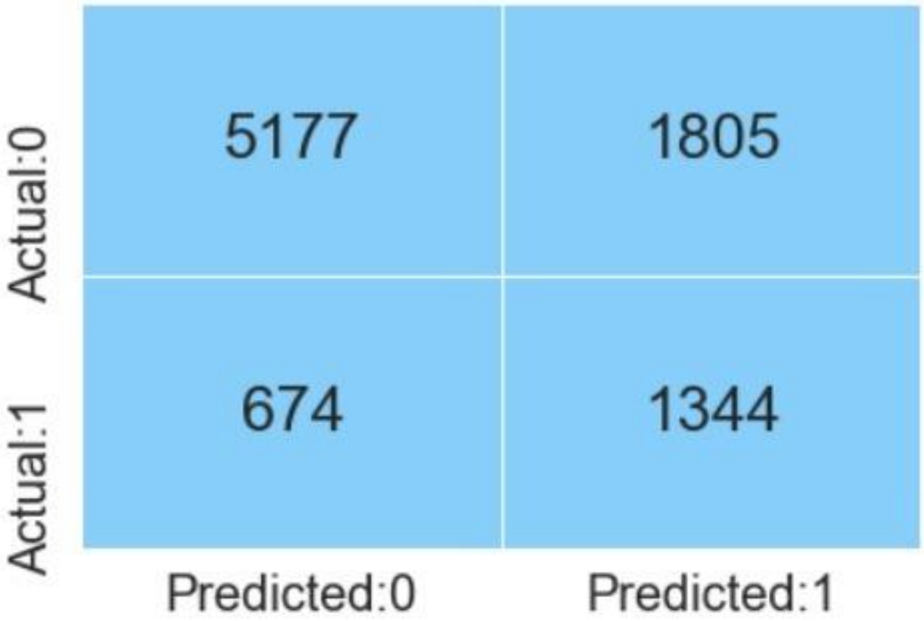
c)ROC Curve:

From below graph, we can see that the blue line (Actual Test) is farther away from no predictive value line i.e. (red line) and the AUC score is 0.7423 which is also greater than last model.



Random Forest with Grid Search :

a) Confusion Matrix: As we can see from below total of 6521 data points are predicted correctly and around 2479 data points are wrongly predicted.

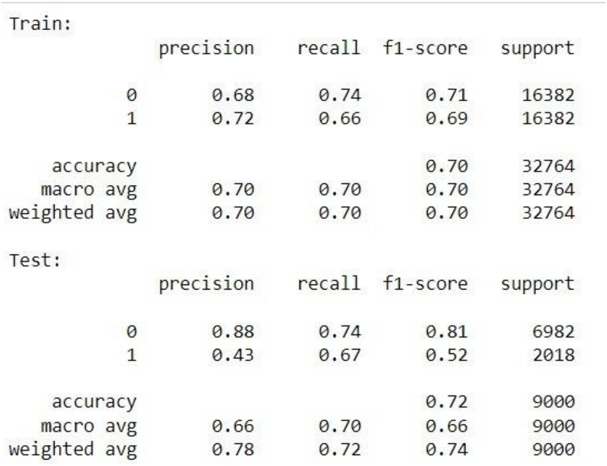


	Predicted:0	Predicted:1
Actual:0	5177	1805
Actual:1	674	1344

b) Train and Test Classification Report:

As we can see from below report the train results and precision and recall(0.72 and 0.66 respectively) But the accuracy is 0.70.

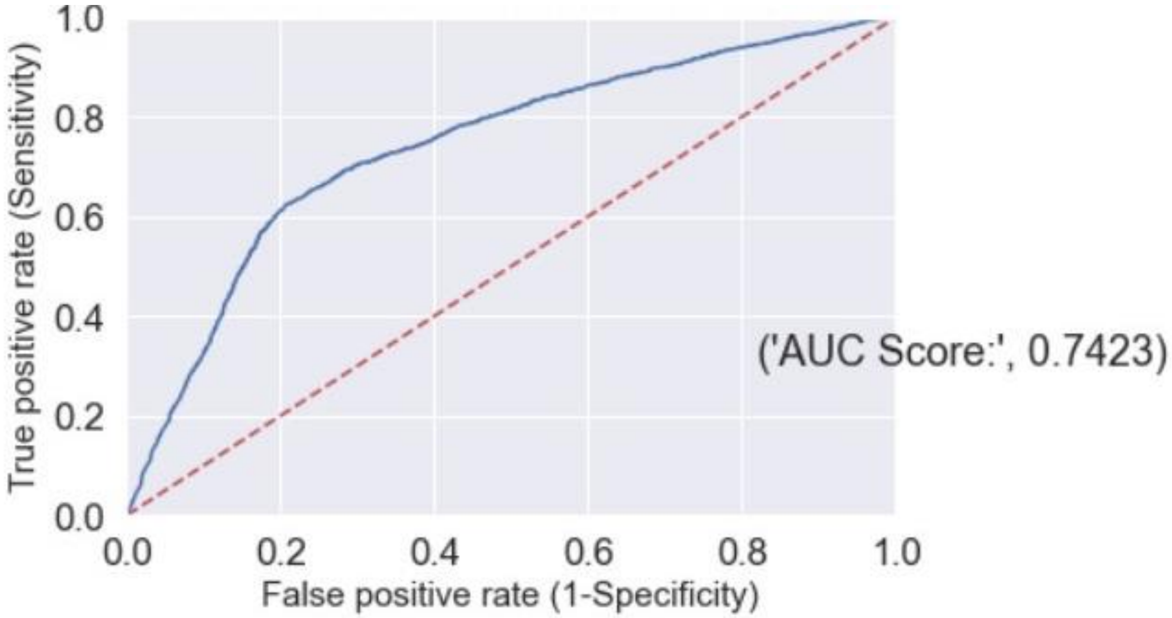
As we can see from below report the test results and precision and recall(0.43 and 0.67 respectively) But the accuracy is 0.72.



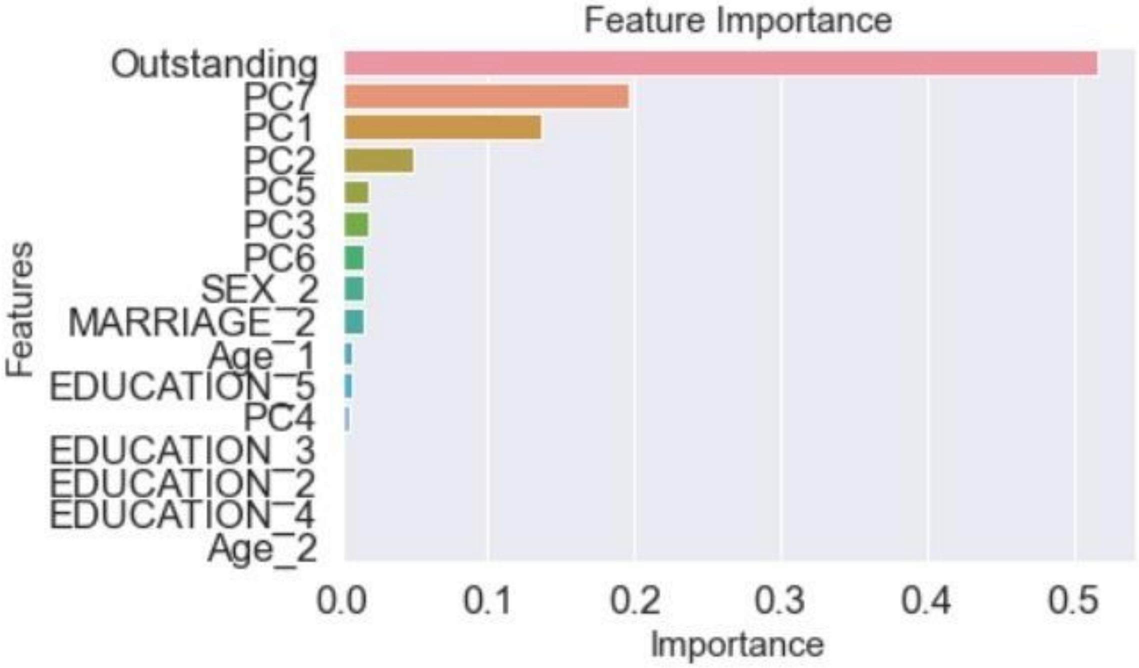
```
Train:
              precision    recall  f1-score   support

           0       0.68      0.74      0.71     16382
           1       0.72      0.66      0.69     16382

    accuracy                           0.70     32764
   macro avg       0.70      0.70      0.70     32764
weighted avg       0.70      0.70      0.70     32764

Test:
              precision    recall  f1-score   support

           0       0.88      0.74      0.81      6982
           1       0.43      0.67      0.52      2018

    accuracy                           0.72      9000
   macro avg       0.66      0.70      0.66      9000
weighted avg       0.78      0.72      0.74      9000
```

c)ROC Curve:

From below graph, we can see that the blue line (Actual Test) is farther away from no predictive value line i.e. (red line) and the AUC score is 0.7423 which is also greater than last model.

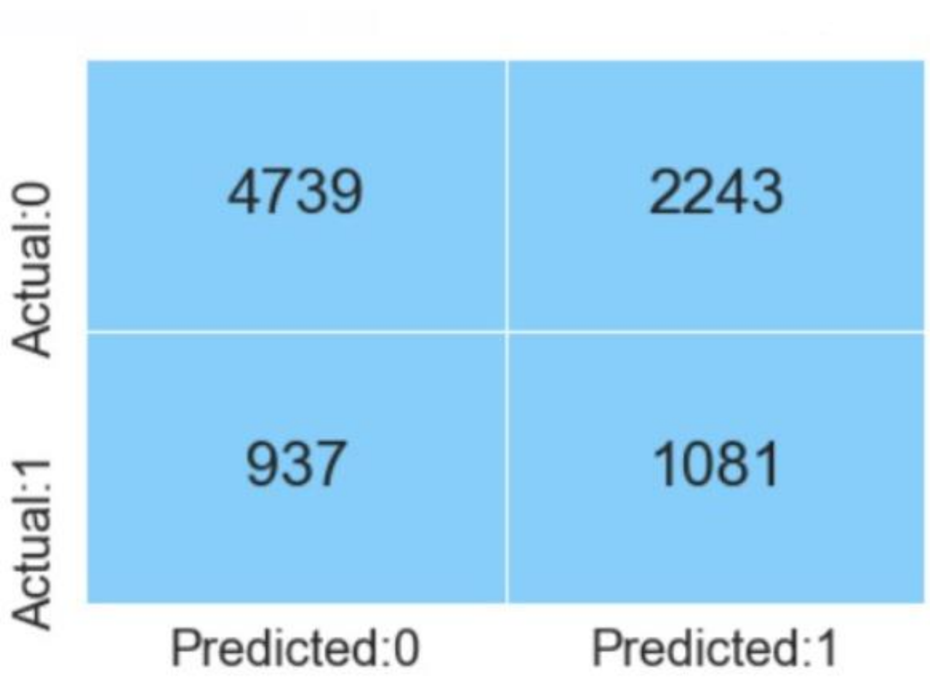


d) as we can see from the horizontal bar plot 'Outstanding 'variable is most important feature



KNN

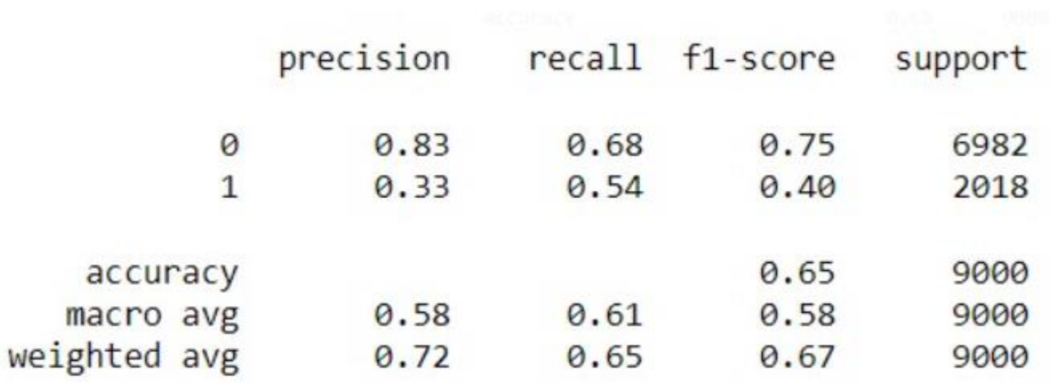
a) Confusion Matrix: As we can see from below total of 5820 data points are predicted correctly and around 3180 data points are wrongly predicted.



	Predicted:0	Predicted:1
Actual:0	4739	2243
Actual:1	937	1081

b) Test Classification Report:

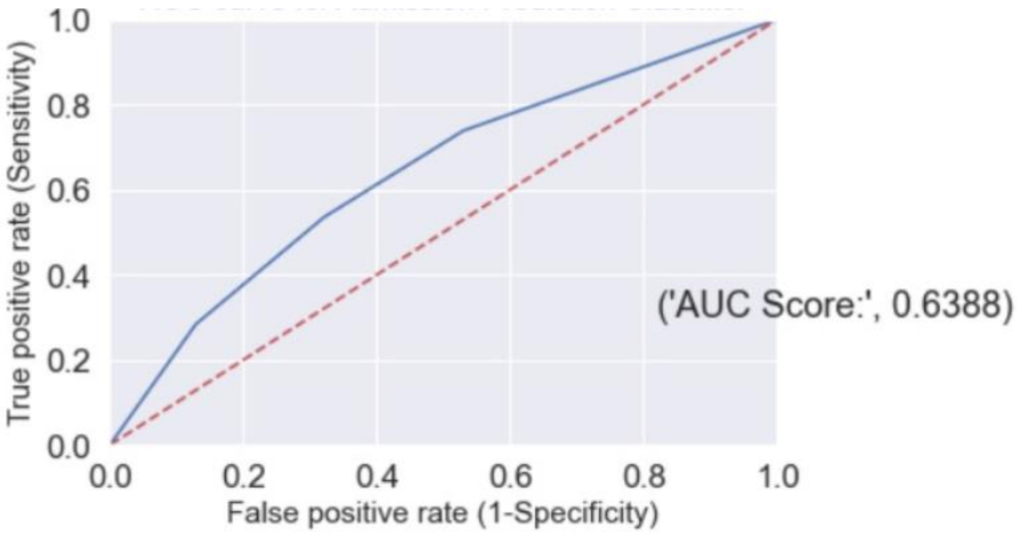
As we can see from below report the test results, precision and recall(0.33 and 0.54 respectively) and the accuracy is 0.65.



```
              precision    recall  f1-score   support

           0       0.83      0.68      0.75      6982
           1       0.33      0.54      0.40      2018

    accuracy                           0.65      9000
   macro avg       0.58      0.61      0.58      9000
weighted avg       0.72      0.65      0.67      9000
```

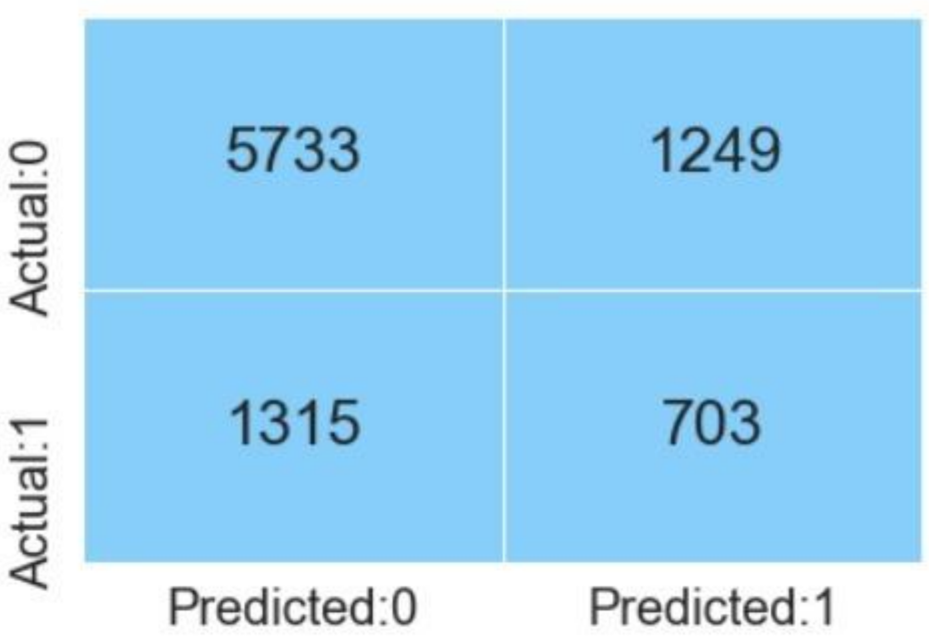
c)ROC Curve:

From below graph, we can see that the AUC score is 0.6388.which is less than the last model.



KNN with grid

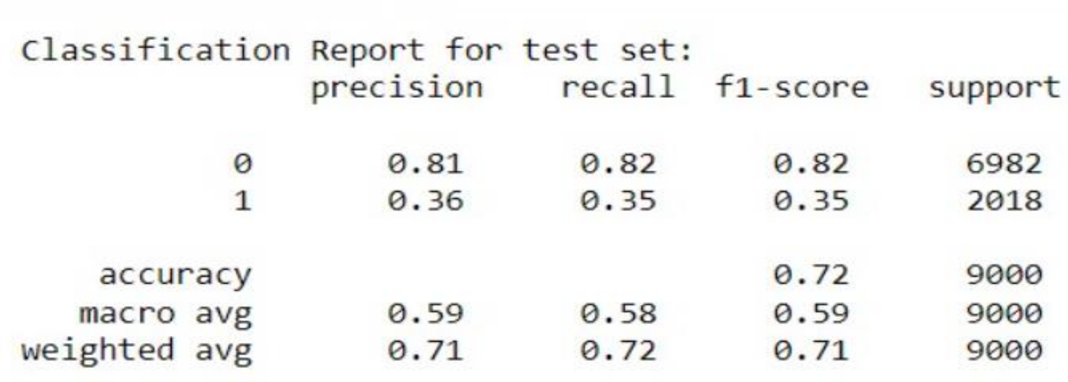
a) Confusion Matrix: As we can see from below total of 6436 data points are predicted correctly and around 2564 data points are wrongly predicted.



	Predicted:0	Predicted:1
Actual:0	5733	1249
Actual:1	1315	703

b) Test Classification Report:

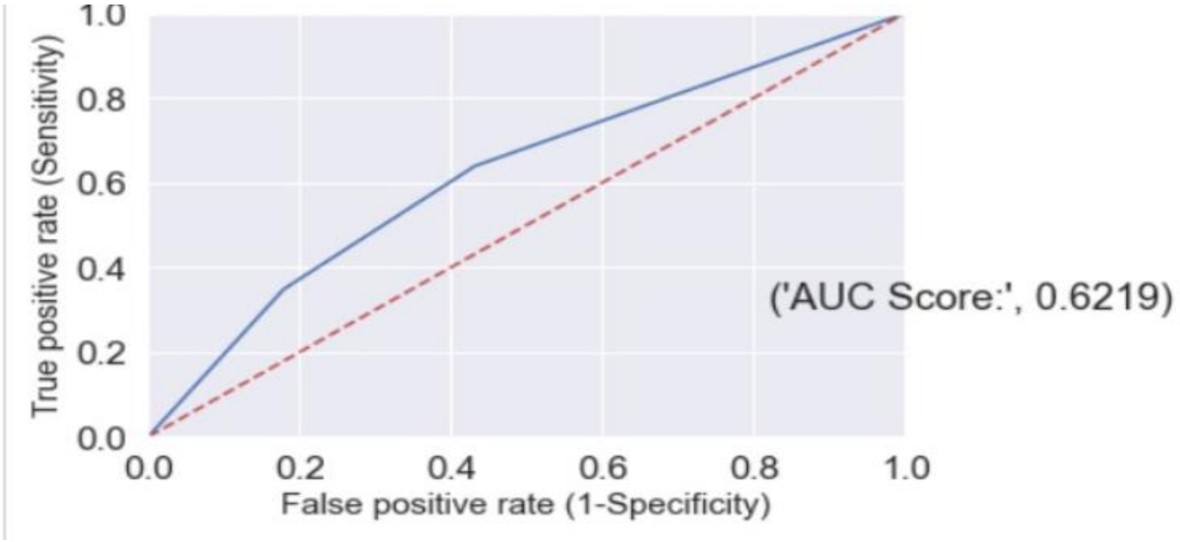
As we can see from below report the test results, precision and recall(0.36 and 0.35 respectively) and the accuracy is 0.72.



```
Classification Report for test set:
              precision    recall  f1-score   support

           0       0.81      0.82      0.82      6982
           1       0.36      0.35      0.35      2018

    accuracy                           0.72      9000
   macro avg       0.59      0.58      0.59      9000
weighted avg       0.71      0.72      0.71      9000
```

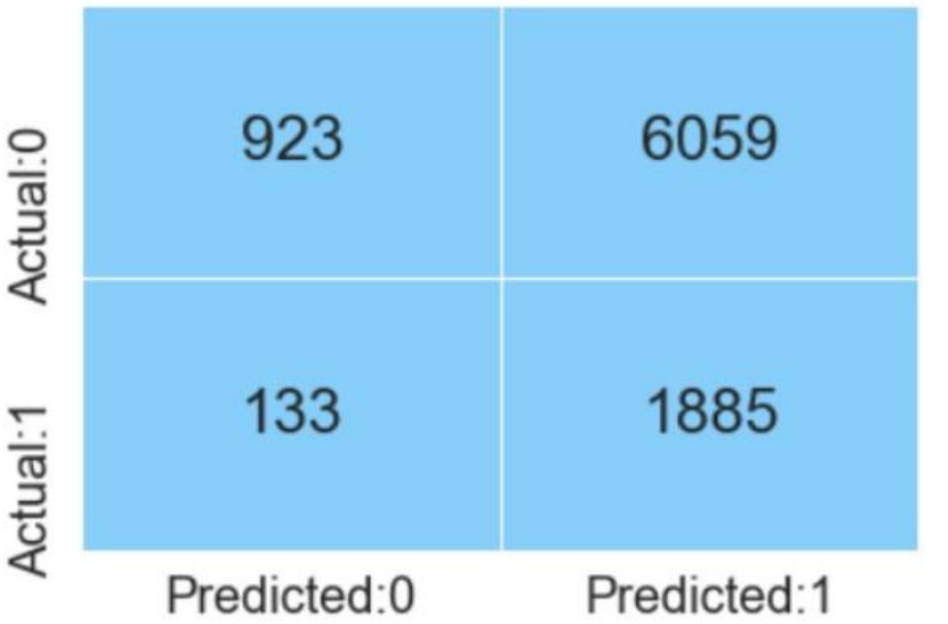
c)ROC Curve:

From below graph, we can see that the AUC score is 0.6219.which is less than the last model.



GNB

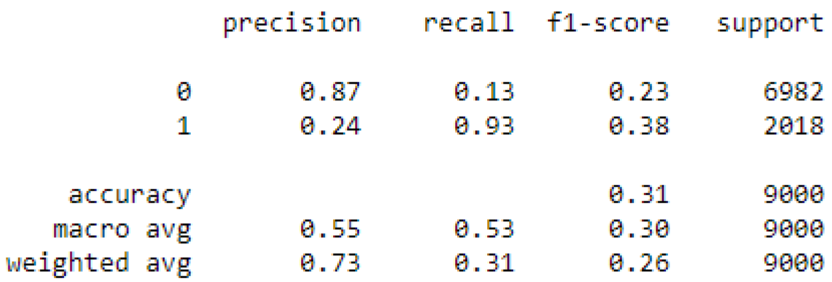
a) Confusion Matrix: As we can see from below total of 2808 data points are predicted correctly and around 6192 data points are wrongly predicted



	Predicted:0	Predicted:1
Actual:0	923	6059
Actual:1	133	1885

b) Test Classification Report:

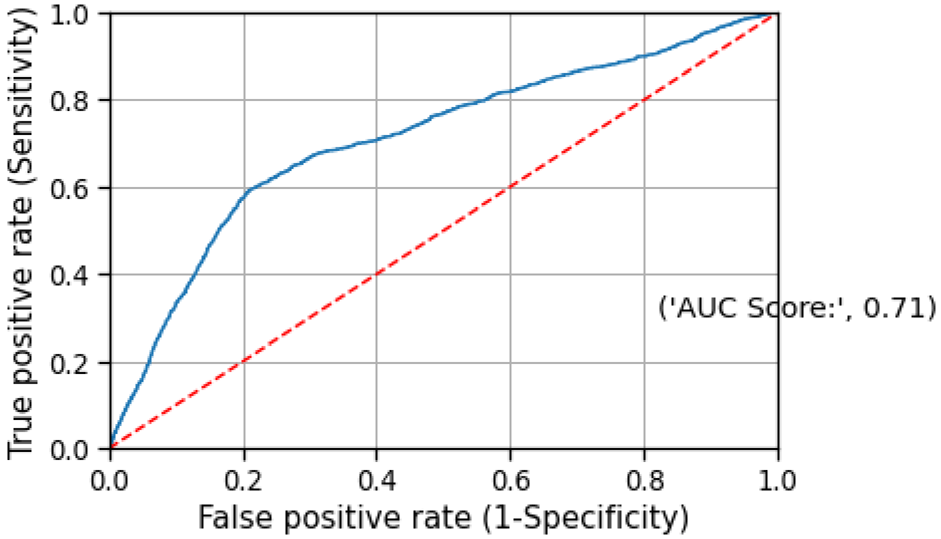
As we can see from below report the test results, precision and recall(0.24 and 0.93 respectively) and the accuracy is 0.31.



```
              precision    recall  f1-score   support

           0       0.87      0.13      0.23      6982
           1       0.24      0.93      0.38      2018

    accuracy                           0.31      9000
   macro avg       0.55      0.53      0.30      9000
weighted avg       0.73      0.31      0.26      9000
```

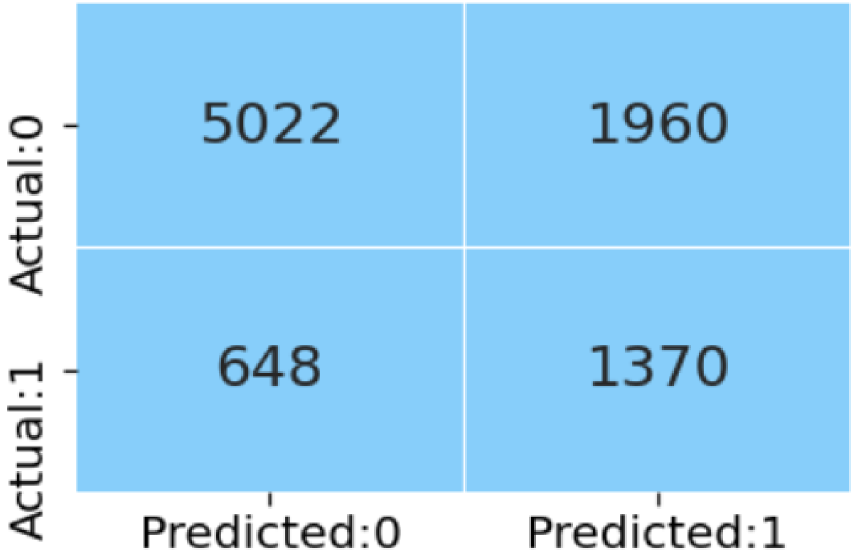
c)ROC Curve:

From below graph, we can see that the AUC score is 0.71.which is more than the last model.



adaboosting

a) Confusion Matrix: As we can see from below total of 6392 data points are predicted correctly and around 2608 data points are wrongly predicted

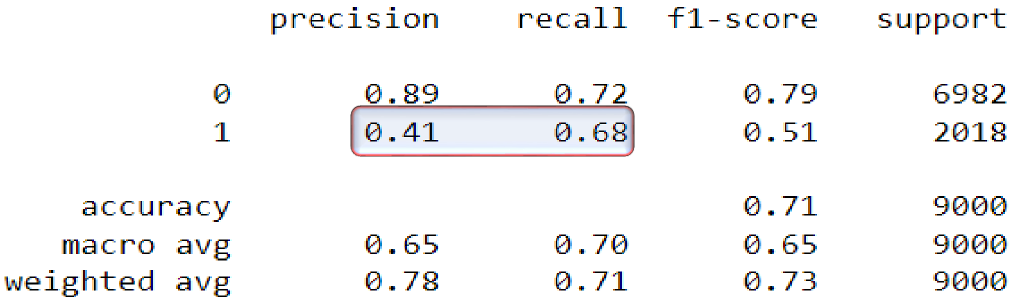


	Predicted:0	Predicted:1
Actual:0	5022	1960
Actual:1	648	1370

b) Test Classification Report:

As we can see from below report the test results, precision and recall(0.41 and 0.68 respectively) and the accuracy is 0.71.

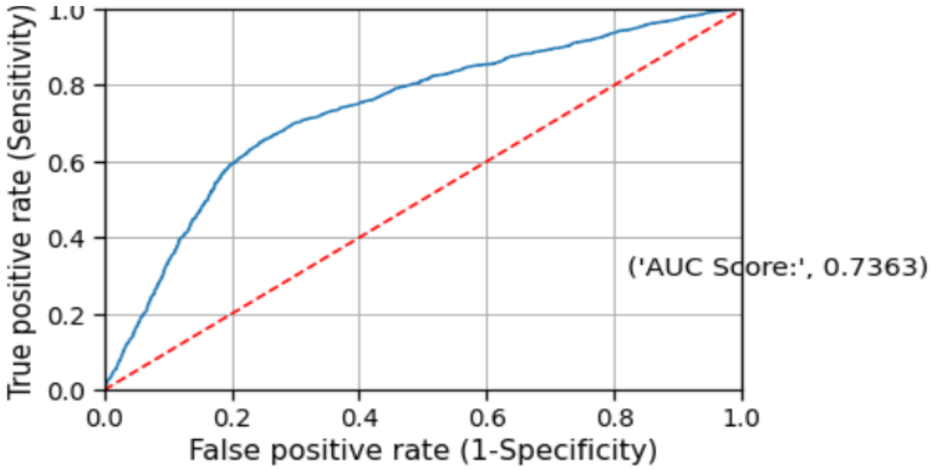
test score



```
              precision    recall  f1-score   support

           0       0.89      0.72      0.79      6982
           1       0.41      0.68      0.51      2018

    accuracy                           0.71      9000
   macro avg       0.65      0.70      0.65      9000
weighted avg       0.78      0.71      0.73      9000
```

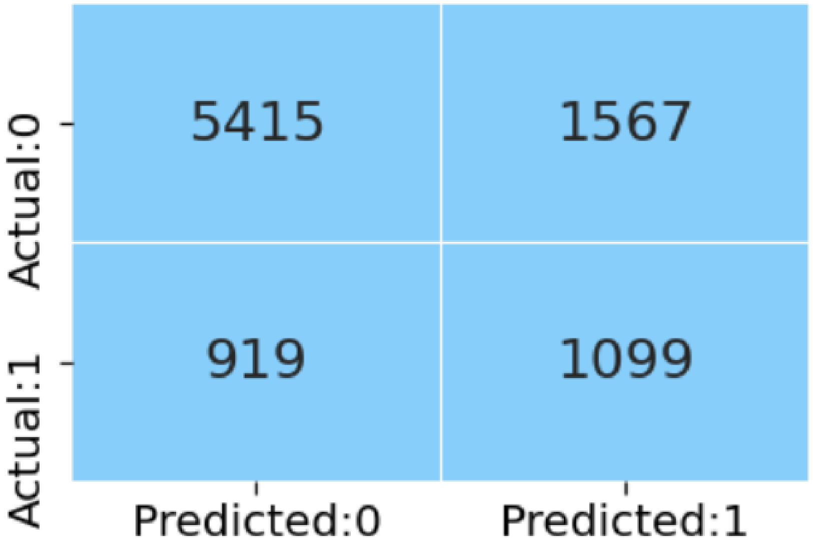
c)ROC Curve:

From below graph, we can see that the AUC score is 0.7363, which is more than the last model



gradient boosting

a) Confusion Matrix: As we can see from below total of 6514 data points are predicted correctly and around 2486 data points are wrongly predicted

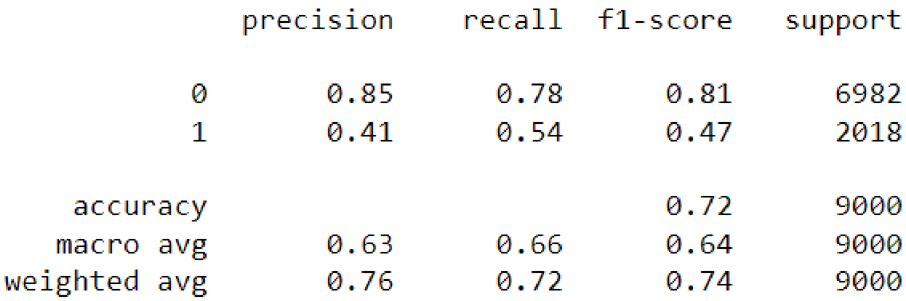


	Predicted:0	Predicted:1
Actual:0	5415	1567
Actual:1	919	1099

b) Test Classification Report:

As we can see from below report the test results, precision and recall(0.41 and 0.54 respectively) and the accuracy is 0.72.

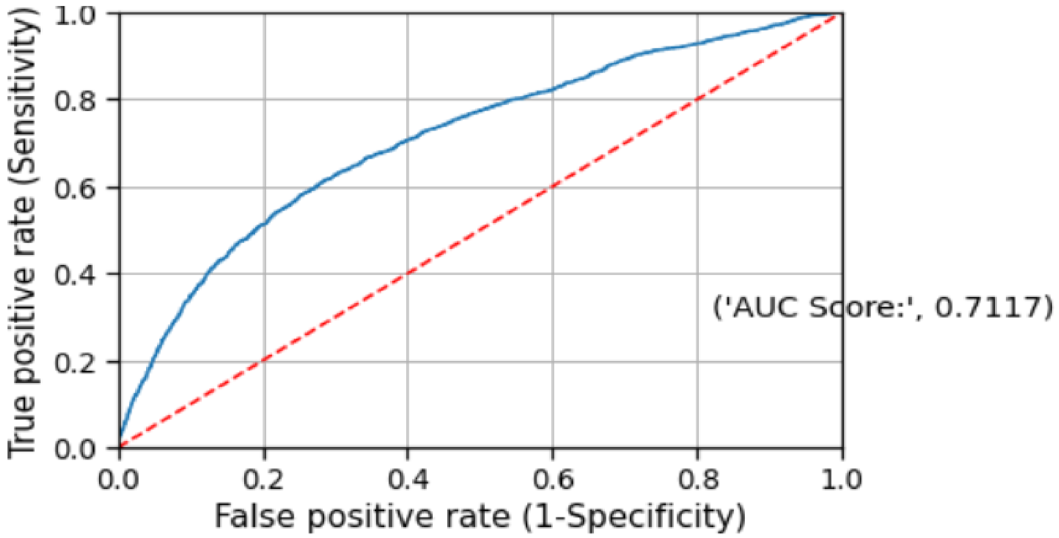
test score



```
              precision    recall  f1-score   support

           0       0.85      0.78      0.81      6982
           1       0.41      0.54      0.47      2018

    accuracy                           0.72      9000
   macro avg       0.63      0.66      0.64      9000
weighted avg       0.76      0.72      0.74      9000
```

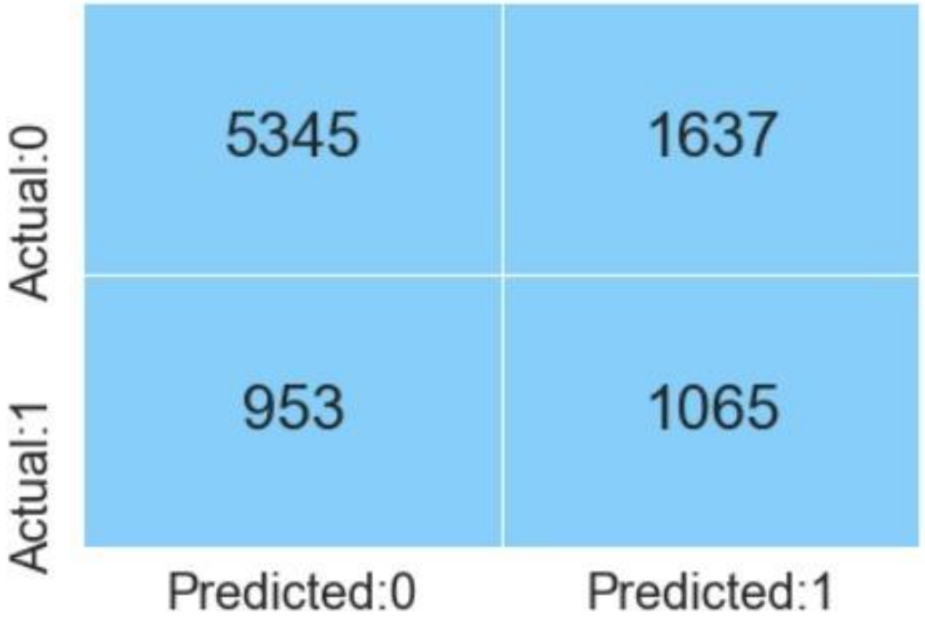
c)ROC Curve:

From below graph, we can see that the AUC score is 0.7117, which is almost similar to the last model.



XGBClassifier

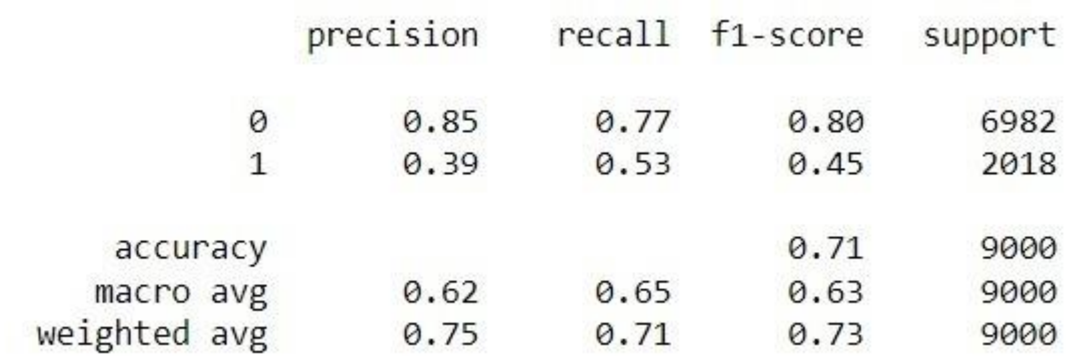
a) Confusion Matrix: As we can see from below total of 6410 data points are predicted correctly and around 2590 data points are wrongly predicted



	Predicted:0	Predicted:1
Actual:0	5345	1637
Actual:1	953	1065

b) Test Classification Report:

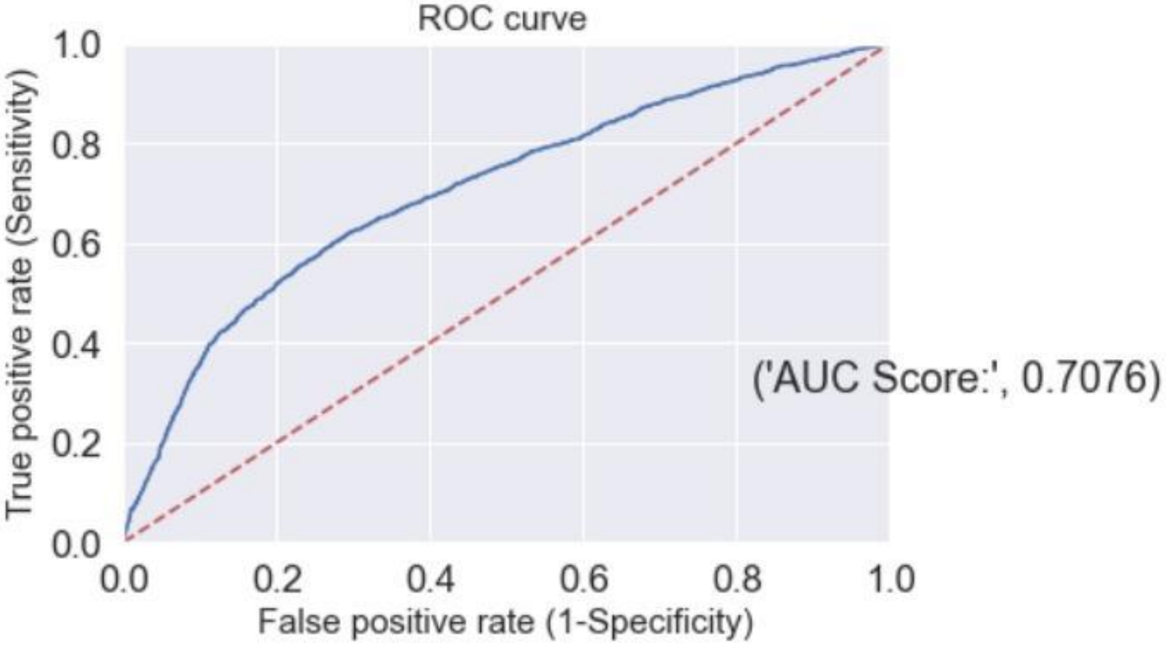
As we can see from below report the test results, precision and recall(0.39 and 0.53 respectively) and the accuracy is 0.71.



```
              precision    recall  f1-score   support

           0       0.85      0.77      0.80      6982
           1       0.39      0.53      0.45      2018

    accuracy                           0.71      9000
   macro avg       0.62      0.65      0.63      9000
weighted avg       0.75      0.71      0.73      9000
```

c)ROC Curve:

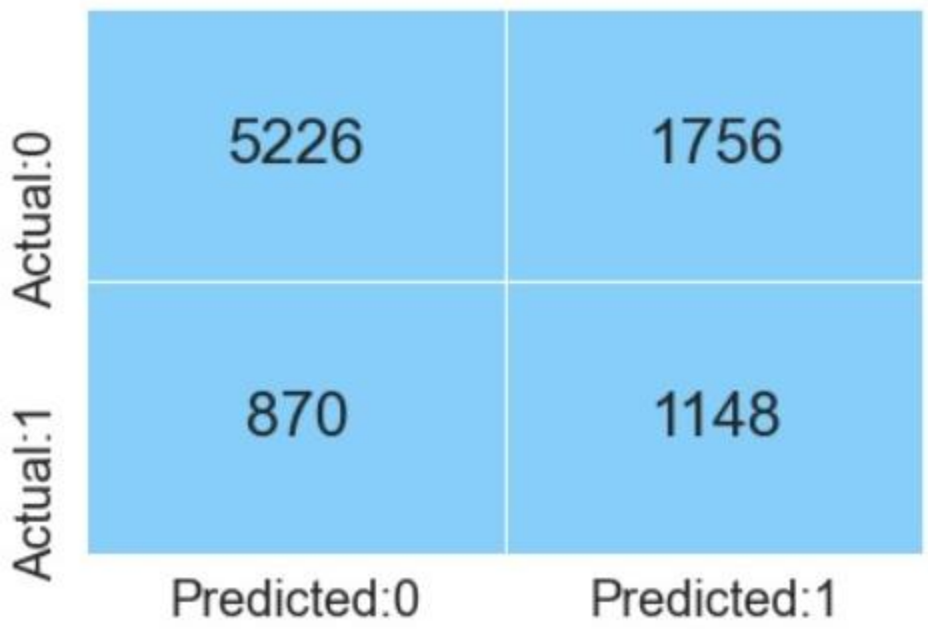
From below graph, we can see that the AUC score is 0.7076, which is slightly less than the last model.



XGBClassifier

With grid search

a) Confusion Matrix: As we can see from below total of 6374 data points are predicted correctly and around 2626 data points are wrongly predicted



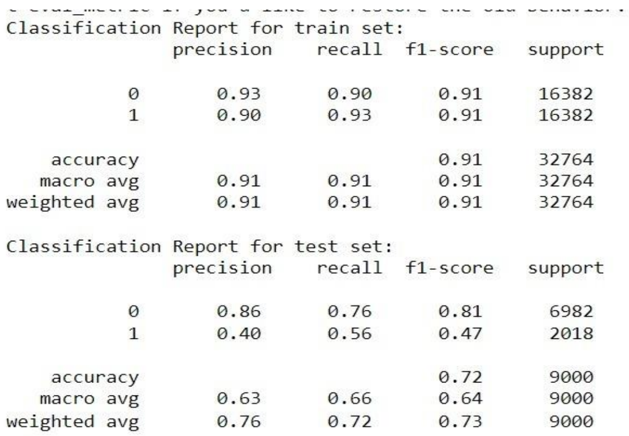
	Predicted:0	Predicted:1
Actual:0	5226	1756
Actual:1	870	1148

b) Test Classification Report:

As we can see from below report the train results, precision and recall(0.93 and 0.90 respectively) and the accuracy is 0.91.

b) test Classification Report:

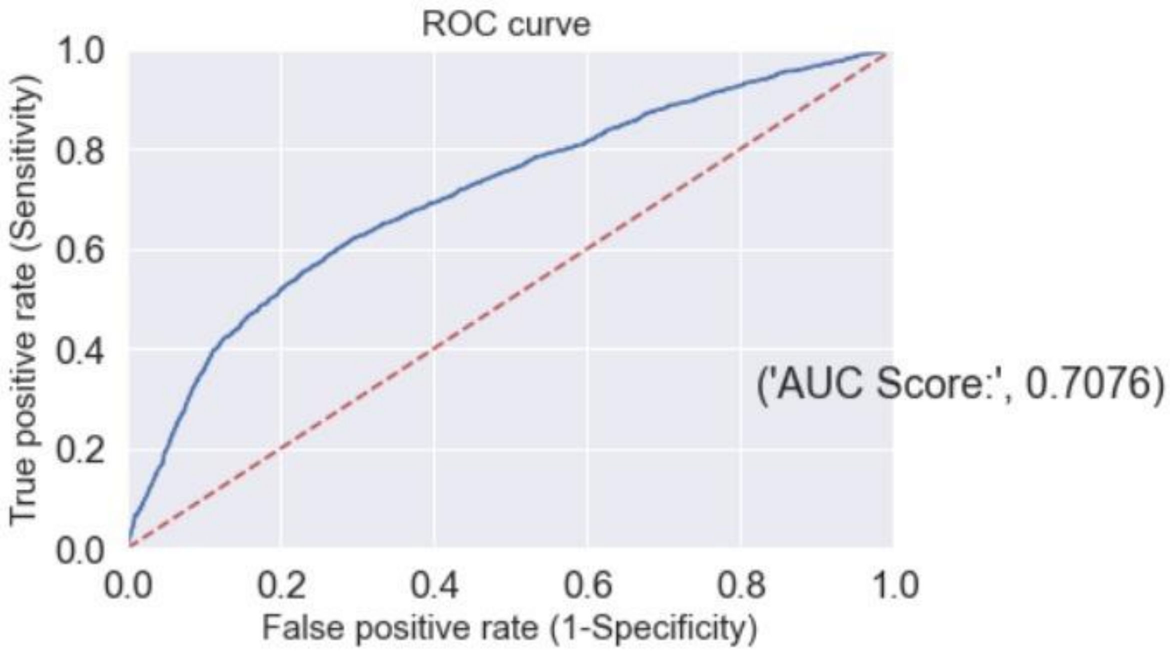
As we can see from below report the test results, precision and recall(0.56 and 0.40 respectively) and the accuracy is 0.72.



```
e eval_metric if you'd like to restore the old behavior.
Classification Report for train set:
               precision    recall  f1-score   support

           0       0.93      0.90      0.91     16382
           1       0.90      0.93      0.91     16382

    accuracy                           0.91     32764
   macro avg       0.91      0.91      0.91     32764
weighted avg       0.91      0.91      0.91     32764

Classification Report for test set:
               precision    recall  f1-score   support

           0       0.86      0.76      0.81      6982
           1       0.40      0.56      0.47      2018

    accuracy                           0.72      9000
   macro avg       0.63      0.66      0.64      9000
weighted avg       0.76      0.72      0.73      9000
```

c)ROC Curve:

From below graph, we can see that the AUC score is 0.7076, which is same as last model.



After performing above all models, we observed that Random Forest with hyper parameter tuning gave the better results. Though the Recall and Precision still haven't been balanced, the model is not overfitting on the train data as per the accuracy results.

The AUC Score is 0.7423, which is better than the other models. We're calculating the area between the blue curved line and pink dotted line. This area is a number between 0 and 1, zero meaning the model predicted all of the data incorrectly, and one meaning the model predicted all of the data correctly. Our model is pretty good at 0.7423.

Random forest, like its name implies, consists of a large number of individual decision trees that operate as an ensemble. Each individual tree in the random forest spits out a class prediction and the class with the most votes becomes our model’s prediction

We might get better than this with more number of estimators in the Grid Search(due to lower processing power of our laptops, we couldn’t do that.).

From the Classification report from the Random Forest model, due to class imbalance it is mostly identifying the class ‘0’, even after applying the SMOTE.

The Recall and Precision values tell us, how are we predicting the classes ‘0’s and ‘1’s i.e., non defaulters and defaulters. Here, we need to predict both of them correctly to not end up in the losses.

Since, identifying the potential defaulter as non defaulter might lead to the credit sanction and end up at loss.

In the other case, identifying the potential non defaulter as defaulter might result in losing the loyal customers.

At the end of the day, having the ability to predict 95% (recall score) of potential defaults would save a-lot of money on credit card charge-offs. Obviously, real-world application is more nuanced, but this modeling process is a step in the right direction.

LIMITATIONS, CHALLENGES AND SCOPE

Limitations of Data

Few of the limitations are: -

1. The dataset belongs to Taiwan and consists of data of only 30000 card holders details. The model will be more robust if the data would have belonged from different regions of the world.

2. Also, the duration of data collected is from April, 2005 and to September, 2005

2017. Due to this there isn't even distribution of the data.

Challenges

Few of the challenges faced are: -

1. High cardinality results in huge training effort in model tuning due to increase in model complexity (i.e. more number of features)

2. We also faced challenges on robust model tuning on all the models. Due to computational limitations, we are limited to using Randomised Search, and Grid Search as hyper parameter tuning techniques instead of using HyperOpt etc.

Scope

Scope for some future work is: -

1. Perform more hyper parameter tuning techniques for the XGB model since due to lower processing power of our laptops, we couldn't do that.

2. Exploring Google collab as an option for model training and tuning with faster lead time.

3. Exploring some robust data sampling technique as part of choosing smaller sample (a true representation of population data) from the population data.

4. Train the model again once more data comes in.

5. Try to work on more balanced data and in order to achieve better recall and precision.