CAPSTONE PROJECT

FINAL SUBMISSION

BUSINESS REPORT

Name and Batch – Shraddha Suman Guru JAN 22A

CONTENTS

LIST OF FIGURES

LIST OF TABLES

Introduction

Defining problem statement

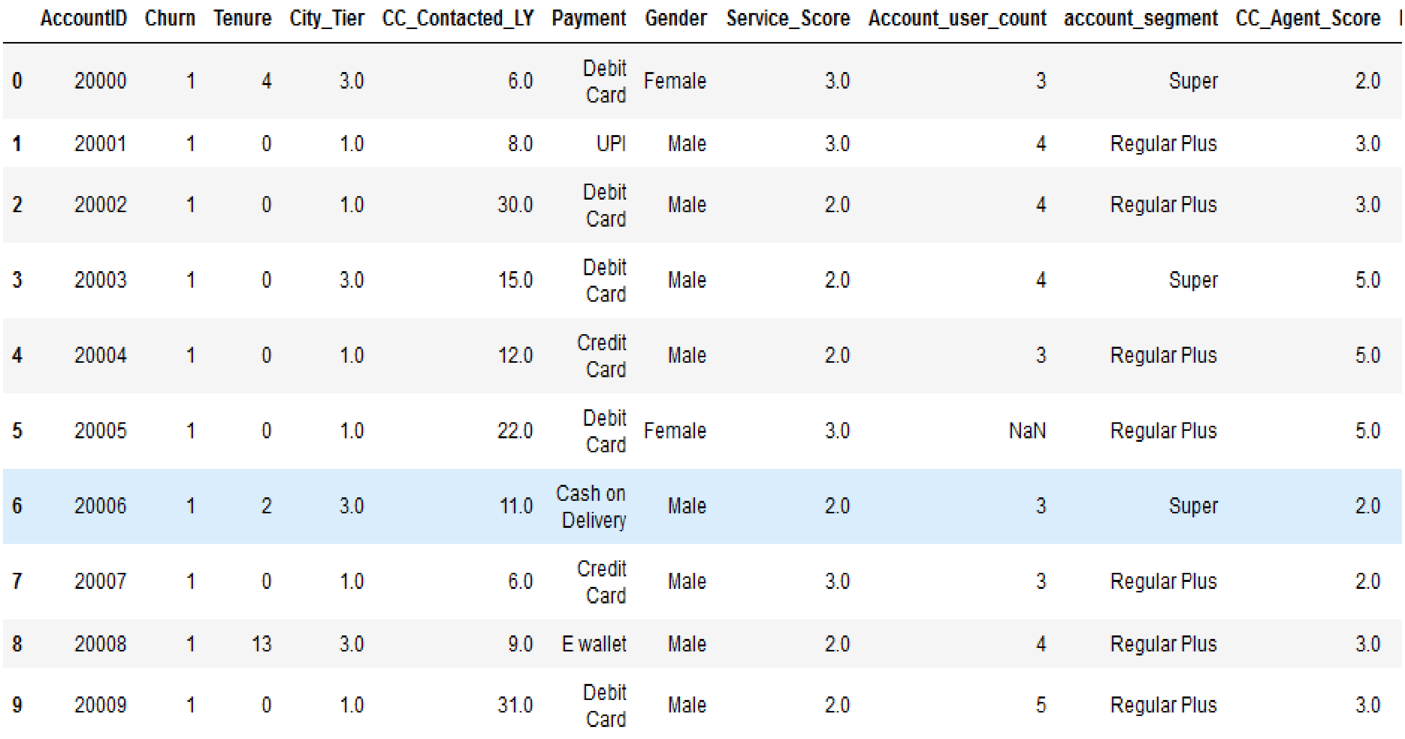
An E Commerce company or DTH (you can choose either of these two domains) provider is facing a lot of competition in the current market and it has become a challenge to retain the existing customers in the current situation. Hence, the company wants to develop a model through which they can do churn prediction of the accounts and provide segmented offers to the potential churners. In this company, account churn is a major thing because 1 account can have multiple customers. Hence by losing one account the company might be losing more than one customer.

Dataset for Problem: Customer Churn Data

Data Dictionary

Variable	Description
AccountID	account unique identifier
Churn	account churn flag (Target)
Tenure	Tenure of account
City_Tier	Tier of primary customer's city
CC_Contacted_L12m	How many times all the customers of the account has contacted customer care in last 12months
Payment	Preferred Payment mode of the customers in the account
Gender	Gender of the primary customer of the account
Service_Score	Satisfaction score given by customers of the account on service provided by company
Account_user_count	Number of customers tagged with this account
account_segment	Account segmentation on the basis of spend
CC_Agent_Score	Satisfaction score given by customers of the account on customer care service provided by company
Marital_Status	Marital status of the primary customer of the account
rev_per_month	Monthly average revenue generated by account in last 12 months
Complain_l12m	Any complaints has been raised by account in last 12 months
rev_growth_yoy	revenue growth percentage of the account (last 12 months vs last 24 to 13 month)
coupon_used_l12m	How many times customers have used coupons to do the payment in last 12 months
Day_Since_CC_connect	Number of days since no customers in the account has contacted the customer care
cashback_l12m	Monthly average cashback generated by account in last 12 months
Login_device	Preferred login device of the customers in the account

Need of the study/project



	AccountID	Churn	Tenure	City_Tier	CC_Contacted_LY	Payment	Gender	Service_Score	Account_user_count	account_segment	CC_Agent_Score
0	20000	1	4	3.0	6.0	Debit Card	Female	3.0	3	Super	2.0
1	20001	1	0	1.0	8.0	UPI	Male	3.0	4	Regular Plus	3.0
2	20002	1	0	1.0	30.0	Debit Card	Male	2.0	4	Regular Plus	3.0
3	20003	1	0	3.0	15.0	Debit Card	Male	2.0	4	Super	5.0
4	20004	1	0	1.0	12.0	Credit Card	Male	2.0	3	Regular Plus	5.0
5	20005	1	0	1.0	22.0	Debit Card	Female	3.0	NaN	Regular Plus	5.0
6	20006	1	2	3.0	11.0	Cash on Delivery	Male	2.0	3	Super	2.0
7	20007	1	0	1.0	6.0	Credit Card	Male	3.0	3	Regular Plus	2.0
8	20008	1	13	3.0	9.0	E wallet	Male	2.0	4	Regular Plus	3.0
9	20009	1	0	1.0	31.0	Debit Card	Male	2.0	5	Regular Plus	3.0

Table 1- Dataset

The above table represents the data that we are going to work on. We are provided with variables like tenure, customers contacted in the past 12 months, Gender of the customer, Marital Status etc. Using these variables we have to calculate the target variable ‘Churn’.

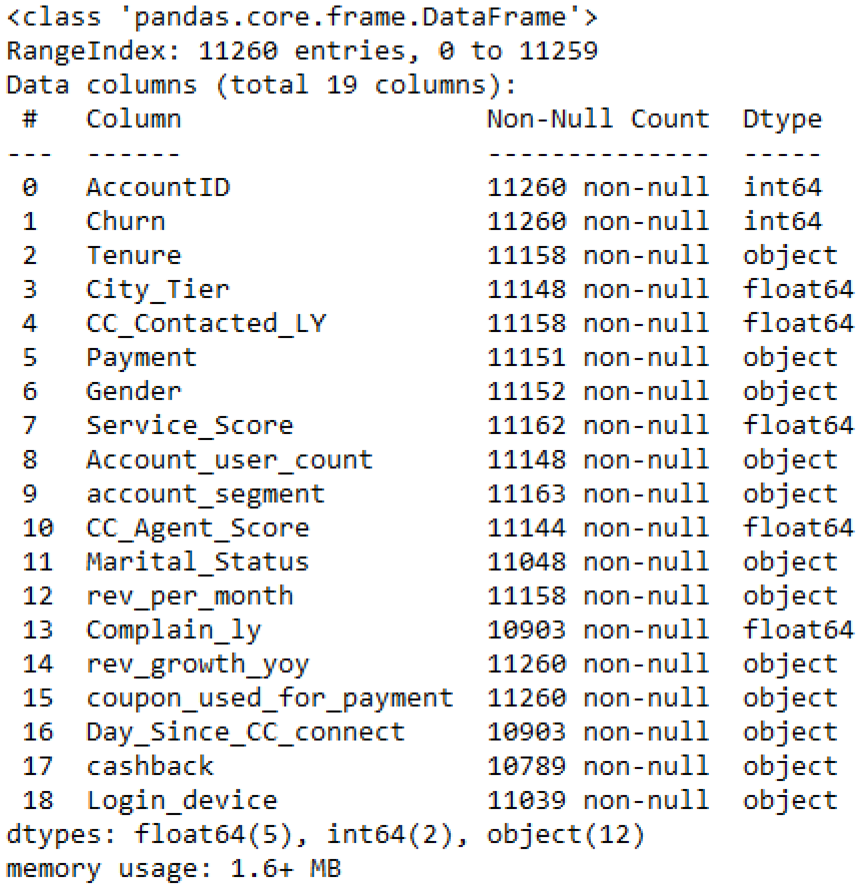
As the above table shows, the data has a total of 19 columns and 11260 rows

Understanding business/social opportunity

We have been assigned to develop a churn prediction model for this company and provide business recommendations on the campaign.

EDA and Business Implication

Both visual and non-visual understanding of the data

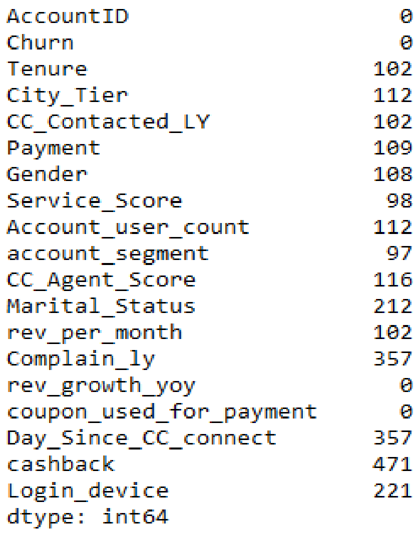


```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 11260 entries, 0 to 11259
Data columns (total 19 columns):
 #   Column                   Non-Null Count  Dtype  
---  ------                   --------------  -----  
 0   AccountID                11260 non-null  int64  
 1   Churn                    11260 non-null  int64  
 2   Tenure                   11158 non-null  object 
 3   City_Tier                11148 non-null  float64
 4   CC_Contacted_LY          11158 non-null  float64
 5   Payment                  11151 non-null  object 
 6   Gender                   11152 non-null  object 
 7   Service_Score            11162 non-null  float64
 8   Account_user_count       11148 non-null  object 
 9   account_segment          11163 non-null  object 
 10  CC_Agent_Score           11144 non-null  float64
 11  Marital_Status           11048 non-null  object 
 12  rev_per_month            11158 non-null  object 
 13  Complain_ly              10903 non-null  float64
 14  rev_growth_yoy           11260 non-null  object 
 15  coupon_used_for_payment  11260 non-null  object 
 16  Day_Since_CC_connect     10903 non-null  object 
 17  cashback                 10789 non-null  object 
 18  Login_device             11039 non-null  object 
dtypes: float64(5), int64(2), object(12)
memory usage: 1.6+ MB
```

Table 2- Info of the dataset

So we can observe from the above table that the dataset contains seven numerical datatypes and twelve categorical datatypes. We can also observe that some of the variables that should be numerical in nature are actually categorical type .For example, the attributes like Tenure, account_user_count, rev_per_month etc. This means something is suspicious within these variables and we need to clean the data.

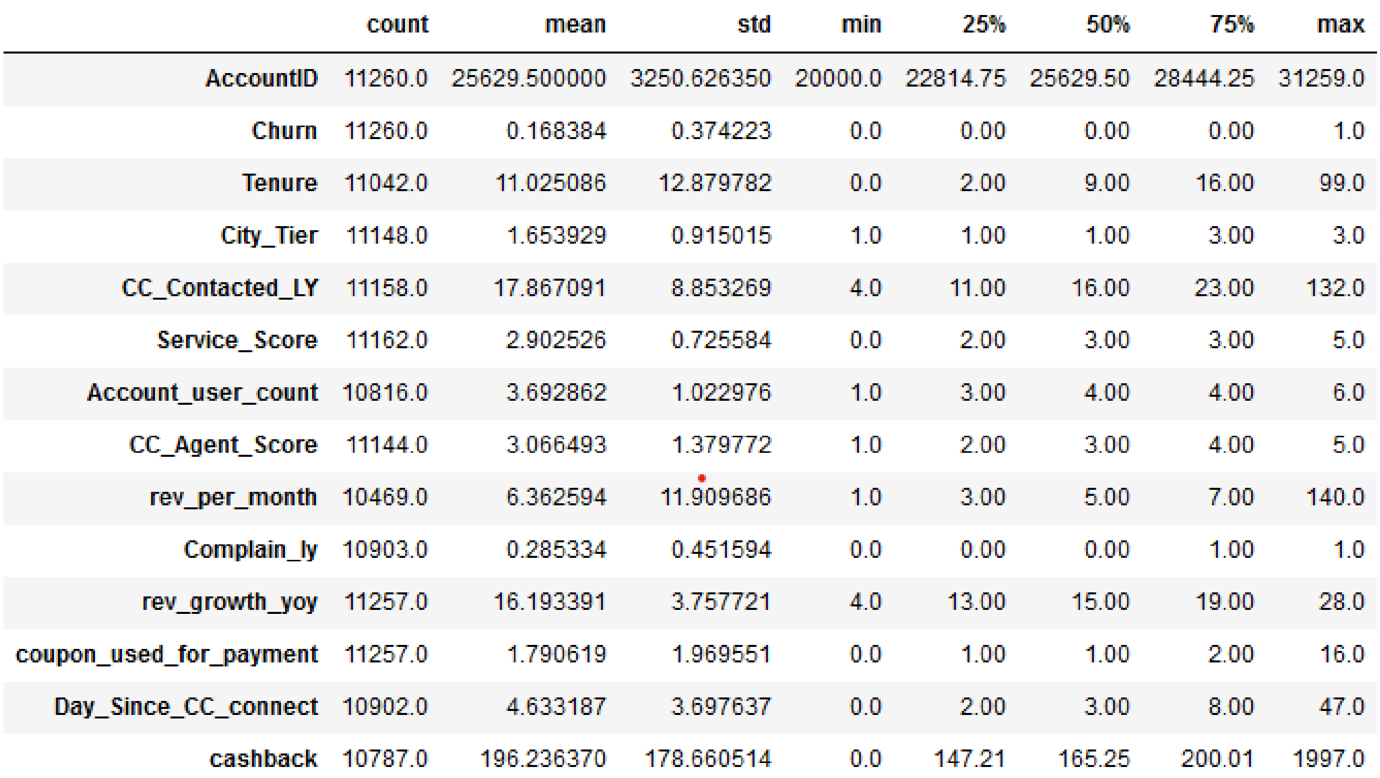
This dataset has 19 columns and 11260 rows.
Let’s check the number of nulls per column.



```
AccountID                    0
Churn                        0
Tenure                     102
City_Tier                  112
CC_Contacted_LY            102
Payment                    109
Gender                     108
Service_Score               98
Account_user_count         112
account_segment             97
CC_Agent_Score             116
Marital_Status             212
rev_per_month              102
Complain_ly                357
rev_growth_yoy               0
coupon_used_for_payment      0
Day_Since_CC_connect       357
cashback                   471
Login_device               221
dtype: int64
```

So we have null values in the data. The total number of null values present in the dataset is 2676.
The number of rows multiplied by number of columns is given by the size function in Python which is also called as the total number of cells present in the dataset. So the total number of cells are 213940
The null values cover 2.04% of the data.

Also, we have no duplicated rows.



	count	mean	std	min	25%	50%	75%	max
AccountID	11260.0	25629.500000	3250.626350	20000.0	22814.75	25629.50	28444.25	31259.0
Churn	11260.0	0.168384	0.374223	0.0	0.00	0.00	0.00	1.0
Tenure	11042.0	11.025086	12.879782	0.0	2.00	9.00	16.00	99.0
City_Tier	11148.0	1.653929	0.915015	1.0	1.00	1.00	3.00	3.0
CC_Contacted_LY	11158.0	17.867091	8.853269	4.0	11.00	16.00	23.00	132.0
Service_Score	11162.0	2.902526	0.725584	0.0	2.00	3.00	3.00	5.0
Account_user_count	10816.0	3.692862	1.022976	1.0	3.00	4.00	4.00	6.0
CC_Agent_Score	11144.0	3.066493	1.379772	1.0	2.00	3.00	4.00	5.0
rev_per_month	10469.0	6.362594	11.909686	1.0	3.00	5.00	7.00	140.0
Complain_ly	10903.0	0.285334	0.451594	0.0	0.00	0.00	1.00	1.0
rev_growth_yoy	11257.0	16.193391	3.757721	4.0	13.00	15.00	19.00	28.0
coupon_used_for_payment	11257.0	1.790619	1.969551	0.0	1.00	1.00	2.00	16.0
Day_Since_CC_connect	10902.0	4.633187	3.697637	0.0	2.00	3.00	8.00	47.0
cashback	10787.0	196.236370	178.660514	0.0	147.21	165.25	200.01	1997.0

Table 3- Description of numeric attributes

Inferences

(i) The mean value of Tenure is around 11 years and the standard deviation is around 12 years which is significantly high. The minimum value of tenure is 0 years and the maximum is 99 years which is surely an outlier. As mean is greater than median, the distribution is right skewed.

(ii) Around 18 customers contacted on an average on the past 12 months with a standard deviation of 9 customers. The difference between minimum and maximum customers contacted is high and the distribution is right skewed.

(iii) The revenue per month has an average of 6.4L with a standard deviation of 12L. The revenue ranges from a minimum of 1L per month to a maximum of 140L per month. The distribution is right skewed.

(iv) The average cash back is 196 rupees with a standard deviation of 179 rupees. The cash back ranges from a minimum of 0 rupees to a maximum of 2k rupees. The distribution is right skewed.

Univariate and Bivariate Analysis

Let's have a look on the histogram of the numeric attributes.

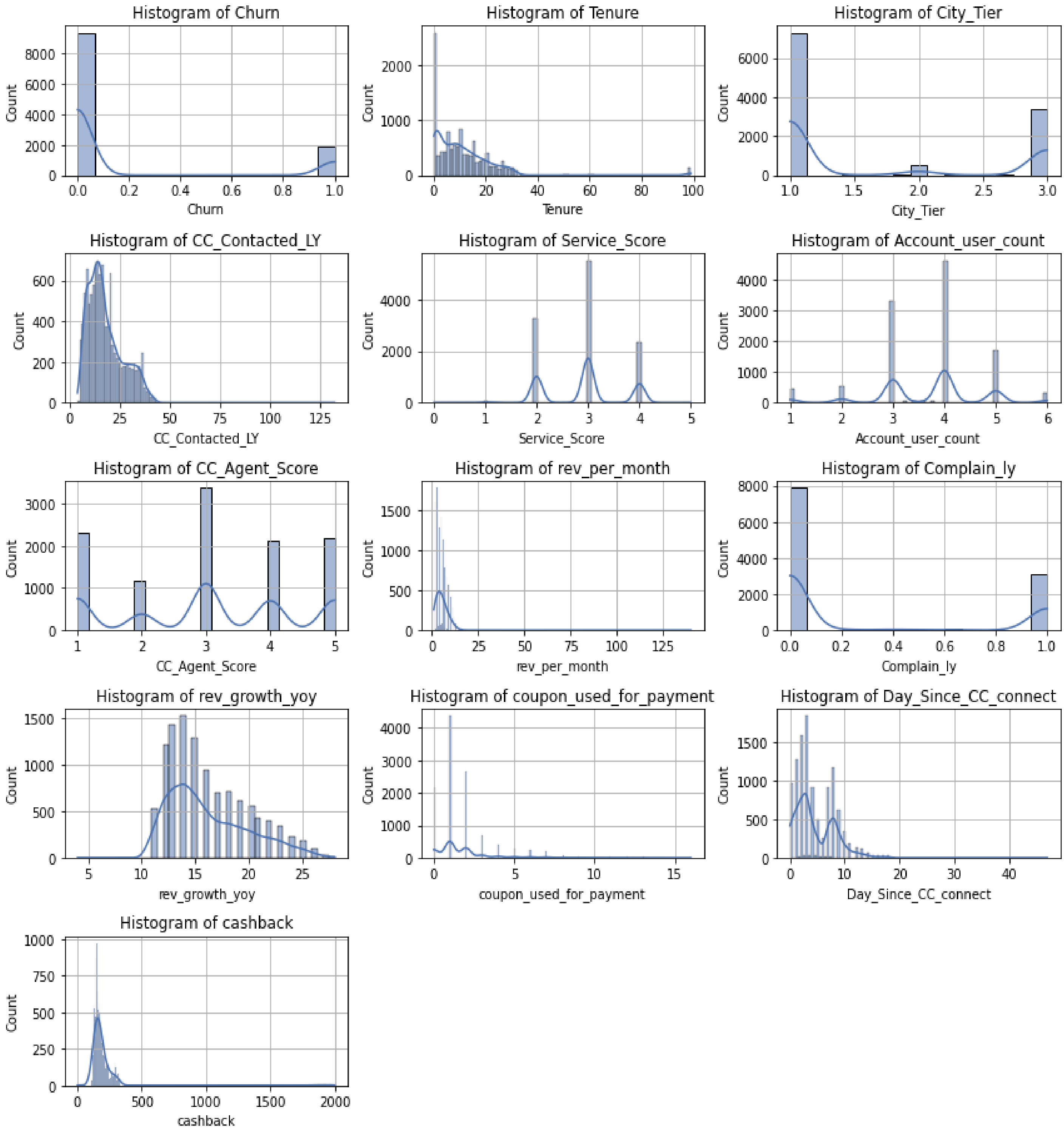


Figure 1- Histogram of numeric attributes

Inferences

(i) Variables like 'Tenure', 'CC_contacted_last_year', 'rev_per_month', 'rev_growth_yoy' and 'cashback' have right skewed distribution.

(ii) We can also observe variables like service_score, cc_agent_score and complaint_ly which are based on some categorical data.

a. The most marked service score for both service_score and cc_agent_score is 3. But significant number of people have provided a score 5 in cc_agent_score .

b. Around 3000 complaints have been submitted by the customers in the past 12 months

(iii) The number of customers on the verge of being churned is very less compared to the regular customers.

Let's have a look on the boxplots of the numeric attributes.

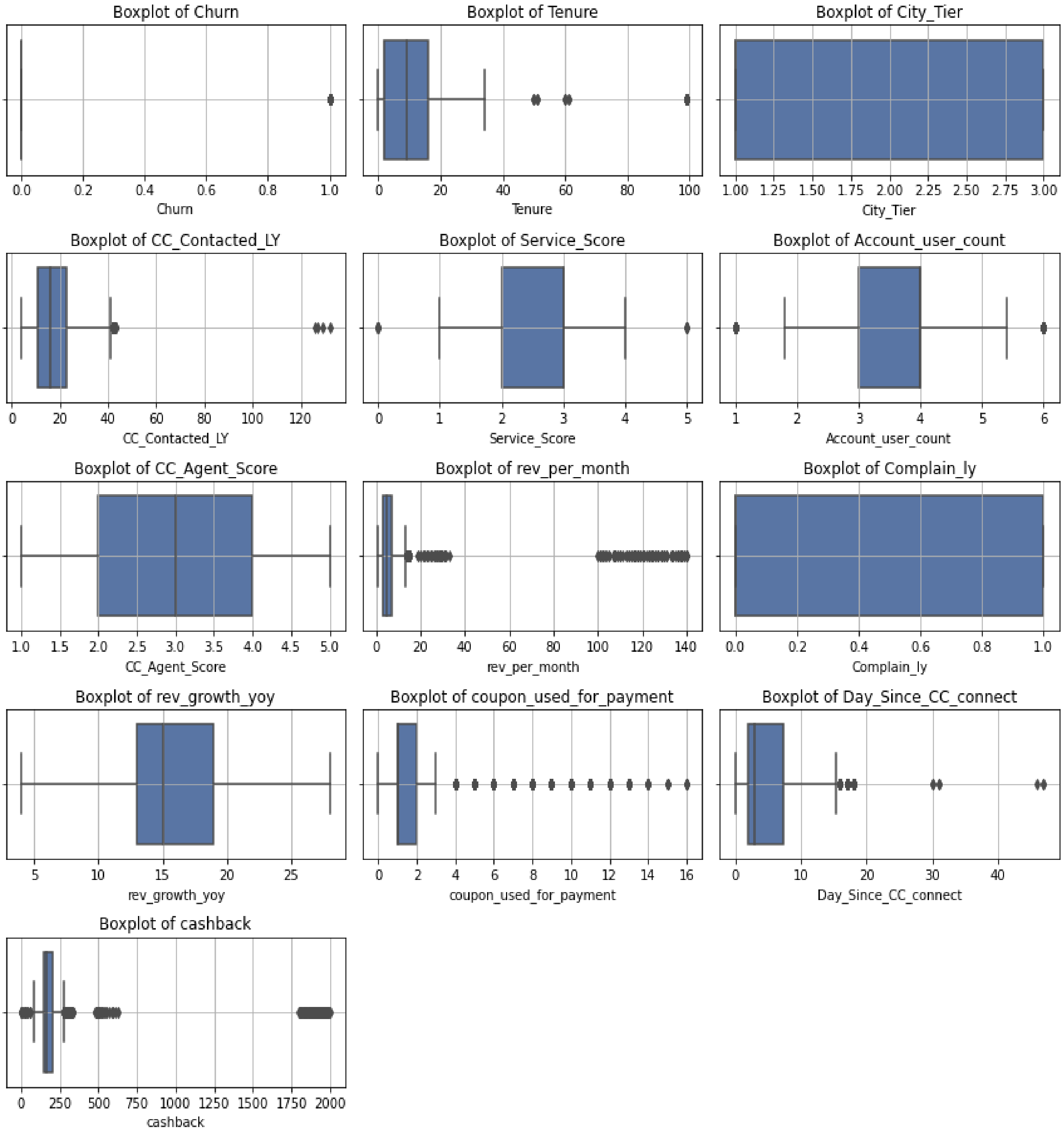


Figure 2- Boxplot of numeric attributes

Inferences

(i) 'cashback' has outliers both on its upper and lower values.

(ii) 'Tenure', 'CC_contacted_last_year', 'rev_per_month' and 'rev_growth_yoy' have outliers on their upper values.

(iii) We can observe outliers for service_score but those are not actually outliers. Those are categories or segment values made for the customers to rate the company and these outliers do not affect the performance of the data.

(iv) Tenure has a median value of around 10 years.

Let's have a look on the countplot of the categorical attributes.

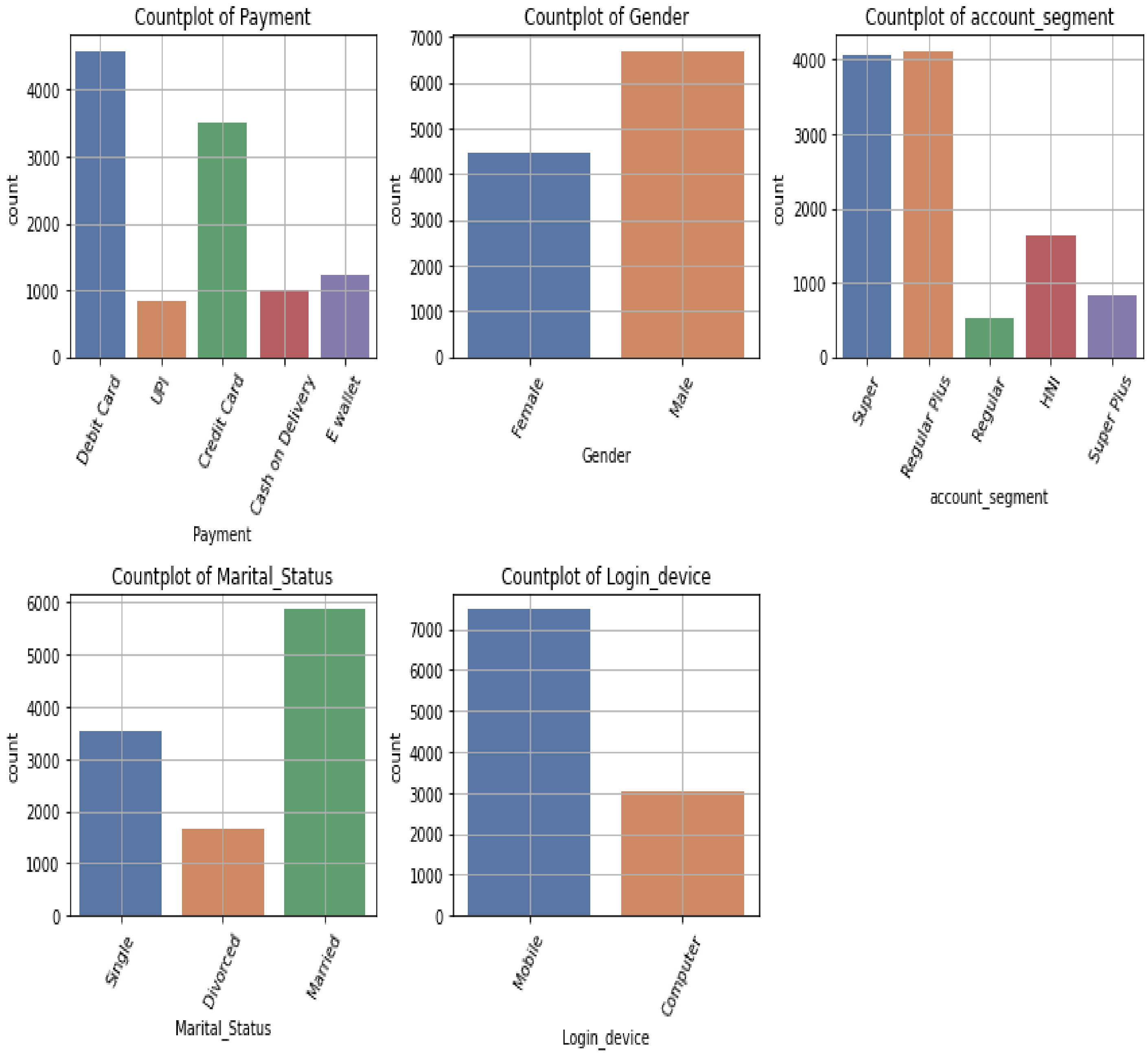


Figure 3- Countplot of Categorical attributes

Inferences

(i) The use of mobile is around three times more than computer for login device.

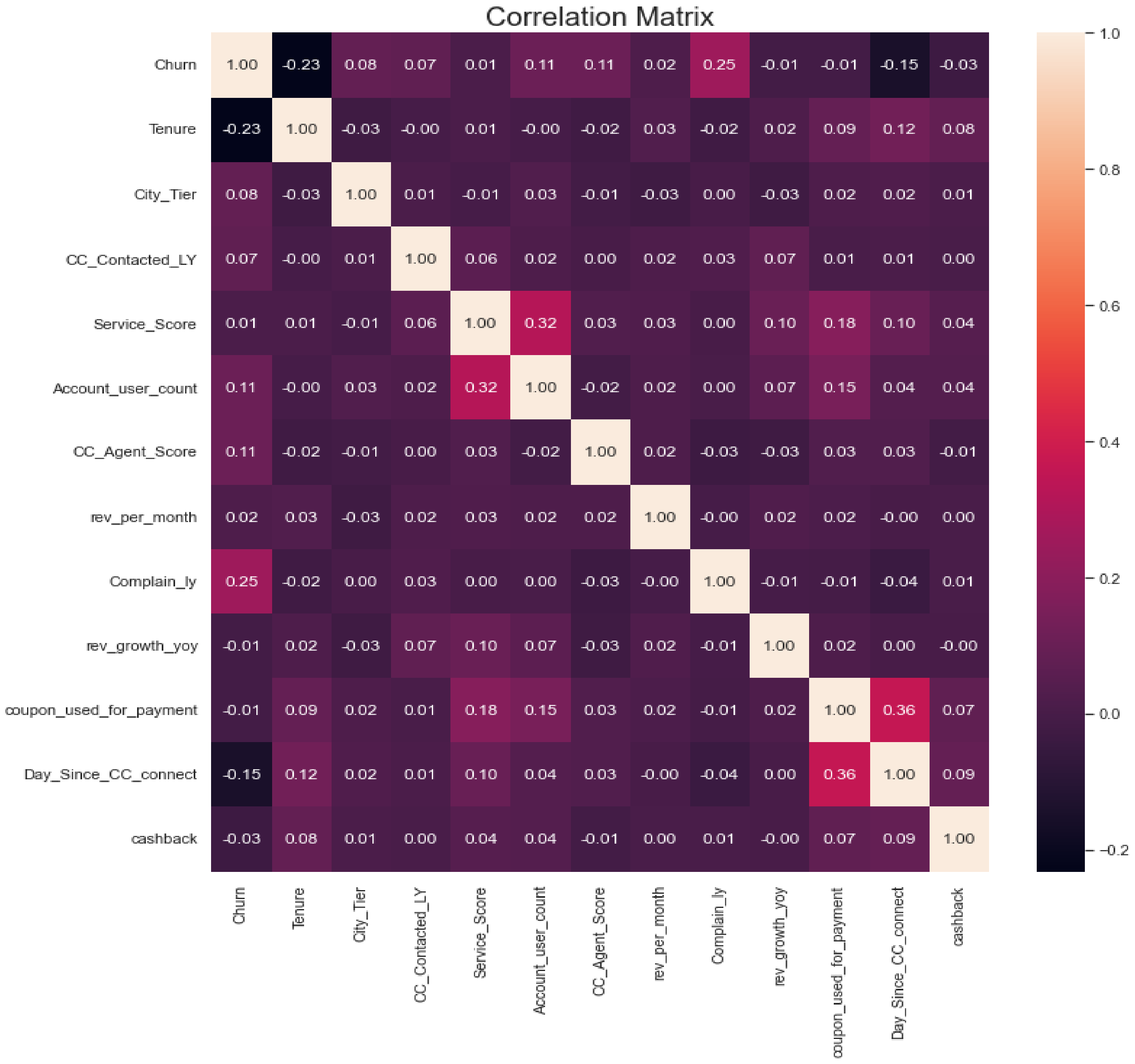
(ii) The number of males are higher than the females in Gender category.

(iii) Most of the customers like to pay through their debit cards and credit cards.

(iv) Most of the customers are Super and Regular Plus based on their money spent on purchases.

(v) Marital Status category shows that most of the married couples visit to buy their products.

Let’s have a look on the correlation matrix of the numeric attributes.



	Churn	Tenure	City_Tier	CC_Contacted_LY	Service_Score	Account_user_count	CC_Agent_Score	rev_per_month	Complain_ly	rev_growth_yoy	coupon_used_for_payment	Day_Since_CC_connect	cashback
Churn	1.00	-0.23	0.08	0.07	0.01	0.11	0.11	0.02	0.25	-0.01	-0.01	-0.15	-0.03
Tenure	-0.23	1.00	-0.03	-0.00	0.01	-0.00	-0.02	0.03	-0.02	0.02	0.09	0.12	0.08
City_Tier	0.08	-0.03	1.00	0.01	-0.01	0.03	-0.01	-0.03	0.00	-0.03	0.02	0.02	0.01
CC_Contacted_LY	0.07	-0.00	0.01	1.00	0.06	0.02	0.00	0.02	0.03	0.07	0.01	0.01	0.00
Service_Score	0.01	0.01	-0.01	0.06	1.00	0.32	0.03	0.03	0.00	0.10	0.18	0.10	0.04
Account_user_count	0.11	-0.00	0.03	0.02	0.32	1.00	-0.02	0.02	0.00	0.07	0.15	0.04	0.04
CC_Agent_Score	0.11	-0.02	-0.01	0.00	0.03	-0.02	1.00	0.02	-0.03	-0.03	0.03	0.03	-0.01
rev_per_month	0.02	0.03	-0.03	0.02	0.03	0.02	0.02	1.00	-0.00	0.02	0.02	-0.00	0.00
Complain_ly	0.25	-0.02	0.00	0.03	0.00	0.00	-0.03	-0.00	1.00	-0.01	-0.01	-0.04	0.01
rev_growth_yoy	-0.01	0.02	-0.03	0.07	0.10	0.07	-0.03	0.02	-0.01	1.00	0.02	0.00	-0.00
coupon_used_for_payment	-0.01	0.09	0.02	0.01	0.18	0.15	0.03	0.02	-0.01	0.02	1.00	0.36	0.07
Day_Since_CC_connect	-0.15	0.12	0.02	0.01	0.10	0.04	0.03	-0.00	-0.04	0.00	0.36	1.00	0.09
cashback	-0.03	0.08	0.01	0.00	0.04	0.04	-0.01	0.00	0.01	-0.00	0.07	0.09	1.00

Figure 4- Correlation Matrix

Inferences

(i) It seems that all of the variables have a low correlation with the target variable 'Churn'.

(ii) 'Day_Since_CC_connect' and 'Tenure' have a low inverse correlation with the target variable 'Churn' which means if value of one of these two variables increases then significantly the value of target variable 'Churn' decreases.

Let’s have a look on the pairplot of the numeric attributes.

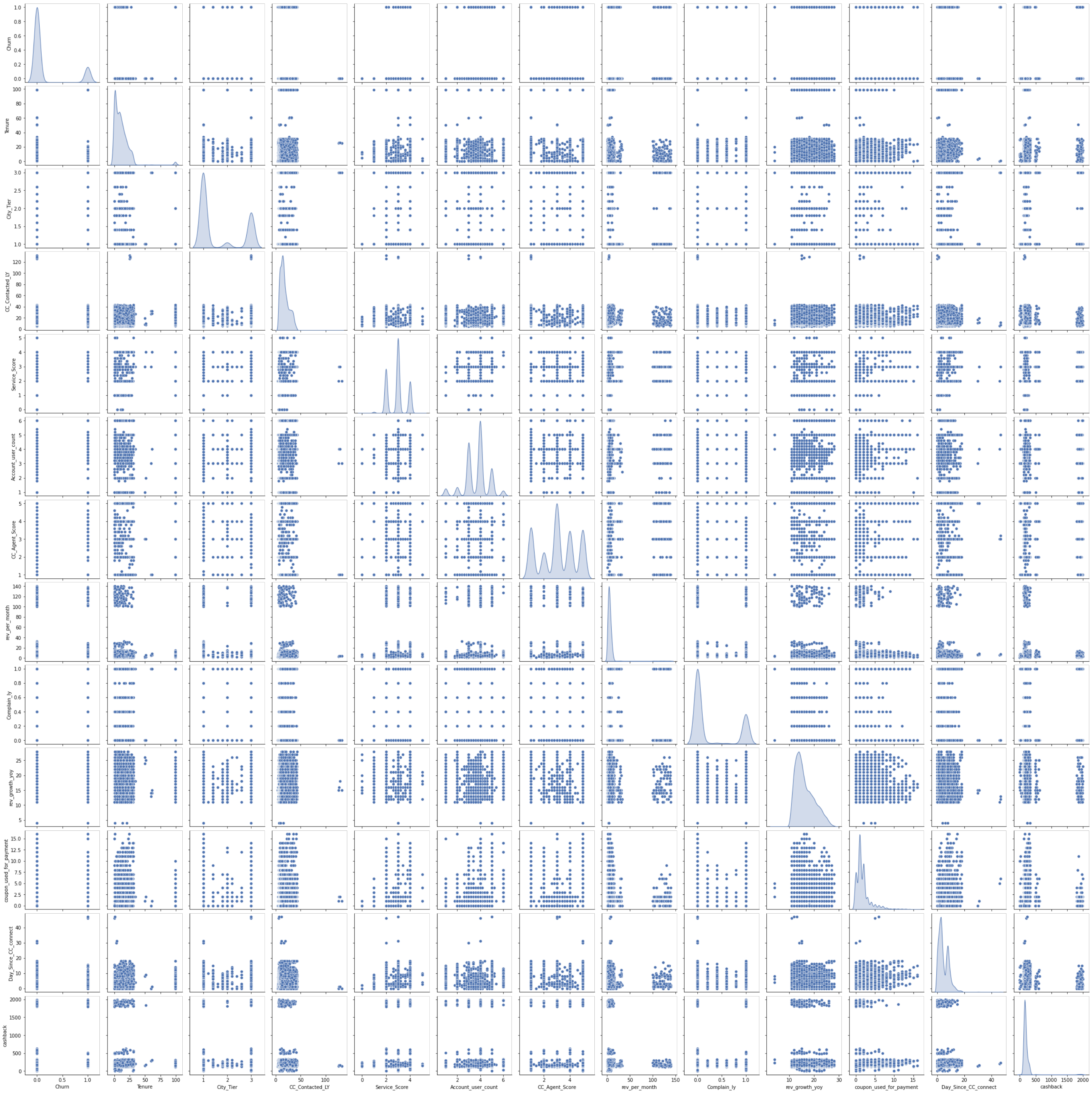


Figure 5- Pairplot of numeric attributes

Inferences

(i) The distribution of the attributes ‘Tenure’, ‘CC_contacted_last_year’, ‘rev_per_month’, ‘rev_growth_yoy’ and ‘cashback’ look normal.

(ii) Some of the attributes have a cloud like distribution which shows the presence of a low correlation among the variables.

Let's have a look on the boxplot of Churn vs Tenure

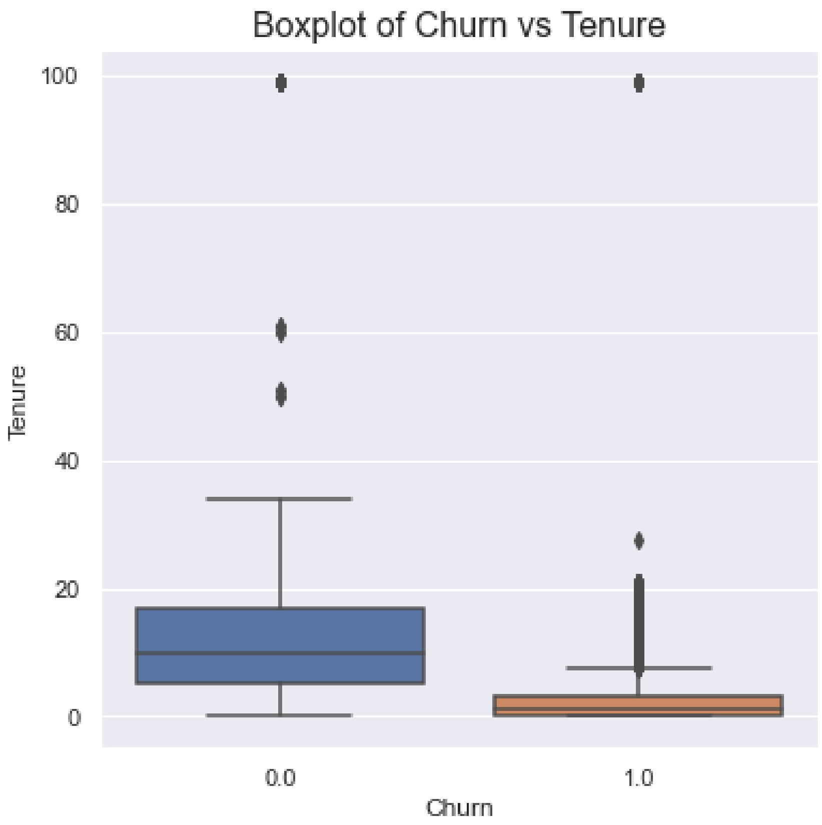


Figure 6- Boxplot of Churn vs Tenure

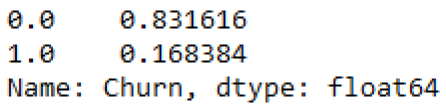
Inferences

(i) We can observe that employees have a high median tenure for customers who are not at the verge of churning but a very low median tenure of employees where customers are at the verge of churning.

(ii) The company should try providing basic and useful information to the employees having low tenure of how to deal with customers.

Business insights from EDA

The data is highly imbalanced. The customers on the verge of churning are only 17% of the total customers.



```
0.0    0.831616
1.0    0.168384
Name: Churn, dtype: float64
```

There are various techniques to handle imbalanced data. One of the techniques is

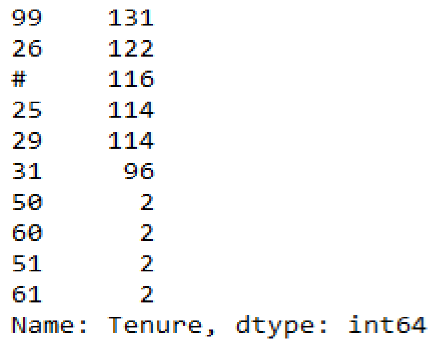
- **SMOTE** – SMOTE is otherwise known as **Synthetic Minority Oversampling Technique**. This is a very simple and effective algorithm which is often preferred over the Shuffle Split. The idea here is to generate new data points with same properties as that of minority class thus increasing its size and reducing biases towards majority class.

Business Insights from EDA

- Most of the married couples visit to buy the products from the company which means that the company should be able to provide products to both male and female that is products should be available according to the needs of the customers.
- As tenure increases, churn decreases because 'Tenure' has a low inverse correlation with 'Churn'. We can also recommend the company to hire experienced trainees who can help in churn control.
- The company should try providing basic and useful information to the employees having low tenure of how to deal with customers.
- The company should not spend most of their time and value on each and every one of these customers on the verge of being churned. This is because number of customers on the verge of being churned is five times less than the regular customers.

Data Cleaning and Pre-processing

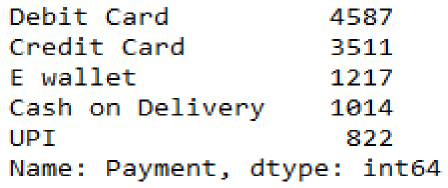
Let's look at the attribute 'Tenure'.



```
99    131
26    122
#     116
25    114
29    114
31     96
50      2
60      2
51      2
61      2
Name: Tenure, dtype: int64
```

We can observe an unwanted data is present in the variable that is '#' which has 116 values under it. Let's convert this into null values using np.nan() function.

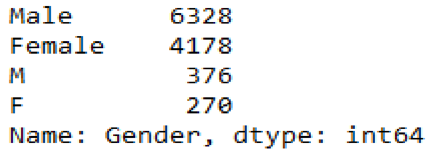
Let's look at the attribute 'Payment'.



```
Debit Card          4587
Credit Card         3511
E wallet            1217
Cash on Delivery    1014
UPI                  822
Name: Payment, dtype: int64
```

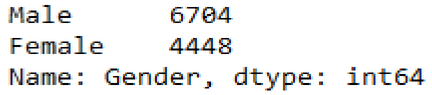
This seems alright.

Let's look at the attribute 'Gender'.



```
Male      6328
Female    4178
M          376
F          270
Name: Gender, dtype: int64
```

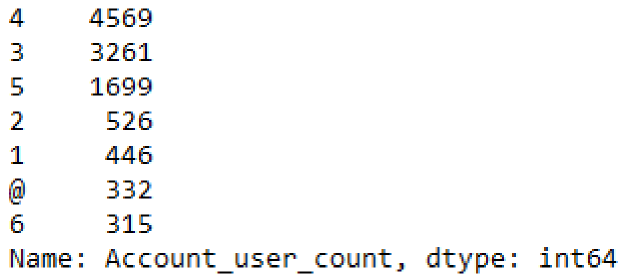
This looks as if there has been a typing mistake. Let's replace 'M' with Male and 'F' with Female.



```
Male      6704
Female    4448
Name: Gender, dtype: int64
```

Now this variable looks good.

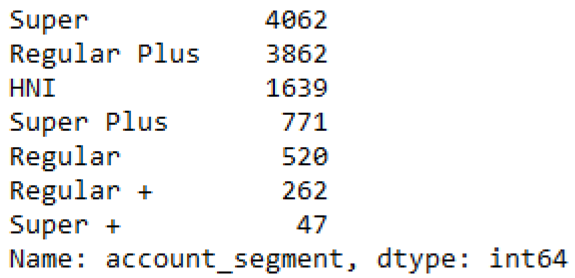
Let’s look at the attribute ‘Account_user_count’.



```
4    4569
3    3261
5    1699
2     526
1     446
@     332
6     315
Name: Account_user_count, dtype: int64
```

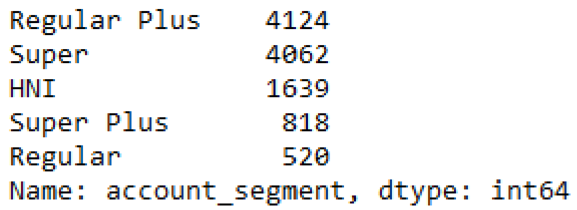
We can observe an unwanted data is present in the variable that is ‘@’ which has 332 values under it. Let’s convert this into null values using np.nan() function.

Let’s look at the attribute ‘account_segment’.



```
Super           4062
Regular Plus    3862
HNI             1639
Super Plus       771
Regular          520
Regular +        262
Super +           47
Name: account_segment, dtype: int64
```

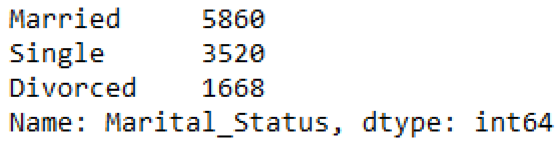
Again there has been a typing mistake. Let’s replace ‘Regular +’ with Regular Plus and ‘Super +’ with Super Plus.



```
Regular Plus    4124
Super           4062
HNI             1639
Super Plus       818
Regular          520
Name: account_segment, dtype: int64
```

This is good to use now.

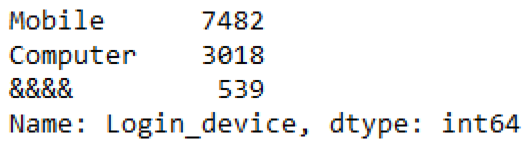
Let’s look at the attribute ‘Marital_Status’



```
Married     5860
Single      3520
Divorced    1668
Name: Marital_Status, dtype: int64
```

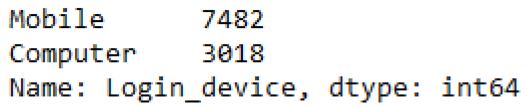
This looks good.

Let’s look at the attribute ‘Login_device’



```
Mobile      7482
Computer    3018
&&&&         539
Name: Login_device, dtype: int64
```

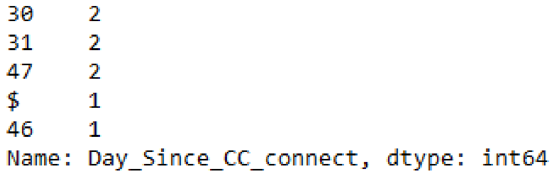
Let’s clean the unwanted data present in the variable that is ‘&&&&’ which has 539 values under it. Let’s convert this into null values using np.nan() function.



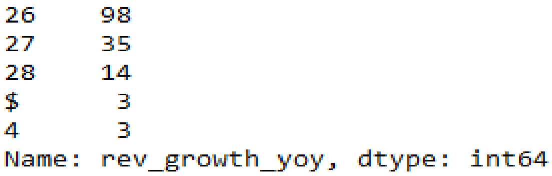
```
Mobile      7482
Computer    3018
Name: Login_device, dtype: int64
```

This looks good.

Let’s have a look on the bunch of other variables having unwanted data. Let’s clean them and convert them to null values.

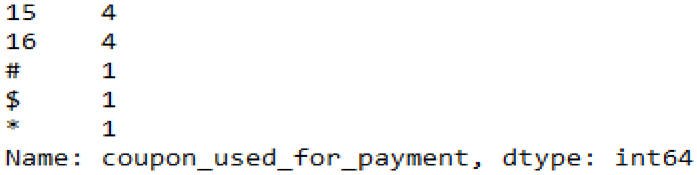


```
30    2
31    2
47    2
$     1
46    1
Name: Day_Since_CC_connect, dtype: int64
```



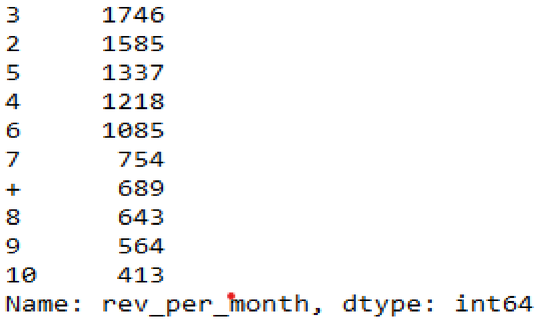
```
26     98
27     35
28     14
$       3
4       3
Name: rev_growth_yoy, dtype: int64
```

The above tables have ‘$’ as an unwanted data.



```
15     4
16     4
#      1
$      1
*      1
Name: coupon_used_for_payment, dtype: int64
```

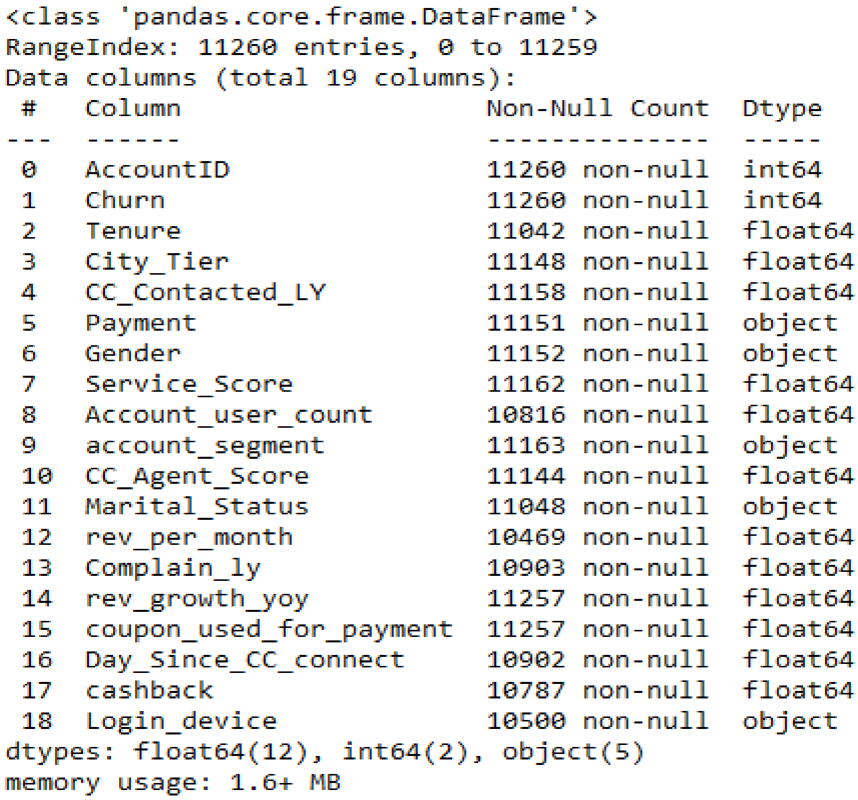
The above table has three types of unwanted data.



```
3      1746
2      1585
5      1337
4      1218
6      1085
7       754
+       689
8       643
9       564
10      413
Name: rev_per_month, dtype: int64
```

The above table has ‘+’ as an unwanted data.

So hence we are done cleaning the dataset which completes our first step in EDA.



```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 11260 entries, 0 to 11259
Data columns (total 19 columns):
 #   Column                   Non-Null Count  Dtype  
---  ------                   --------------  -----  
 0   AccountID                11260 non-null  int64  
 1   Churn                    11260 non-null  int64  
 2   Tenure                   11042 non-null  float64
 3   City_Tier                11148 non-null  float64
 4   CC_Contacted_LY          11158 non-null  float64
 5   Payment                  11151 non-null  object 
 6   Gender                   11152 non-null  object 
 7   Service_Score            11162 non-null  float64
 8   Account_user_count       10816 non-null  float64
 9   account_segment          11163 non-null  object 
 10  CC_Agent_Score           11144 non-null  float64
 11  Marital_Status           11048 non-null  object 
 12  rev_per_month            10469 non-null  float64
 13  Complain_ly              10903 non-null  float64
 14  rev_growth_yoy           11257 non-null  float64
 15  coupon_used_for_payment  11257 non-null  float64
 16  Day_Since_CC_connect     10902 non-null  float64
 17  cashback                 10787 non-null  float64
 18  Login_device             10500 non-null  object 
dtypes: float64(12), int64(2), object(5)
memory usage: 1.6+ MB
```

Hence we can observe that we have removed unwanted data and have the numerical variables back which were in the categorical format due to presence of unwanted data.

Identifying and treating missing values

The total amount of nulls present after cleaning the dataset are 4361.
Let’s inspect the null values visually.

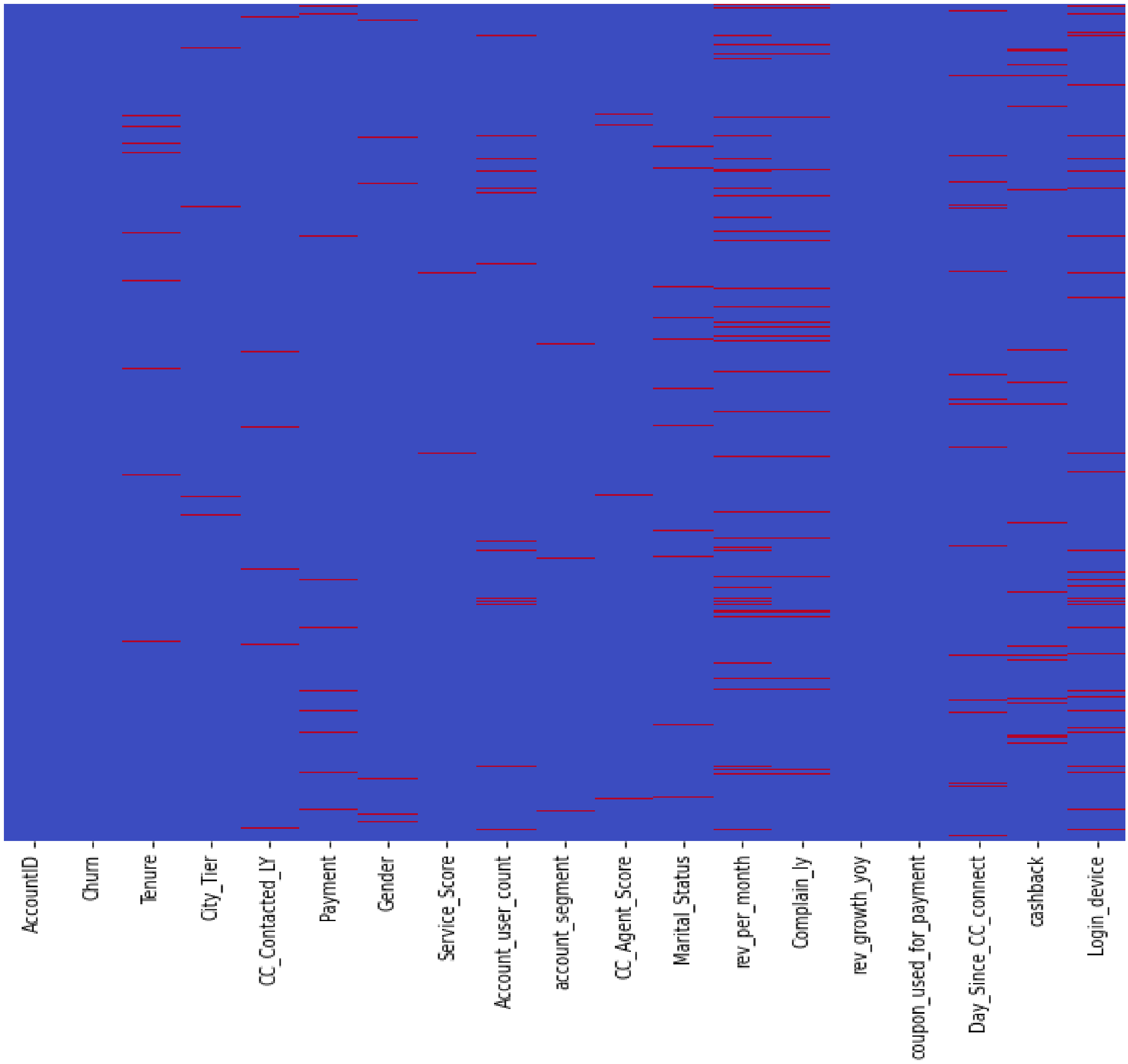


Figure 7- Visualising null values

The red lines in the above figure shows the missing or null values in the dataset.

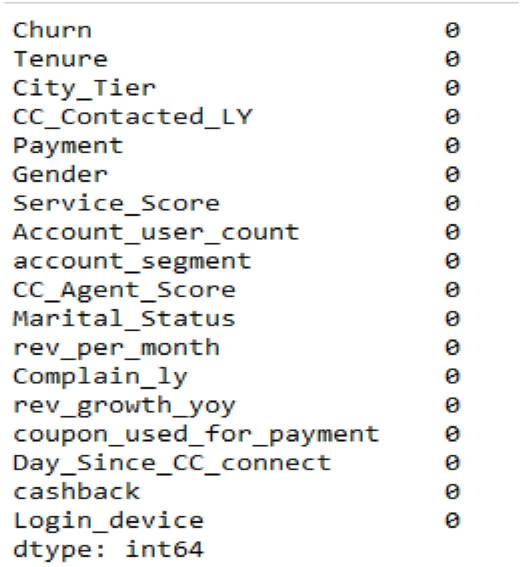
Removing the 'AccountID' attribute as it is insignificant for use in the dataset.

As we have imputed the categorical variables, now it’s time to impute the numerical variables.

Let’s impute the missing values in the numerical variables using KNN Imputer (k Nearest Neighbors) Imputer and choosing the hyper parameter n_neighbors equal to 5. KNN imputer is better in comparison to mean, mode and median computation.

```
(n_neighbors=5)
```

After imputing null values with KNN Imputer, we observe that there is no missing value in the data.



```
Churn                      0
Tenure                     0
City_Tier                  0
CC_Contacted_LY            0
Payment                    0
Gender                     0
Service_Score              0
Account_user_count         0
account_segment            0
CC_Agent_Score             0
Marital_Status             0
rev_per_month              0
Complain_ly                0
rev_growth_yoy             0
coupon_used_for_payment    0
Day_Since_CC_connect       0
cashback                   0
Login_device               0
dtype: int64
```

Outlier treatment

As mentioned earlier that there are outliers for service_score but those are not actually outliers. Those are categories or segment values made for the customers to rate the company and these outliers do not affect the performance of the data. So let’s move ahead by treating outliers of some selected numeric attributes.

The selected numeric attributes are **Tenure, CC_Contacted_LY, rev_per_month, coupon_used_for_payment, Day_Since_CC_connect** and **cashback.**

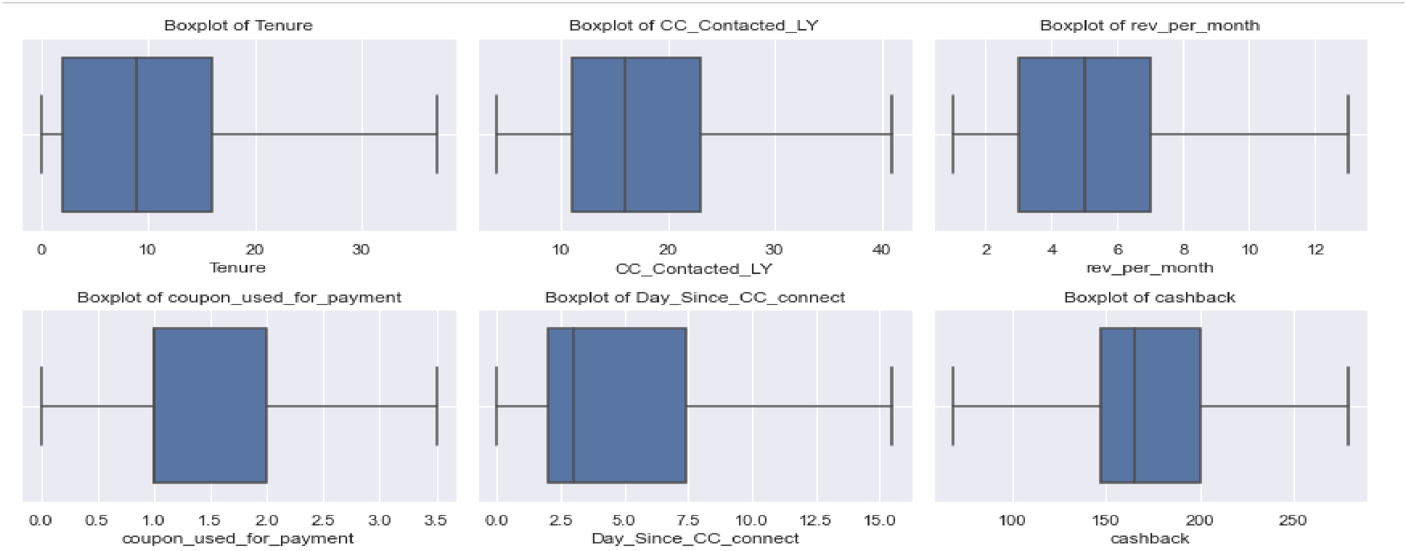


Figure 8- Boxplot of selected variables after treating outliers

We can observe that the outliers for the selected variables have been treated. Now let’s see the whole boxplot of all attributes again.

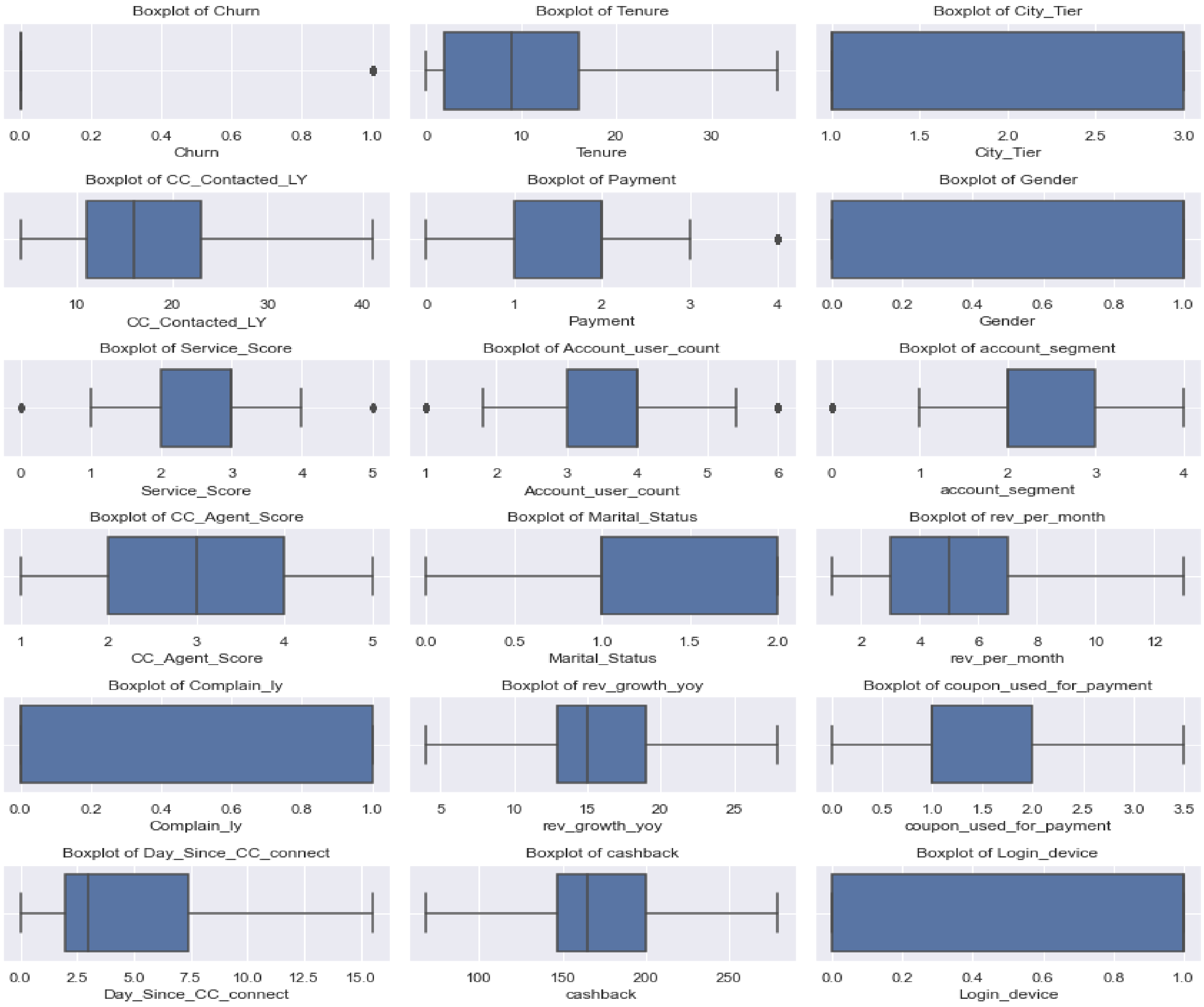
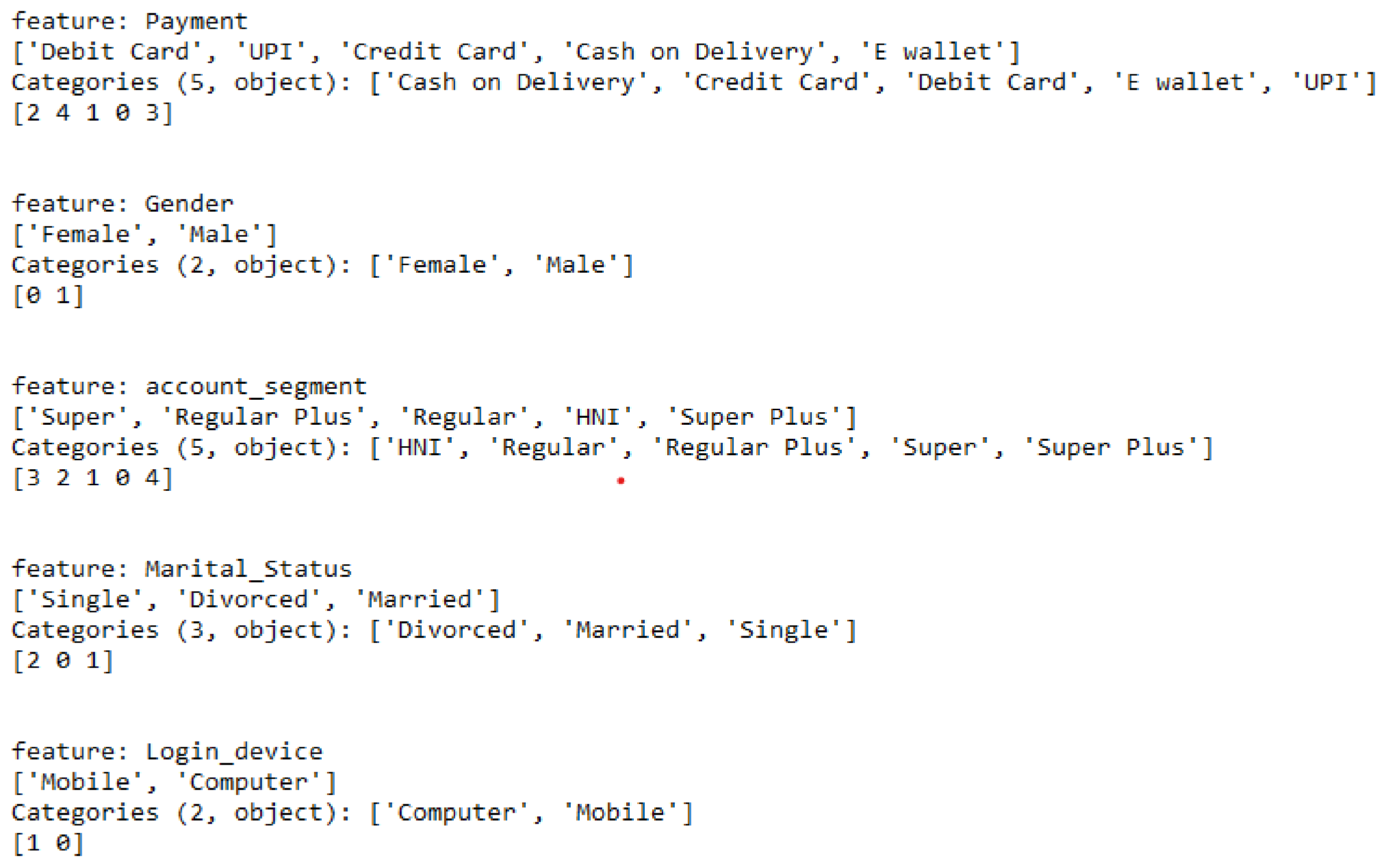


Figure 9- Boxplot of numerical attributes after treating outliers

Variable transformation

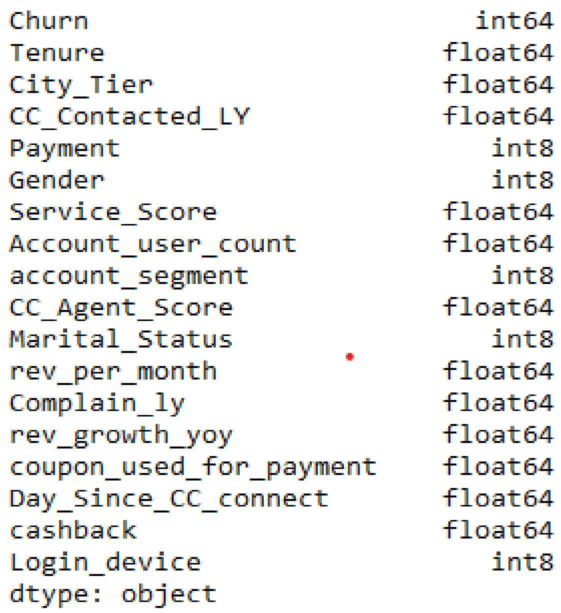
Let’s encode the categorical data using Categorical encoder. Categorical encoder is a encoding technique used to encode categories in a alphabetical manner. It provides integers based on the alphabetical sorting.

Encoding is necessary because most of the models do not account to work on categorical labels and throw an error during the process. Let’s observe the below figure to see how it encodes.



```
feature: Payment
['Debit Card', 'UPI', 'Credit Card', 'Cash on Delivery', 'E wallet']
Categories (5, object): ['Cash on Delivery', 'Credit Card', 'Debit Card', 'E wallet', 'UPI']
[2 4 1 0 3]


feature: Gender
['Female', 'Male']
Categories (2, object): ['Female', 'Male']
[0 1]


feature: account_segment
['Super', 'Regular Plus', 'Regular', 'HNI', 'Super Plus']
Categories (5, object): ['HNI', 'Regular', 'Regular Plus', 'Super', 'Super Plus']
[3 2 1 0 4]


feature: Marital_Status
['Single', 'Divorced', 'Married']
Categories (3, object): ['Divorced', 'Married', 'Single']
[2 0 1]


feature: Login_device
['Mobile', 'Computer']
Categories (2, object): ['Computer', 'Mobile']
[1 0]
```

We can observe that the encoder has encoded in an alphabetical manner.



```
Churn                        int64
Tenure                     float64
City_Tier                  float64
CC_Contacted_LY            float64
Payment                       int8
Gender                        int8
Service_Score              float64
Account_user_count         float64
account_segment               int8
CC_Agent_Score             float64
Marital_Status                int8
rev_per_month              float64
Complain_ly                float64
rev_growth_yoy             float64
coupon_used_for_payment    float64
Day_Since_CC_connect       float64
cashback                   float64
Login_device                  int8
dtype: object
```

We can observe that the categorical variables like ‘Payment’, ‘Gender’, ‘account_segment’, ‘Marital_status’ and ‘Login_device have changed into integer data types.

Now the data is ready for its next step, model building.

Insights from KNN Clustering

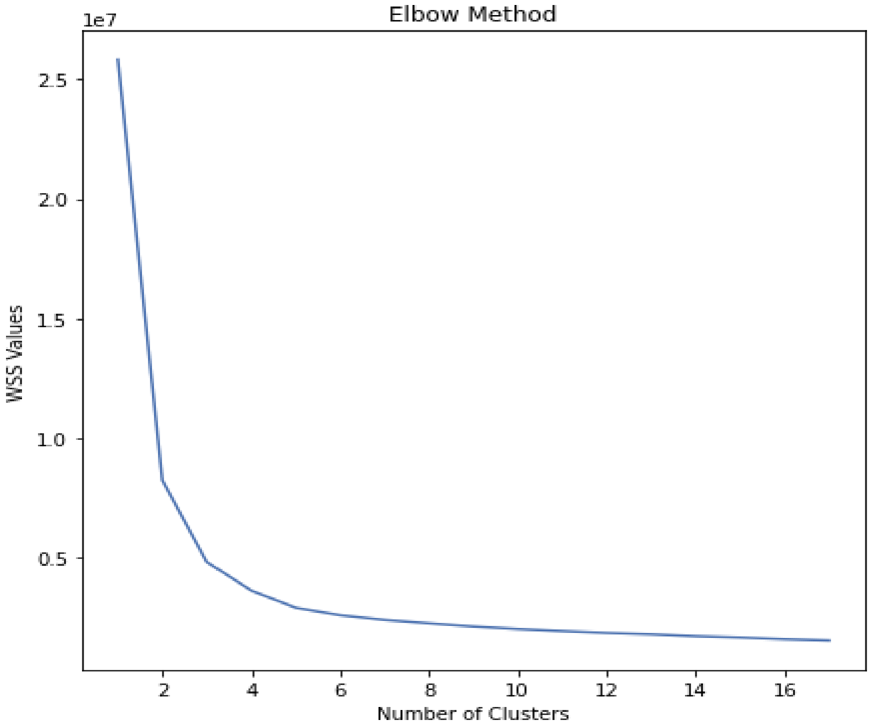
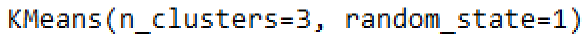


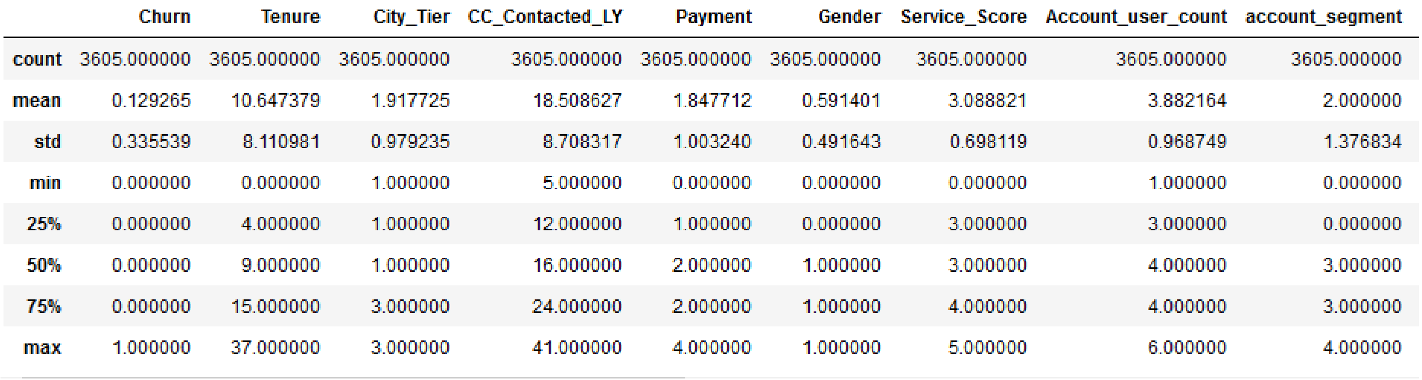
Figure 10- Elbow Curve

Number of clusters to be chosen from the above curve is equal to 3 i.e. k=3.



```
KMeans(n_clusters=3, random_state=1)
```

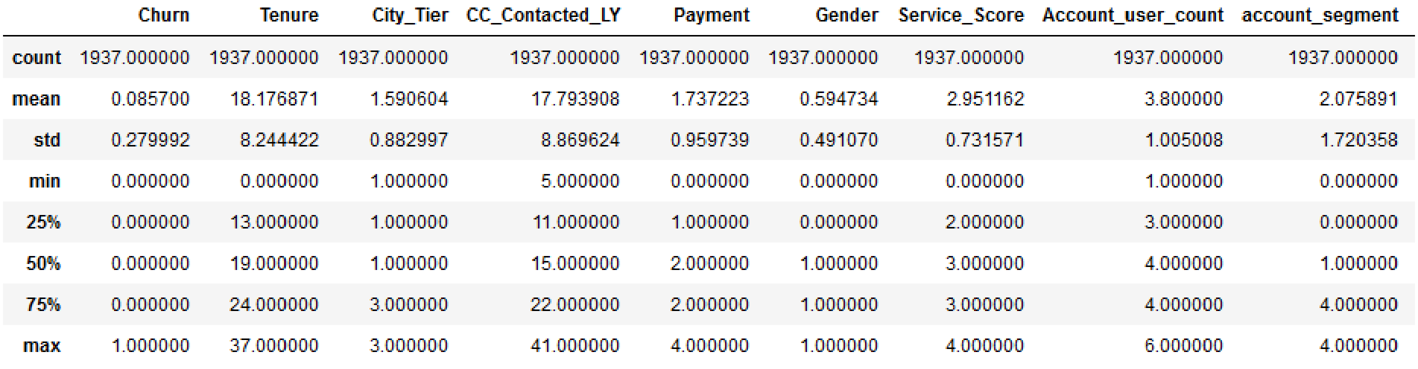
Fitting the K means clustering model and choosing the number of clusters to be 3 with a random state of 1.



	Churn	Tenure	City_Tier	CC_Contacted_LY	Payment	Gender	Service_Score	Account_user_count	account_segment
count	3605.000000	3605.000000	3605.000000	3605.000000	3605.000000	3605.000000	3605.000000	3605.000000	3605.000000
mean	0.129265	10.647379	1.917725	18.508627	1.847712	0.591401	3.088821	3.882164	2.000000
std	0.335539	8.110981	0.979235	8.708317	1.003240	0.491643	0.698119	0.968749	1.376834
min	0.000000	0.000000	1.000000	5.000000	0.000000	0.000000	0.000000	1.000000	0.000000
25%	0.000000	4.000000	1.000000	12.000000	1.000000	0.000000	3.000000	3.000000	0.000000
50%	0.000000	9.000000	1.000000	16.000000	2.000000	1.000000	3.000000	4.000000	3.000000
75%	0.000000	15.000000	3.000000	24.000000	2.000000	1.000000	4.000000	4.000000	3.000000
max	1.000000	37.000000	3.000000	41.000000	4.000000	1.000000	5.000000	6.000000	4.000000

Table 4- Clustering Segment 1

- 3 segments from Clustering
- Mean Values from Segment 1
  - Tenure for first segment is 11 years, revenue (per month) is 5.57 L, Cash back is Rs. 188 and days since no customer contacted CC is 5 days.

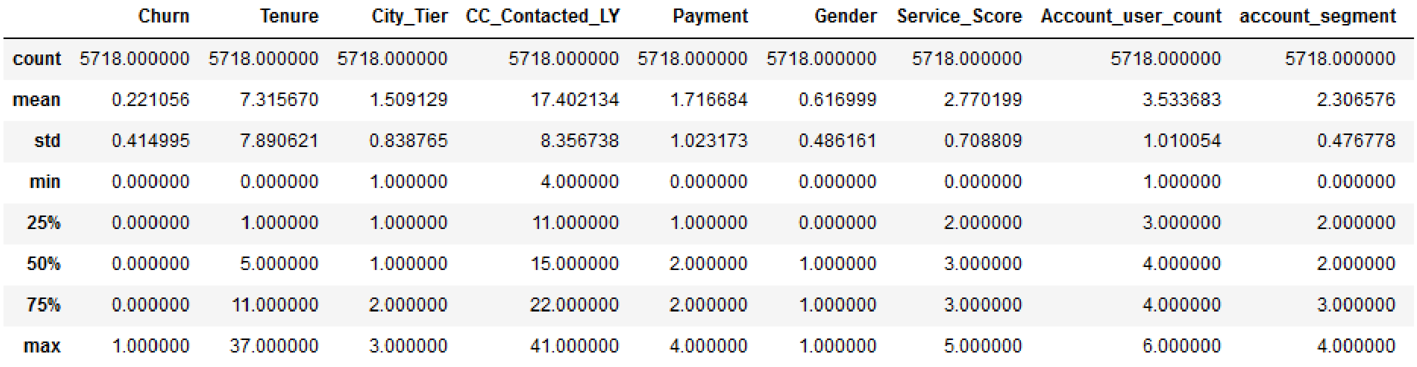


	Churn	Tenure	City_Tier	CC_Contacted_LY	Payment	Gender	Service_Score	Account_user_count	account_segment
count	1937.000000	1937.000000	1937.000000	1937.000000	1937.000000	1937.000000	1937.000000	1937.000000	1937.000000
mean	0.085700	18.176871	1.590604	17.793908	1.737223	0.594734	2.951162	3.800000	2.075891
std	0.279992	8.244422	0.882997	8.869624	0.959739	0.491070	0.731571	1.005008	1.720358
min	0.000000	0.000000	1.000000	5.000000	0.000000	0.000000	0.000000	1.000000	0.000000
25%	0.000000	13.000000	1.000000	11.000000	1.000000	0.000000	2.000000	3.000000	0.000000
50%	0.000000	19.000000	1.000000	15.000000	2.000000	1.000000	3.000000	4.000000	1.000000
75%	0.000000	24.000000	3.000000	22.000000	2.000000	1.000000	3.000000	4.000000	4.000000
max	1.000000	37.000000	3.000000	41.000000	4.000000	1.000000	4.000000	6.000000	4.000000

Table 5 - Clustering Segment 2

- Mean Values of Segment 2

  - Tenure for the second segment is 18 years, revenue (per month) is 6.08 L, cash back is Rs. 263 and days since no customer contacted CC is 7 days.



	Churn	Tenure	City_Tier	CC_Contacted_LY	Payment	Gender	Service_Score	Account_user_count	account_segment
count	5718.000000	5718.000000	5718.000000	5718.000000	5718.000000	5718.000000	5718.000000	5718.000000	5718.000000
mean	0.221056	7.315670	1.509129	17.402134	1.716684	0.616999	2.770199	3.533683	2.306576
std	0.414995	7.890621	0.838765	8.356738	1.023173	0.486161	0.708809	1.010054	0.476778
min	0.000000	0.000000	1.000000	4.000000	0.000000	0.000000	0.000000	1.000000	0.000000
25%	0.000000	1.000000	1.000000	11.000000	1.000000	0.000000	2.000000	3.000000	2.000000
50%	0.000000	5.000000	1.000000	15.000000	2.000000	1.000000	3.000000	4.000000	2.000000
75%	0.000000	11.000000	2.000000	22.000000	2.000000	1.000000	3.000000	4.000000	3.000000
max	1.000000	37.000000	3.000000	41.000000	4.000000	1.000000	5.000000	6.000000	4.000000

Table 6- Clustering Segment 3

- Mean Values of Segment 3

  - Tenure for the third segment is 7 years, revenue (per month) is 4.83 L, cash back is Rs. 144 and days since no customer contacted CC is 4 days.

The silhouette score is 0.4443 which is good value.

Model building and interpretation

80-20 percent ratio split is used for the data and a random state (=1) is introduced in train test split. A 80/20 split is done to ensure a good accuracy of the model.

CART model

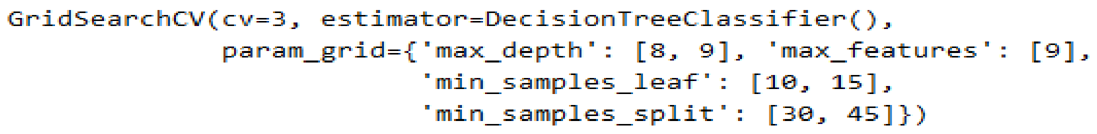
Let's run the model using Grid Search CV by building a cross validation table. This CV table contains hyper parameters for max depth, minimum sample leaf, minimum sample split and maximum features.

max depth- It is the number of nodes along the longest path from the root node down to the farthest leaf node. Growing decision tree to its fullest overfits the model. Hence the hyper parameter chosen is 8 or 9.

minimum sample leaf- It is the minimum number of samples required to be at a leaf node. It is usually selected at 1-3 percent of all records. The hyper parameter is chosen at random as 10 or 15.

minimum sample split- It is the minimum number of samples required to split an internal node. It is chosen to be usually three times of the chosen value of minimum sample leaf. Hyper parameter is chosen to be 30 or 45 in accordance with minimum sample leaf.

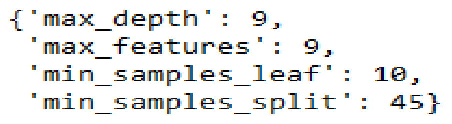
max features- It is usually a restriction given on high number of features to improve accuracy of the model. It is a hyper parameter which is usually chosen around half the number of available features. Hence the hyper parameter chosen is 9 as we have 18 features in the data as of now after dropping the 'Account ID' column.



```
GridSearchCV(cv=3, estimator=DecisionTreeClassifier(),
             param_grid={'max_depth': [8, 9], 'max_features': [9],
                         'min_samples_leaf': [10, 15],
                         'min_samples_split': [30, 45]})
```

'CV' is nothing but an internal cross validation technique which is used to calculate the score for each combination of parameters in the grid. Here in this case cv is equal to 3.

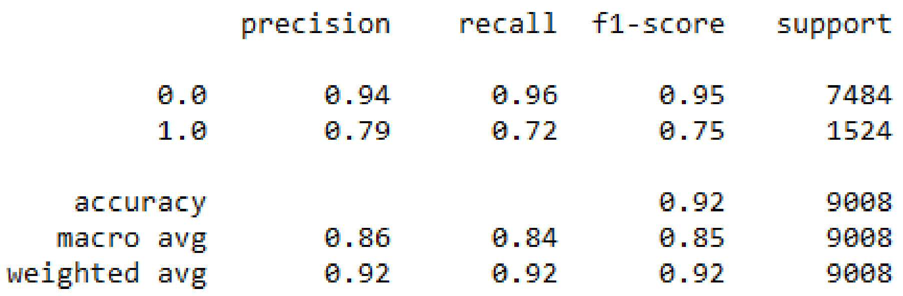
The best parameters chosen by the model are as below:



```
{'max_depth': 9,
 'max_features': 9,
 'min_samples_leaf': 10,
 'min_samples_split': 45}
```

The accuracy score of the CART model (Train) is 0.91907 or 91.90%.

The bigger and most important problem that will be faced by the company is, when the model predicts that the customer is not at the verge of churning but actually he is at the verge of churning. That's when we are going to lose a precious customer. Such errors are called Type-2 error or also called as False Negatives. Such errors are determined by the recall score of the model. **Hence recall scores have a high significance for this dataset.**



```
              precision    recall  f1-score   support

         0.0       0.94      0.96      0.95      7484
         1.0       0.79      0.72      0.75      1524

    accuracy                           0.92      9008
   macro avg       0.86      0.84      0.85      9008
weighted avg       0.92      0.92      0.92      9008
```

Table 7- *Classification Report CART (Train) Model*

We can observe that class 1 (customers on the verge of churning) has moderate values of all three precision, recall and f1 score. The most important is the recall score or the false negatives here.

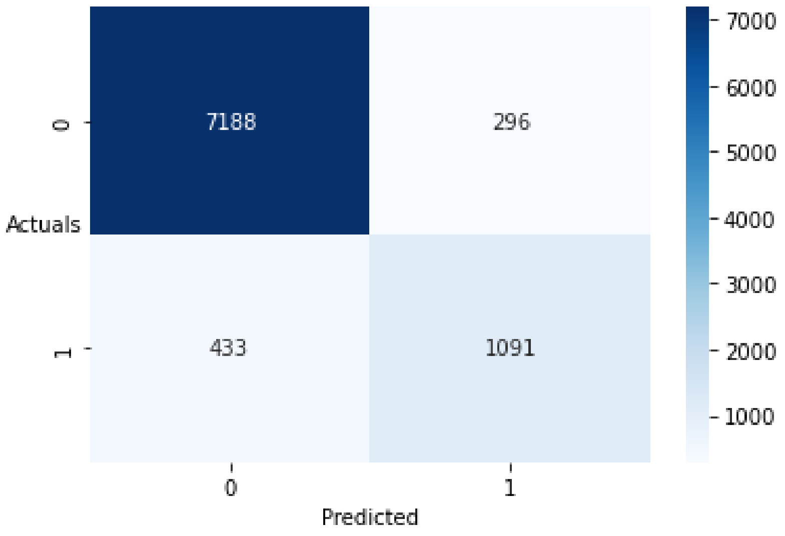


Table 8- Confusion Matrix *CART (Train) Model*

Here the number of false negatives is 433. The model can be more accurate if the number of false negatives can be reduced. Formula for Recall is given by:

Recall = TruePositives / (TruePositives + FalseNegatives)

The roc-auc score for the CART Model (Train) is 0.95013 which is pretty high. Also the ROC Curve is very steep.

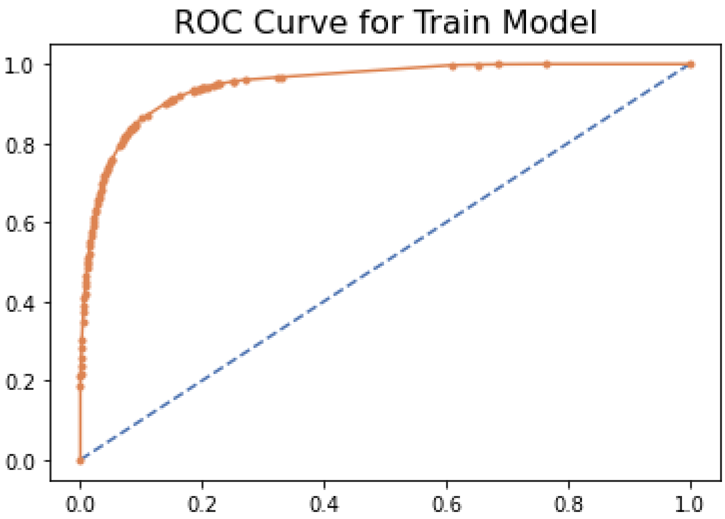


Figure 11- ROC Curve *CART (Train) Model*

Interpretation of the model:

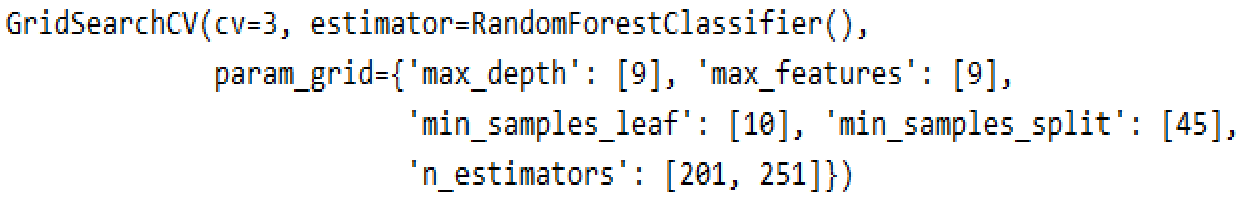
- This CART model performed normally for this dataset without any over fitting.
- Although the recall score is less in value.
- Recall score is high.

Random Forest model

Let’s run the model using Grid Search CV by building a cross validation table. This CV table contains hyper parameters for max depth, minimum sample leaf, minimum sample split, maximum features and n estimators as an additional parameter.

n estimators- It is usually the number of trees we want to build. Higher the number of trees, better the performance but slower the code. We have randomly chosen the hyper parameter for this to be 201 or 251.

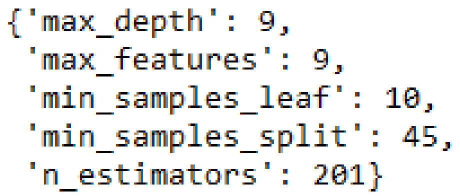
As from the above decision tree model, we found the best parameters. Let's choose the hyper parameter for n estimators at random and find the best value.



```
GridSearchCV(cv=3, estimator=RandomForestClassifier(),
             param_grid={'max_depth': [9], 'max_features': [9],
                         'min_samples_leaf': [10], 'min_samples_split': [45],
                         'n_estimators': [201, 251]})
```

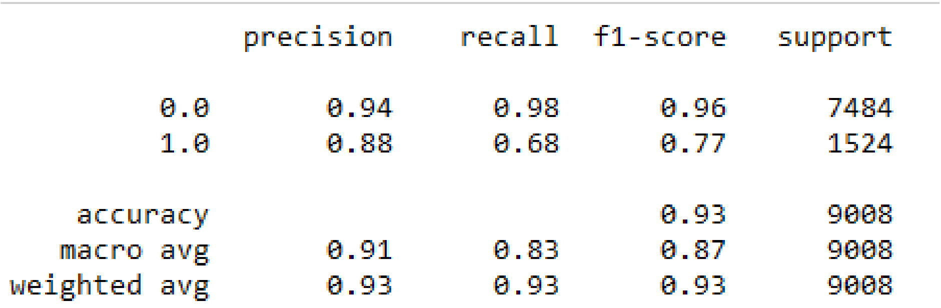
Here in this case cv is equal to 3.

The best parameters chosen by the model are as below:



```
{'max_depth': 9,
 'max_features': 9,
 'min_samples_leaf': 10,
 'min_samples_split': 45,
 'n_estimators': 201}
```

The accuracy score of the Random Forest Model (Train) is 0.93117 or 93.12%.



```
              precision    recall  f1-score   support

         0.0       0.94      0.98      0.96      7484
         1.0       0.88      0.68      0.77      1524

    accuracy                           0.93      9008
   macro avg       0.91      0.83      0.87      9008
weighted avg       0.93      0.93      0.93      9008
```

Table 9- *Classification Report RF (Train) Model*

Here in class 1 (customers on the verge of churning) precision has a high score which is not needed to us. But the recall score has a moderate value of 0.68

The below confusion matrix has 481 false negatives which is higher than number of false negatives for the CART model

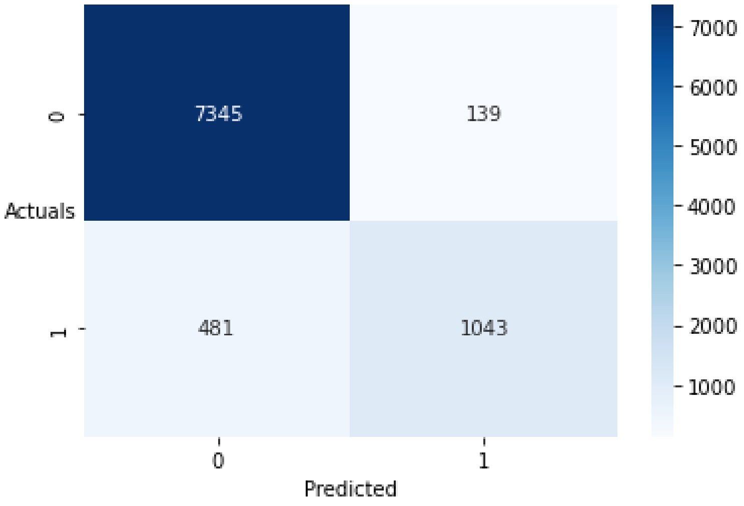


Table 10- Confusion Matrix RF (Train) Model

The roc-auc score for the Random Forest Model (Train) is 0.973 or 97.3%. Also the ROC Curve is highly steep.

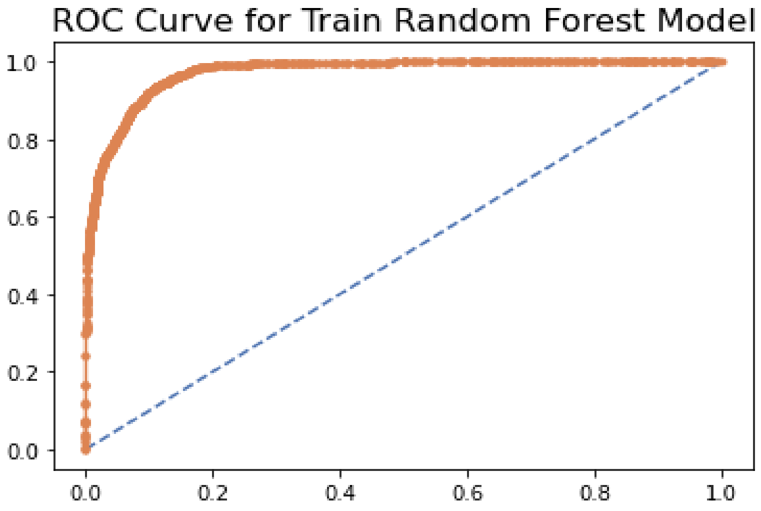


Figure 12- ROC Curve RF (Train) Model

Interpretation of the model:

- This RF model performed normally for this dataset without any over fitting.
- Although the recall score is less in value for this dataset.
- Recall score is high.

Artificial Neural Network model

Let’s run the model using Grid Search CV by building a cross validation table. This CV table contains hyper parameters for hidden layer sizes, activation, solver, tolerance and maximum iterations.

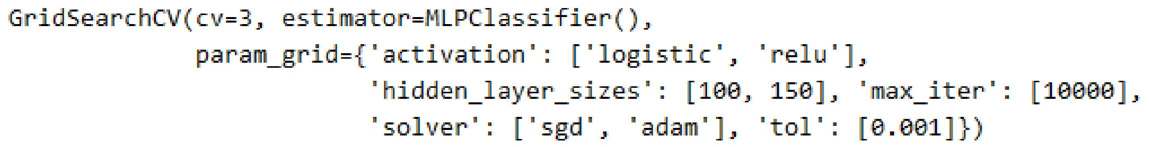
Hidden layer sizes are chosen on the basis of getting a better model. Sizes are allowed to be chosen as either 100 or 150.

Both the activation formulas are chosen i.e. logistic and relu (Rectified Linear Units)

For solver sgd (Stochastic Gradient Descent) and adam methods are chosen to get an optimal model.

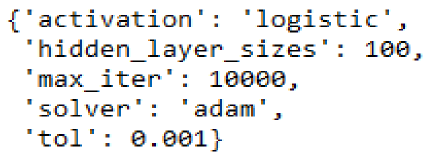
Higher the tolerance value of the model, the faster the speed of the model but lower the accuracy. Lower the tolerance value of the model, higher the accuracy but slower the speed of the model. Hence, for this problem tolerance value is 0.001

For the model to reach optimal point has to run multiple iterations. So, the selected number of iterations is 10000.



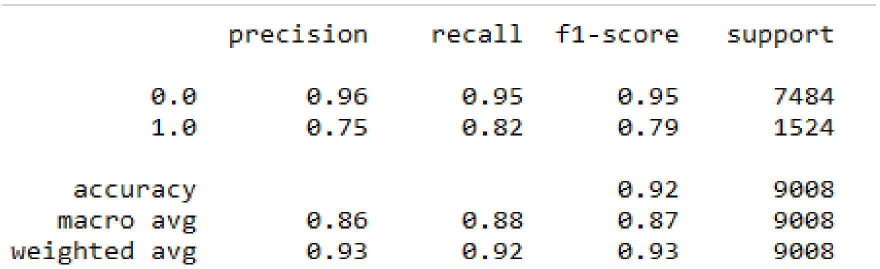
```
GridSearchCV(cv=3, estimator=MLPClassifier(),
             param_grid={'activation': ['logistic', 'relu'],
                         'hidden_layer_sizes': [100, 150], 'max_iter': [10000],
                         'solver': ['sgd', 'adam'], 'tol': [0.001]})
```

The best parameters chosen by the model are as below:



```
{'activation': 'logistic',
 'hidden_layer_sizes': 100,
 'max_iter': 10000,
 'solver': 'adam',
 'tol': 0.001}
```

The accuracy score of the ANN Model (Train) is 0.92451 or 92.45%.



```
              precision    recall  f1-score   support

         0.0       0.96      0.95      0.95      7484
         1.0       0.75      0.82      0.79      1524

    accuracy                           0.92      9008
   macro avg       0.86      0.88      0.87      9008
weighted avg       0.93      0.92      0.93      9008
```

Table 11- Classification Report ANN (Train) Model

This has a good recall score. The recall score is 0.82 and other precision and F1 score are moderate in nature. The below confusion matrix has 272 false negatives which is significantly lower than both the CART and RF models.

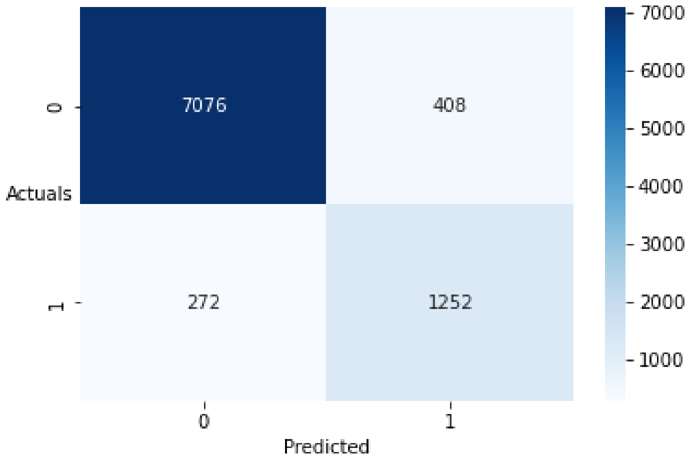


Table 12- Confusion Matrix ANN (Train) Model

The roc-auc score for the ANN Model (Train) is 0.96269 or 96.27%. Also the ROC Curve is very highly steep.

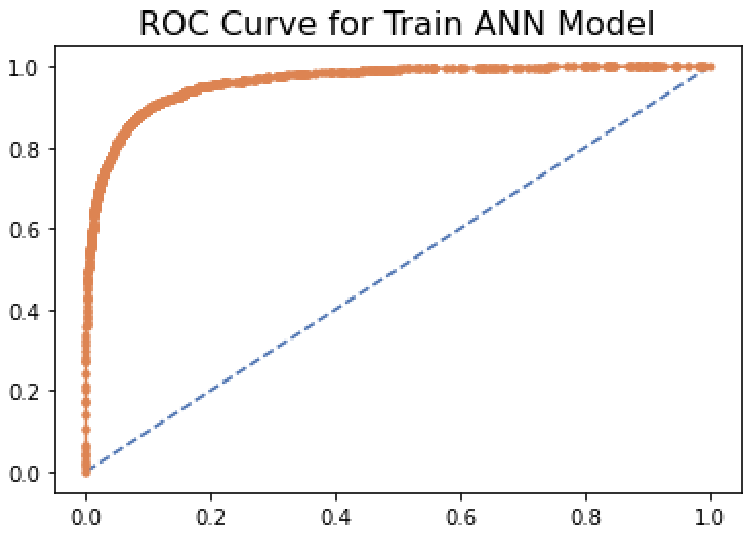


Figure 13- ROC Curve ANN (Train) Model

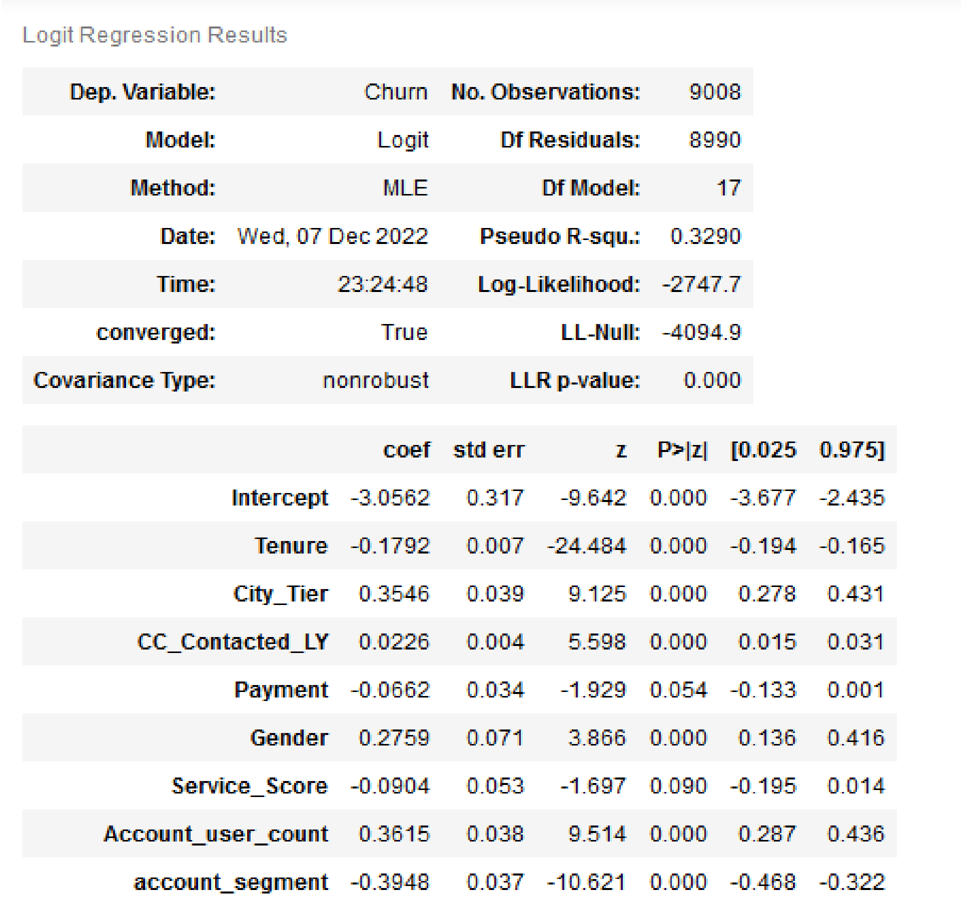
Interpretation of the model:

- This ANN model performed normally for this dataset without any over fitting.
- The recall score is high for this dataset.

Backward Elimination Feature using Logistic Regression

Stats model requires labeled data, therefore, concatenating the y variable to the train set. Now we have all the previous 18 columns in our new train dataset.

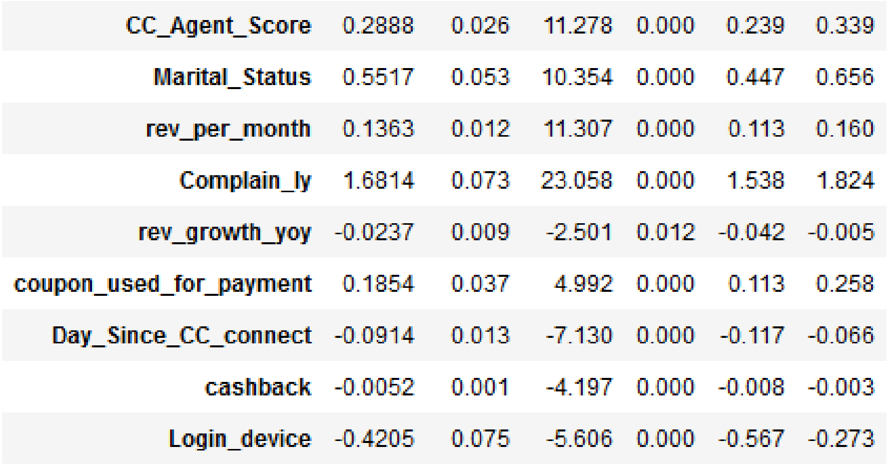
Let's create our first model using all the columns. Let's look at the summary table.



Logit Regression Results

Dep. Variable:	Churn	**No. Observations:**	9008
Model:	Logit	**Df Residuals:**	8990
Method:	MLE	**Df Model:**	17
Date:	Wed, 07 Dec 2022	**Pseudo R-squ.:**	0.3290
Time:	23:24:48	**Log-Likelihood:**	-2747.7
converged:	True	**LL-Null:**	-4094.9
Covariance Type:	nonrobust	**LLR p-value:**	0.000

	coef	std err	z	P>|z|	[0.025	0.975]
Intercept	-3.0562	0.317	-9.642	0.000	-3.677	-2.435
Tenure	-0.1792	0.007	-24.484	0.000	-0.194	-0.165
City_Tier	0.3546	0.039	9.125	0.000	0.278	0.431
CC_Contacted_LY	0.0226	0.004	5.598	0.000	0.015	0.031
Payment	-0.0662	0.034	-1.929	0.054	-0.133	0.001
Gender	0.2759	0.071	3.866	0.000	0.136	0.416
Service_Score	-0.0904	0.053	-1.697	0.090	-0.195	0.014
Account_user_count	0.3615	0.038	9.514	0.000	0.287	0.436
account_segment	-0.3948	0.037	-10.621	0.000	-0.468	-0.322



CC_Agent_Score	0.2888	0.026	11.278	0.000	0.239	0.339
Marital_Status	0.5517	0.053	10.354	0.000	0.447	0.656
rev_per_month	0.1363	0.012	11.307	0.000	0.113	0.160
Complain_ly	1.6814	0.073	23.058	0.000	1.538	1.824
rev_growth_yoy	-0.0237	0.009	-2.501	0.012	-0.042	-0.005
coupon_used_for_payment	0.1854	0.037	4.992	0.000	0.113	0.258
Day_Since_CC_connect	-0.0914	0.013	-7.130	0.000	-0.117	-0.066
cashback	-0.0052	0.001	-4.197	0.000	-0.008	-0.003
Login_device	-0.4205	0.075	-5.606	0.000	-0.567	-0.273

Table 13- Summary Table 1

The value of intercept in this case is -3.056.

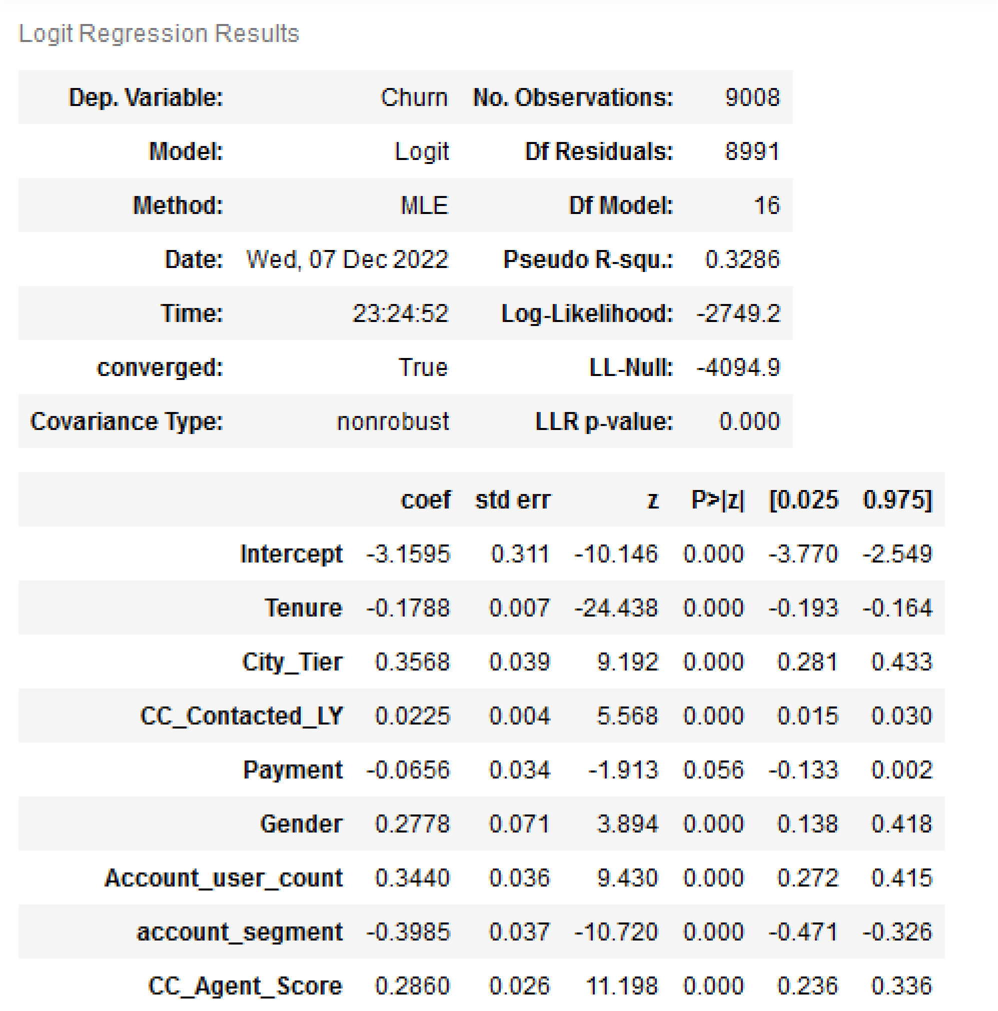
Hypothesis Testing

(i) The Null Hypothesis (H0) states that there is no relation between Churn and one of the variables and the Alternative Hypothesis (Ha) states that there is a relation between Churn and one of the variables.

(ii) At 95% confidence level if the p value is greater than 0.05, we reject H0 which means there is a relation between dependent (Churn) and independent variable.

So we choose to drop those columns having a p-value greater than 0.05.

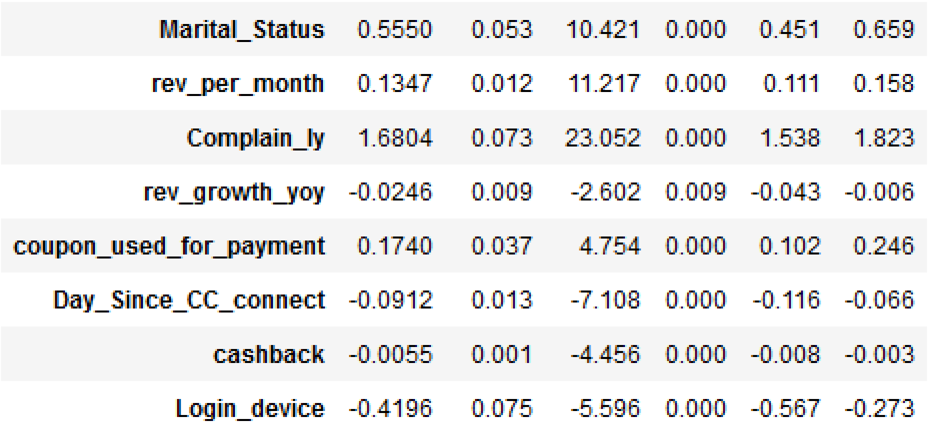
Removing Service_score as it has a probability value greater than 0.05 which means it has a relation with the dependent variable and creating another summary table to check again whether any variable has correlation.



Logit Regression Results

Dep. Variable:	Churn	No. Observations:	9008
Model:	Logit	Df Residuals:	8991
Method:	MLE	Df Model:	16
Date:	Wed, 07 Dec 2022	Pseudo R-squ.:	0.3286
Time:	23:24:52	Log-Likelihood:	-2749.2
converged:	True	LL-Null:	-4094.9
Covariance Type:	nonrobust	LLR p-value:	0.000

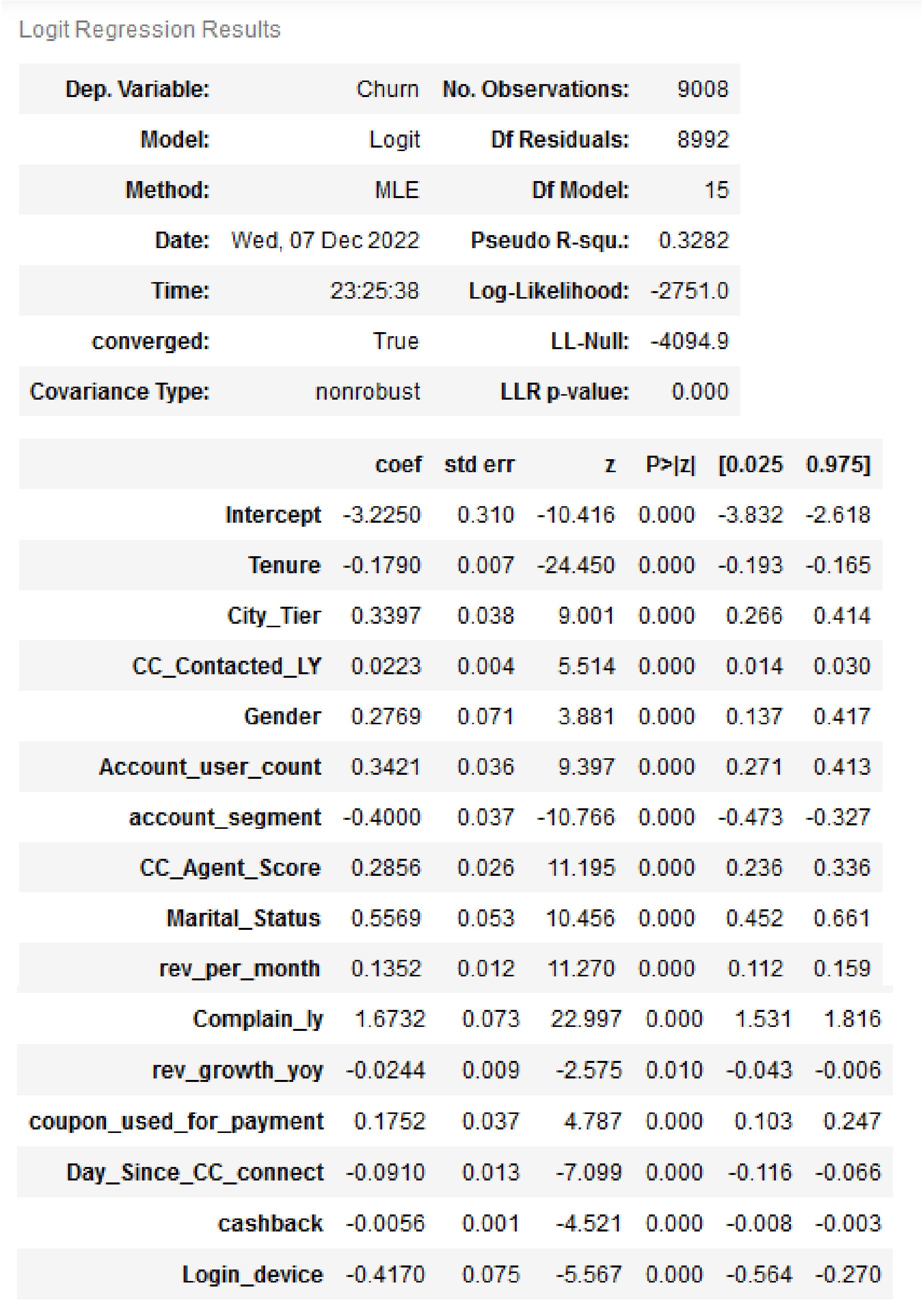
	coef	std err	z	P>|z|	[0.025	0.975]
Intercept	-3.1595	0.311	-10.146	0.000	-3.770	-2.549
Tenure	-0.1788	0.007	-24.438	0.000	-0.193	-0.164
City_Tier	0.3568	0.039	9.192	0.000	0.281	0.433
CC_Contacted_LY	0.0225	0.004	5.568	0.000	0.015	0.030
Payment	-0.0656	0.034	-1.913	0.056	-0.133	0.002
Gender	0.2778	0.071	3.894	0.000	0.138	0.418
Account_user_count	0.3440	0.036	9.430	0.000	0.272	0.415
account_segment	-0.3985	0.037	-10.720	0.000	-0.471	-0.326
CC_Agent_Score	0.2860	0.026	11.198	0.000	0.236	0.336



Marital_Status	0.5550	0.053	10.421	0.000	0.451	0.659
rev_per_month	0.1347	0.012	11.217	0.000	0.111	0.158
Complain_ly	1.6804	0.073	23.052	0.000	1.538	1.823
rev_growth_yoy	-0.0246	0.009	-2.602	0.009	-0.043	-0.006
coupon_used_for_payment	0.1740	0.037	4.754	0.000	0.102	0.246
Day_Since_CC_connect	-0.0912	0.013	-7.108	0.000	-0.116	-0.066
cashback	-0.0055	0.001	-4.456	0.000	-0.008	-0.003
Login_device	-0.4196	0.075	-5.596	0.000	-0.567	-0.273

Table 14- Summary Table 2

The value of intercept in this case is -3.16. Some of the coefficients in the variables are either positive or negative. Removing payment as it has a probability value greater than 0.05 which means it has a relation with the dependent variable and creating another summary table to check again whether any variable has correlation.



Logit Regression Results

Dep. Variable:	Churn	No. Observations:	9008
Model:	Logit	Df Residuals:	8992
Method:	MLE	Df Model:	15
Date:	Wed, 07 Dec 2022	Pseudo R-squ.:	0.3282
Time:	23:25:38	Log-Likelihood:	-2751.0
converged:	True	LL-Null:	-4094.9
Covariance Type:	nonrobust	LLR p-value:	0.000

	coef	std err	z	P>|z|	[0.025	0.975]
Intercept	-3.2250	0.310	-10.416	0.000	-3.832	-2.618
Tenure	-0.1790	0.007	-24.450	0.000	-0.193	-0.165
City_Tier	0.3397	0.038	9.001	0.000	0.266	0.414
CC_Contacted_LY	0.0223	0.004	5.514	0.000	0.014	0.030
Gender	0.2769	0.071	3.881	0.000	0.137	0.417
Account_user_count	0.3421	0.036	9.397	0.000	0.271	0.413
account_segment	-0.4000	0.037	-10.766	0.000	-0.473	-0.327
CC_Agent_Score	0.2856	0.026	11.195	0.000	0.236	0.336
Marital_Status	0.5569	0.053	10.456	0.000	0.452	0.661
rev_per_month	0.1352	0.012	11.270	0.000	0.112	0.159
Complain_ly	1.6732	0.073	22.997	0.000	1.531	1.816
rev_growth_yoy	-0.0244	0.009	-2.575	0.010	-0.043	-0.006
coupon_used_for_payment	0.1752	0.037	4.787	0.000	0.103	0.247
Day_Since_CC_connect	-0.0910	0.013	-7.099	0.000	-0.116	-0.066
cashback	-0.0056	0.001	-4.521	0.000	-0.008	-0.003
Login_device	-0.4170	0.075	-5.567	0.000	-0.564	-0.270

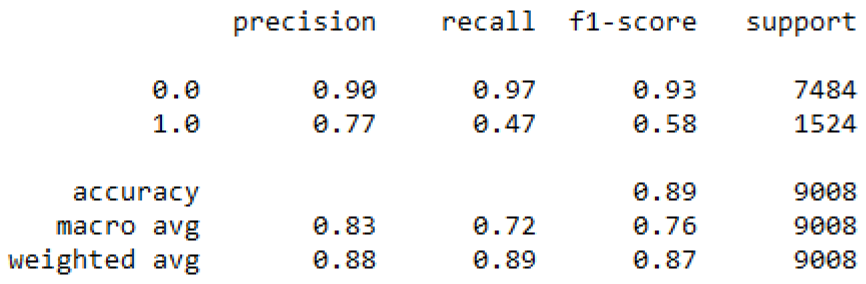
Table 15- Summary Table 3

There is no more variable which has a probability greater than 0.05. Now every feature is significant.

We can observe that the value of intercept in this case is -3.23. The variables like ‘tenure’, ‘account segment’, rev growth yoy’, ‘days since cc connect’, ‘cashback’ and ‘login device’ have a negative coefficient which means they have a negative correlation with the dependent variable.

Similarly except all the variables (having negative coefficient) the variables with positive coefficient impacts the dependent variable with a positive correlation. ‘Complain ly’ has the highest positive correlation with the dependent variable and ‘Login device’ has the highest negative correlation with the dependent variable.

Now let’s take a look at the model building.



```
              precision    recall  f1-score   support

         0.0       0.90      0.97      0.93      7484
         1.0       0.77      0.47      0.58      1524

    accuracy                           0.89      9008
   macro avg       0.83      0.72      0.76      9008
weighted avg       0.88      0.89      0.87      9008
```

Table 16- Classification Report Logit (Train) Model

We can observe a very low value of recall in the classification report. The number of false negatives In the confusion matrix is very high (= 814). The accuracy of the train model is 0.89 or 89%. This model performed poorly for the given dataset.

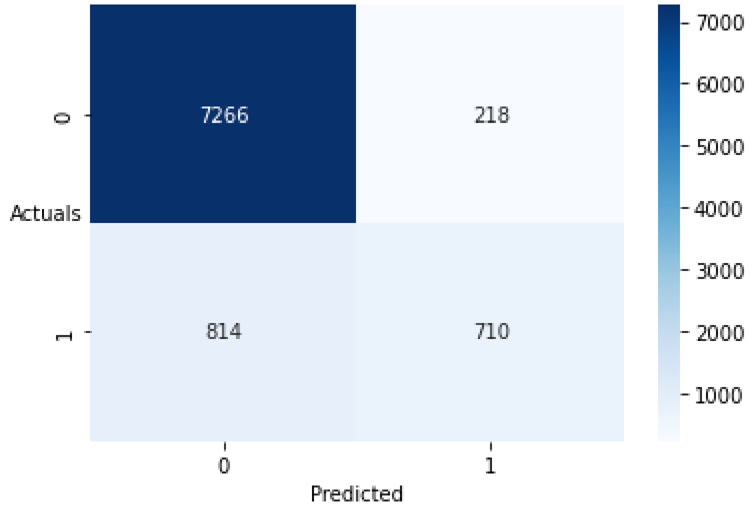


Table 17- Confusion Matrix Logit (Train) Model

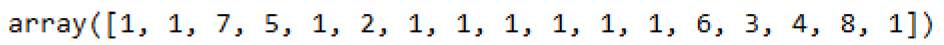
Interpretation of the model:

- This model performed poorly for this dataset.
- The recall score is low.

Logistic Regression using SMOTE

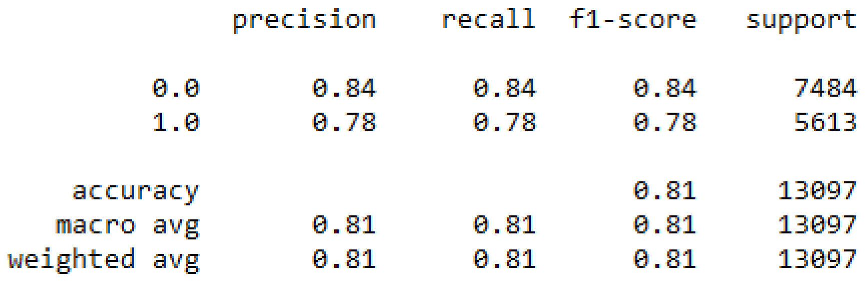
Let’s append the Logistic Regression function into RFE (Recursive Feature Elimination) model in a parameter called estimator and choose the number of features to be 10. This numerical value (= 10) for n features is chosen after playing around with the hyper parameters of the model. Now let’s fit the train dataset into RFE function.

The selector ranking provided by the model is as given below:



```
array([1, 1, 7, 5, 1, 2, 1, 1, 1, 1, 1, 1, 6, 3, 4, 8, 1])
```

The numerical value 1 is repeated 10 times.
The accuracy score of the LR Model using SMOTE (Train) is 0.81171 or 81.17% which is a moderate score.



```
              precision    recall  f1-score   support

         0.0       0.84      0.84      0.84      7484
         1.0       0.78      0.78      0.78      5613

    accuracy                           0.81     13097
   macro avg       0.81      0.81      0.81     13097
weighted avg       0.81      0.81      0.81     13097
```

Table 18- Classification Report LR (Train) Model

All the three measures that are precision, recall and f1 score in the classification report are same. The recall score is pretty good for this model as compared to other model having a poor performance.
The below confusion matrix has 1233 false negatives which is quite high because SMOTE model generates synthetic samples for the train data.

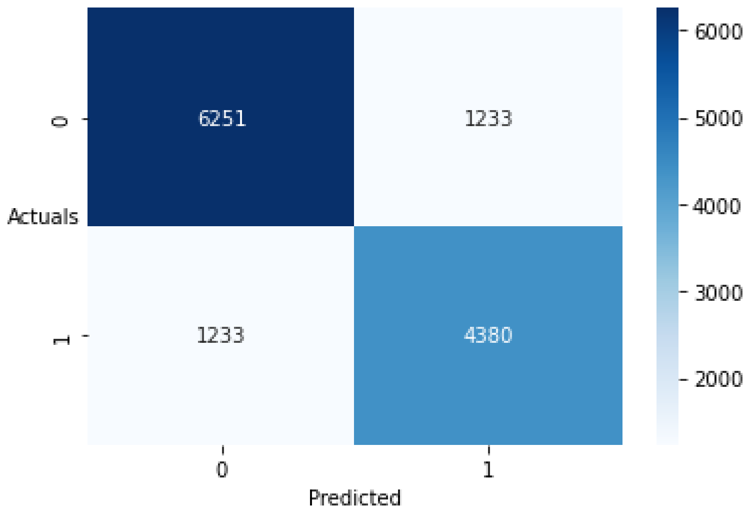


Table 19- Confusion Matrix LR (Train) Model

The roc-auc score for the LR model using SMOTE (Train) is 0.87343 or 87.34%. The roc curve is highly steep as in the figure below.

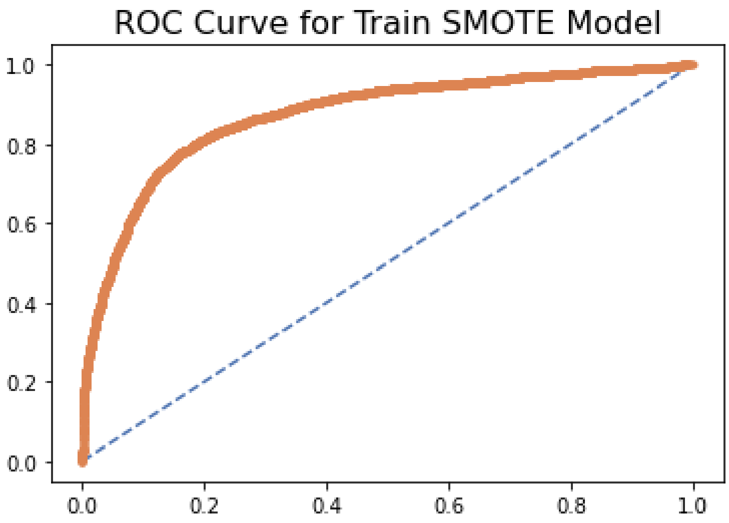


Figure 14- ROC Curve LR (Train) Model

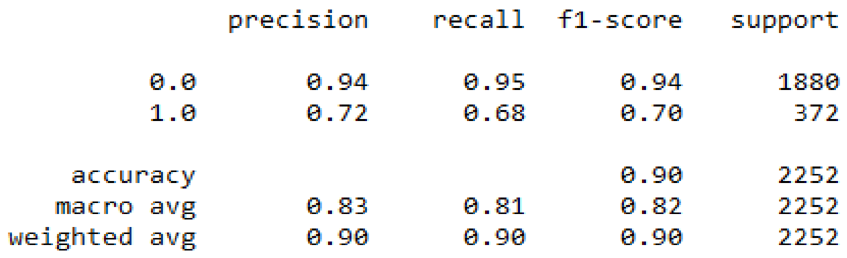
Interpretation of the model:

- This model balanced the imbalanced class
- Also this model performed well for this dataset.
- The recall score is moderate in nature which is better than other models having poor recall scores.

Model validation

CART Model

The accuracy score of the CART model (Test) is 0.90453 or 90.45% which is significant.



```
              precision    recall  f1-score   support

         0.0       0.94      0.95      0.94      1880
         1.0       0.72      0.68      0.70       372

    accuracy                           0.90      2252
   macro avg       0.83      0.81      0.82      2252
weighted avg       0.90      0.90      0.90      2252
```

Table 20- Classification Report CART (Test) Model

The classification report of the test model has a moderate recall value. This recall value is no good or beneficial for the company. The number of false negatives for the test model is 119. Reducing the number of false negatives will help us achieve a better recall value.

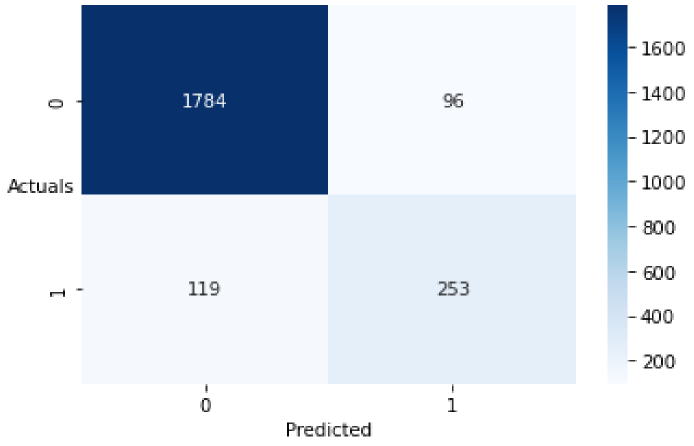


Table 21- Confusion Matrix CART (Test) Model

The roc-auc score for the CART Model (Test) is 0.92945 or 92.95%. The ROC curve is highly steep.

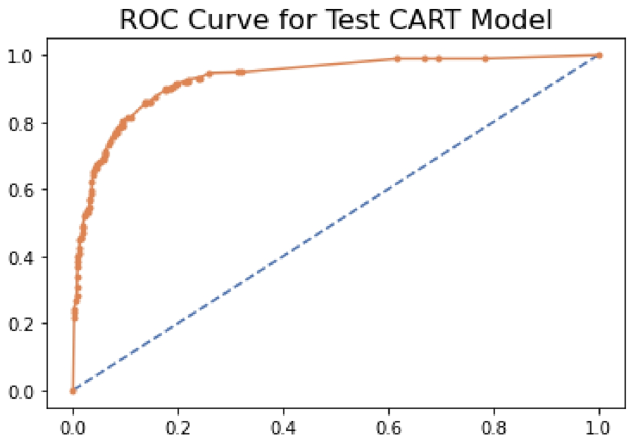


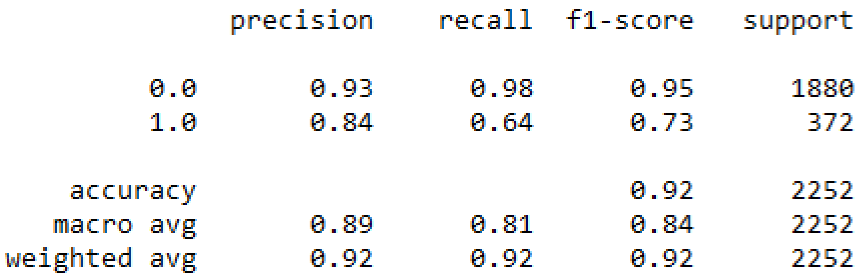
Figure 15- ROC Curve CART (Test) Model

Interpretation of the model:

- This model performed in a poor manner for this dataset.
- This test model has a poor recall score.

Random Forest model

The accuracy score of the Random Forest Model (Test) is 0.92096 or 92.10%.



```
              precision    recall  f1-score   support

         0.0       0.93      0.98      0.95      1880
         1.0       0.84      0.64      0.73       372

    accuracy                           0.92      2252
   macro avg       0.89      0.81      0.84      2252
weighted avg       0.92      0.92      0.92      2252
```

Table 22- Classification Report RF (Test) Model

The recall value of the RF model (Test) is poorer than the CART (Test) model. It has good precision value but it is not useful to us for this case.

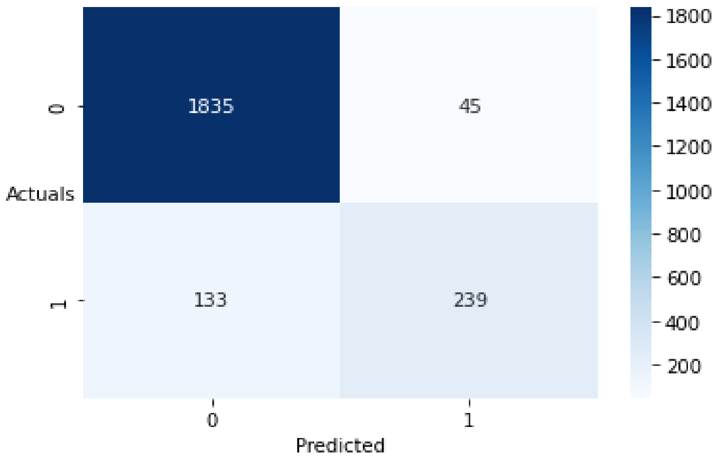


Table 23- Confusion Matrix RF (Test) Model

This model has a higher number of false negatives than the CART model and hence gives a lower recall value.
The roc-auc score for the Random Forest Model (Test) is 0.95759 or 95.76%. The curve for RF model is more steeper than the CART (Test) model.

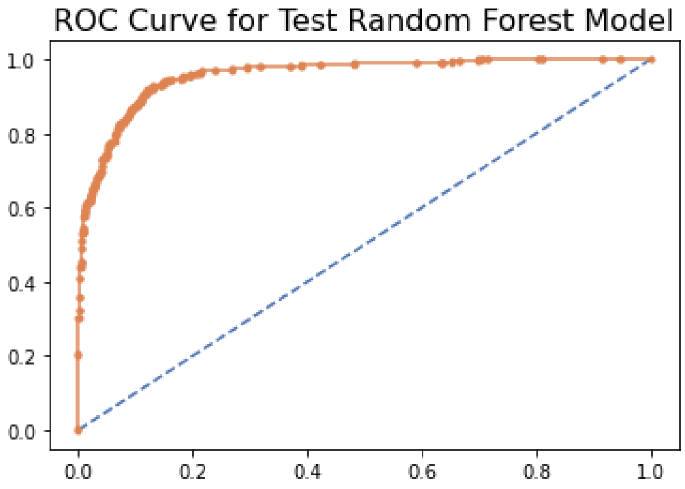


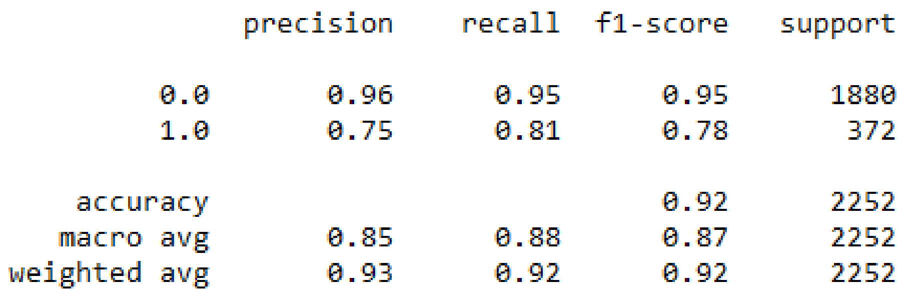
Figure 16- ROC Curve RF (Test) Model

Interpretation of the model:

- This model performed in a poor manner but has a significant roc auc score.
- This test model has a poor recall score.

Artificial Neural Network model

The accuracy score of the ANN Model (Test) is 0.92318 or 92.32%.



```
              precision    recall  f1-score   support

         0.0       0.96      0.95      0.95      1880
         1.0       0.75      0.81      0.78       372

    accuracy                           0.92      2252
   macro avg       0.85      0.88      0.87      2252
weighted avg       0.93      0.92      0.92      2252
```

Table 24- Classification Report ANN (Test) Model

This ANN test model has a good recall score than RF and CART models which means this significantly performs better than both the models. The precision and f1 scores are also moderate in value.
The confusion matrix below also shows that the number of false negatives is significantly lower giving a higher recall score.

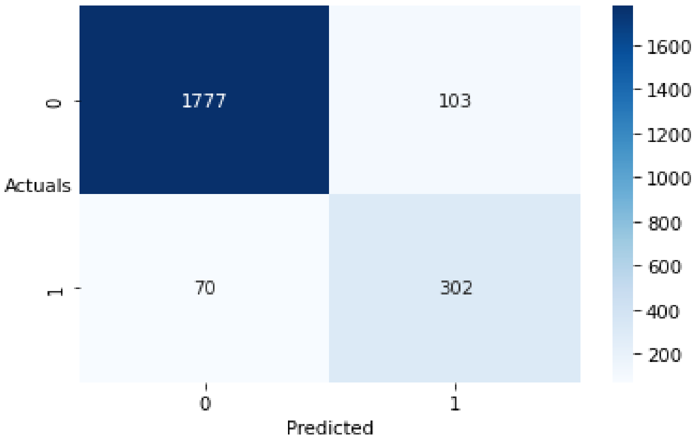


Table 25- Confusion Matrix ANN (Test) Model

The roc-auc score for the ANN Model (Test) is 0.94878 or 94.88%. The ROC curve is also pretty steep.

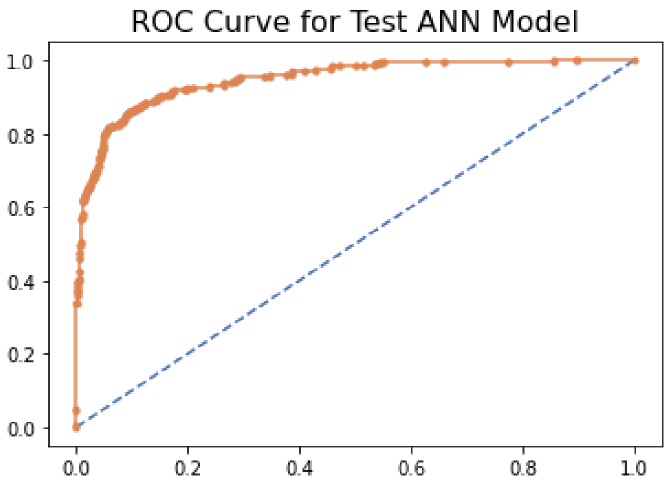
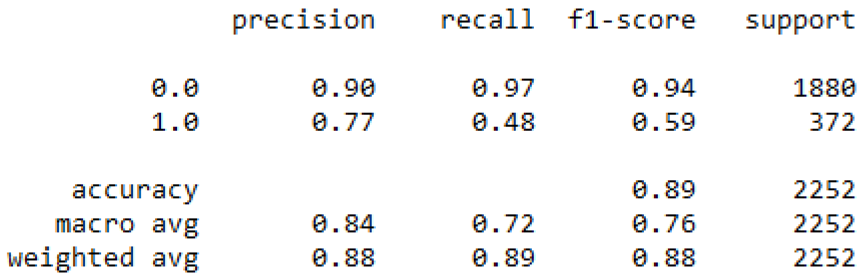


Figure 17- ROC Curve ANN (Test) Model

Interpretation of the model:

- This model performed well for this dataset.
- This model has a significant roc auc score.
- It also has a good recall score.

Backward Elimination Feature using Logistic Regression



```
              precision    recall  f1-score   support

         0.0       0.90      0.97      0.94      1880
         1.0       0.77      0.48      0.59       372

    accuracy                           0.89      2252
   macro avg       0.84      0.72      0.76      2252
weighted avg       0.88      0.89      0.88      2252
```

Table 26- Classification Report Logit (Test) Model

The above classification report shows that this model has a very poor recall score. However the precision score is quite good but it is not useful for this dataset. The accuracy score is 0.89 which is exactly same with the accuracy of train data of this Logit model.

The below confusion matrix shows that the number of false negatives are high which gives a poor recall value in this model.

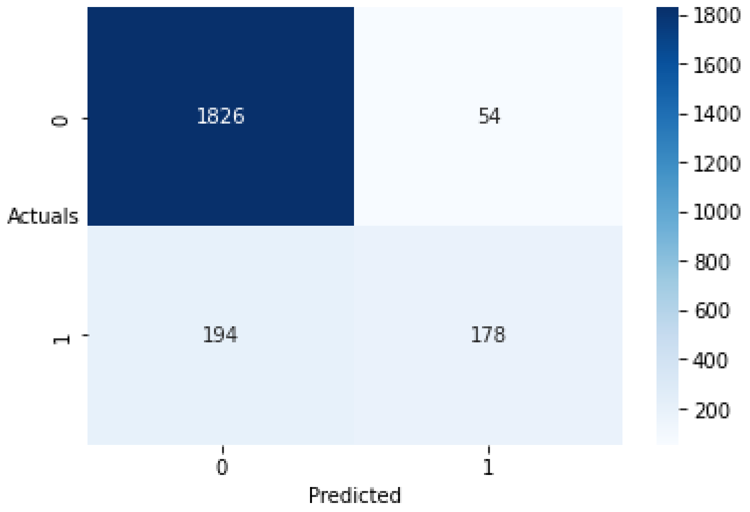


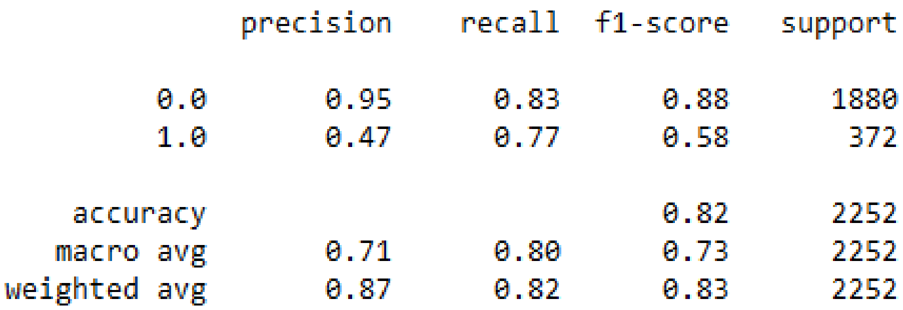
Table 27- Confusion Matrix Logit (Test) Model

Interpretation of the model:

- This model performed poorly for this dataset.
- It also has a poor recall score.

SMOTE using Logistic Regression

The accuracy score of the SMOTE Model (Test) is 0.81883 or 81.83%.



```
              precision    recall  f1-score   support

         0.0       0.95      0.83      0.88      1880
         1.0       0.47      0.77      0.58       372

    accuracy                           0.82      2252
   macro avg       0.71      0.80      0.73      2252
weighted avg       0.87      0.82      0.83      2252
```

Table 28- Classification Report SMOTE (Test) Model

The above classification report shows that this model has a good and moderate recall score for class 1. However the accuracy of the model is quite unsatisfactory.

The below confusion matrix shows that the number of false negatives are low which gives a good recall value in this model.

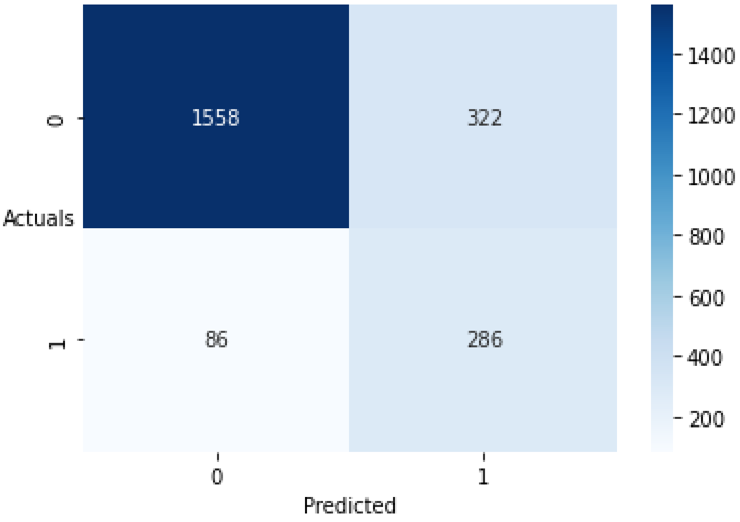


Table 29- Confusion Matrix SMOTE (Test) Model

The roc-auc score for the SMOTE Model (Train) is 0.86413 or 86.41%. The ROC curve is also steep as in the figure below.

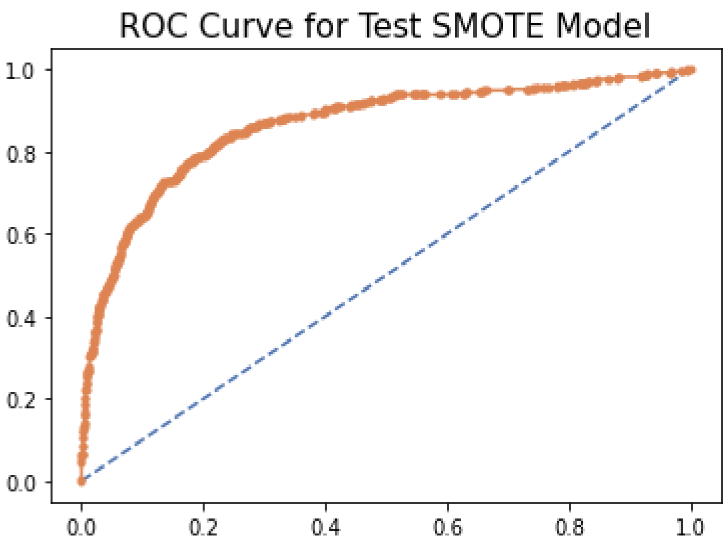
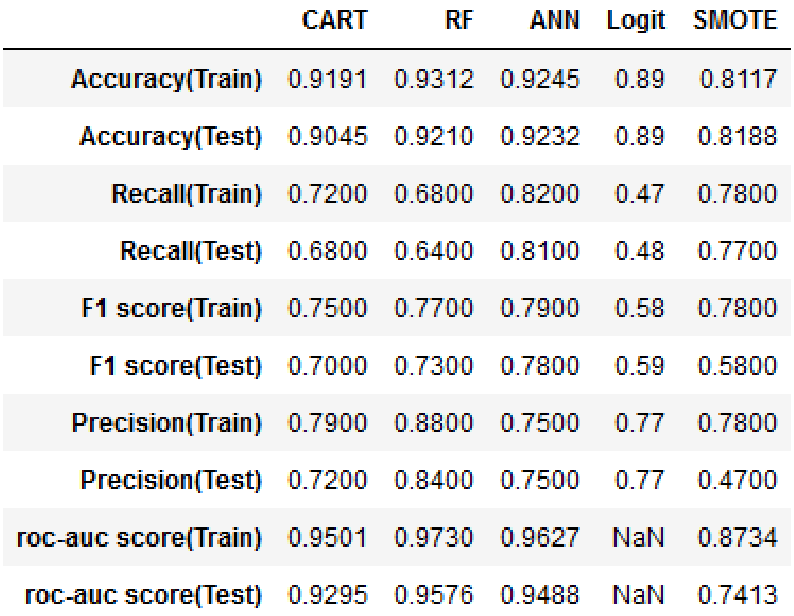


Figure 18- ROC Curve SMOTE (Test) Model

Interpretation of the model:

- This model performed poorly for this dataset.
- But it also has a moderate recall score which is quite better than other models.

Comparison Table of Different models



	CART	RF	ANN	Logit	SMOTE
Accuracy(Train)	0.9191	0.9312	0.9245	0.89	0.8117
Accuracy(Test)	0.9045	0.9210	0.9232	0.89	0.8188
Recall(Train)	0.7200	0.6800	0.8200	0.47	0.7800
Recall(Test)	0.6800	0.6400	0.8100	0.48	0.7700
F1 score(Train)	0.7500	0.7700	0.7900	0.58	0.7800
F1 score(Test)	0.7000	0.7300	0.7800	0.59	0.5800
Precision(Train)	0.7900	0.8800	0.7500	0.77	0.7800
Precision(Test)	0.7200	0.8400	0.7500	0.77	0.4700
roc-auc score(Train)	0.9501	0.9730	0.9627	NaN	0.8734
roc-auc score(Test)	0.9295	0.9576	0.9488	NaN	0.7413

Table 30- Comparison Table of Different models

Here Logit model is none other than 'Backward Elimination Feature using Logistic Regression' model.
We can interpret the following:

- The ANN turns out to be the **most optimum** model. It has the highest recall value and accuracy value. Judging from the fact that the model does not over fit, it performs way better than all of the other models.
- Out of all models Logit model does an under fitting. It performs better in the test model than the train model. But its performance is poor in overall terms. Hence this model is unfit for this dataset.
- The second highest recall value is provided by the SMOTE model but it has poor values of precision, f1 score, and roc auc score. The model is quite over fitting but provided a good recall value.
- The recall values given by CART and RF models are quite poor.

Final interpretation / recommendation

The most optimum model is the ANN model due to its lead recall value and its overall performance metrics.

Implications on business:

- A good recall value will imply the retention of customers who are on the verge of being churned. On the other hand, if the recall value is poor then it may lead to huge losses and downfall of reputation of the company.
- ANN model provides a good recall value which will help the business in retention of customers who are on the verge of being churned.
- Retention of customers will also forge customer loyalty into the business.
- This can also link a possible chance of attracting more new customers into the business

Recommendation:

- Company should provide internship programme for the employees having low tenure.

- Engage employees with profound knowledge and moderate or high tenure.
- Company can provide discounts and lucrative combos to improve their sales.
- Company can bring in new and diverse products to attract the attention of customers.