BANKING PROJECT

Probability of Default

JUNE 16, 2024

SHUBHAM KUMAR

CONTENT Page No.

Introduction

Defining Problem Statement: -

As we all know in present times almost everyone is carrying a credit card in order to pay their bills. Credit cards also offers certain benefits in terms of discounts, offers while making a payment which has increased the credit card sales and usage. With increase in usage of credit card we also get more risk related to non-payment of credit card bills. Hence we need to study the behavioral pattern of the past customers and create a model such that we can identify such customers who can default and hence not issue a credit card or issue with limited balance.

Need of the Study: -

The credit card Industry of the banking domain has always been a major concern for the bank in terms of identifying legitimate customers. There is a strong need for risk prediction especially in financial industry to help manage uncertainty. Banking Operations are something that we all come across in our daily lives. In recent years the use of credit cards has become very popular as it is one of the most convenient payment options for everyone.

However, the convince does come with its own risk for the bank. As the number of customers using credit cards increases, more efforts need to be taken to consider managing the risk involved in terms of delinquency or default.

Understanding Business Opportunity: -

The overall objective of the risk management is to utilize the past behavioral information of the customers - financial spending, utilization and understand the patterns to make sound decisions for optimizing the risk of default customers.

The traditional approach to building a credit risk model, wherein the probability of default is to be estimated, utilizes a Logistic Regression methodology which not only gives accuracy rate but also has easily interpretable results. But here we will also use certain other machine learning techniques (such as Random Forest, Bagging, Boosting) in order to get better model.

Data Report

Data Collection is the process of collecting information from relevant sources in order to find a solution to the given problem

Data Collection in terms of time: - Data is collected of past 24 months or 2 years.

Data Collection in terms of Frequency: - Data Frequency means the rate at which the data is collected. Data is collected in terms of Quarterly, half yearly and annually basis

- In terms of Month: - 0 to 3 months & 0 to 6 months & 0 to 12 months
- In terms of Years: - Past 2 years.

Data Collection Methodology: - Data collection methodology means how the data are collected. Here we have records of customers for past 2 years.

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

- There are 99976 rows and 36 columns.
- Out of 36 columns, 3 columns are of Object datatype and 33 columns are Float Datatype

First 5 rows of the Dataset: -

Columns/Attributes	0	1	2	3	4
userid	4567129	2635118	4804232	1442693	4575322
default	0	0	0	0	0
acct_amt_added_12_24m	0	0	0	0	0
acct_days_in_dc_12_24m	0	0	0	NaN	0
acct_days_in_rem_12_24m	0	0	0	NaN	0
acct_days_in_term_12_24m	0	0	0	NaN	0
acct_incoming_debt_vs_paid_0_24m	0	NaN	NaN	NaN	NaN
acct_status	1	1	NaN	NaN	NaN
acct_worst_status_0_3m	1	1	NaN	NaN	NaN
acct_worst_status_12_24m	NaN	1	NaN	NaN	NaN
acct_worst_status_3_6m	1	1	NaN	NaN	NaN
acct_worst_status_6_12m	NaN	1	NaN	NaN	NaN
age	20	50	22	36	25
avg_payment_span_0_12m	12.692308	25.833333	20	4.6875	13
avg_payment_span_0_3m	8.333333	25	18	4.888889	13
merchant_category	Dietary supplements	Books & Magazines	Diversified entertainment	Diversified entertainment	Electronic equipment & Related accessories
merchant_group	Health & Beauty	Entertainment	Entertainment	Entertainment	Electronics
has_paid	1	1	1	1	1
max_paid_inv_0_12m	31638	13749	29890	40040	7100
max_paid_inv_0_24m	31638	13749	29890	40040	7100
name_in_email	no_match	F+L	L1+F	F1+L	F+L
num_active_div_by_paid_inv_0_12m	0.153846	0	0.071429	0.03125	0
num_active_inv	2	0	1	1	0
num_arch_dc_0_12m	0	0	0	0	0
num_arch_dc_12_24m	0	0	0	0	0
num_arch_ok_0_12m	13	9	11	31	1
num_arch_ok_12_24m	14	19	0	21	0
num_arch_rem_0_12m	0	3	3	0	0
status_max_archived_0_6_months	1	1	1	1	1
status_max_archived_0_12_months	1	2	2	1	1
status_max_archived_0_24_months	1	2	2	1	1
recovery_debt	0	0	0	0	0
sum_capital_paid_acct_0_12m	0	0	0	0	0
sum_capital_paid_acct_12_24m	0	0	0	0	0
sum_paid_inv_0_12m	178839	49014	124839	324676	7100
time_hours	9.653333	13.181389	11.561944	15.751111	12.698611

Fig. 2.A

First Five Rows of the Dataset

Last 5 rows of the Dataset: -

Column/ Attributes	89971	89972	89973	89974	89975
userid	1545432	3061692	1535658	2142946	3344745
default	0	0	0	0	0
acct_amt_added_12_24m	0	0	0	0	0
acct_days_in_dc_12_24m	0	0	0	0	NaN
acct_days_in_rem_12_24m	0	0	0	0	NaN
acct_days_in_term_12_24m	0	0	0	0	NaN
acct_incoming_debt_vs_paid_0_24m	NaN	NaN	NaN	NaN	NaN
acct_status	NaN	NaN	1	NaN	NaN
acct_worst_status_0_3m	NaN	NaN	1	NaN	NaN
acct_worst_status_12_24m	NaN	NaN	1	NaN	NaN
acct_worst_status_3_6m	NaN	NaN	1	NaN	NaN
acct_worst_status_6_12m	NaN	NaN	1	NaN	NaN
age	70	25	34	51	22
avg_payment_span_0_12m	NaN	10.1667	13.5556	13.4	34.5
avg_payment_span_0_3m	NaN	8	15	12.5	NaN
merchant_category	Concept stores & Miscellaneous	Diversified entertainment	Youthful Shoes & Clothing	Books & Magazines	Sports gear & Outdoor
merchant_group	Leisure	Entertainment	Clothing & Shoes	Entertain	Leisure
has_paid	NaN	1	1	1	NaN
max_paid_inv_0_12m	NaN	2,380.00	10,790.00	4,580.00	NaN
max_paid_inv_0_24m	NaN	2,380.00	10,790.00	4,580.00	NaN
name_in_email	NaN	Nick	F	Nick	NaN
num_active_div_by_paid_inv_0_12m	NaN	0	0	0	NaN
num_active_inv	NaN	0	0	0	NaN
num_arch_dc_0_12m	NaN	0	0	0	NaN
num_arch_dc_12_24m	NaN	0	0	0	NaN
num_arch_ok_0_12m	NaN	6	9	5	NaN
num_arch_ok_12_24m	NaN	9	0	1	NaN
num_arch_rem_0_12m	NaN	0	0	0	NaN
status_max_archived_0_6_months	NaN	1	1	1	NaN
status_max_archived_0_12_months	NaN	1	1	1	NaN
status_max_archived_0_24_months	NaN	1	1	1	NaN
recovery_debt	NaN	0	0	0	NaN
sum_capital_paid_acct_0_12m	NaN	0	0	0	NaN
sum_capital_paid_acct_12_24m	NaN	0	0	0	NaN
sum_paid_inv_0_12m	NaN	6,535.00	47,306.00	13,530.00	NaN
time_hours	NaN	11.8467	18.6819	11.9644	NaN

Fig. 2.B

Last Five Rows of the Dataset

Descriptive Statistics of the dataset: -

Column Names	count	mean	std	min	25%	50%	75%	max
userid	99,976.00	2,998,976.73	1,154,177.40	1,000,053.00	2,000,266.75	2,998,832.00	4,000,640.75	4,999,868.00
default	89,976.00	0.0143	0.1188	0	0	0	0	1
acct_amt_added_12_24m	99,976.00	12,255.15	35,481.48	0	0	0	4,937.25	1,128,775.00
acct_days_in_dc_12_24m	88,140.00	0.223	5.8081	0	0	0	0	365
acct_days_in_rem_12_24m	88,140.00	5.0446	22.864	0	0	0	0	365
acct_days_in_term_12_24m	88,140.00	0.2869	2.9299	0	0	0	0	97
acct_incoming_debt_vs_paid_0_24m	40,661.00	1.3313	26.4823	0	0	0.1521	0.663	3,914.00
acct_status	45,603.00	1.0422	0.2027	1	1	1	1	4
acct_worst_status_0_3m	45,603.00	1.1729	0.4201	1	1	1	1	4
acct_worst_status_12_24m	33,215.00	1.3373	0.575	1	1	1	2	4
acct_worst_status_3_6m	42,274.00	1.1853	0.4433	1	1	1	1	4
acct_worst_status_6_12m	39,626.00	1.2531	0.5056	1	1	1	1	4
age	99,976.00	36.0163	13.0013	18	25	34	45	100
avg_payment_span_0_12m	76,140.00	17.9715	12.7511	0	10.8	14.9091	21	260
avg_payment_span_0_3m	50,671.00	14.9898	10.2974	0	8.4	13	18.2857	87
has_paid	88,942.00	0.8658	0.3409	0	1	1	1	1
max_paid_inv_0_12m	88,942.00	9,362.82	13,672.46	0	2,390.00	6,170.00	11,400.00	279,000.00
max_paid_inv_0_24m	88,942.00	11,419.73	15,431.75	0	3,685.00	7,720.00	13,865.00	538,500.00
num_active_div_by_paid_inv_0_12m	70,051.00	0.1121	0.2881	0	0	0	0.1	9
num_active_inv	88,942.00	0.626	1.6105	0	0	0	1	47
num_arch_dc_0_12m	88,942.00	0.0626	0.3821	0	0	0	0	17
num_arch_dc_12_24m	88,942.00	0.0592	0.3623	0	0	0	0	13
num_arch_ok_0_12m	88,942.00	7.7865	16.7429	0	0	3	8	261
num_arch_ok_12_24m	88,942.00	6.8467	16.068	0	0	2	7	313
num_arch_rem_0_12m	88,942.00	0.484	1.3956	0	0	0	0	42
status_max_archived_0_6_months	88,942.00	0.8217	0.7166	0	0	1	1	3
status_max_archived_0_12_months	88,942.00	1.0742	0.7764	0	1	1	2	5
status_max_archived_0_24_months	88,942.00	1.2482	0.8205	0	1	1	2	5
recovery_debt	88,942.00	3.602	116.2115	0	0	0	0	16,411.00
sum_capital_paid_acct_0_12m	88,942.00	10,860.38	26,630.74	0	0	0	8,960.75	571,475.00
sum_capital_paid_acct_12_24m	88,942.00	6,615.02	19,243.90	0	0	0	102.75	341,859.00
sum_paid_inv_0_12m	88,942.00	41,036.37	94,596.85	0	3,396.50	17,057.50	45,739.50	2,962,870.00
time_hours	88,942.00	15.3418	5.0306	0.0003	11.6319	15.8083	19.5542	23.9997

Fig. 2.B

Descriptive Statistics of the Dataset

Understanding the variables: -

- **UsersId**: - It specifies the userid of the customers
- **Default**: - It gives the default status of the customer. 0 means Non-Defaulters & 1 means Defaulters.
- **acct_amt_added_12_24m**: - It tells us about the purchase made in between last 24 months and 12 months. For example, taking present date as 12-05-24. It means purchase made between 12-05-22 and 12-05-23.
- **acct_days_in_dc_12_24m**: - Debit collection status means if a user has not paid even a minimum amount of the bill in given time period after bill generation, the account moves into Debit collection status. Time period is same as above. Logical if an account has stayed in debit collection status is high then chances to default is also high.
- **acct_days_in_rem_12_24m**: - Reminder Status means if an account user has not paid a minimum account until due date, the bank starts sending reminder. The date from which the first reminder minus the date when payment was done gives us the days the account has stayed in reminder status.
- **acct_days_in_term_12_24m: -** Terminated status means the number of days the account has been terminated due to non-payment of minimum of the credit card bill.
- **acct_incoming_debt_vs_paid_0_24m: -** The ratio of the amount collected out of the total debt in an account by an agency to the total debt amount of the account in the previous 24 months from the current date.
- **acct_status: -** Account Status. 1 means is active and zero means inactive
- **acct_worst_status_0_3m / 12_24m / 3_6m / 6_12m: -** Worst Status: If a Customer has not even paid a minimum amount of the bill for more than 30 days post the due date, then the account goes into a state called as worst date.
- **avg_payment_span_0_12m / 0_3m: -** The average payment span that the customer has taken in days after the credit card bill got generated in the last one year and last 3 months.
- **Merchant_category & merchant_group:** - It tells us about the merchant category and group to which the credit card was use for.
- **has_paid:** - Customer has paid their last credit card bill or not
- **max_paid_inv_0_12m / 0_24m**: - it tells gives us about the maximum credit card bill amount that has been paid by the customer in the last one year and 2 years. These 2 variables tells us about the customer expenditure capacity.
- **num_active_div_by_paid_inv_0_12m:** - Ratio of the number of unpaid bills to the paid bills in the last one year.
- **num_active_inv:** - Number of active invoice
- **num_arch_dc_0_12m / 12_24m**: - Achieved purchase means we have purchased something and return product. number of archived purchases that were in debt collection status in the last one year or between last 24 and 12 months.

- **num_arch_ok_0_12m / 12_24m: -** number of archived purchases that were paid in the last one year or between last 24 and 12 months.
- **num_arch_rem_0_12m**: - number of archived purchases that were in the reminder status in the last one year.
- **status_max_archived_0_6_months / 0_12_months / 0_24_months**: - maximum number of times the account was in archived status in the last 6 months / 1 year /2 years.
- **recovery_debt**: - Total amount that has been recovered out of total debit amount on the account
- **sum_capital_paid_acct_0_12m / 12_24m**: - sum of principal balance paid on account in the last one year / between last 24 and 12 months.
- **sum_paid_inv_0_12m: -** The total amount of the paid invoices in the last one year.
- **time_hours**: - The total hours spent by the customer in purchases made using the credit card.

Exploratory data analysis

Univariate Analysis

Default: -

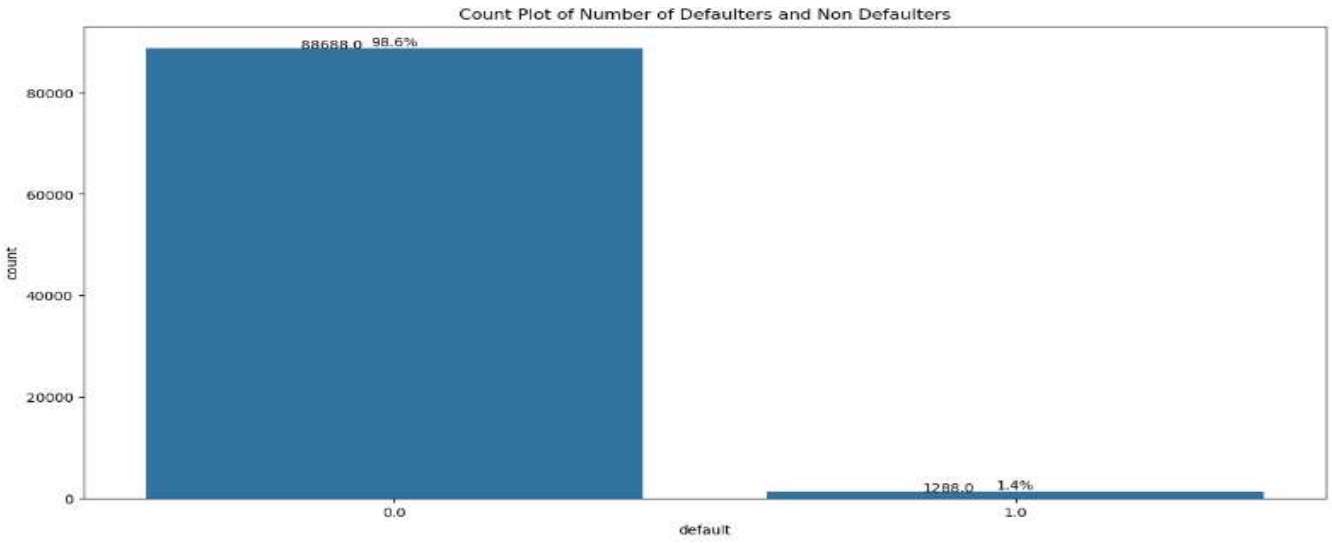


Fig. 3.1

- Out of total credit card user only 1.4% or 1288 of users have defaulted

Age_group: -

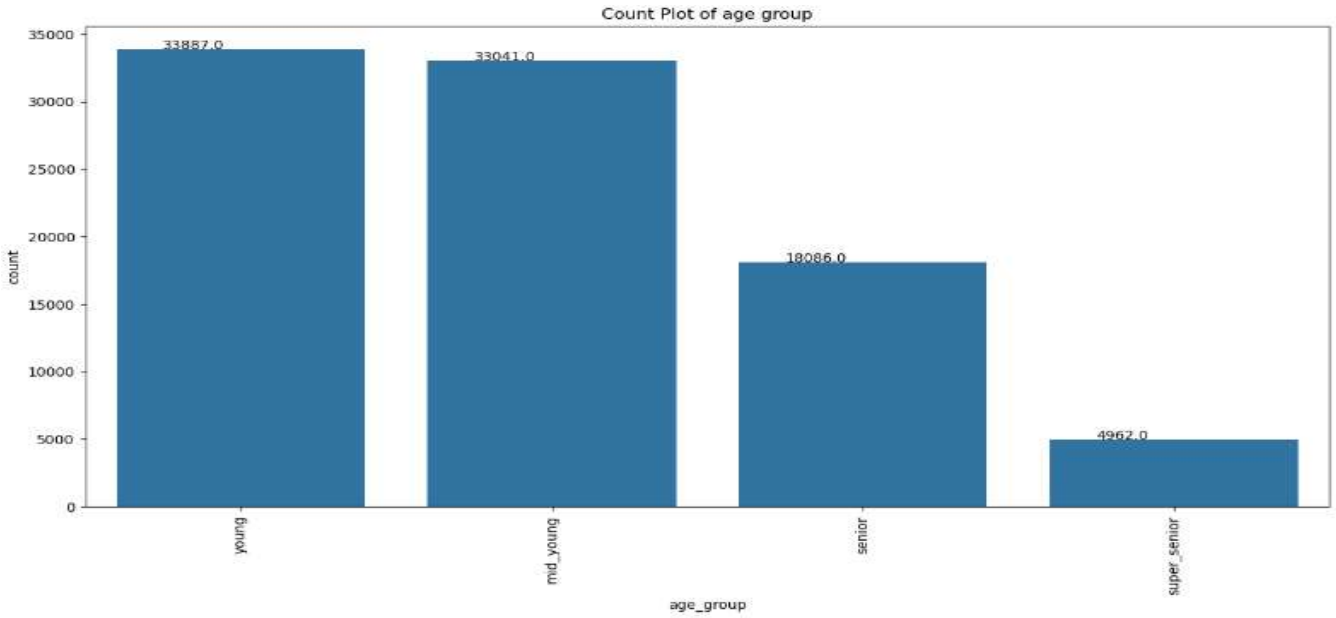


Fig. 3.1

- More than 70 % of Customer are either Young (18 - 30) or Mid Young (30 – 45)

Merchant Group: -

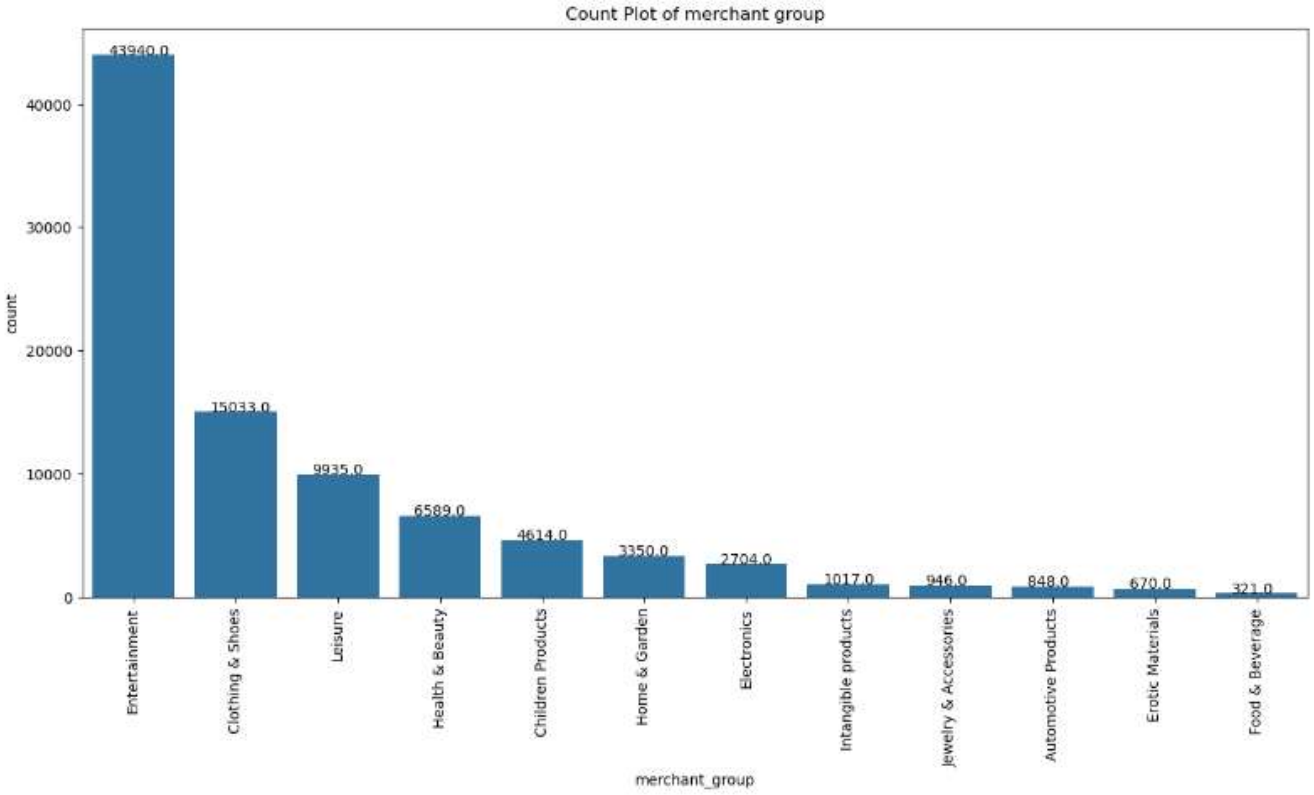


Fig. 3.2

Count Plot of Merchant Group

Inferences: -

- Credit cards were used most at merchant group of Entertainment i.e., 50% followed by Clothing &store
- Credit card were least used in Food & Beverages followed by Erotic Materials

Mechant Category: -

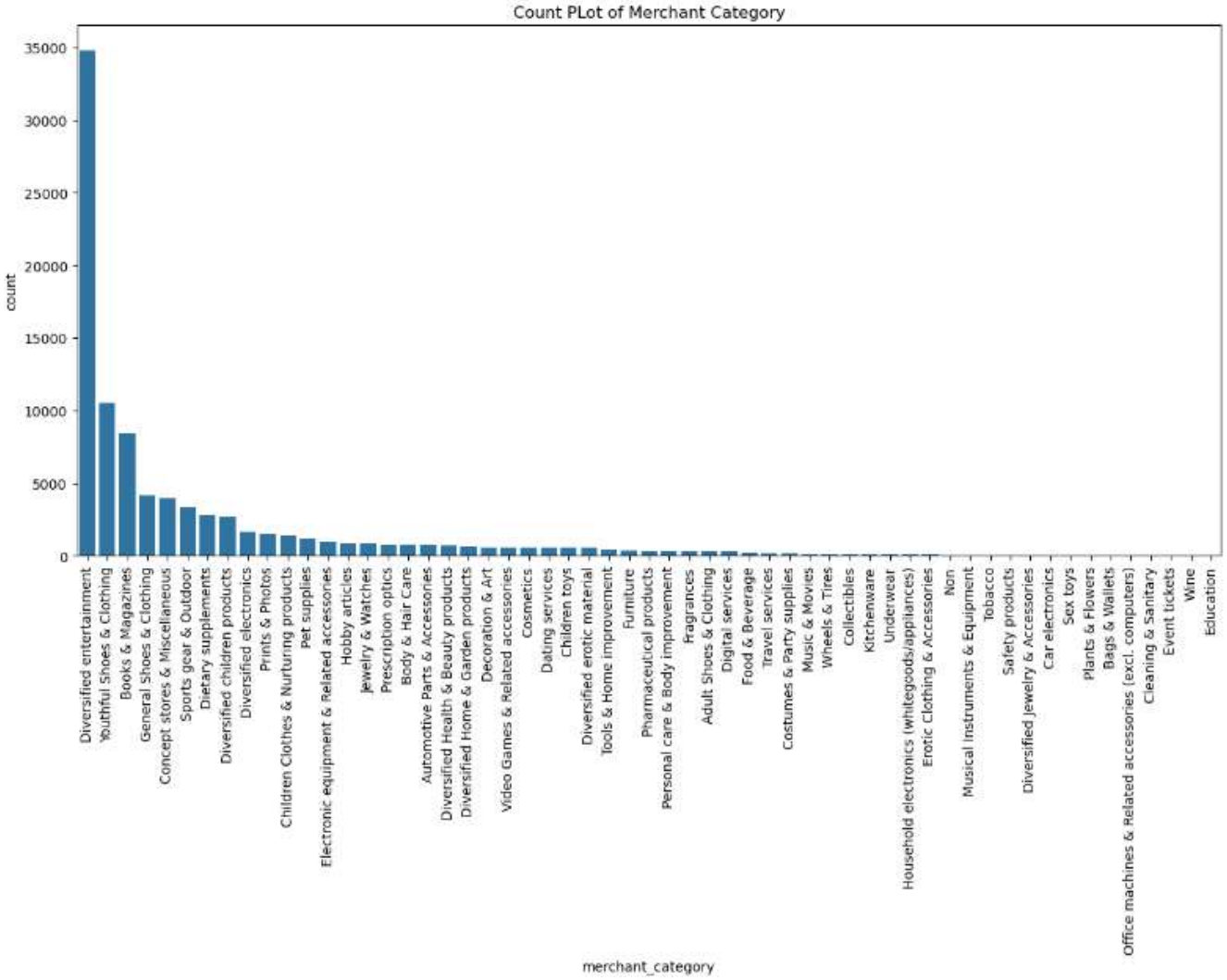


Fig. 3.3

Count Plot of Merchant Category

Inferences: -

- Credit Card were used most for Diversified Entertainment purpose followed by Youthful Shoe & Clothing.
- Credit card were used least for Education followed by Wine.

Acct_status: -

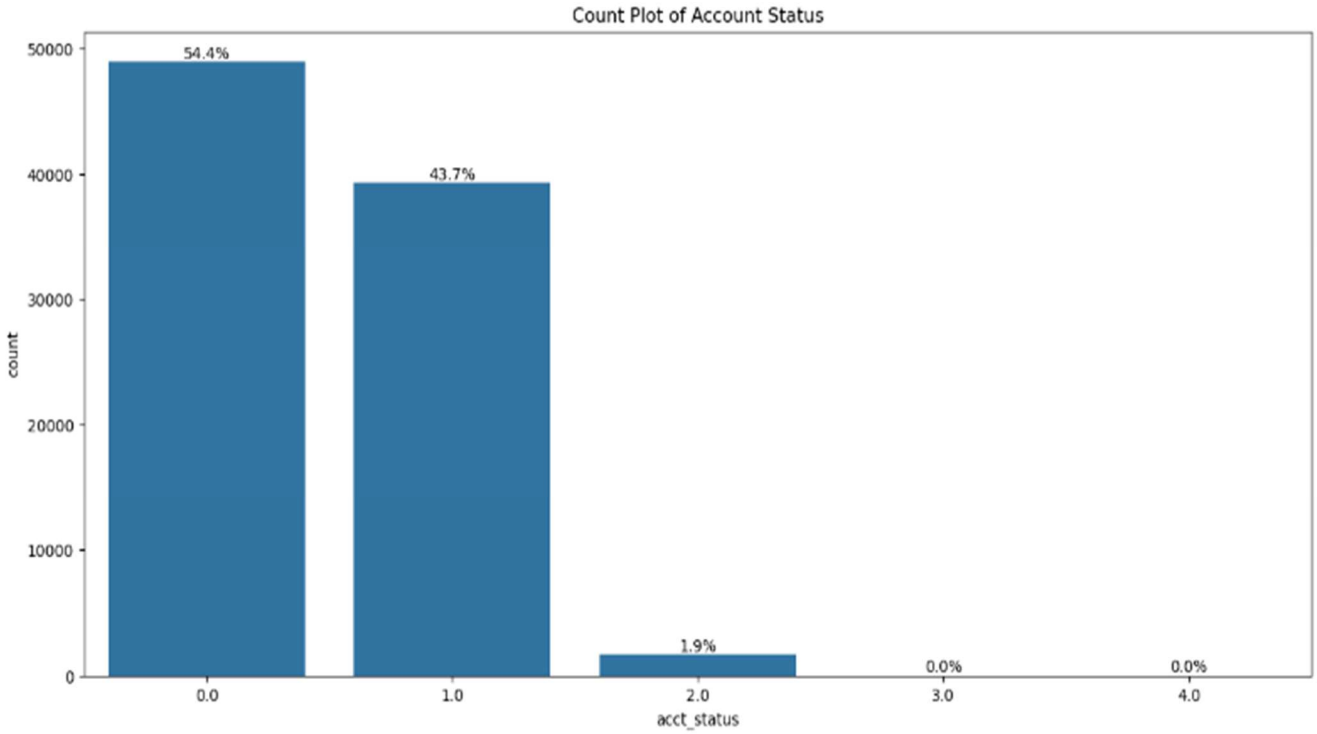


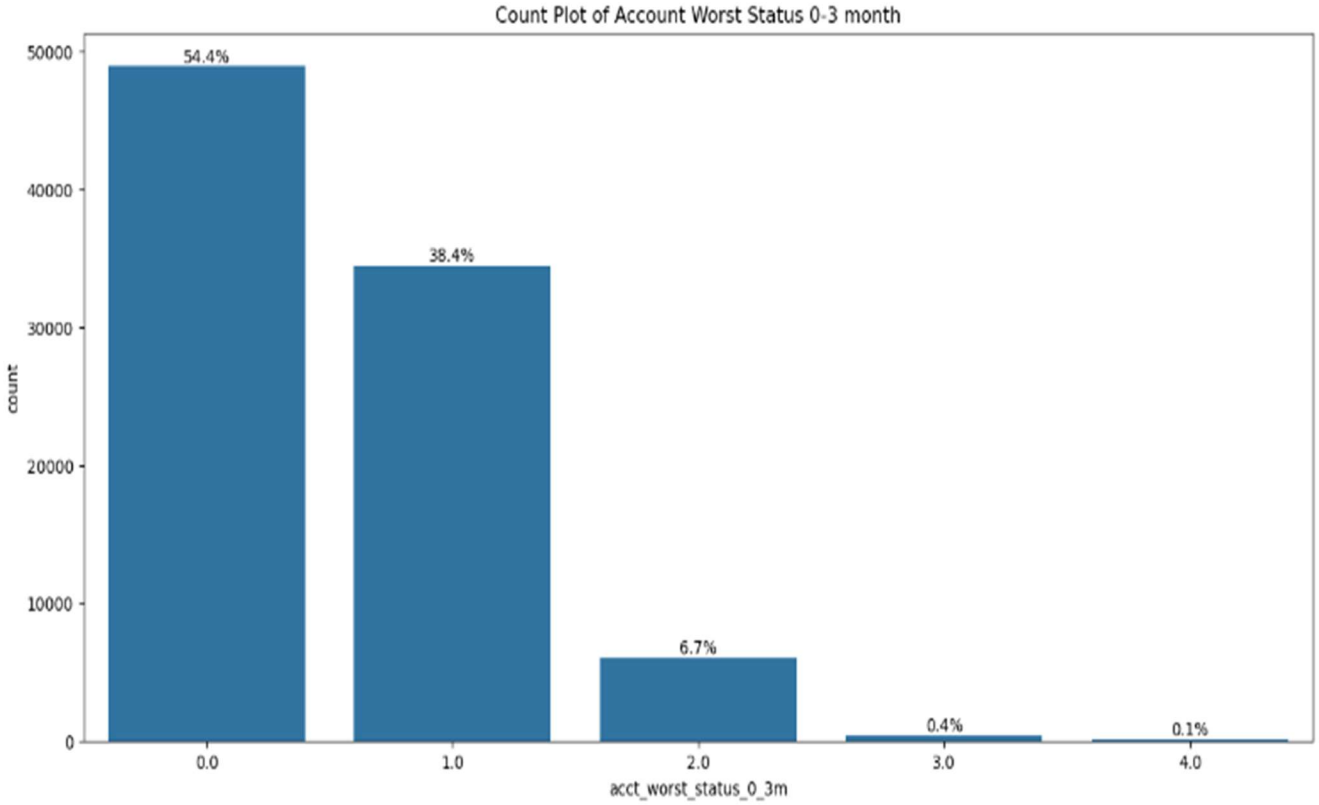
Fig 3.4

Count Plot of Account Status

Inferences: -

- 0 is considered as Inactive Account. 54 % of account are in Inactive Status
- 1-2-3-4 are considered as an active account . 46 % of account are in Active Status.

Acct_worst_status_0_3m: -



3.4
Count Plot of Acct_worst_status_0_3m

Inferences: -

- 54.4% accout were inactive.
- Out of remaining 43.6% of accounts, 38.4% account has low account worst status
- 7.1% of accounts have medium account worst status
- 0.1% of account had worst account status

Acct_worst_status_12_24m: -

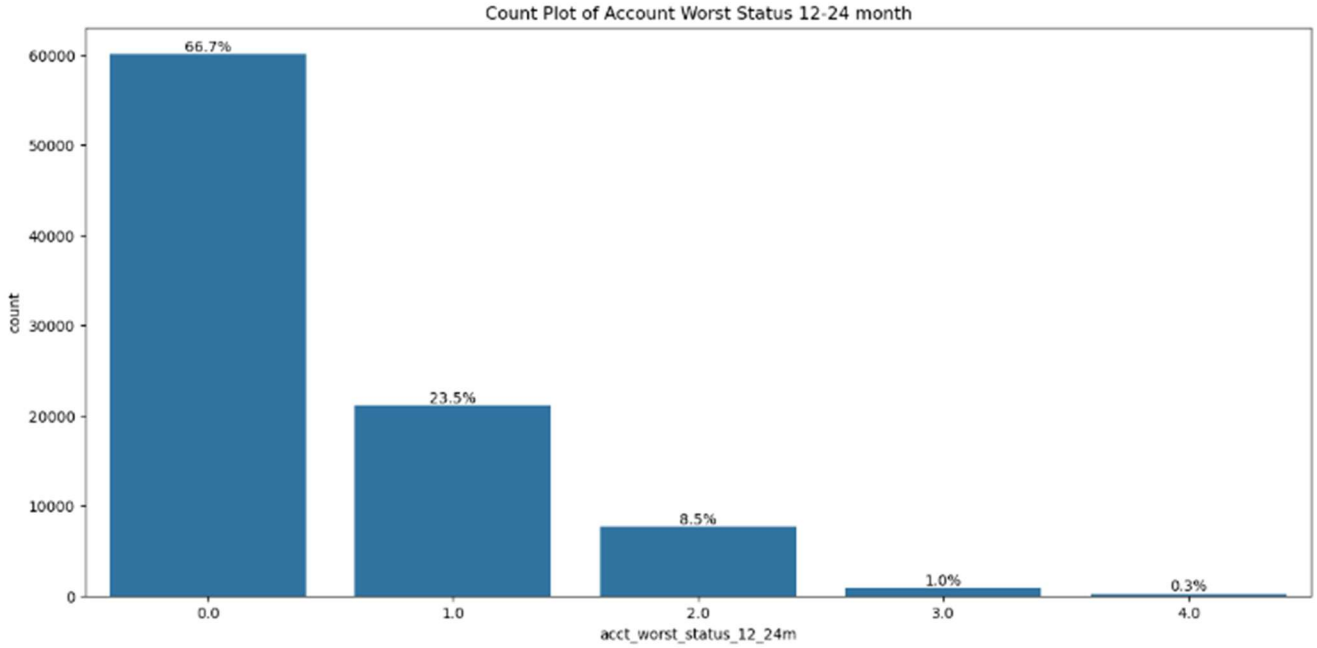


Fig 3.5
Count Plot of Acct_worst_status_12_24m

Inferences: -

- In between of last 24 to 12 months, 66.7% of account has remained inactive.
- 23.5% of account has low account worst status
- 9.5% of account has medium account worst status
- 0.3% of account has high account worst status

Acct_worst_status_3_6m: -

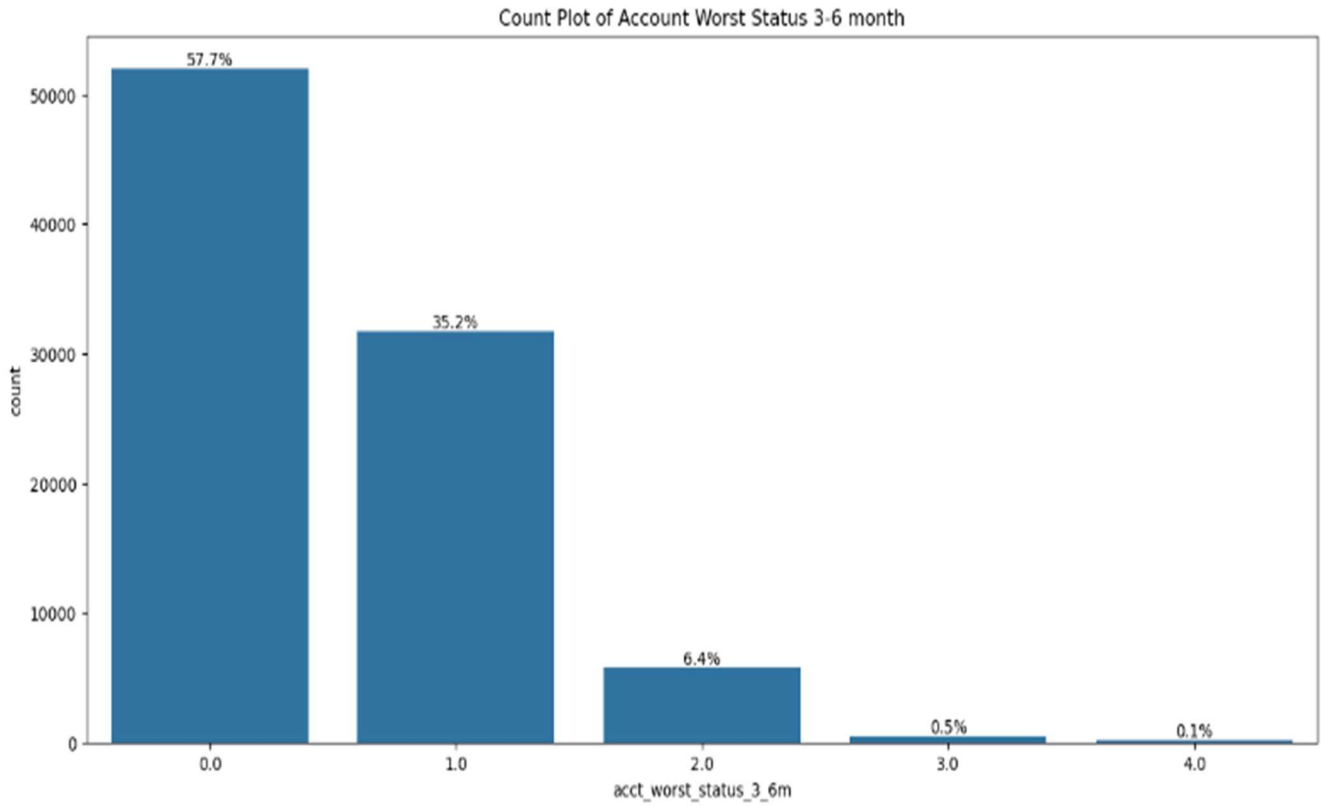


Fig 3.6
Count Plot of Acct_worst_status_3_6m

Inferences: -

- In between of last 6 to 3 months, 57.7% of account has remained inactive.
- 35.2% of account has low account worst status
- 6.9% of account has medium account worst status
- 0.1% of account has high account worst status

Acct_worst_status_6_12m: -

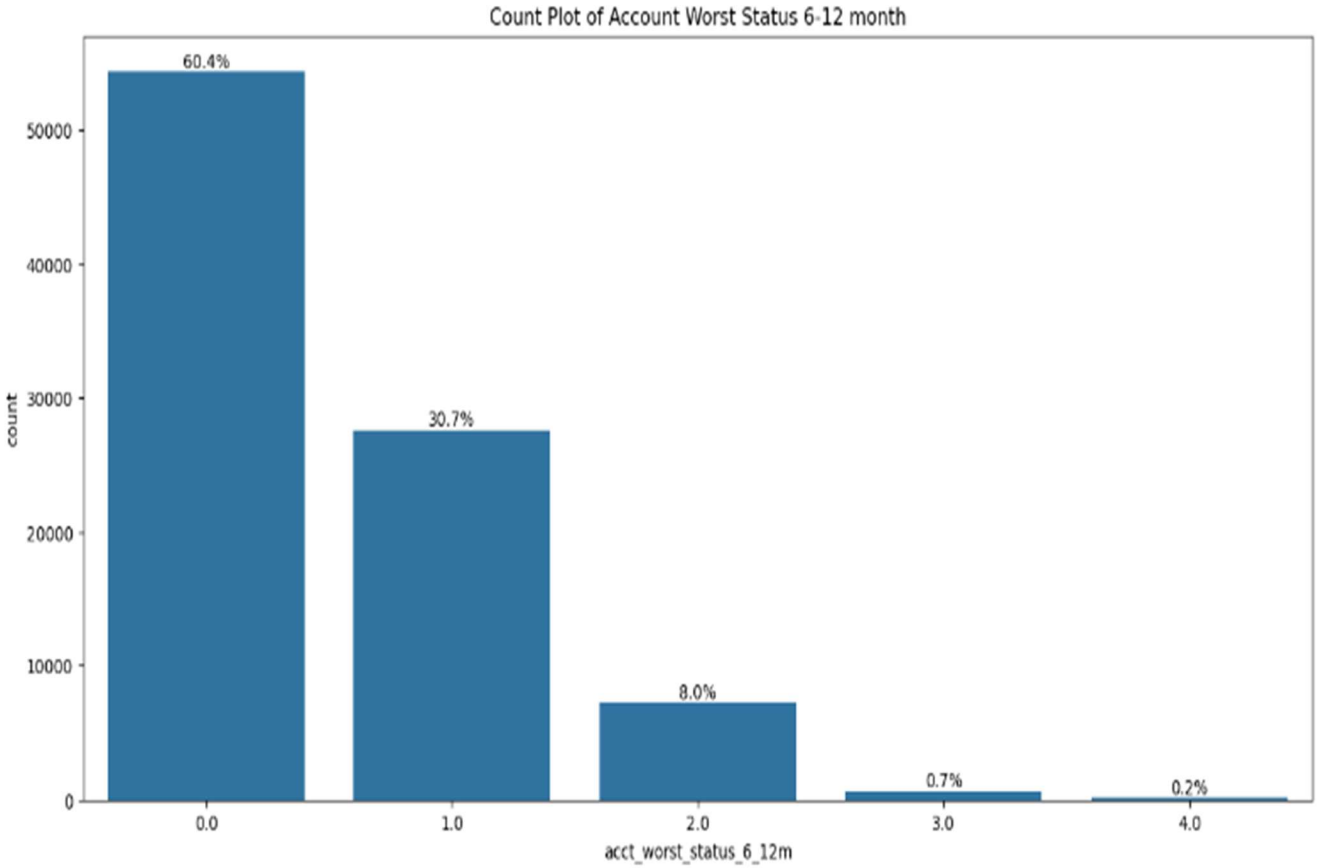


Fig 3.7
Count Plot of Acct_worst_status_6_12m

Inferences: -

- In between of last 6 to 3 months, 60.4% of account has remained inactive.
- 30.7% of account has low account worst status
- 8.7% of account has medium account worst status
- 0.2% of account has high account worst status

Has_paid: -

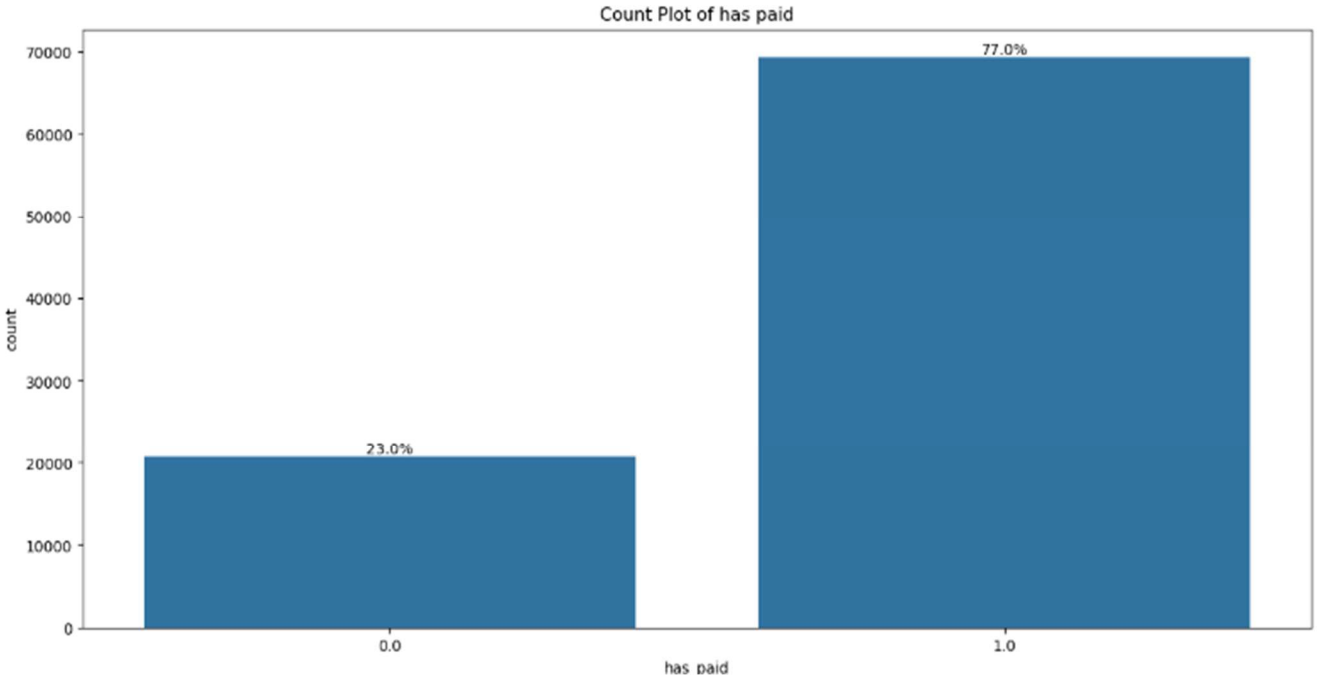


Fig 3.8
Count Plot of has_paid

Inferences: -

- 77% of customer has paid the last credit card bill
- 23% of customer has not paid the credit card bill

Distribution Plot of Numberical variable (continious)

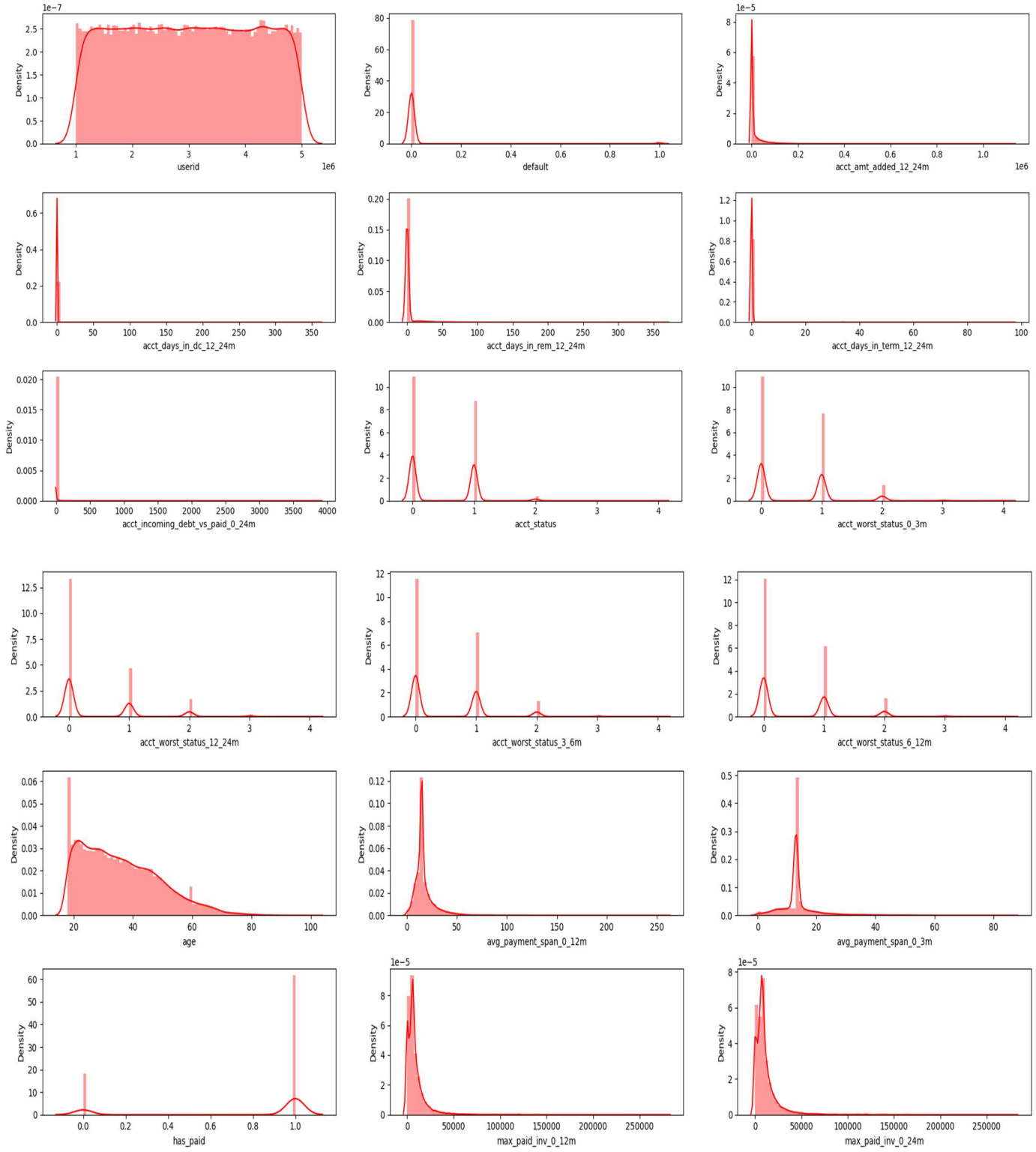


Fig. 3.9A

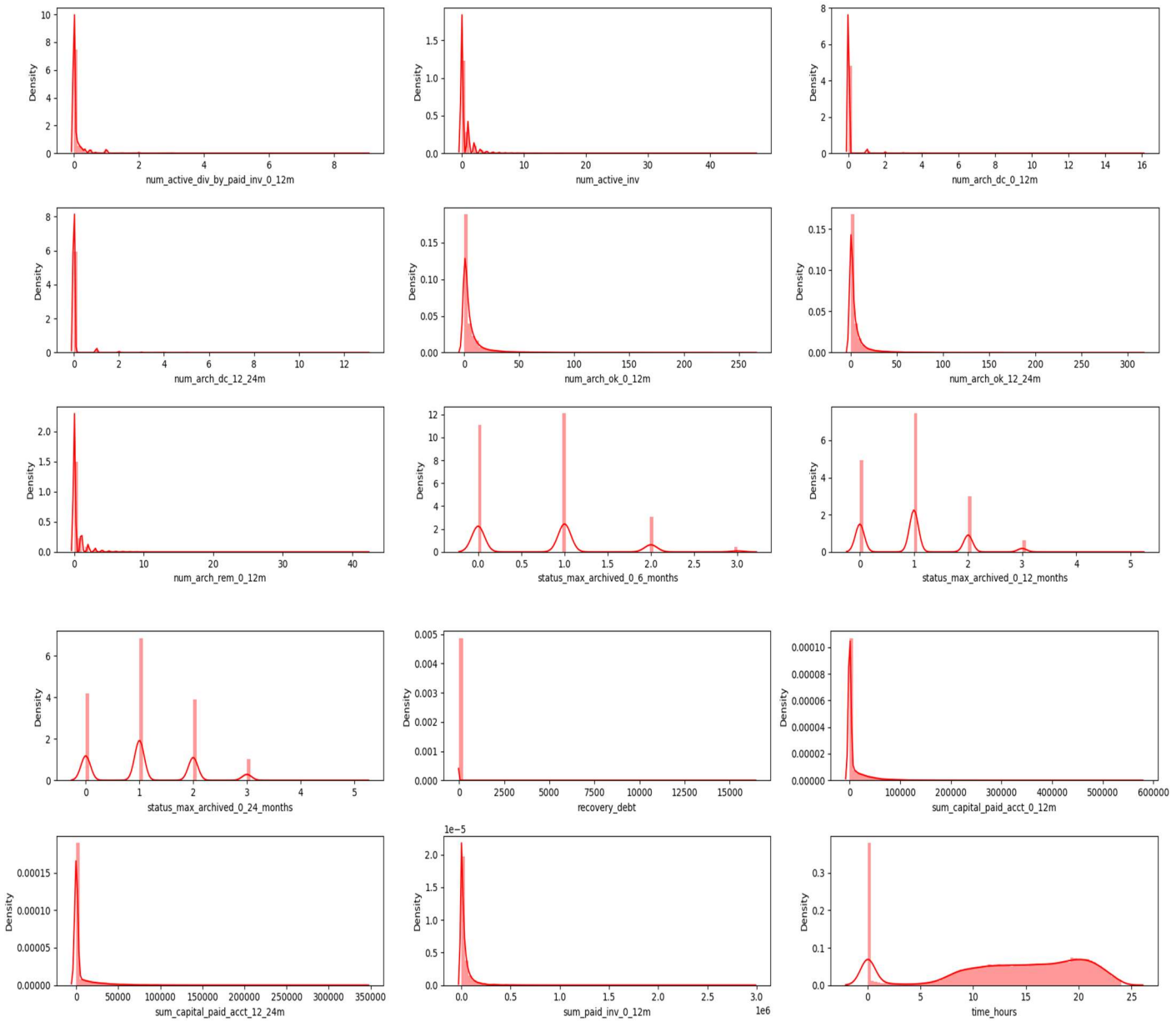


Fig 3.9B
Distribution Plot of all Numeric Variables

Inferences: -

- **Almost all the numeric continious variables are right skewed.**
- **Most of the data points are accumalated in left side of plots indicating possible outliers in the variables.**

Boxplot of all the numeric variables: -

Fig. 3.10

- **Almost all the numeric continuous variables are having outliers**

Bivariate Analysis

1. Default Vs Acct_amt_added_12_24m

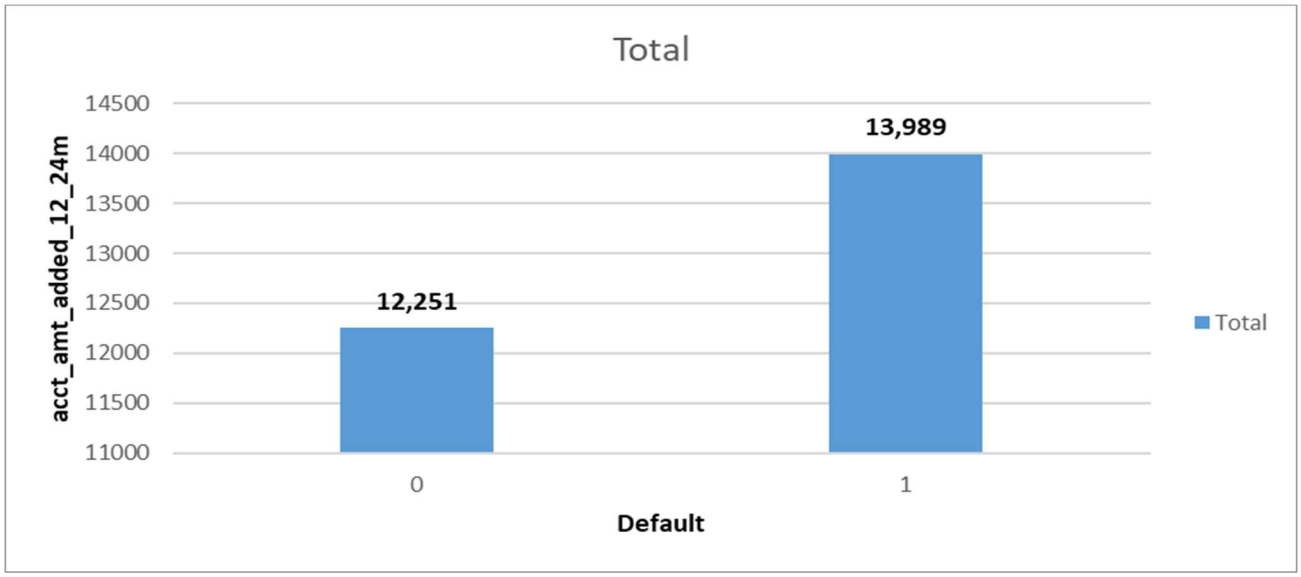


Fig. 3.11A

- Average acct_amt_added_12-24m for Defaults is 13989 whereas for Non Defaults it is 12251

2. Default Vs acct_days_in_dc_12_24m

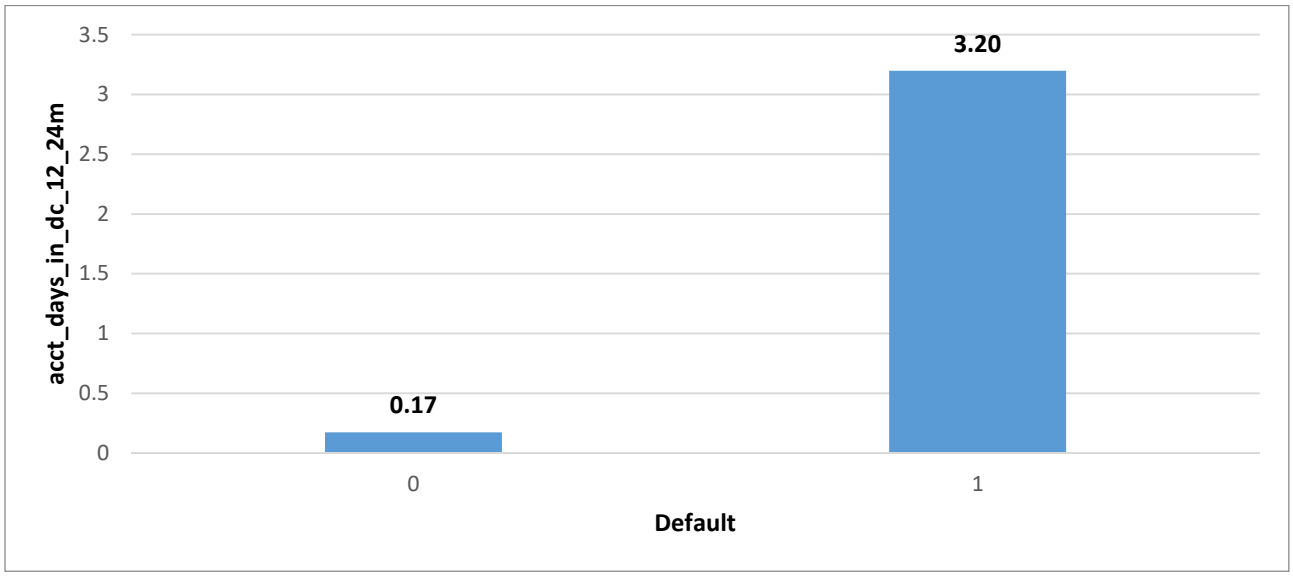


Fig. 3.11B

- Probability to Default Increases if the account has been in debit credit for more than 3 days

3. Default Vs Average of acct_days_in_term_12_24m

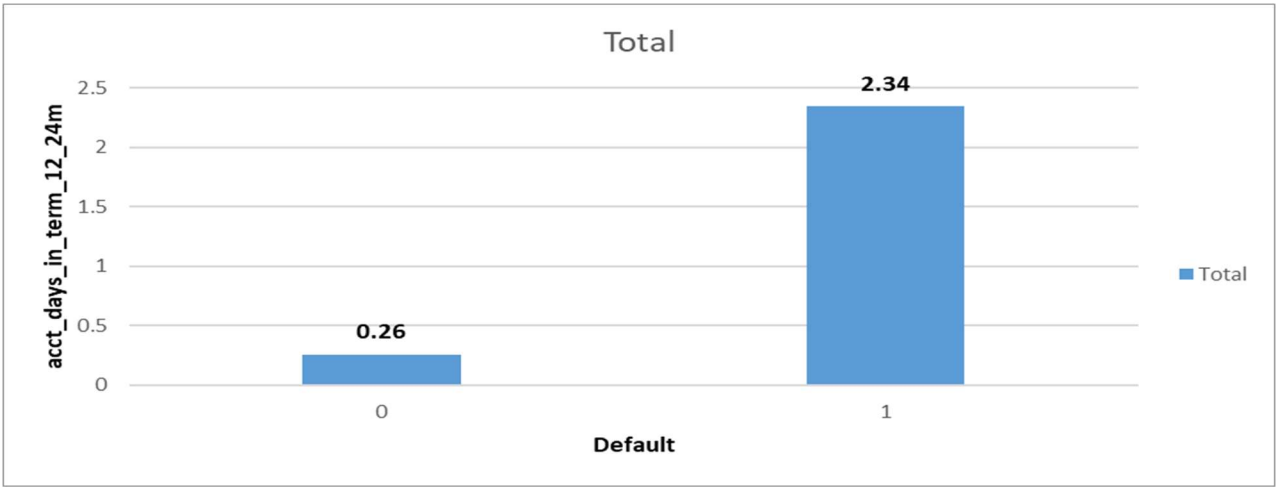


Fig. 3.11C

- Probability to Default Increases if the account has been in terminated for more than 2 days

4. <u>Default Vs merchant_group</u>

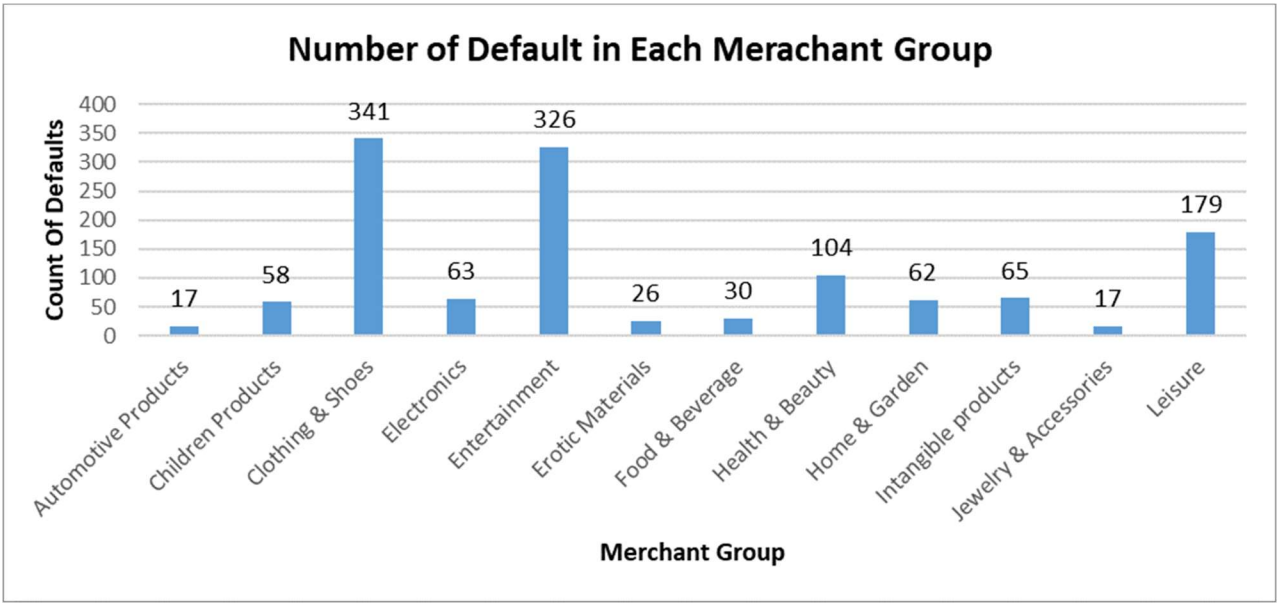


Fig. 3.11D

- 'Clothing & Shoe' and Entertainment are 2 highest subcategory group which contributes to more than 50 % 0f Default

Multivariate Analysis

Heatmap of all the Numeric Variables: -

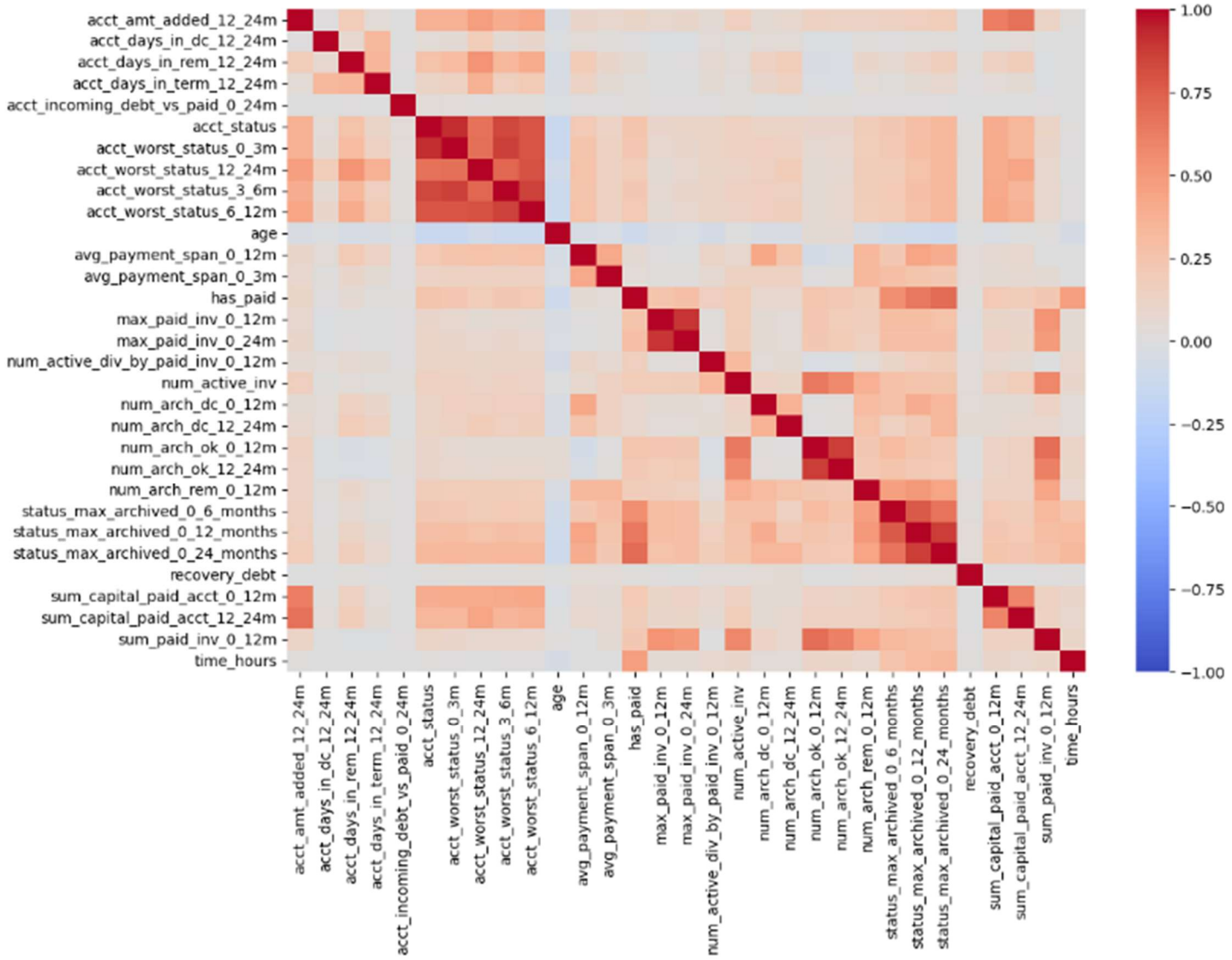


Fig 3.11
Correlation Plot of all Numeric Variables

Inferences: -

- Acct_status, acct_worst_status_0_3m, acct_worst_status_12_24m, acct_worst_status_3_6m, acct_worst_status_6_12m have high positive correlation.
- Max_paid_inv_0_12_m and max_paid_inv_0_24M also have high positive correlation.
- Status_max_achieved_0_6_months, Status_max_achieved_0_12_months & Status_max_achieved_0_24_months also have high positive correlation
- Has_paid has some correlation **with** Status_max_achieved_0_6_months, Status_max_achieved_0_12_months & Status_max_achieved_0_24_months

Data Cleaning & Data Preprocessing

Missing Value treatment

There are certainly many missing values in our dataset.

- In Default columns we have total of 10002 missing values. We have dropped the rows where default column has missing values as imputing missing value of target variable is not a good choice for model. Also default missing value is almost 10% of dataset. By dropping these 10% of data we will have 90 % of pure target variable data.
- In all other columns we have imputed missing values using either 0 or median except for merchant_category column where we have imputed using logical category

Column Names	No. of Null Entries	Imptation Using
userid	0	
default	0	
acct_amt_added_12_24m	0	
acct_days_in_dc_12_24m	10683	0
acct_days_in_rem_12_24m	10683	0
acct_days_in_term_12_24m	10683	0
acct_incoming_debt_vs_paid_0_24m	53357	0
acct_status	48934	0
acct_worst_status_0_3m	48934	0
acct_worst_status_12_24m	60055	0
acct_worst_status_3_6m	51938	0
acct_worst_status_6_12m	54313	0
age	0	
avg_payment_span_0_12m	21468	Median
avg_payment_span_0_3m	44382	Median
merchant_category	0	
merchant_group	9	Logical
has_paid	9944	0

Column Names	No. of Null Entries	Imptation Using
max_paid_inv_0_12m	9944	Median
max_paid_inv_0_24m	9944	Median
num_active_div_by_paid_inv_0_12m	26936	Median
num_active_inv	9944	0
num_arch_dc_0_12m	9944	0
num_arch_dc_12_24m	9944	0
num_arch_ok_0_12m	9944	0
num_arch_ok_12_24m	9944	0
num_arch_rem_0_12m	9944	0
status_max_archived_0_6_months	9944	0
status_max_archived_0_12_months	9944	0
status_max_archived_0_24_months	9944	0
recovery_debt	9944	0
sum_capital_paid_acct_0_12m	9944	0
sum_capital_paid_acct_12_24m	9944	0
sum_paid_inv_0_12m	9944	0
time_hours	9944	0

Outliers Treatment

- Logistic Regression is **sensitive** to outliers.
- Decision Tree Model or Ensemble Technique Model are **insensitive** to outliers.
- So we will treat these outliers for Logistic Regression model.
- Outliers Treatment can be done using **IQR method**
- **We have used IQR Method to treat outliers as all the variable are having skewed distribution or we can say not normal distribution**
- IQR Stands for Inter Quartile Range
- In IQR method we all the values which are above Upper Fence are replaced with Upper Limit and all the values which are below Lower Fence are replaced with Lower Limit
- IQR is calculated using First Quartile (Q1) and Third Quartile (Q3).

IQR = Q3 – Q1

- Upper Limit and Lower Limit are calculated using: -

Upper Limit = Q3 + (1.5 * IQR) **Lower Limit = Q1 – (1.5 * IQR)**

Variable Transformation

- LDA is a distance based algorithm which means all the variables needs to be in same scale to create robust model.
- Scaling or Variable transformation is an important technique to create robust models using logistic regression. Because the predictors are linear in the log of the odds, it is often helpful to transform the continuous variables to create a more linear relationship.
- For Variables or Dataset Transformation we have used **Log Transfromation.**
- For Decision Tree Model or Ensemble Technique Model scaling or variable transformation is not required because the tree structure will remain the same with or without the transformation.

Addition of new variables

Since it is a classification problem it is better to have variables with different categories rather than number hence we have added a new variable age_group, where we have divided age into 4 groups: -

- 18 - 30 as ‘Young’
- 30 - 45 as ‘Mid Young’
- 45 - 60 as ‘Senior’
- 60 - 100 as ‘Super Senior’

Removal of unwanted variables

- We have removed 3 variables which are not relevant fir our models, userid , name_in_email & age.
  1. Userid: - UserID are distinct for each customer. Hence we have dropped
  2. Name_in_email: - name_in_email is simply name written in the email which doesn't add any relevance in our model. Hence we have dropped
  3. Age: - We have created a new column age_group from age. Hence we have dropped it.

Converting Categorical Variables into Numeric Variables

We have 3 columns (namely: - merchant_group, merchant_category, age_group) which are object data type which we needed to convert into numeric.

- In order to convert these object datatypes columns into numeric datatype columns we have used **LaberEncoder**. LabelEncoder assigns value to each categories of the column.
- Sample data before and after conversion using LabelEncoder : -

```
Entertainment           43940
Clothing & Shoes        15033
Leisure                  9935
Health & Beauty          6589
Children Products        4614
Home & Garden            3350
Electronics              2704
Intangible products      1017
Jewelry & Accessories     946
Automotive Products       848
Erotic Materials          670
Food & Beverage           330
Name: merchant_group, dtype: int64
```

```
4     43940
2     15033
11     9935
7      6589
1      4614
8      3350
3      2704
9      1017
10      946
0       848
5       670
6       330
Name: merchant_group, dtype: int64
```

MODEL SELECTION

As we know in this dataset we have determine whether a customer is going to default or not based on past data of the customers.

Since we have to predicts discrete class labels of the customer (Default or Non Default), it is a classification model. We have several Classification Model from which we have selected 4 models in order to choose most optimum model: -

1. **Logistic Regression**: - It is ideal for prediction of binary output. (ex: - Yes/No, 1/0, Pass/Fail)

2. **Linear Discriminant Analysis**: - It reduces the dimensionality of data by capturing the most relevant features (topics) while discarding noise.

3. **K Nearest Neighbor Model: -** It is widely disposable in real-life scenarios since it is non-parametric, meaning it does not make any underlying assumptions about the distribution of data (as opposed to other algorithms such as GMM, which assume a Gaussian distribution of the given data

4. **Decision Tree**: - Unlike many other machine learning algorithms, decision trees work effectively with non-linear data. They can handle complex relationships between features

5. **Adaptive Boosting**: - AdaBoost is a powerful technique in machine learning that improves model performance by combining multiple weak learners

Model building and interpretation

LOGISTIC REGRESSION MODEL

Logistic regression is used for binary classification where we use sigmoid function, that takes input as independent variables and produces a probability value between 0 and 1.

Key Points:

- Logistic regression predicts the output of a categorical dependent variable. Therefore, the outcome must be a categorical or discrete value.
- It can be either Yes or No, 0 or 1, true or False, etc. but instead of giving the exact value as 0 and 1, it gives the probabilistic values which lie between 0 and 1.
- In Logistic regression, instead of fitting a regression line, we fit an "S" shaped logistic function, which predicts two maximum values (0 or 1)
- It maps any real value into another value within a range of 0 and 1. The value of the logistic regression must be between 0 and 1, which cannot go beyond this limit, so it forms a curve like the "S" form. The S-form curve is called the Sigmoid function or the logistic function.

SIMPLE LOGISTIC REGRESSION MODEL

We have used the scaled data for the Logistic Regression

First we have created a simple Logistic Regression model and calculated the Confusion Matrix and Classification Report on both train and test data: -

Classification Report performing simple Logistic Regression on Train Data

	precision	recall	f1-score	support
0.0	0.99	1.00	0.99	62081
1.0	0.35	0.03	0.05	902
accuracy			0.99	62983
macro avg	0.67	0.51	0.52	62983
weighted avg	0.98	0.99	0.98	62983

Classification Report performing simple Logistic Regression on Test Data

	precision	recall	f1-score	support
0.0	0.99	1.00	0.99	26607
1.0	0.39	0.04	0.07	386
accuracy			0.99	26993
macro avg	0.69	0.52	0.53	26993
weighted avg	0.98	0.99	0.98	26993

Fig. 4.1A

Classification Report with simple Logistic Regression (Train and Test Data)

INFERNCES: -

- Our major concern is for prediction of defaults (1)
- We can see for both train and test data our model has recall for default (1) very less i.e., 0.03 & 0.04, hence we can say our model is very poor in prediction of defaults.

Note: - We can see that model Recall of 1 is very low for both train and test. Hence the Simple Logistic Regression model is the weak or poor model

The default threshold for the logistic regression model is 0.5. i.e., all the prediction probability less than 0.5 are predicted as 0 and all the prediction probability more than 0.5 are predicted as 1. So here we tried to change the threshold value of prediction probability and do the prediction based on that cutoff threshold.

In order to find a cutoff threshold we had used np.argmax(). This function will take the difference between all the sequential TPR (True Positive Rate) and FPR (False Positive Rate) and finds out the maximum difference between the two for both train and test.

For train Optimum Threshold Cutoff = 0.018833

For Test Optimum Threshold Cutoff = 0.013922

Classification Report and Confusion Matrix After shifting the Optimum Cutoff Threshold for Train data: -

Logistic Regression Classification Report(threshold=0.01883) on Train Data

	precision	recall	f1-score	support
0.0	1.00	0.72	0.84	62081
1.0	0.04	0.81	0.08	902
accuracy			0.72	62983
macro avg	0.52	0.77	0.46	62983
weighted avg	0.98	0.72	0.83	62983

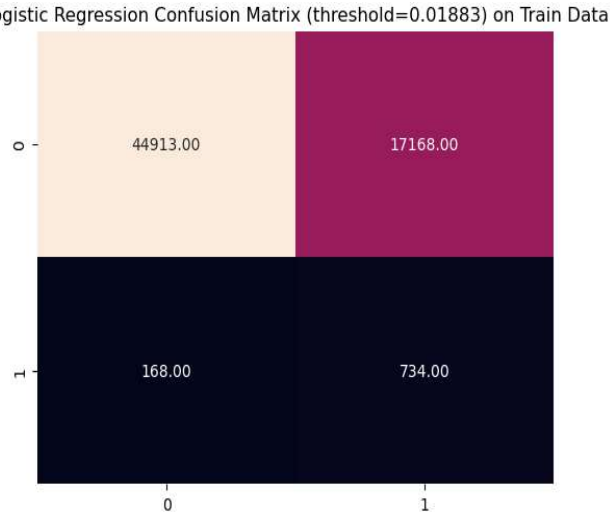


Fig. 4.1B

Classification Report and Confusion Matrix of Train Data (Cutoff Threshold = 0.018833)

INFERNCES: -

- For Train Data out of 902 number of 1's(Default), 734 were actually predicted as 1's (Default)
- Also 168 numbers of 1's (Default) were predicted as 0's and 17168 number of 0's(Non-Default) were predicted as 1's(Default)
- Recall of 1's(Yes) = 0.81
- Precision of 1's (Yes) = 0.04
- Accuracy = 0.72
- Recall of 0's(No) = 0.72
- Precision of 0's(No) = 1
- ROC AUC Score = 0.769

Classification Report and Confusion Matrix After shifting the Optimum Cutoff Threshold for Test data

Logistic Regression Classification Report threshold=0.0139 on Test Data

	precision	recall	f1-score	support
0.0	1.00	0.71	0.83	26607
1.0	0.04	0.84	0.08	386
accuracy			0.71	26993
macro avg	0.52	0.78	0.45	26993
weighted avg	0.98	0.71	0.82	26993

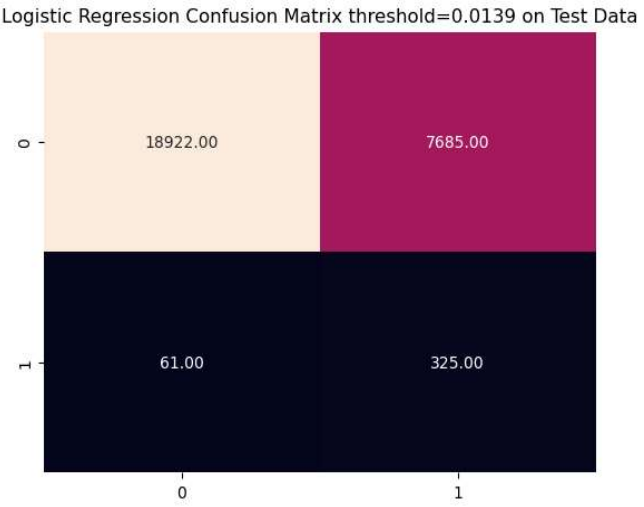


Fig. 4.1C

Classification Report and Confusion Matrix of Test Data (Cutoff Threshold = 0.013922)

Inference: -

- For Test Data out of 386 number of 1's(Default), 325 were actually predicted as 1's (Default)
- Also 61 numbers of 1's (Default) were predicted as 0's and 7685 number of 0's(Non-Default) were predicted as 1's(Yes)
- Recall of 1's (Default) = 0.84
- Precision of 1's (Default) = 0.04
- Accuracy = 0.71
- Recall of 0's (Non - Default) = 0.71
- Precision of 0's (Non - Default) = 1
- ROC AUC score = 0.77

For Both Train and Test Data after shifting the cutoff threshold, we can see that our model performance to detect defaulters have increased significantly.

Note: - We can see that model Recall of 1 is very good for both train and test. Hence the Simple Logistic Regression model with optimum cutoff threshold is good model

Logistic Regression Model with GridSearchCV

GridSearchCV is a hyper tuning parameter which is used to find the best suitable parameters for the model.

Best Parameters after hyper tuning: -

```
LogisticRegression
LogisticRegression(max_iter=1000, penalty='l1', random_state=1, solver='saga',
                   tol=0.001)
```

Fig. 4.1D
Best Parameters after GridSearchCV

Logistic Regression Classification Report and Confusion matrix (with GridSearchCV) on Train Data

Logistic Regression Classification Report(with GridSearchCV) on Train Data

	precision	recall	f1-score	support
0.0	0.99	1.00	0.99	62081
1.0	0.34	0.02	0.05	902
accuracy			0.99	62983
macro avg	0.66	0.51	0.52	62983
weighted avg	0.98	0.99	0.98	62983

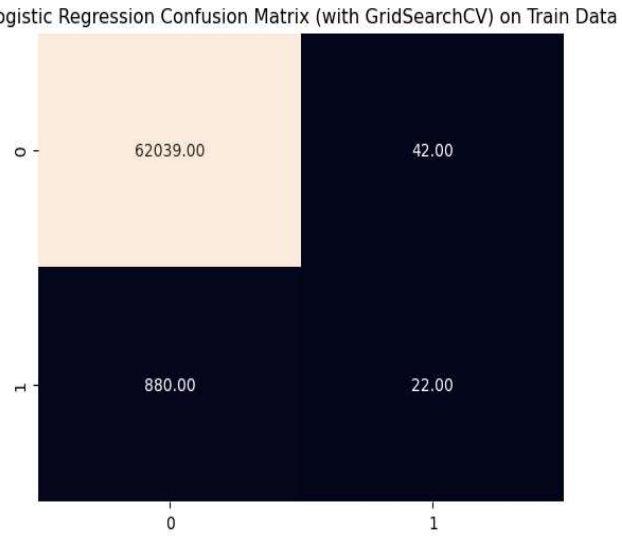


Fig. 4.1E
LR Model Classification Report and Confusion Matrix with GridSearchCV on Train Data

INFERNCES: -

- For Train Data out of 902 number of 1's(Default), only 22 were actually predicted as 1's (Default)
- Also 880 numbers of 1's (Default) were predicted as 0's and 42 number of 0's(Non-Default) were predicted as 1's(Default)
- Recall of 1's(Yes) = 0.02
- Precision of 1's (Yes) = 0.34
- Accuracy = 0.99
- Recall of 0's(No) = 1
- Precision of 0's(No) = 99
- ROC AUC Score = 0.512

Logistic Regression Classification Report and Confusion matrix (with GridSearchCV) on Test Data

Logistic Regression Classification Report(with GridSearchCV) on Test Data

	precision	recall	f1-score	support
0.0	0.99	1.00	0.99	26607
1.0	0.40	0.04	0.08	386
accuracy			0.99	26993
macro avg	0.70	0.52	0.54	26993
weighted avg	0.98	0.99	0.98	26993

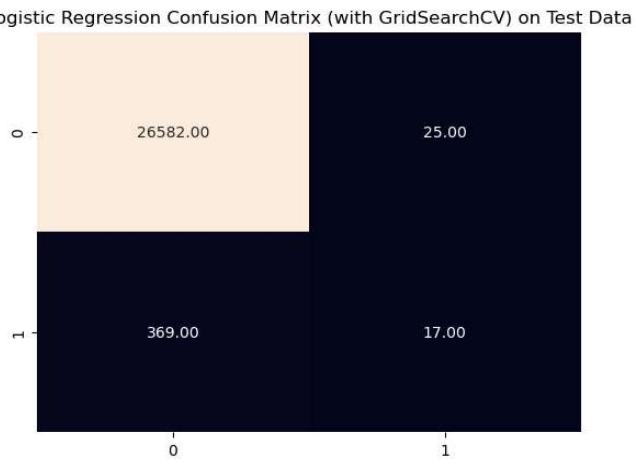


Fig. 4.1F

LR Model Classification Report and Confusion Matrix with GridSearchCV on Test Data

Inference: -

- For Test Data out of 386 number of 1's(Default), only 17 were actually predicted as 1's (Default)
- Also 369 numbers of 1's (Default) were predicted as 0's and 25 number of 0's(Non-Default) were predicted as 1's(Yes)
- Recall of 1's (Default) = 0.04 Recall of 0's (Non - Default) = 1
- Precision of 1's (Default) = 0.40 Precision of 0's (Non - Default) = 0.99
- Accuracy = 0.99 ROC AUC Score = 0.522

We can see from result of both train and test data of LR model with GridSearchCV that model performance to predict defaulters are really poor. Let's try shifting the cutoff threshold which is by default 0.50 for LR model.

Note: - We can see that model Recall of 1 is very low for both train and test. Hence the Logistic Regression model with GridSearchCV is poor or weak model

For Train Optimum Threshold Cutoff = 0.0173

For Test Optimum Threshold Cutoff = 0.0168

Classification Report and Confusion Matrix with Optimum Cutoff Threshold = 0.0173 for Train data

Logistic Regression Model with GridSearch CV (cutoff threshold = 0.017283)

	precision	recall	f1-score	support
0.0	1.00	0.78	0.87	62081
1.0	0.05	0.78	0.09	902
accuracy			0.78	62983
macro avg	0.52	0.78	0.48	62983
weighted avg	0.98	0.78	0.86	62983

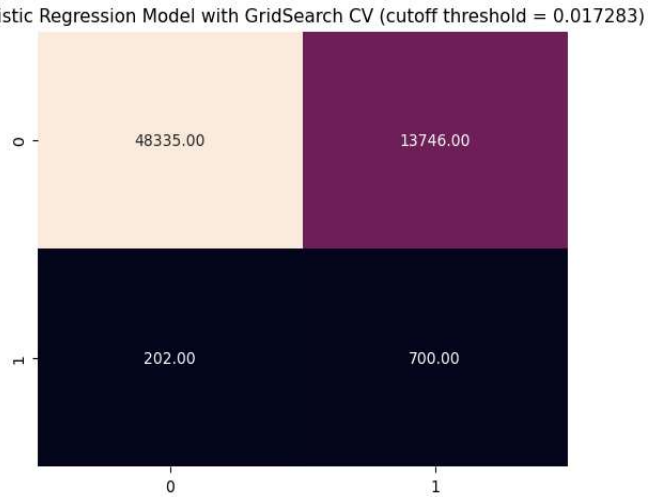


Fig. 4.1G

LR Model Classification Report and Confusion Matrix with GridSearchCV (cutoff=0.0173) on Train Data

INFERNCES: -

- For Train Data out of 902 number of 1's(Default), only 700 were actually predicted as 1's (Default)
- Also 202 numbers of 1's (Default) were predicted as 0's and 13746 number of 0's(Non-Default) were predicted as 1's(Default)
- Recall of 1's(Yes) = 0.78 Recall of 0's(No) = 0.78
- Precision of 1's (Yes) = 0.05 Precision of 0's(No) = 1
- Accuracy = 0.78 ROC AUC score = 0.77
-

Classification Report and Confusion Matrix with Optimum Cutoff Threshold = 0.0168 for Test data

Logistic Regression Model with GridSearch CV (cutoff= 0.0168) on test data

	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

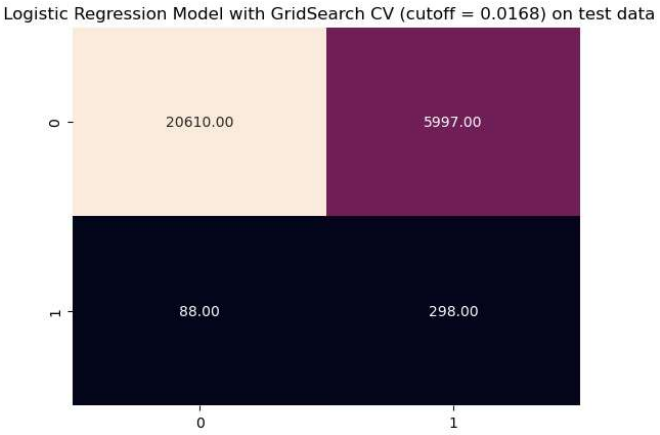


Fig. 4.1H

LR Model Classification Report and Confusion Matrix with GridSearchCV (cutoff=0.0168) on Test Data

Inference: -

- For Test Data out of 386 number of 1's(Default), only 298 were actually predicted as 1's (Default)
- Also 88 numbers of 1's (Default) were predicted as 0's and 5997 number of 0's(Non-Default) were predicted as 1's(Yes)
- Recall of 1's (Default) = 0.77 Recall of 0's (Non - Default) = 0.77
- Precision of 1's (Default) = 0.05 Precision of 0's (Non - Default) = 1
- Accuracy = 0.77 ROC AUC Score = 0.77

From all the Logistic Regression Model i.e., with and without GridSearchCV we can see that after shifting the cutoff threshold model are predicting defaulters much better than cutoff = 0.5

Note: - We can see that model Recall of 1 is very good for both train and test. Hence the Simple Logistic Regression model with optimum cutoff threshold is good model

LINEAR DISRIMINAT ANALYSIS

LDA is a dimensionality reduction technique primarily utilized in supervised classification problems

A. Intercept of the LDA Model

```
The Intercept of the LDA model :-  [-6.05268659]
```

B. Coefficient of each Independent variable for LDA Equation: -

```
The coefficient each varaible of the LDA model :-
 [[-0.057  0.444  0.293  0.381 -0.005 -0.45   1.436 -0.669 -0.261  0.332
   1.026 -0.192  0.13  -0.07  -0.003  0.065 -0.217  0.529  0.061  0.558
   0.466  0.163 -0.194 -0.193 -0.114 -0.701 -0.091  0.129 -0.072  0.009
  -0.007  0.057  0.175]]
```

C. LDA model Equation: -

Equation :

DEFAULT = [-6.053] +(-0.057)* ACCT_AMT_ADDED_12_24M +(0.444)* ACCT_DAYS_IN_DC_12_24M +(0.293)* ACCT_DAYS_IN_REM_12_24M + (0.381)* ACCT_DAYS_IN_TERM_12_24M +(-0.005)* ACCT_INCOMING_DEBT_VS_PAID_0_24M +(-0.45)* ACCT_STATUS +(1.436)* ACCT_WORST_STATUS_0_3M +(-0.669)* ACCT_WORST_STATUS_12_24M +(-0.261)* ACCT_WORST_STATUS_3_6M +(0.332)* ACCT_WORST_STATUS_6_12M +(1.026)* AVG_PAYMENT_SPAN_0_12M +(-0.192)* AVG_PAYMENT_SPAN_0_3M +(0.13)* MERCHANT_CATEGORY +(-0.07)* MERCHANT_GROUP +(-0.003)* HAS_PAID +(0.065)* MAX_PAID_INV_0_12M +(-0.217)* MAX_PAID_INV_0_24M +(0.529)* NUM_ACTIVE_DIV_BY_PAID_INV_0_12M + (0.061)* NUM_ACTIVE_INV +(0.558)* NUM_ARCH_DC_0_12M +(0.466)* NUM_ARCH_DC_12_24M +(0.163)* NUM_ARCH_OK_0_12M +(-0.194)* NUM_ARCH_OK_12_24M +(-0.193)* NUM_ARCH_REM_0_12M +(-0.114)* STATUS_MAX_ARCHIVED_0_6_MONTHS +(-0.701)* STATUS_MAX_ARCHIVED_0_12_MONTHS +(-0.091)* STATUS_MAX_ARCHIVED_0_24_MONTHS +(0.129)* RECOVERY_DEBT +(-0.072)* SUM_CAPITAL_PAID_ACCT_0_12M +(0.009)* SUM_CAPITAL_PAID_ACCT_12_24M +(-0.007)* SUM_PAID_INV_0_12M +(0.057)* TIME_HOURS +(0.175)* AGE_GROUP

LDA Confusion Matrix and Classification Report on train data

Linear Disriminat Analysis Classification Report on Train Data

	precision	recall	f1-score	support
0.0	0.99	0.99	0.99	62081
1.0	0.16	0.19	0.18	902
accuracy			0.97	62983
macro avg	0.58	0.59	0.58	62983
weighted avg	0.98	0.97	0.98	62983

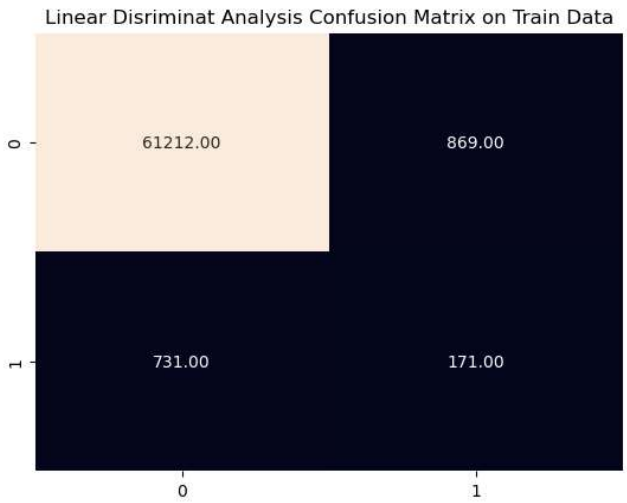


Fig. 5.1A
LDA Model Classification Report and Confusion Matrix on Train Data

INFERNCES: -

- For Train Data out of 902 number of 1's(Default), only 171 were actually predicted as 1's (Default)
- Also 731 numbers of 1's (Default) were predicted as 0's and 869 number of 0's(Non-Default) were predicted as 1's(Default)
- Recall of 1's(Yes) = 0.19
- Precision of 1's (Yes) = 0.16
- Accuracy = 0.97
- Recall of 0's(No) = 0.99
- Precision of 0's(No) = 0.99
- ROC AUC score = 0.59

LDA Confusion Matrix and Classification Report on test data

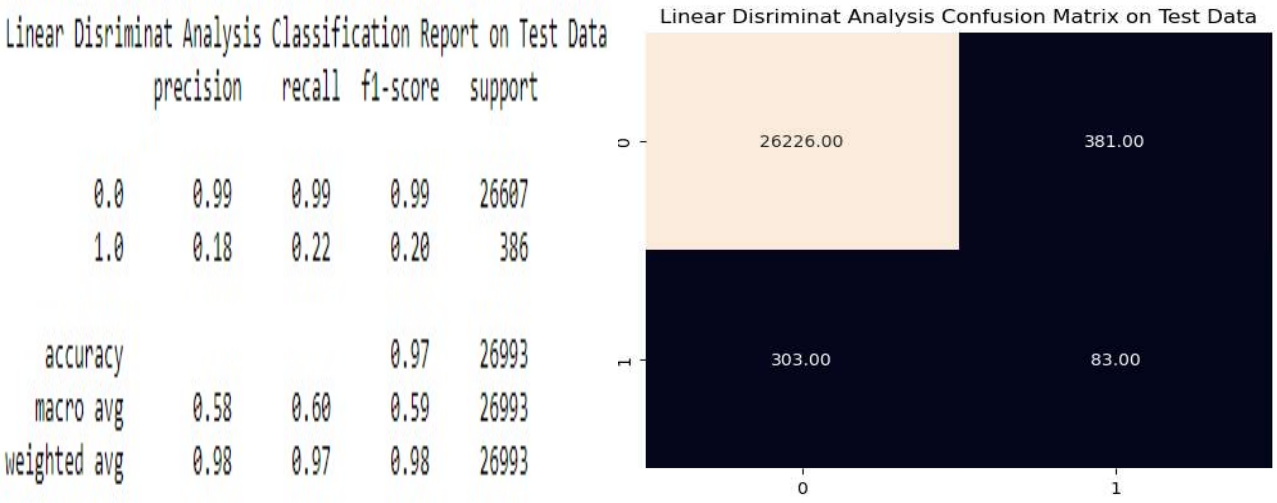


Fig. 5.1B
LDA Model Classification Report and Confusion Matrix on Test Data

Inference: -

- For Test Data out of 386 number of 1's(Default), only 83 were actually predicted as 1's (Default)
- Also 303 numbers of 1's (Default) were predicted as 0's and 381 number of 0's(Non-Default) were predicted as 1's(Yes)
- Recall of 1's (Default) = 0.22
- Precision of 1's (Default) = 0.18
- Accuracy = 0.97
- Recall of 0's (Non - Default) = 0.99
- Precision of 0's (Non - Default) = 0.99
- ROC AUC Score = 0.60

Note: - We can see that model Recall of 1 is very low for both train and test. Hence the Linear Discriminant Analysis model is the weak or poor model

LDA Model ROC AUC Score and ROC plot for Train and Test data

```
ROC AUC Score train Data =  0.59
ROC AUC Score test Data =  0.6
```

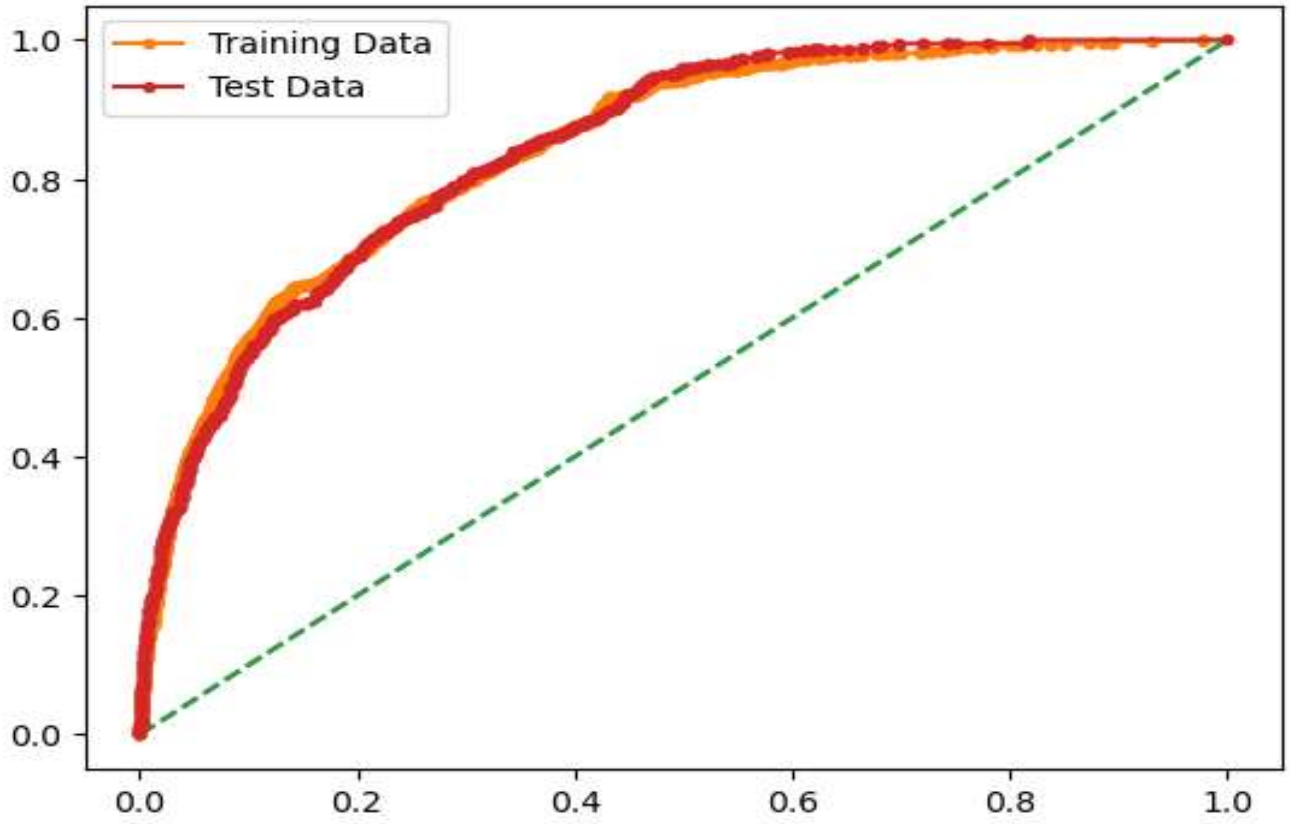


Fig. 5.1C

K-NEAREST NEIGHBOUR (KNN) Model

The k-nearest neighbors (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. a class label is assigned on the basis of a majority vote—i.e. the label that is most frequently represented around a given data point is used.

In KNN Model the data needs to be scaled as it is a Distance Based Algorithm.

KNN Model Confusion Matrix and Classification Report on train data

KNN model Classification Report on Train Data

	precision	recall	f1-score	support
0.0	0.99	1.00	0.99	62081
1.0	0.82	0.05	0.10	902
accuracy			0.99	62983
macro avg	0.90	0.53	0.55	62983
weighted avg	0.98	0.99	0.98	62983

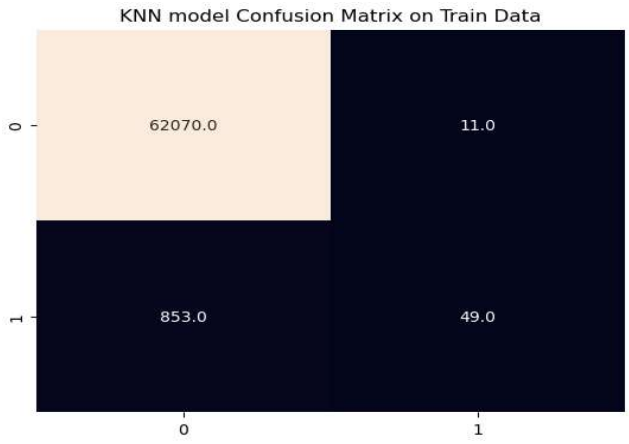


Fig. 6.1A
KNN Model Classification Report and Confusion Matrix on Train Data

INFERNCES: -

- For Train Data out of 902 number of 1's(Default), only 49 were actually predicted as 1's (Default)
- Also 853 numbers of 1's (Default) were predicted as 0's and 11 number of 0's(Non-Default) were predicted as 1's(Default)
- Recall of 1's(Yes) = 0.05
- Precision of 1's (Yes) = 0.82
- Accuracy = 0.99
- Recall of 0's(No) = 1
- Precision of 0's(No) = 0.99
- ROC AUC score = 0.53

KNN Model Confusion Matrix and Classification Report on test data

KNN model Classification Report on Test Data

	precision	recall	f1-score	support
0.0	0.99	1.00	0.99	26607
1.0	0.38	0.02	0.04	386
accuracy			0.99	26993
macro avg	0.68	0.51	0.52	26993
weighted avg	0.98	0.99	0.98	26993

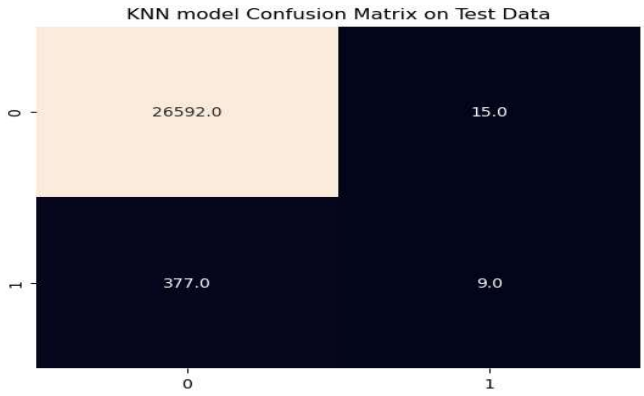


Fig. 6.1B
KNN Model Classification Report and Confusion Matrix on Test Data

Inference: -

- For Test Data out of 386 number of 1's(Default), only 9 were actually predicted as 1's (Default)
- Also 377 numbers of 1's (Default) were predicted as 0's and 15 number of 0's(Non-Default) were predicted as 1's(Yes)
- Recall of 1's (Default) = 0.02 Recall of 0's (Non - Default) = 1
- Precision of 1's (Default) = 0.38 Precision of 0's (Non - Default) = 0.99
- Accuracy = 0.99 ROC AUC Score = 0.51

KNN Model ROC AUC Score and ROC plot for Train and Test data

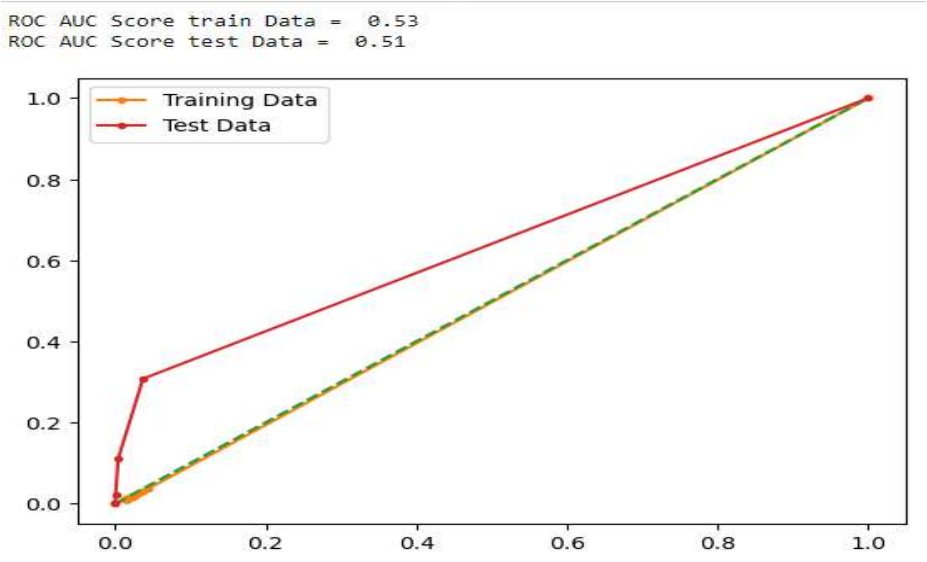


Fig. 6.1C

Note: - We can see that model Recall of 1 is very low for both train and test. Hence the Linear Discriminant Analysis model is the weak or poor model

DECISION TREE

A decision tree is a flowchart-like structure used to make decisions or predictions. It consists of nodes representing decisions or tests on attributes, branches representing the outcome of these decisions, and leaf nodes representing final outcomes or predictions. Each internal node corresponds to a test on an attribute, each branch corresponds to the result of the test, and each leaf node corresponds to a class label or a continuous value.

First we had made a Simple Decision Tree Model: -

Simple Decision Tree Confusion Matrix and Classification Report on train data

Simple Decision Tree Classification Report on Train Data

	precision	recall	f1-score	support
0.0	1.00	1.00	1.00	62076
1.0	1.00	0.94	0.97	907
accuracy			1.00	62983
macro avg	1.00	0.97	0.98	62983
weighted avg	1.00	1.00	1.00	62983

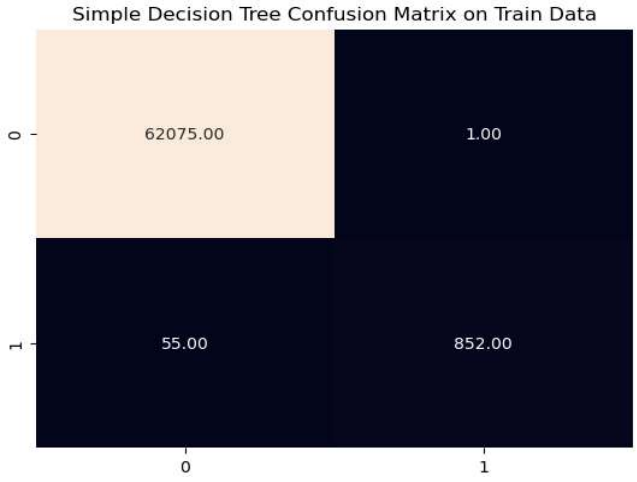


Fig. 7.1A
Simple Decision Tree Model Classification Report and Confusion Matrix on Train Data

INFERNCES: -

- For Train Data out of 902 number of 1's(Default), only 852 were actually predicted as 1's (Default)
- Also 55 numbers of 1's (Default) were predicted as 0's and 1 number of 0's(Non-Default) were predicted as 1's(Default)
- Recall of 1's(Yes) = 0.94
- Recall of 0's(No) = 0.1
- Precision of 1's (Yes) = 1
- Precision of 0's(No) = 1
- Accuracy = 1
- ROC AUC score = 0.97

Note: - As we know Simple Decision Tree Model run until it achieve 100 % pure nodes. Hence it is able to distinguish all the defaulters and non-defaulters. But when we valid this model on test data, it isn't able to distinguish between defaulters and non-defaulters as purely as training model. Let's see.

Simple Decision Tree Confusion Matrix and Classification Report on test data

Simple Decision Tree Classification Report on Test Data

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

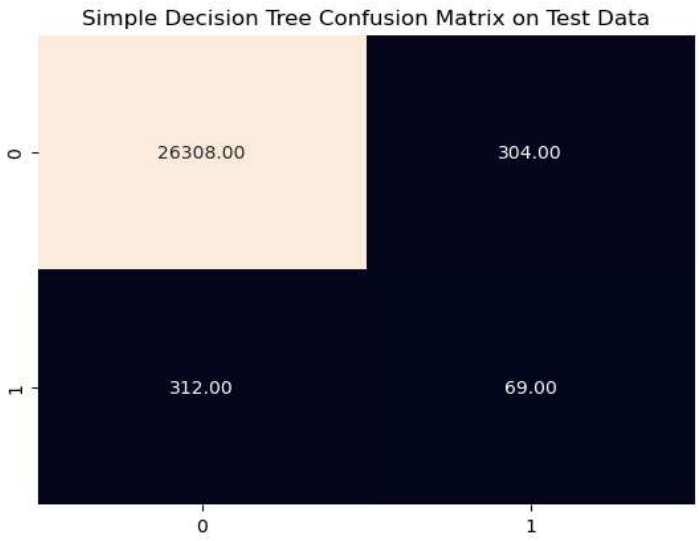


Fig. 7.1B
Simple Decision Tree Model Classification Report and Confusion Matrix on Test Data

Inference: -

- For Test Data out of 386 number of 1's(Default), only 69 were actually predicted as 1's (Default)
- Also 312 numbers of 1's (Default) were predicted as 0's and 304 number of 0's(Non-Default) were predicted as 1's(Yes)
- Recall of 1's (Default) = 0.18
- Precision of 1's (Default) = 0.18
- Accuracy = 0.98
- Recall of 0's (Non - Default) = 0.99
- Precision of 0's (Non - Default) = 0.99
- ROC AUC Score = 0.58

Note: - From above two inferences of train and test we can say that the Simple Decision Tree Model is an Over Fitted Model as it performs very good during training but doesn't perform well at test.

Decision Tree ROC AUC Score and ROC plot for Train and Test data

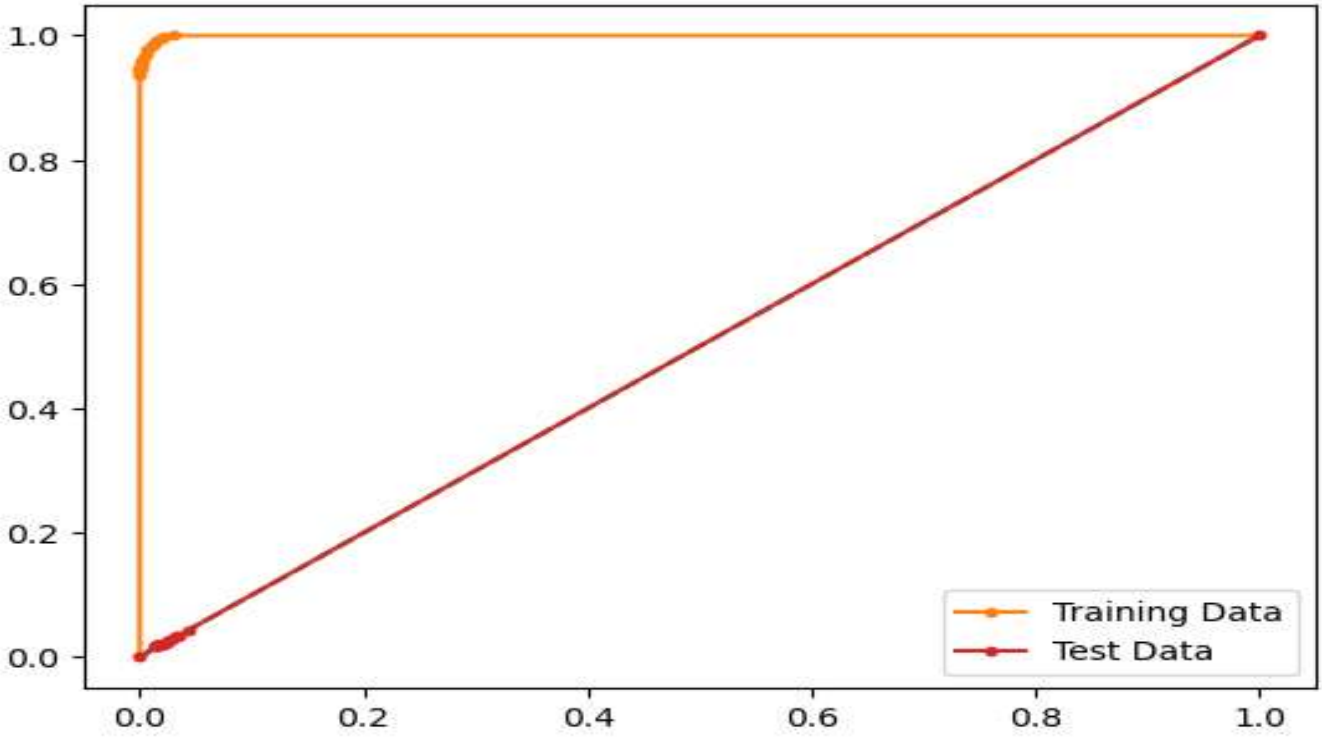


Fig. 7.1C

Decision Tree Model with GridSearchCV

Best Parameters: -

```
DecisionTreeClassifier
DecisionTreeClassifier(ccp_alpha=0.01, max_depth=10, max_features='auto',
                       min_samples_leaf=10, random_state=1)
```

Decision Tree Model (with GridSearchCV) Classification Report for train and test data: -

	precision	recall	f1-score	support
0.0	0.99	1.00	0.99	62076
1.0	0.00	0.00	0.00	907
accuracy			0.99	62983
macro avg	0.49	0.50	0.50	62983
weighted avg	0.97	0.99	0.98	62983

	precision	recall	f1-score	support
0.0	0.99	1.00	0.99	26612
1.0	0.00	0.00	0.00	381
accuracy			0.99	26993
macro avg	0.49	0.50	0.50	26993
weighted avg	0.97	0.99	0.98	26993

Fig. 7.1D

Decision Tree Model (with GridSearchCV) Classification Report for Train and Test Data

We can see from classification report that the Decision Tree Model (with GridSearchCV) is not able classify Defaulters from Non-Defaulters. It can be due to under sampling of the Defaulters i.e., Number of Defaulter's with respect to Defaulters are very low due to which is isn't not bale to distinguish Defaulters from Non-Defaulters. There are 88688 number of Non-Defaulters and only 1288 Defaulters in a dataset.

Here in order to increase the number of Defaulters without messing with the original data we had used **SMOTE (Synthetic Minority Oversampling Technique).**

SMOTE aims to balance class distribution by randomly increasing minority class examples by replicating them. SMOTE synthesizes new minority instances between existing minority instances. It generates the virtual training records by linear interpolation for the minority class. These synthetic training records are generated by randomly selecting one or more of the k-nearest neighbors for each example in the minority class.

SMOTE is only carried out on training data.

Decision Tree Model (GridSearchCV) (SMOTE) on training data: -

Decision Tree Model (with SMOTE) Classification Report on Train Data

	precision	recall	f1-score	support
0.0	0.82	0.67	0.74	62076
1.0	0.72	0.85	0.78	62076
accuracy			0.76	124152
macro avg	0.77	0.76	0.76	124152
weighted avg	0.77	0.76	0.76	124152

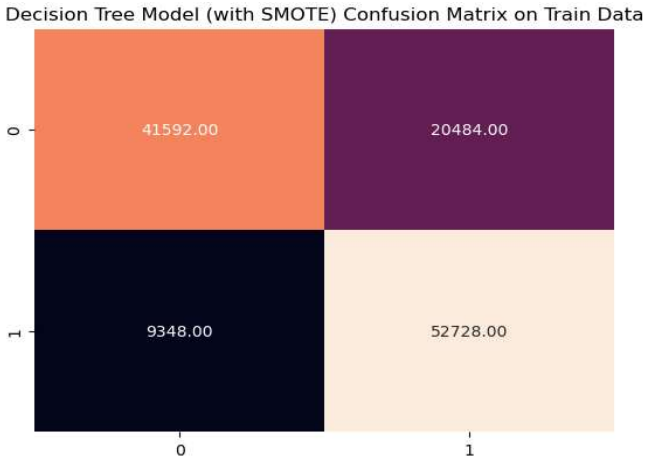


Fig. 7.1E
Decision Tree Model (with SMOTE) Classification Report for Train Data

INFERNCES: -

- For Train Data out of 62076 number of 1's(Default), only 52728 were actually predicted as 1's (Default)
- Also 9348 numbers of 1's (Default) were predicted as 0's and 20484 number of 0's(Non-Default) were predicted as 1's(Default)
- Recall of 1's(Yes) = 0.85
- Recall of 0's(No) = 0.67
- Precision of 1's (Yes) = 0.72
- Precision of 0's(No) = 0.82
- Accuracy = 0.76
- ROC AUC score = 0.76

Decision Tree Model (after SMOTE) on test data: -

Decision Tree Model (after SMOTE) Classification Report on Test Data

	precision	recall	f1-score	support
0.0	0.99	0.67	0.80	26612
1.0	0.03	0.73	0.06	381
accuracy			0.67	26993
macro avg	0.51	0.70	0.43	26993
weighted avg	0.98	0.67	0.79	26993

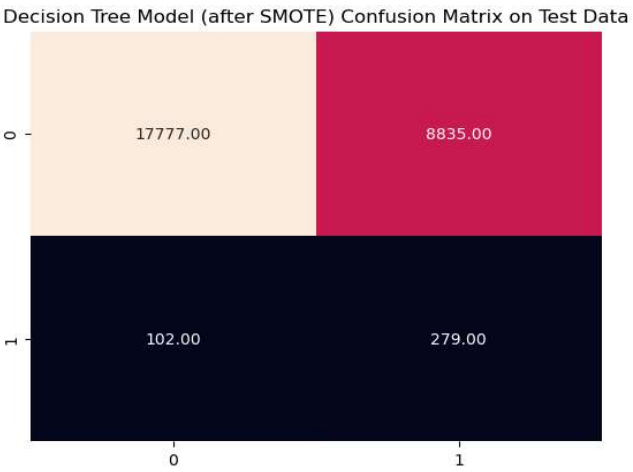


Fig. 7.1F
Decision Tree Model (after SMOTE) Classification Report on Test Data

Inference: -

- For Test Data out of 386 number of 1's(Default), 279 were actually predicted as 1's (Default)
- Also 102 numbers of 1's (Default) were predicted as 0's and 8835 number of 0's(Non-Default) were predicted as 1's(Yes)
- Recall of 1's (Default) = 0.73 Recall of 0's (Non - Default) = 0.67
- Precision of 1's (Default) = 0.0.03 Precision of 0's (Non - Default) = 0.99
- Accuracy = 0.67 ROC AUC Score = 0.70

Decision Tree Model (after SMOTE) ROC AUC Score and ROC plot for Train and Test data

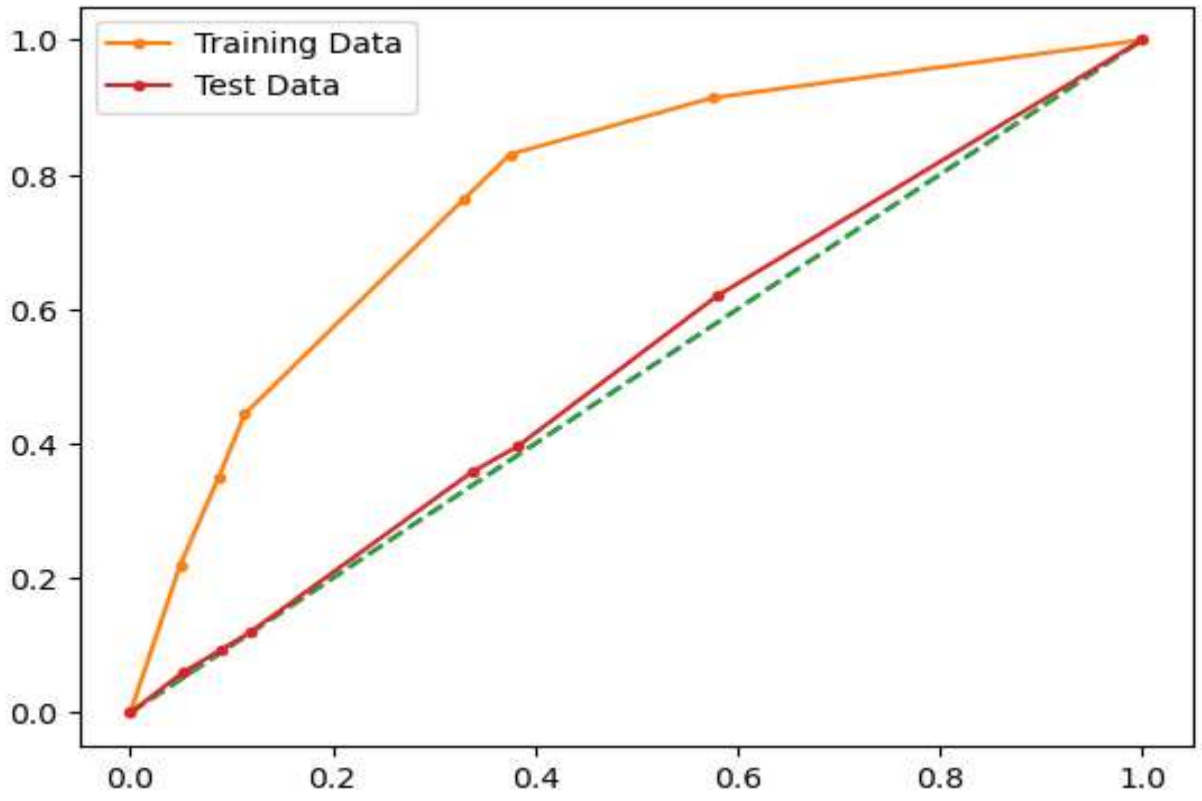


Fig. 7.1G

Note: - We can see that model Recall of 1 is very good train and test. Hence the Decision Tree Model (with SMOTE) model is good model.

Ensemble Modelling

Boosting: -

Boosting is an ensemble modeling technique that attempts to build a strong classifier from the number of weak classifiers. It is done by building a model by using weak models in series. Firstly, a model is built from the training data. Then the second model is built which tries to correct the errors present in the first model. This procedure is continued and models are added until either the complete training data set is predicted correctly or the maximum number of models are added.

There are 2 Types of Boosting: -

1. **Gradient Boosting**: - It is a boosting technique that builds a final model from the sum of several weak learning algorithms that were trained on the same dataset. It operates on the idea of stage wise addition.
2. **Adaptive Boosting**: - Adaptive Boost is a boosting algorithm that also works on the principle of the stage wise addition method where multiple weak learners are used for getting strong learners.

Here we will be performing Adaptive Boosting: -

Adaptive Boosting Model Confusion Matrix and Classification Report on train data

Adaptive Boosting Model Classification Report on Train Data

	precision	recall	f1-score	support
0.0	0.99	1.00	0.99	62076
1.0	0.23	0.05	0.08	907
accuracy			0.98	62983
macro avg	0.61	0.52	0.54	62983
weighted avg	0.98	0.98	0.98	62983

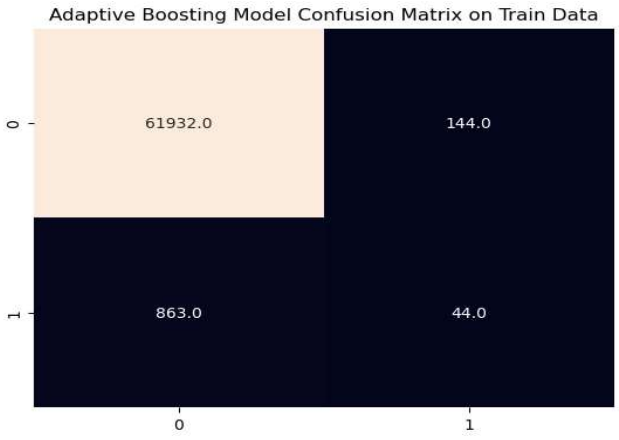


Fig. 8.1A
Adaptive Boosting Classification Report and Confusion Matrix for Train Data

INFERNCES: -

- For Train Data out of 907 number of 1's(Default), only 44 were actually predicted as 1's (Default)
- Also 844 numbers of 1's (Default) were predicted as 0's and 144 number of 0's(Non-Default) were predicted as 1's(Default)
- Recall of 1's(Yes) = 0.05
- Precision of 1's (Yes) = 0.23
- Accuracy = 0.98
- Recall of 0's(No) = 1
- Precision of 0's(No) = 0.99
- ROC AUC score = 0.52

Adaptive Boosting Model Confusion Matrix and Classification Report on test data

Adaptive Boosting Model Classification Report on Test Data

	precision	recall	f1-score	support
0.0	0.99	1.00	0.99	26612
1.0	0.22	0.04	0.06	381
accuracy			0.98	26993
macro avg	0.60	0.52	0.53	26993
weighted avg	0.98	0.98	0.98	26993

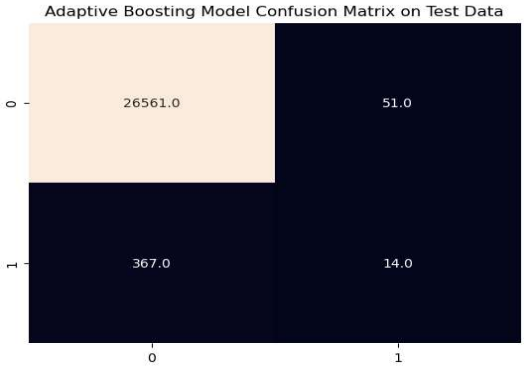


Fig. 8.1B
Adaptive Boosting Classification Report and Confusion Matrix for Test Data

Inference: -

- For Test Data out of 386 number of 1's(Default), only 14 were actually predicted as 1's (Default)
- Also 367 numbers of 1's (Default) were predicted as 0's and 51 number of 0's(Non-Default) were predicted as 1's(Yes)
- Recall of 1's (Default) = 0.04
- Recall of 0's (Non - Default) = 1
- Precision of 1's (Default) = 0.22
- Precision of 0's (Non - Default) = 0.99
- Accuracy = 0.98
- ROC AUC Score = 0.517

We can see that the our model is not performing better in either train or test as recall of default is only 0.05 and 0.04 respectively

Adaptive Boosting Model ROC AUC Score and ROC plot for Train and Test data

ROC AUC Score train Data = 0.523
ROC AUC Score test Data = 0.517

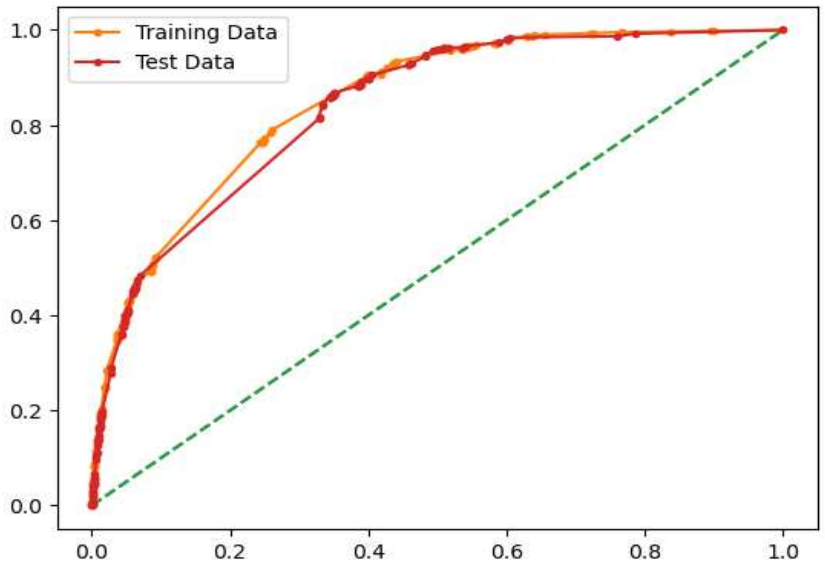


Fig. 8.1C

Note: - We can see that model Recall of 1 is very low for both train and test. Hence the Simple Adaptive Boosting model is the weak or poor model

We can see from classification report that the Adaptive Boosting Model, it is able to classify Non-Defaulters as Non-Defaulters but it isn't able to classify Defaulters as Defaulters. It is due to under sampling of the Defaulters i.e., Number of Defaulter's with respect to Defaulters are very low i.e., 88688 number of Non-Defaulters and only 1288 number of Defaulters due to which is not able to distinguish Defaulters from Non-Defaulters.

Here in order to increase the number of Defaulters without messing with the original data we had used SMOTE (Synthetic Minority Oversampling Technique).

SMOTE aims to balance class distribution by randomly increasing minority class examples by replicating them. SMOTE synthesizes new minority instances between existing minority instances. It generates the virtual training records by linear interpolation for the minority class. These synthetic training records are generated by randomly selecting one or more of the k-nearest neighbors for each example in the minority class.

SMOTE is only carried out on training data

Train Dataset Before SMOTE: -

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

Train Dataset After performing SMOTE: -

```
After OverSampling, the shape of train_X: (124152, 33)
After OverSampling, the shape of train_y: (124152,)

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

Adaptive Boosting Model (with SMOTE) on training data: -

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

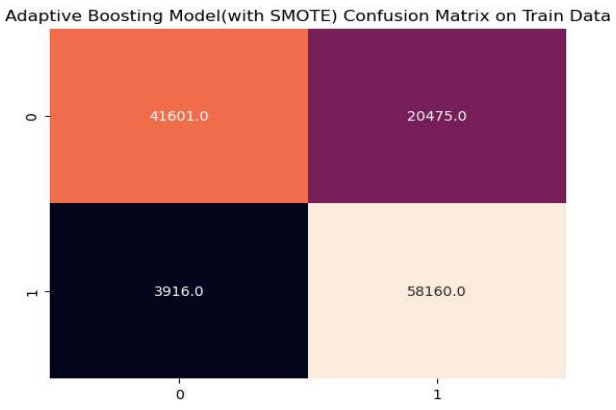


Fig. 8.1D
Adaptive Boosting (with SMOTE) Classification Report and Confusion Matrix for Train Data

INFERNCES: -

- For Train Data out of 62076 number of 1's(Default), only 58160 were actually predicted as 1's (Default)
- Also 3916 numbers of 1's (Default) were predicted as 0's and 20475 number of 0's(Non-Default) were predicted as 1's(Default)
- Recall of 1's(Yes) = 0.74 Recall of 0's(No) = 0.67
- Precision of 1's (Yes) = 0.74 Precision of 0's(No) = 0.91
- Accuracy = 0.80 ROC AUC score = 0.80

Adaptive Boosting Model (After SMOTE) on test data: -

Adaptive Boosting Model (after SMOTE) Classification Report on Test Data

	precision	recall	f1-score	support
0.0	1.00	0.67	0.80	26612
1.0	0.03	0.82	0.07	381
accuracy			0.67	26993
macro avg	0.52	0.74	0.43	26993
weighted avg	0.98	0.67	0.79	26993

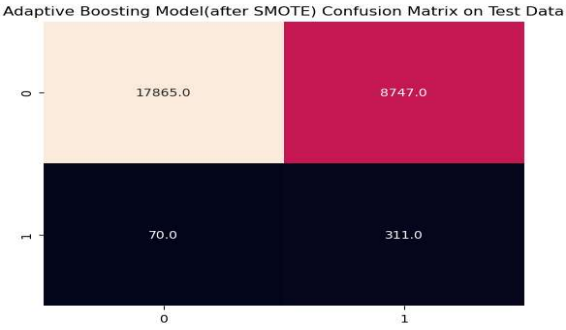


Fig. 8.1E

Adaptive Boosting (after SMOTE) Classification Report and Confusion Matrix for Test Data

Inference: -

- For Test Data out of 386 number of 1's(Default), 311 were actually predicted as 1's (Default)
- Also 70 numbers of 1's (Default) were predicted as 0's and 8747 number of 0's(Non-Default) were predicted as 1's(Yes)
- Recall of 1's (Default) = 0.82 Recall of 0's (Non - Default) = 0.67
- Precision of 1's (Default) = 0.03 Precision of 0's (Non - Default) = 1
- Accuracy = 0.67 ROC AUC Score = 0.74

Adaptive Boosting Model (with SMOTE) ROC AUC Score and ROC plot for Train and Test data

```
ROC AUC Score train Data =  0.8
ROC AUC Score test Data =  0.74
```

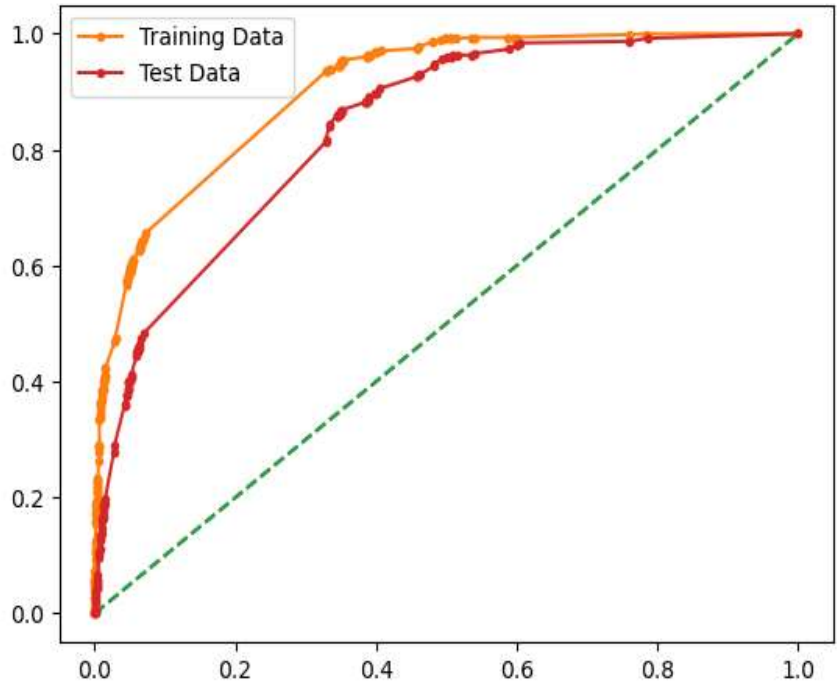


Fig. 8.1F

Note: - We can see that model Recall of 1 is very good for both train and test. Hence the Adaptive Boosting model (with SMOTE) is the good/strong model.

Model Validation

In this Dataset the prime concern is to detect Defaulters , so the Recall will be the most Important parameter in order to choose the best model followed by ROC AUC Score , F1 Score and Accuracy.

Recall: - The recall is calculated as the ratio between the numbers of Positive samples correctly classified as Positive to the total number of Positive samples. The recall measures the model's ability to detect positive samples. The higher the recall, the more positive samples detected

RECALL = True Positive TP /(True Positive TP + False Negative FN)

ROC AUC Score: - ROC AUC (Area under the ROC Curve) is a measure of performance across all possible classification thresholds

F1 score: - The F1 score is calculated as the harmonic mean of precision and recall

F1 Score= 2 * (Precision * Recall) / (Precision + Recall)

Accuracy: - Accuracy is used to measure the performance of the model. It is the ratio of Total correct instances to the total instances

Accuracy = True Positive TP + True Negative TN / Total Sample

where

TP = Default Predicted as Default, FN = Default Predicted as Non Default

FP = Non Default Predicted as Default TN = Non Default Predicted as Non Default

Note: - In this problem we have to reduce the Type 1 Error which is False Negative which will subsequently increase the recall

Comparision Between Different Classification Model						
Model Performance Parameter		**Recall**	**ROC AUC Score**	**F1 Score**	**Accuracy**	**Ranking**
Simple Logistic Regression	**Train**	0.81	0.769	0.08	0.72	2
Optimum Cutoff Threshold	**Test**	0.84	0.77	0.08	0.71	
Logistic Regression (GridSearchCV)	**Train**	0.78	0.77	0.09	0.78	4
Optimum Cutoff Threshold	**Test**	0.77	0.77	0.09	0.77	
LDA Model	**Train**	0.19	0.59	0.18	0.97	5
	Test	0.22	0.6	0.2	0.97	
KNN Model	**Train**	0.05	0.53	0.1	0.99	7
	Test	0.02	0.51	0.04	0.99	
Simple Decision Tree	**Train**	0.94	0.97	0.97	1	8
	Test	0.18	0.58	0.18	0.98	
Decision Tree (SMOTE)	**Train**	0.85	0.76	0.78	0.76	3
	Test	0.73	0.7	0.06	0.67	
Adaptive Boosting	**Train**	0.05	0.52	0.08	0.98	6
	Test	0.04	0.517	0.06	0.98	
Adaptive Boosting (SMOTE)	**Train**	0.94	0.8	0.83	0.8	1
	Test	0.82	0.74	0.07	0.67	

Fig. 9.1A

From Above table we see that highest Recall if of the Adaptive Boosting Model with SMOTE . It also has highest ROC AUC Score as well as F1 score. This model is able to classify most number Defaulters as Defaulters.

Next best model is the Simple Logistic Regression Model with Optimum Cutoff Threshold or Decision Tree with SMOTE as it has the next highest Recall, ROC AUC score and F1 score

Business Recommendation

- We can review the policy for the credit card given to customer who are in age group of 18 to 30 or we can say young customer as they possess high chances of default.
- If customer is taking longer time than the usual pattern of making payment, we can reduce the credit card limit.
- We need to review the policy on turning the credit card account into worst status i.e., we can either reduce the balance limit or closely monitor those accounts which has chances to turn into worst status.
- Probability to Default increases if the customer credit card has been terminated for 2 days in the past so we can reduce the card limit of those customer whose account has been terminated in the past
- We can reduce the number of active invoice for a credit card.
- If acct_incoming_debt_vs_paid is increasing for any account, we can put hold on those card until full payment.
- If num_active_div_by_paid_inv_0_12m is greater than 0.2, than customer probability to default increases. For such customer we can give reminder once there num_active_div_by_paid_inv increase from 0.2.
- If most of the credit card payments are coming from either youth shoes/clothing or entertainment, we need to closely monitor such customers card as those might be used by some young person in customer's family
- There are some Credit Card users who has high number of active invoices but they have not defaulted. These problems can be due to non-updating of the data in the report. Hence we need to update and maintain record regularly.
- We can also provide some gift coupons to the customers who are paying their bills on time a uses card very frequently.
- We can review the card limit based on recent expenditure and payment pattern of the customers.