Customer Churn Prediction Report

Prepared by

Divya Sarika Amjuri

11 July 2021

1 Introduction to the business problem

1.1 Problem statement

An E-Commerce company wants to develop a model through which they can do
- Churn prediction of the accounts and
- Provide segmented offers to the potential churners.

Our goal is to develop a churn prediction model for this company and provide business recommendations for a campaign aimed at customers to reduce churn.

The campaign suggestion should be unique. The campaign offer should be in such a way that it doesn't give a lot of free/subsidized material/products to avoid any losses to the company

1.2 Need of the study/project

- Companies are facing a lot of competition in the current market and it has become a challenge to retain the existing customers in the current situation
- Account churn is a major thing for this company because
    - Each account can have multiple customers
    - Losing one account means losing multiple customers

Due to the above points, developing a churn prediction model for this company and providing business recommendations to prepare the campaign to retain customers is important.

1.3 Understanding business/social opportunity

- An E Commerce company 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
- We have to develop a churn prediction model for this company and provide business recommendations for the campaign aimed at retaining customers
- Campaign suggestions should be unique and it should be very clear on the campaign offer that this campaign doesn't give a lot of free (or subsidized) stuff thereby making a loss to the company

2 Data Report

2.1 Collection of data in terms of time, frequency and methodology

Today businesses are connected to their clients, customers, users, employees, vendors, and sometimes even their competitors. Data can tell a story about any of these relationships. With this information, organizations can improve almost any aspect of their operations.

The right data collection method can mean the difference between useful insights and time-wasting misdirection.

Organizations have several tools at their disposal for primary data collection. The methods range from traditional and simple, such as a face-to-face interview, to more sophisticated ways to collect and analyze data.

The most commonly used methods are:

- Published literature sources
- Surveys (email and mail)
- Interviews (telephone, face-to-face or focus group)
- Observations
- Documents and records
- Experiments.

The data collection methods in research methodology:

- Interviews
- Questionnaires and surveys
- Observations
- Documents and records
- Focus groups
- Oral histories

Qualitative data collection looks at several factors to provide a depth of understanding to raw data.

2.1.1 Primary Data

Primary data refers to data obtained directly from individuals, objects or processes. Quantitative or qualitative data can be collected using this approach. Such data is usually collected solely for the research problem we will study.

2.1.2 Secondary Data

When we collect data after another researcher or agency that initially gathered it makes it available, we are gathering secondary data. Examples of secondary data are census data published by the US Census Bureau, stock prices data published by CNN and salaries data published by the Bureau of Labor Statistics.

2.1.3 Methods Employed in Primary Data Collection

When we decide to conduct original research, the data we gather can be quantitative or qualitative. Generally, we collect quantitative data through sample surveys, experiments and observational studies. We obtain qualitative data through focus groups, in-depth interviews and case studies.

2.1.4 Observational Data Collection Methods

In an observational data collection method, we acquire data by observing any relationships that may be present in the phenomenon we are studying. There are four types of observational methods that are available to us as a researcher:

- Cross-sectional
- Case-control
- Cohort
- Ecological

2.1.5 Experiments

An experiment is a data collection method where we change some variables and observe their effect on other variables. The variables that we manipulate are referred to as independent while the variables that change as a result of manipulation are dependent variables.

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

This dataset has 11260 observations and 19 attributes.

- 5 attributes named *City_Tier*, *CC_Contacted_LY*, *Service_Score*, *CC_Agent_Score*, *Complain_ly* are of **float64** type
- 12 attributes named *Tenure, Payment, Gender, Account_user_count, account_segment , Marital_Status, rev_per_month ,rev_growth_yoy, coupon_used_for_payment ,Day_Since_CC_connect, cashback* and *Login_device* are of **object** type
- 2 attributes named *AccountId* and *Churn* are of **integer** type

We can see the number of rows and columns from the Figure No.1 and Figure No.2 shows us the datatypes of attributes

```
data.shape
```

```
(11260, 19)
```

Figure No. 1

```
data.dtypes
```

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

Figure No. 2

We have dropped the column *AccountId* because it's not necessary for our model. After dropping that column the shape of the dataset is as shown below in Figure No. 3

```
data.shape
```

```
(11260, 18)
```

Figure No. 3

Let's start the data exploration step with the head function to look at the first 5 rows, shown in Figure No. 4 and Figure No. 5

```
data = pd.read_excel('Customer Churn Data.xlsx',sheet_name = 'Data for DSBA')
data.head(5)
```

	AccountID	Churn	Tenure	City_Tier	CC_Contacted_LY	Payment	Gender	Service_Score	Account_user_count	account_segment	CC_Agent_Score	Mar
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	

Figure No. 4

```
data = pd.read_excel('Customer Churn Data.xlsx',sheet_name = 'Data for DSBA')
data.head(5)
```

CC_Agent_Score	Marital_Status	rev_per_month	Complain_ly	rev_growth_yoy	coupon_used_for_payment	Day_Since_CC_connect	cashback	Login_device
2.0	Single	9	1.0	11	1	5	159.93	Mobile
3.0	Single	7	1.0	15	0	0	120.9	Mobile
3.0	Single	6	1.0	14	0	3	NaN	Mobile
5.0	Single	8	0.0	23	0	3	134.07	Mobile
5.0	Single	3	0.0	11	1	3	129.6	Mobile

Figure No. 5

We can see that there are some unwanted variables in the dataset, which is the reason for the datatypes of some columns showing as object despite being numerical. Let's do further exploration.

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

By looking at the info as shown in Figure No. 6 given in the next page, we can say that there are some *null* values in the dataset.

- *AccountID* is the account unique identifier and Churn is the account churn flag which is the Target variable
- *Tenure* represents Tenure of account
- *City_Tier* shows us the tier of primary customer's city
- *CC_Contacted_LY* is how many times all the customers of the account have contacted customer care in the last 12 months
- *Payment* is the preferred payment mode of the customers
- *Gender* is the gender of the customer
- *Service_Score* represents satisfaction score for the service provided by company
- *Account_user_count* shows us the number of customers tagged with this account
- *account_segment* is the account segmentation on the basis of spend
- *CC_Agent_Score* is satisfaction score given by customers for customer care service
- *Marital_Status* is whether the customer is married or not
- *rev_per_month* is the monthly average revenue generated by account in last 12 months
- *Complain_ly* shows us if any complaints has been raised by account in last 12 months
- *rev_growth_yoy* is the revenue growth percentage of the account (last 12 months vs last 24 to 13 month)
- *coupon_used_ly* represents how many times customers have used coupons to do the payment in last 12 months
- *Day_Since_CC_connect* is the number of days since no customers in the account has contacted the customer care
- *cashback_ly* is the monthly average cashback generated by account in last 12 months

- *Login_device* is preferred login device of the customers in the account

```
data.info()
```

```
<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
```

Figure No. 6

3 Exploratory data analysis

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

Lets see the churn distribution among variables.

```
data['Churn'].value_counts()
```

```
0    9364
1    1896
Name: Churn, dtype: int64
```

Figure No. 7

- We can say that from Figure No. 7, the number of customers churned is less than customers not churned.

Figure No. 8 and Figure No. 9 represent a bar chart and pie chart which shows the churn distribution.

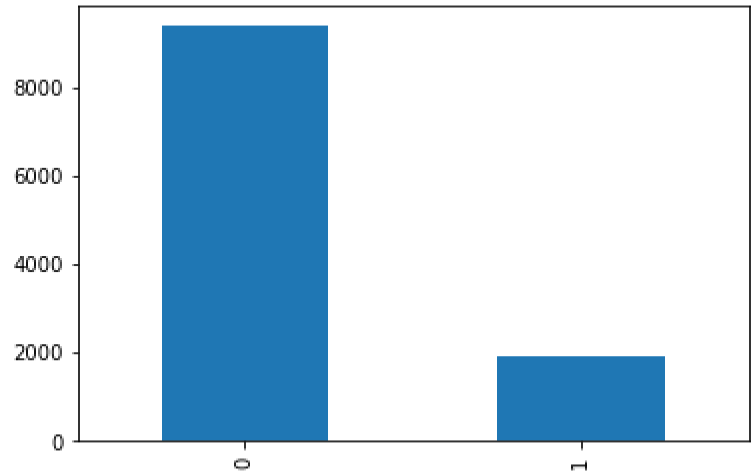


Figure No. 8

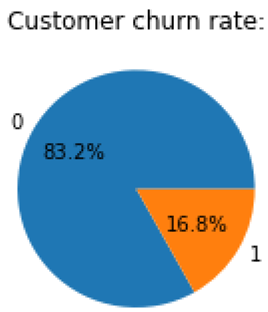


Figure No. 9

Now let's check the distribution of the data.
- The X-axis groups the observations from minimum to maximum along the axis on the basis of the discrete points or class intervals
- The Y-axis measures the frequency of occurrence of observations for each discrete point or class interval.

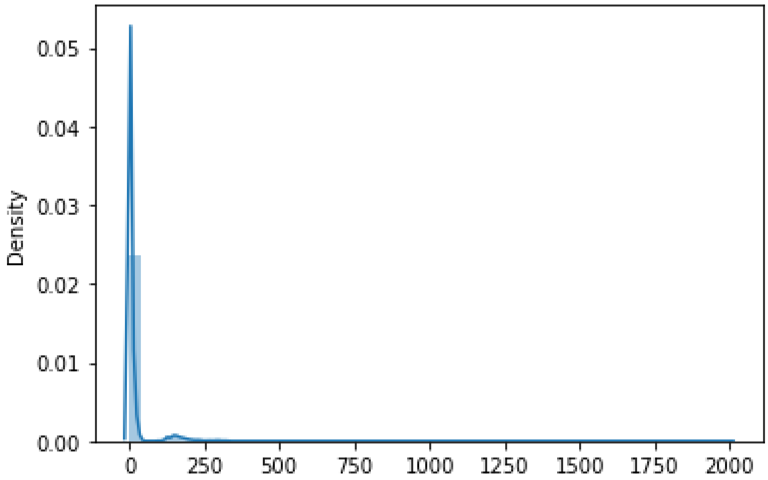
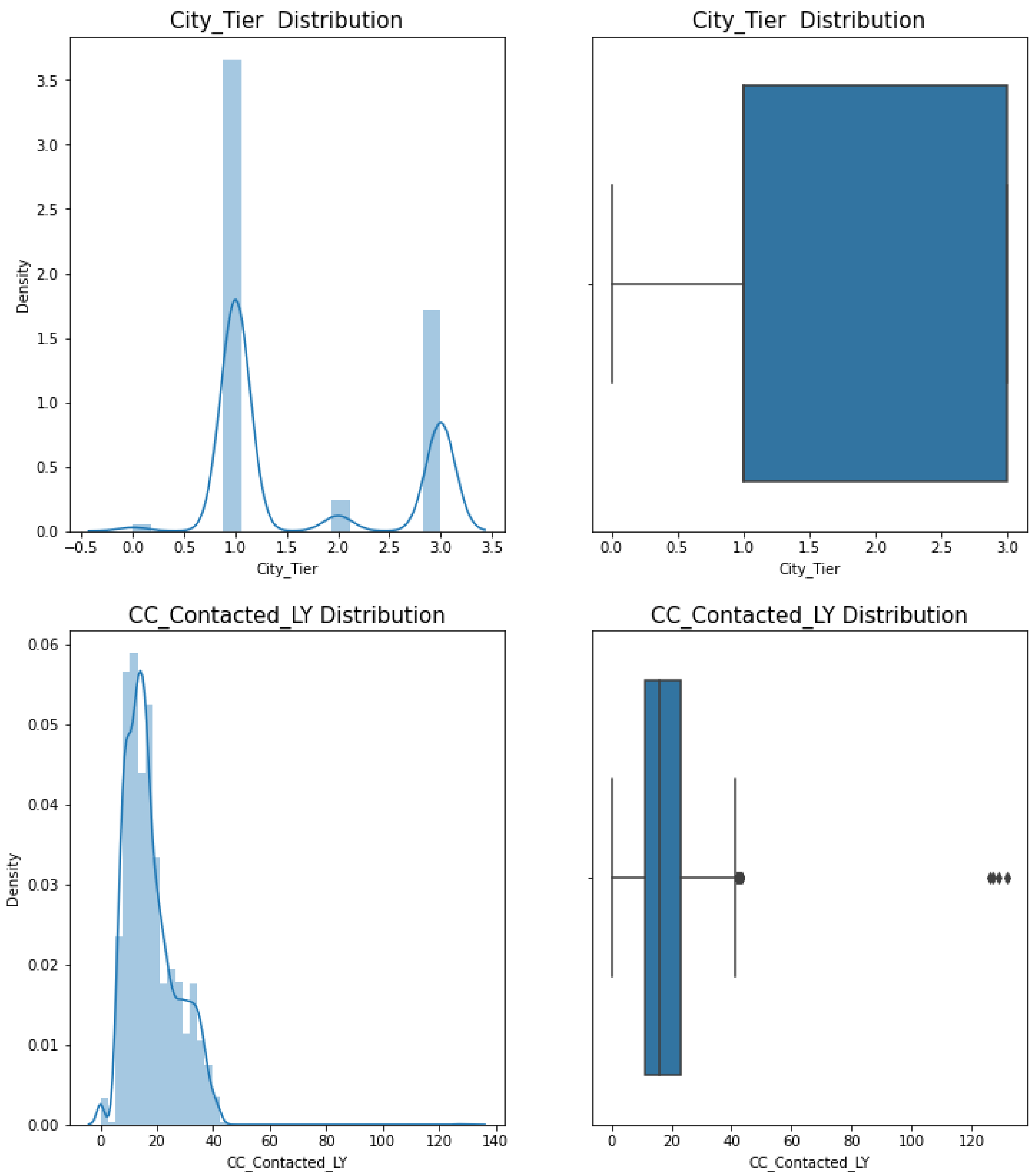


Figure No. 10

Figure No. 10 shows a frequency polygon superimposed on a histogram, which is obtained by using the seaborn package.

Seaborn automatically creates class intervals. The number of bins can also be manually set. We can say that the distribution is right-skewed, also known as positively skewed, which tells us the mean is greater than the median. This is the case because skewed-right data have a few large values that drive the mean upward but do not affect where the exact middle of the data is the median.

Now let's check the distribution and spread for every continuous attribute



City_Tier Distribution
Density
City_Tier
City_Tier Distribution
City_Tier
CC_Contacted_LY Distribution
Density
CC_Contacted_LY
CC_Contacted_LY Distribution
CC_Contacted_LY

Figure No. 11

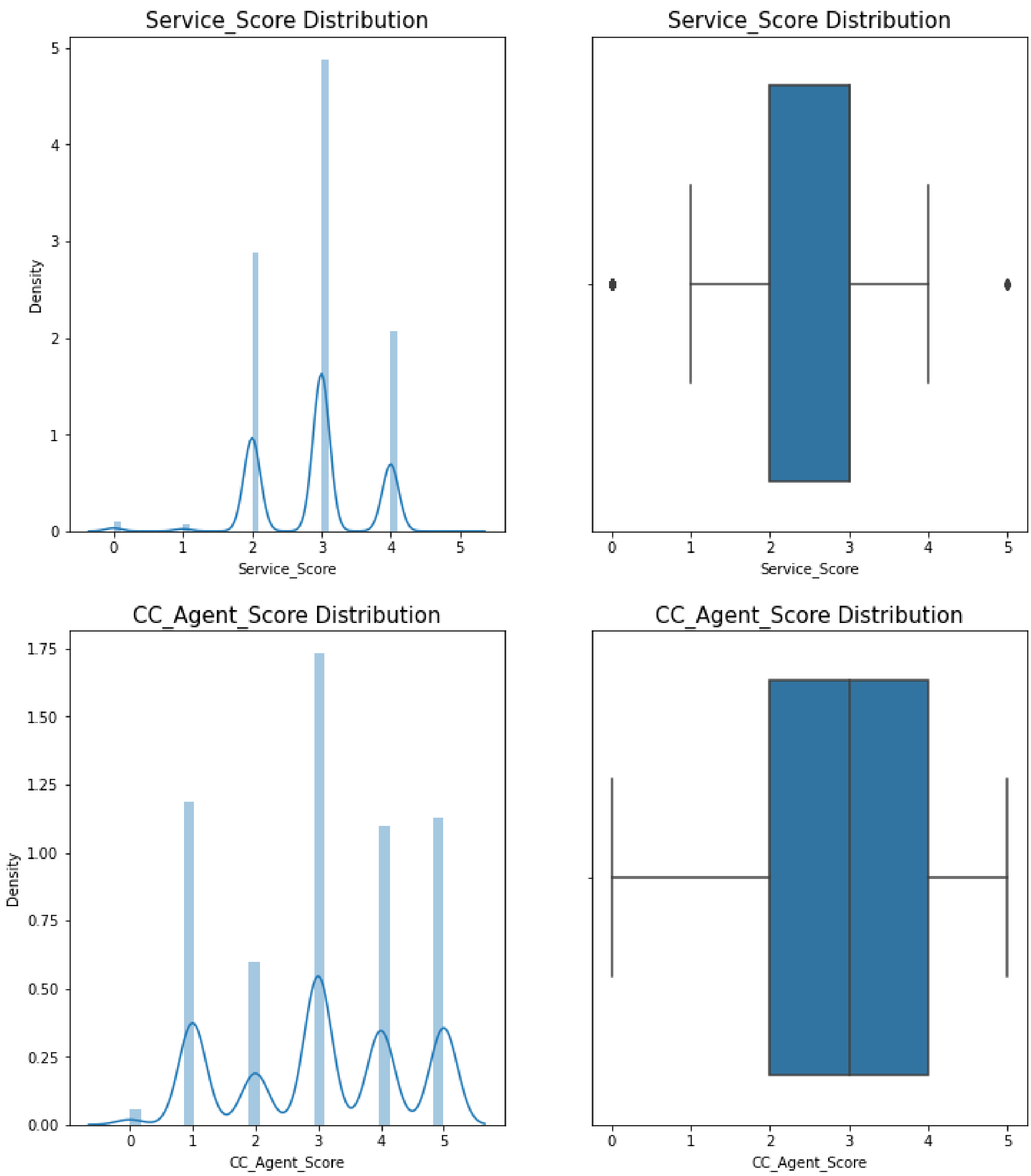

Service_Score Distribution
Density
5
4
3
2
1
0
0
1
2
3
4
5
Service_Score
Service_Score Distribution
0
1
2
3
4
5
Service_Score
CC_Agent_Score Distribution
Density
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
0
1
2
3
4
5
CC_Agent_Score
CC_Agent_Score Distribution
0
1
2
3
4
5
CC_Agent_Score

Figure No. 12

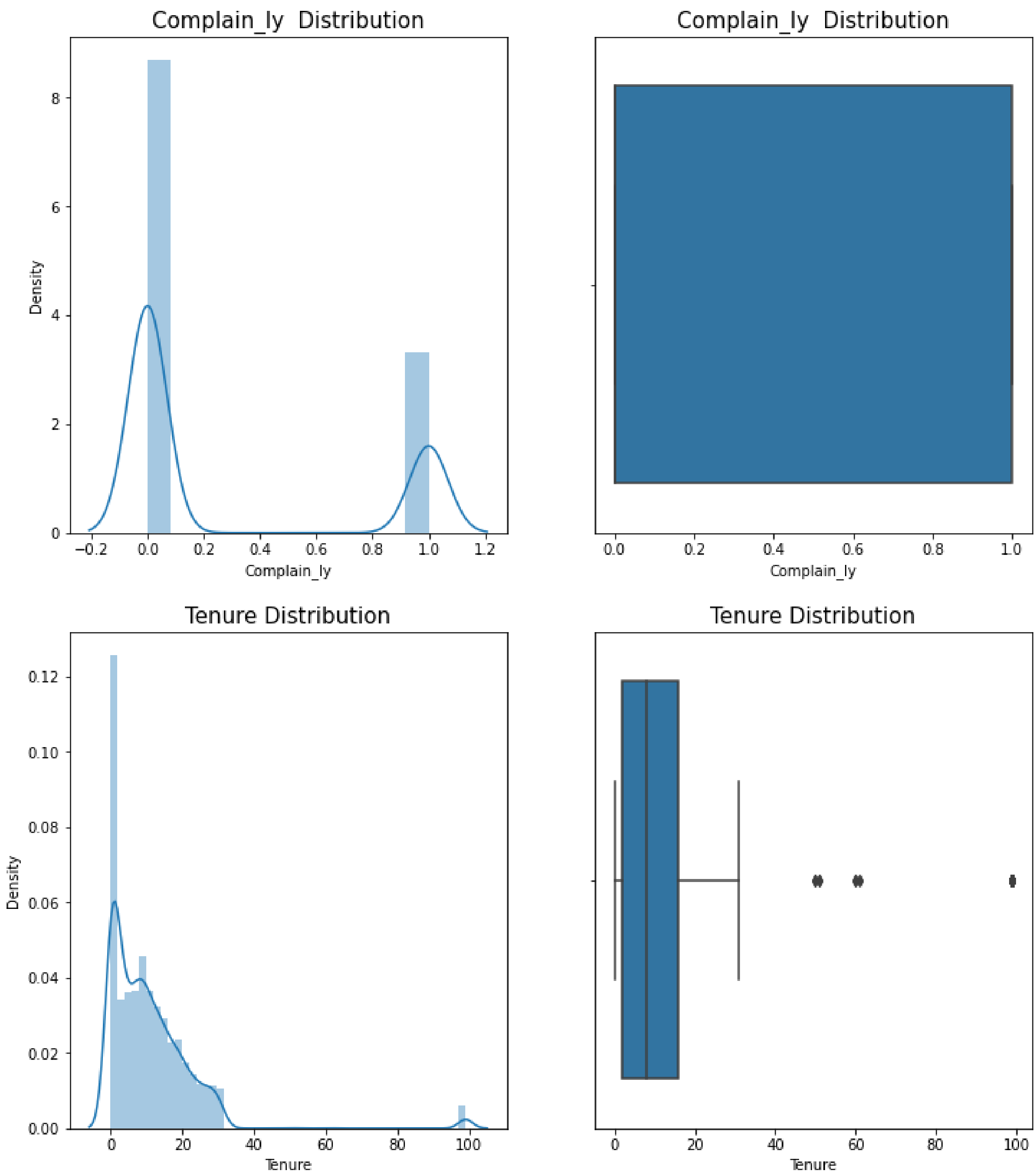
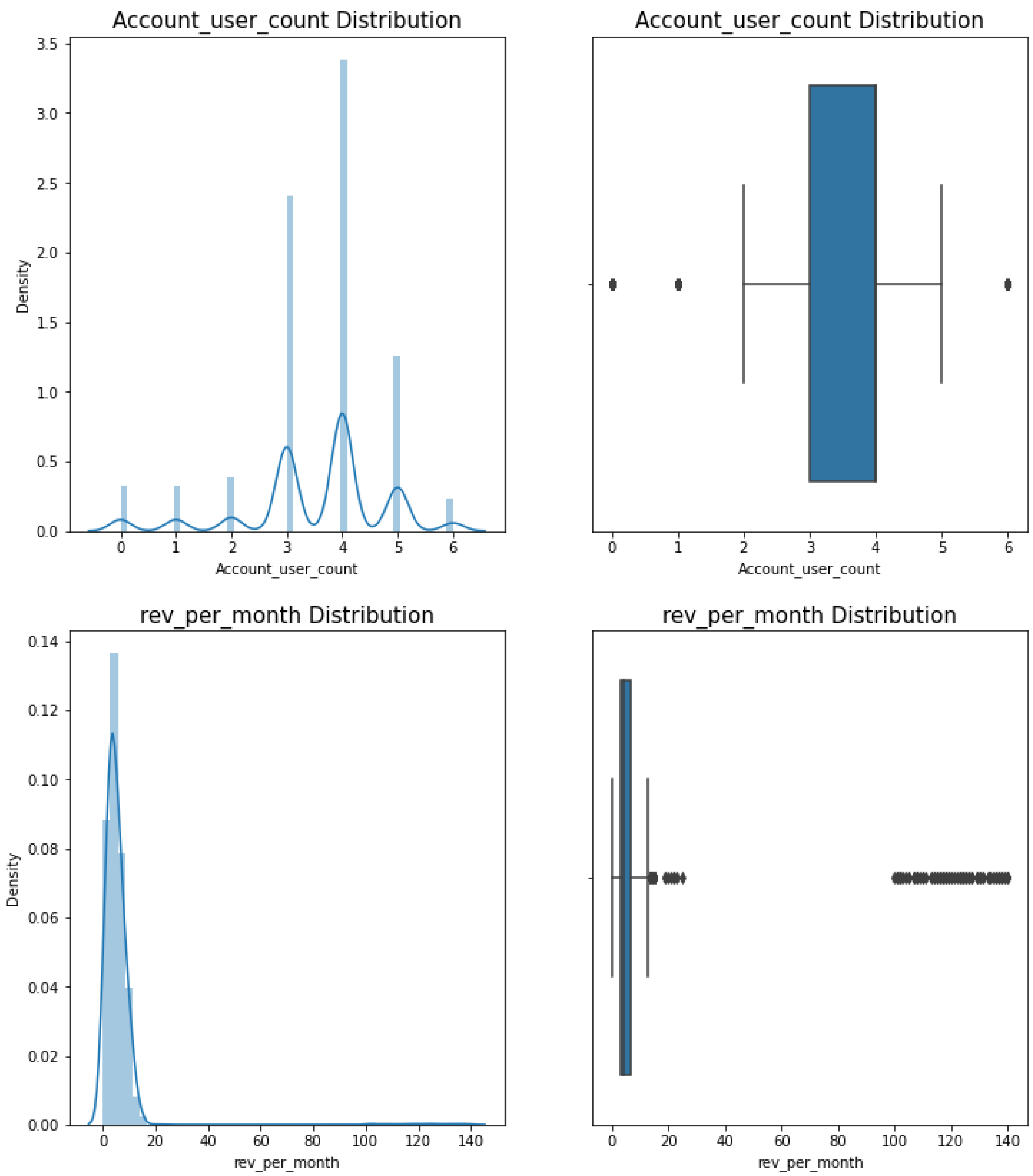

Complain_ly Distribution
Density
Complain_ly
Complain_ly Distribution
Complain_ly
Tenure Distribution
Density
Tenure
Tenure Distribution
Tenure

Figure No. 13



Account_user_count Distribution
Density
Account_user_count
Account_user_count Distribution
Account_user_count
rev_per_month Distribution
Density
rev_per_month
rev_per_month Distribution
rev_per_month

Figure No. 14

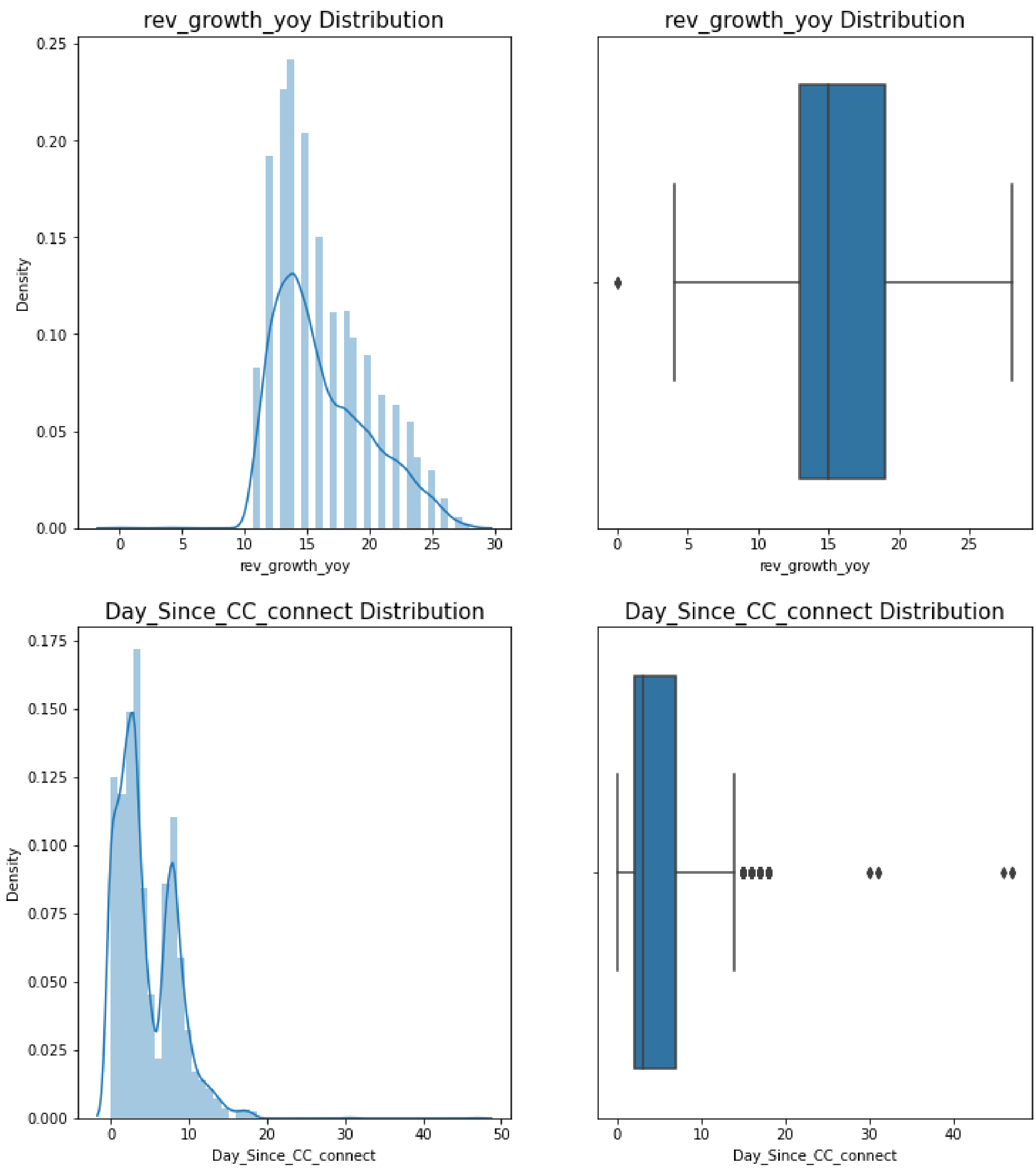

rev_growth_yoy Distribution
Density
rev_growth_yoy
rev_growth_yoy Distribution
rev_growth_yoy
Day_Since_CC_connect Distribution
Density
Day_Since_CC_connect
Day_Since_CC_connect Distribution
Day_Since_CC_connect

Figure No. 15

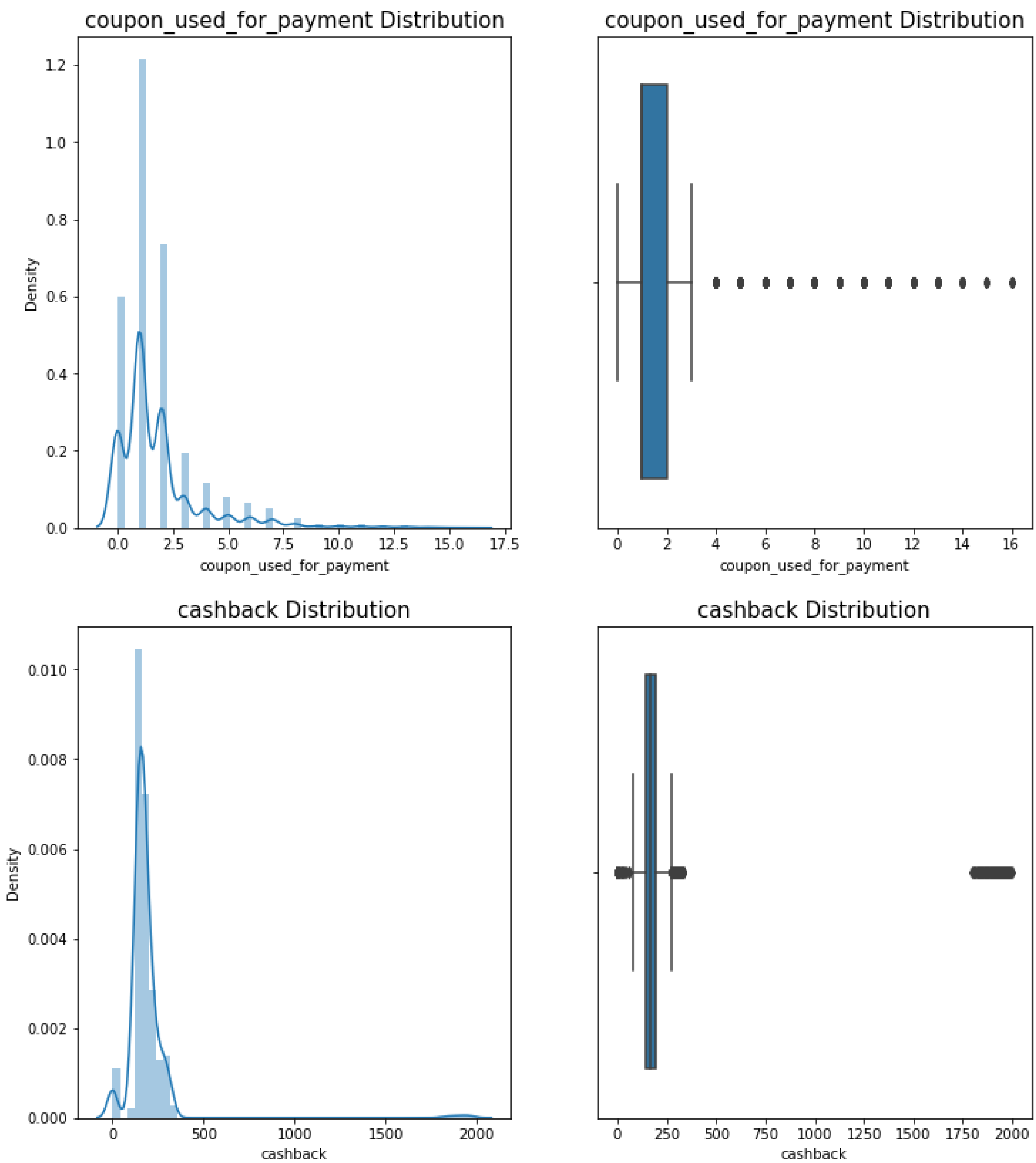


Figure No. 16

Figure No. 11, Figure No.12, Figure No.13, Figure No.14, Figure No.15 and Figure No.16 show the distribution of every continuous variable and also their box plots.

There seems to be multiple peaks in *City_Tier, Service_Score , CC_Agent_Score,Complain_ly, Account_user_count, Day_Since_CC_connect* and *coupon_used_for_payment* which means there seems to be some clusters.

For the variables, *CC_Contacted_LY, Tenure , rev_per_month, rev_growth_yoy* and *cashback*, the distribution is right skewed.

Let’s see the plots which show the Churn rate for each category of the categorical features.

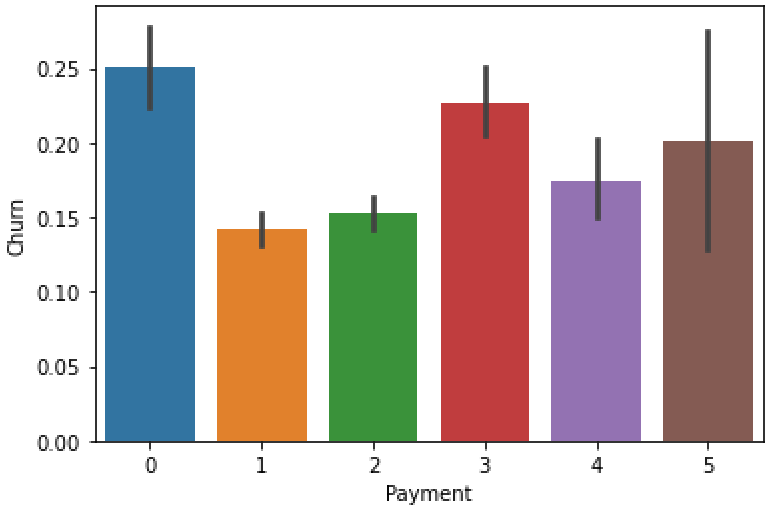


Figure No. 17

From Figure No. 17, the churn rate for different types of payment are shown. Each color shows a different payment mode that is 'Debit Card', 'UPI', 'Credit Card', 'Cash on Delivery' and 'E wallet'. Looks like people with debit cards and ewallet for payment are more churned compared to others.

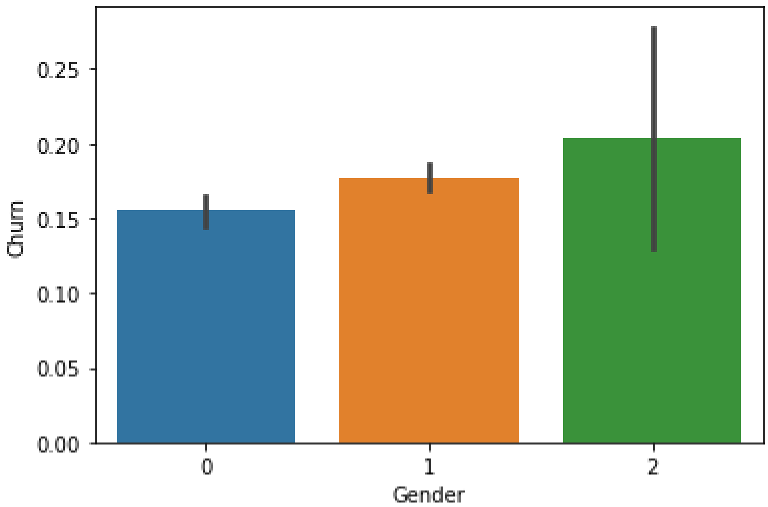


Figure No. 18

Figure No. 18 shows the churn rate for gender. Since *nan* values aren't treated by the time, it's showing 3 values. Green represents *nan* values. O represents female and 1 represents male. We can say that from figure 18, Male are churning more than females.

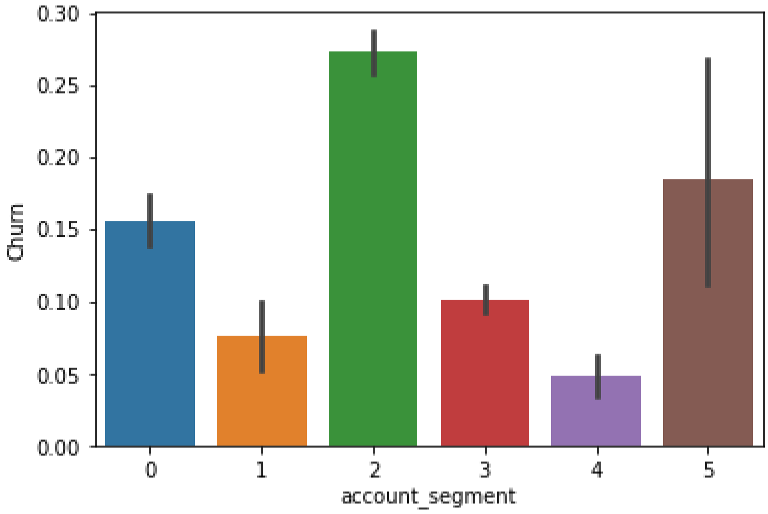


Figure No. 19

Figure No. 19 shows the churn rate for the account segmentation on the basis of spend. The colors represent 'Super', 'Regular Plus', 'Regular', 'HNI', nan, 'Super Plus' respectively. Looks like people with regular are more churning and regular plus are less churning.

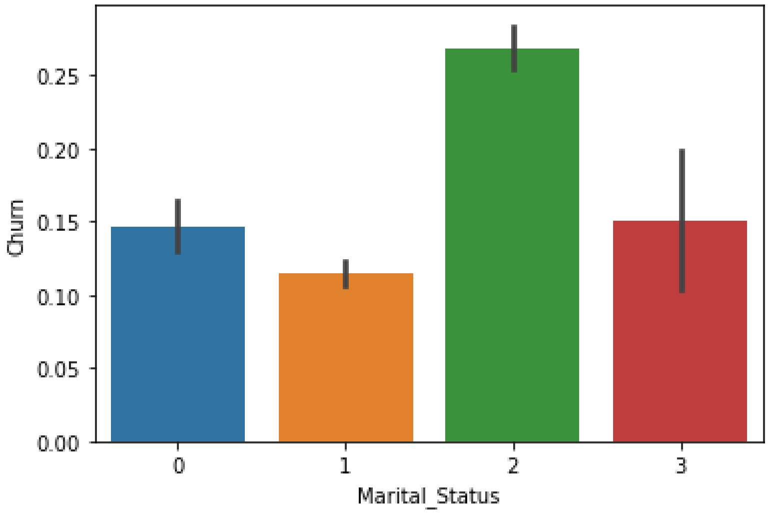


Figure No. 20

Figure No. 20 shows the churn rate for the marital status of the primary customer of the account. The colors represent ‘Single', 'Divorced', 'Married', nan respectively. Looks like people who are married are more churning and divorced are less churning.

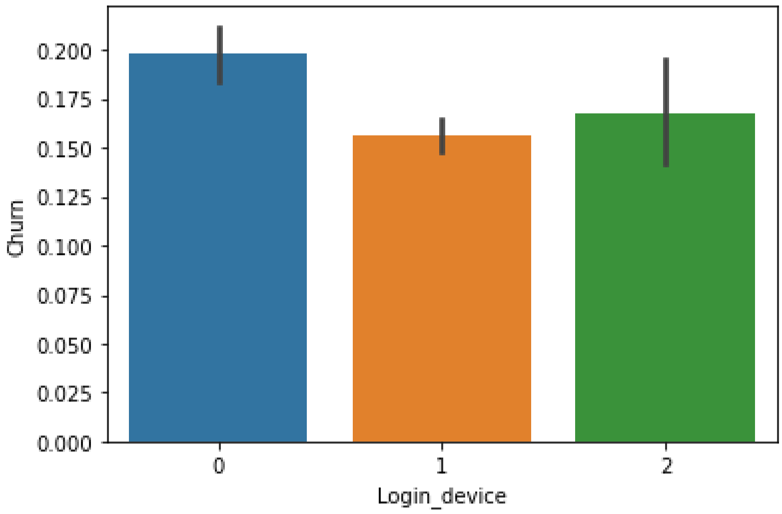


Figure No. 21

Figure No. 21 shows the churn rate for the Preferred login device of the customers in the account. The colors represent 'Mobile', 'Computer', and nan respectively. Looks like people with mobile rather than computers mostly churned.

3.2 Bivariate analysis (relationship between different variables, correlations)

Bivariate analysis is performed to understand interactions between different fields in the dataset or finding interactions between variables. A Scatter plot gives us an idea of the association between two variables

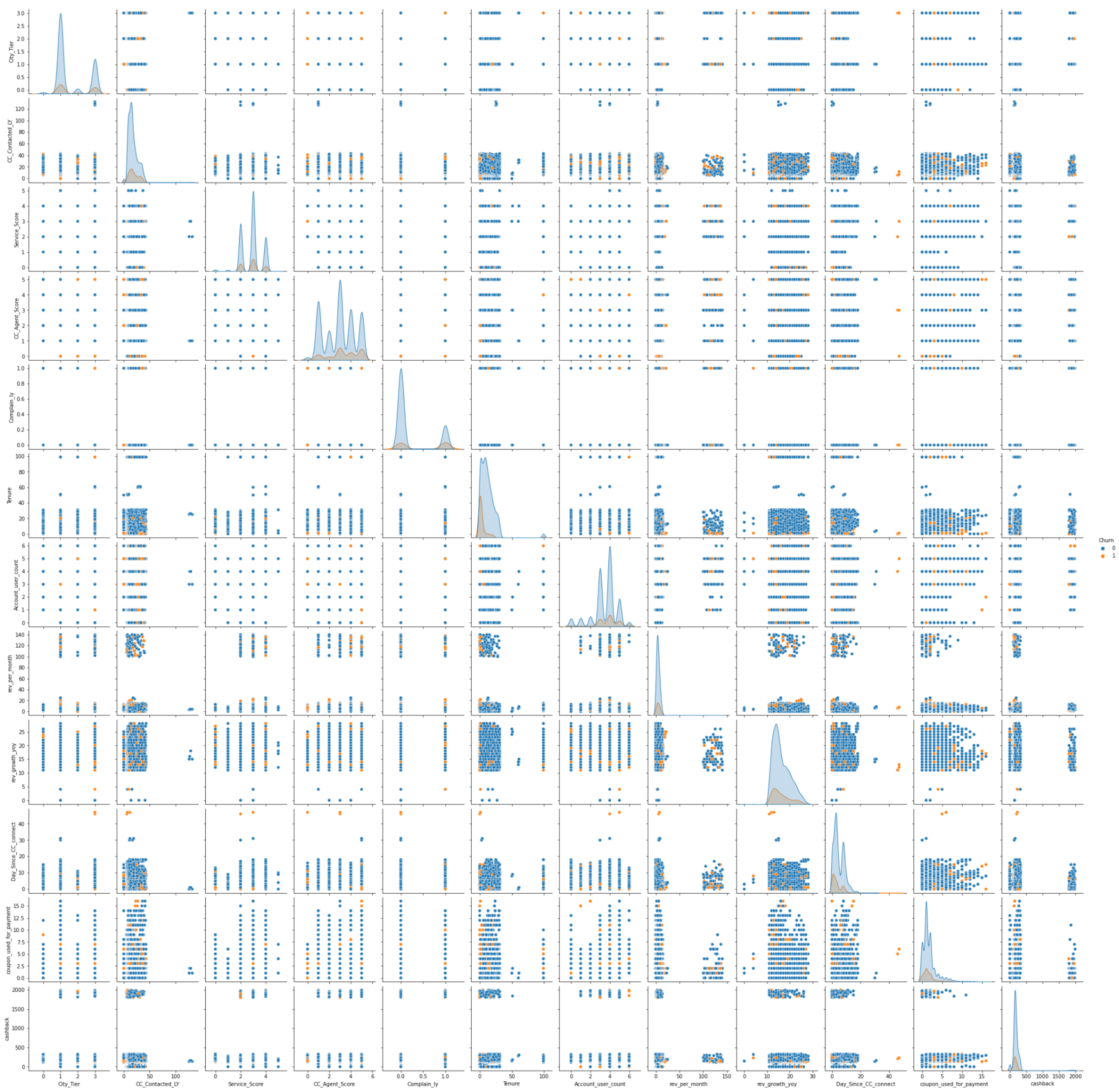


Figure No. 22

Figure No. 22 shows scatter diagrams, which are plotted for all the numerical columns in the dataset. A scatter plot is a visual representation of the degree of correlation between any two columns. The pair plot function in seaborn makes it very easy to generate joint scatter plots for all the columns in the data.

We observe that the same columns have outliers and there seems to be some clusters.

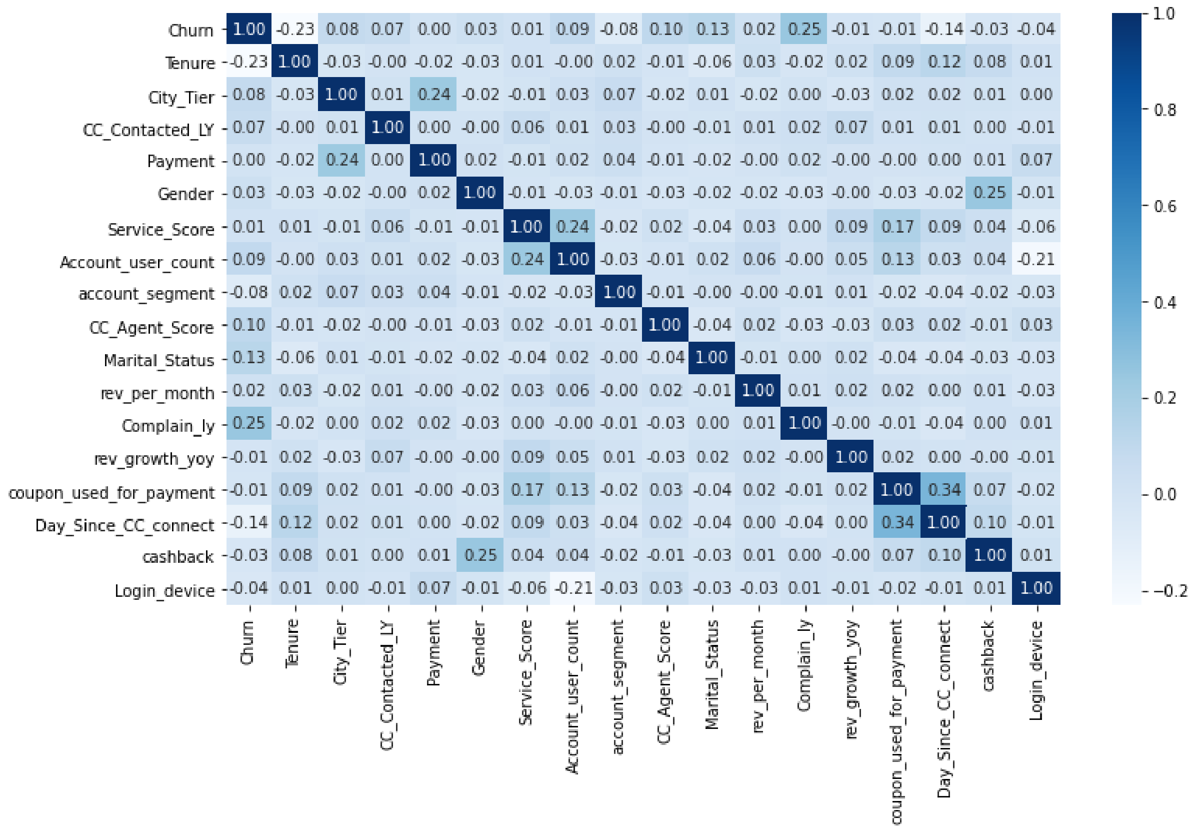


Figure No. 23

Figure No. 23 represents Heat map, which is a graphical representation of data that uses a system of color-coding to represent different values. It shows that all parameters are least correlated with each other.

3.3 Removal of unwanted variables (if applicable)

We can see that from Figure No. 24, there are some unwanted variables in the dataset which leads us to see the wrong datatype for variables as shown in Figure No. 2.

```
data.rev_growth_yoy.unique()
```

```
array([11, 15, 14, 23, 22, 16, 12, 13, 17, 18, 24, 19, 20, 21, 25, 26,
       '$', 4, 27, 28], dtype=object)
```

Figure No. 24

Lets treat these special characters by using the following function as shown in Figure No. 25

```
# a list with all missing value formats

missing_value_formats = ["$","#","*","+", "@","&&&&"]
data = pd.read_excel('Customer Churn Data.xlsx',sheet_name = 'Data for DSBA', na_values = missing_value_formats)
```

Figure No. 25

After treating the special characters, we can see the data types of variables changed into numeric as shown in Figure No. 26.

```
data.dtypes

AccountID                  int64
Churn                      int64
Tenure                   float64
City_Tier                float64
CC_Contacted_LY          float64
Payment                   object
Gender                    object
Service_Score            float64
Account_user_count       float64
account_segment           object
CC_Agent_Score           float64
Marital_Status            object
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              object
dtype: object
```

Figure No. 26

3.4 Missing Value treatment (if applicable)

We can say that from Figure No. 27, dataset has *null* values in almost every column except *churn*, *payment*, *gender, account_segment, marital_status* and *login_device*.

```
data.isnull().sum()
```

```
Churn                      0
Tenure                   218
City_Tier                112
CC_Contacted_LY          102
Payment                    0
Gender                     0
Service_Score             98
Account_user_count       444
account_segment            0
CC_Agent_Score           116
Marital_Status             0
rev_per_month            791
Complain_ly              357
rev_growth_yoy             3
coupon_used_for_payment    3
Day_Since_CC_connect     358
cashback                 473
Login_device               0
dtype: int64
```

Figure No. 27

Lets treat missing values by using replace function. I have treated *null* values by replacing them with 0's. Figure No. 28 shows the dataset after treating missing values.

```
data.isnull().sum().sum()
```

```
0
```

Figure No. 28

3.5 Outlier treatment (if required)

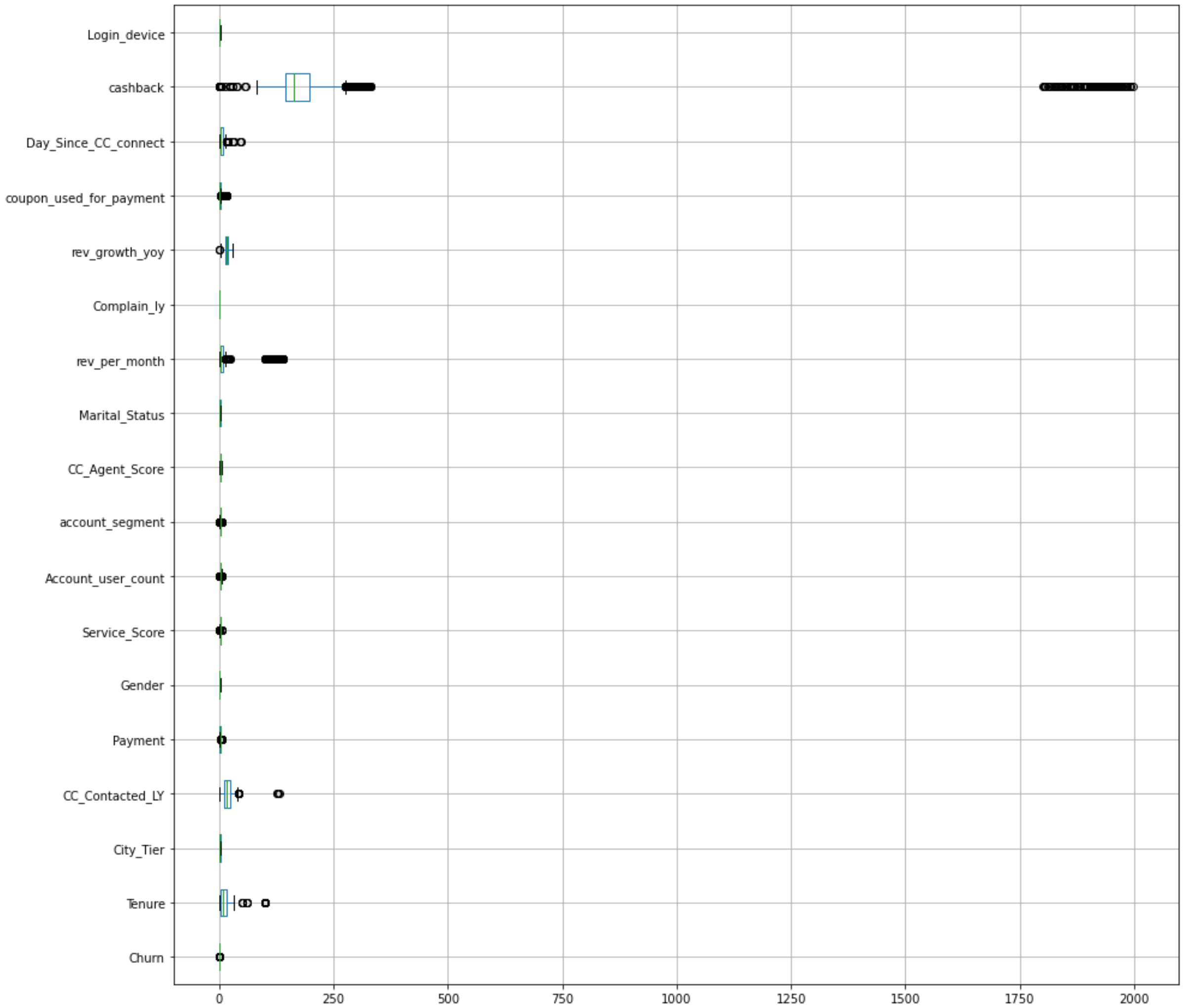


Figure No. 29

From Figure No. 29, there are outliers in almost every continuous variable column except *City_Tier*, *CC_Agent_Score* and *Complain_ly*

Lets treat them by using a custom function as shown below in Figure No. 30, which says that if for a particular column the max value is greater than that assigned max value. Same logic for min value

```
def remove_outlier(col):
    sorted(col)
    Q1,Q3=np.percentile(col,[25,75])
    IQR=Q3-Q1
    lower_range= Q1-(1.5 * IQR)
    upper_range= Q3+(1.5 * IQR)
    return lower_range, upper_range
```

Figure No. 30

Figure No. 31 shows the data after treating outliers. I did not treat outliers for categorical and the target variable.

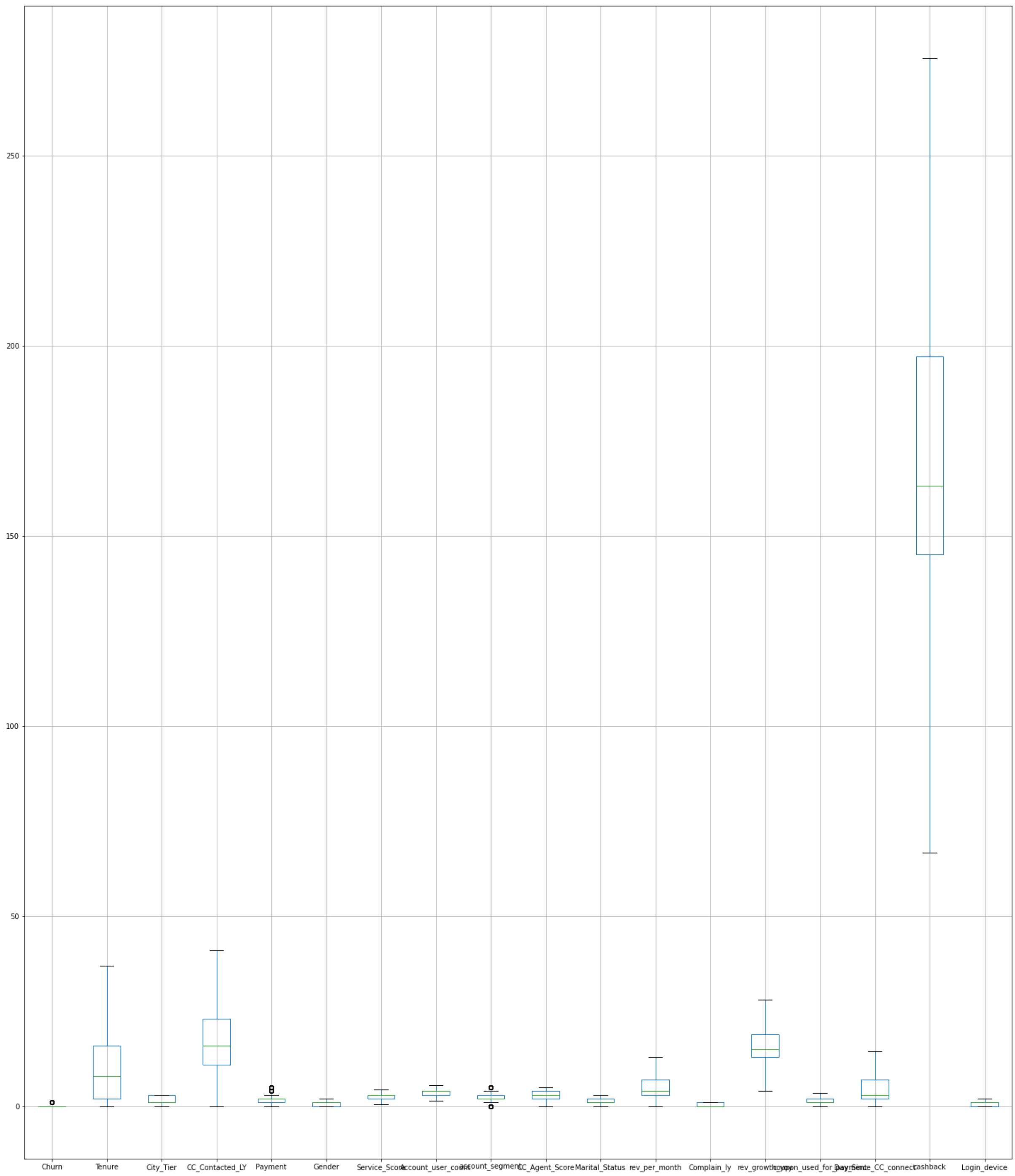


Figure No. 31

3.6 Variable transformation (if applicable)

Let's change the object data types by encoding. Here we are using label encoding in which we replace the categorical value with a numeric value between 0 and the number of classes minus 1. If the categorical variable value contains 5 distinct classes, we use (0, 1, 2, 3, and 4).

```python
#Fetch features of type Object
objFeatures = data.select_dtypes(include="object").columns

#Iterate a loop for features of type object
from sklearn import preprocessing
le = preprocessing.LabelEncoder()

for feat in objFeatures:
    data[feat] = le.fit_transform(data[feat].astype(str))
```

Figure No. 32

Figure No. 32 shows the label encoding function. After label encoding, the datatypes are converted into numerical as shown in Figure No. 33

```
data.dtypes
```

```
Churn                       int64
Tenure                    float64
City_Tier                 float64
CC_Contacted_LY           float64
Payment                     int32
Gender                      int32
Service_Score             float64
Account_user_count        float64
account_segment             int32
CC_Agent_Score            float64
Marital_Status              int32
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                int32
dtype: object
```

Figure No. 33

3.7 Addition of new variables (if required)

Adding new variables is not required but there are some variable names that need to be changed in gender and account_segment columns. After changing the names, the columns looks like below as in Figure No. 34

```python
data.Gender.replace(['F', 'M'], ['Female', 'Male'], inplace=True)
```

```python
data.Gender.unique()
```

```
array(['Female', 'Male', nan], dtype=object)
```

```python
data.account_segment.replace(['Regular +', 'Super +'], ['Regular Plus', 'Super Plus'], inplace=True)
```

```python
data.account_segment.unique()
```

```
array(['Super', 'Regular Plus', 'Regular', 'HNI', nan, 'Super Plus'],
      dtype=object)
```

Figure No. 34

After encoding, let's look at the head of the data and summary statistics

```
data.head()
```

	Churn	Tenure	City_Tier	CC_Contacted_LY	Payment	Gender	Service_Score	Account_user_count	account_segment	CC_Agent_Score	Marital_Status
0	1	4.0	3.0	6.0	2	0	3.0	3.0	3	2.0	2
1	1	0.0	1.0	8.0	4	1	3.0	4.0	2	3.0	2
2	1	0.0	1.0	30.0	2	1	2.0	4.0	2	3.0	2
3	1	0.0	3.0	15.0	2	1	2.0	4.0	3	5.0	2
4	1	0.0	1.0	12.0	1	1	2.0	3.0	2	5.0	2

```
data.head()
```

CC_Agent_Score	Marital_Status	rev_per_month	Complain_ly	rev_growth_yoy	coupon_used_for_payment	Day_Since_CC_connect	cashback	Login_device
2.0	2	9.0	1.0	11.0	1.0	5.0	159.93	1
3.0	2	7.0	1.0	15.0	0.0	0.0	120.90	1
3.0	2	6.0	1.0	14.0	0.0	3.0	NaN	1
5.0	2	8.0	0.0	23.0	0.0	3.0	134.07	1
5.0	2	3.0	0.0	11.0	1.0	3.0	129.60	1

Figure No. 35

Figure No. 35 represents the head of the dataset after encoding

```
data.describe()
```

	Churn	Tenure	City_Tier	CC_Contacted_LY	Payment	Gender	Service_Score	Account_user_count	account_segment	CC
count	11260.000000	11260.000000	11260.000000	11260.000000	11260.000000	11260.000000	11260.000000	11260.000000	11260.000000	
mean	0.168384	10.811634	1.637478	17.705240	1.791208	0.614565	2.877265	3.547247	2.194583	
std	0.374223	12.844640	0.925130	8.974194	1.056286	0.506044	0.771090	1.233610	1.148711	
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	
25%	0.000000	2.000000	1.000000	11.000000	1.000000	0.000000	2.000000	3.000000	2.000000	
50%	0.000000	8.000000	1.000000	16.000000	2.000000	1.000000	3.000000	4.000000	2.000000	
75%	0.000000	16.000000	3.000000	23.000000	2.000000	1.000000	3.000000	4.000000	3.000000	
max	1.000000	99.000000	3.000000	132.000000	5.000000	2.000000	5.000000	6.000000	5.000000	

Figure No. 36

```
data.describe()
```

_Agent_Score	Marital_Status	rev_per_month	Complain_ly	rev_growth_yoy	coupon_used_for_payment	Day_Since_CC_connect	cashback	Login_device
11260.000000	11260.000000	11260.000000	11260.000000	11260.000000	11260.000000	11260.000000	11260.000000	11260.000000
3.034902	1.202131	5.915631	0.276288	16.189076	1.790142	4.485879	187.993048	0.799467
1.407139	0.703736	11.598273	0.447181	3.766505	1.969505	3.728088	179.244067	0.543449
0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
2.000000	1.000000	3.000000	0.000000	13.000000	1.000000	2.000000	145.080000	0.000000
3.000000	1.000000	4.000000	0.000000	15.000000	1.000000	3.000000	163.170000	1.000000
4.000000	2.000000	7.000000	1.000000	19.000000	2.000000	7.000000	197.310000	1.000000
5.000000	3.000000	140.000000	1.000000	28.000000	16.000000	47.000000	1997.000000	2.000000

Figure No. 37

Figure No. 36 and Figure No. 37 shows the summary statistics of the data.

4 Business insights from EDA

4.1 Is the data unbalanced? If so, what can be done? Please explain in the context of the business

After looking at the describe function from Figure No. 36 and Figure No. 37, we can say that

- Data is unbalanced.
- Scaling is necessary
- It helps handle disparities in units
- During long processes, it definitely helps reduce computational expenses
- It helps improve the performance of the model and reduces the values/models from varying widely

We can apply the MinMaxScaler to the dataset directly to normalize the input variables.

```
data.head(5)
```

	Churn	Tenure	City_Tier	CC_Contacted_LY	Payment	Gender	Service_Score	Account_user_count	account_segment	CC_Agent_Score	Marital_Status
0	1	0.108108	1.000000	0.146341	2	0	0.625	0.375	3	0.4	2
1	1	0.000000	0.333333	0.195122	4	1	0.625	0.625	2	0.6	2
2	1	0.000000	0.333333	0.731707	2	1	0.375	0.625	2	0.6	2
3	1	0.000000	1.000000	0.365854	2	1	0.375	0.625	3	1.0	2
4	1	0.000000	0.333333	0.292683	1	1	0.375	0.375	2	1.0	2

Figure No. 38

```
data.head(5)
```

CC_Agent_Score	Marital_Status	rev_per_month	Complain_ly	rev_growth_yoy	coupon_used_for_payment	Day_Since_CC_connect	cashback	Login_device
0.4	2	0.692308	1.0	0.291667	0.285714	0.344828	0.446080	1
0.6	2	0.538462	1.0	0.458333	0.000000	0.000000	0.259262	1
0.6	2	0.461538	1.0	0.416667	0.000000	0.206897	0.000000	1
1.0	2	0.615385	0.0	0.791667	0.000000	0.206897	0.322300	1
1.0	2	0.230769	0.0	0.291667	0.285714	0.206897	0.300905	1

Figure No. 39

Figure No. 38 and Figure No. 39: We can see that the distributions have been adjusted and that the minimum and maximum values for each variable are now a crisp 0.0 and 1.0 respectively.

4.2 Any business insights using clustering (if applicable)

Clustering is done using kmeans clustering. Within the cluster sum of squares reduces as K keeps increasing. `Wss` for `k value` 1 to 11 are shown below in Figure No. 40

```
wss
```

```
[50683.96596321364,
 40244.16049492927,
 33770.95210403609,
 30699.886684956935,
 28793.341489503793,
 27400.185304021506,
 25985.47087757477,
 25358.51447364453,
 24530.35855896802,
 23746.803848831267]
```

Figure No. 40

From Figure No. 41, we can say that after 3 clusters there isn't much difference. To evaluate clusters, silhouette score is used here.

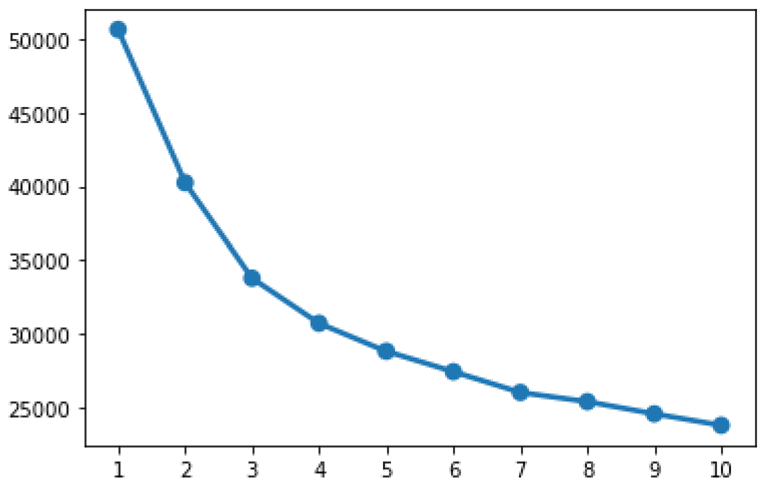


Figure No. 41

Silhouette score is better for 3 clusters than for 4 clusters as shown in Figure No. 42. So, the final clusters will be 3.

```python
from sklearn.metrics import silhouette_samples, silhouette_score
```

```python
# Calculating silhouette_score
silhouette_score(data,labels,random_state=1)
```

```
0.22540715127430286
```

```python
# KMeans with K=4
```

```python
k_means = KMeans(n_clusters = 4,random_state=1)
k_means.fit(data)
labels = k_means.labels_
```

```python
### Cluster evaluation for 4 clusters
```

```python
silhouette_score(data,labels,random_state=1)
```

```
0.15297562970088974
```

Figure No. 42

After appending Clusters to the original dataset, the dataset looks like as below in Figure No. 43 and Figure No. 44

```python
data["Clus_kmeans3"] = labels
data.head()
```

	Churn	Tenure	City_Tier	CC_Contacted_LY	Payment	Gender	Service_Score	Account_user_count	account_segment	CC_Agent_Score	Marital_Status
0	1	0.108108	1.000000	0.146341	2	0	0.625	0.375	3	0.4	2
1	1	0.000000	0.333333	0.195122	4	1	0.625	0.625	2	0.6	2
2	1	0.000000	0.333333	0.731707	2	1	0.375	0.625	2	0.6	2
3	1	0.000000	1.000000	0.365854	2	1	0.375	0.625	3	1.0	2
4	1	0.000000	0.333333	0.292683	1	1	0.375	0.375	2	1.0	2

Figure No. 43

```python
data["Clus_kmeans3"] = labels
data.head()
```

e	Marital_Status	rev_per_month	Complain_ly	rev_growth_yoy	coupon_used_for_payment	Day_Since_CC_connect	cashback	Login_device	Clus_kmeans3
4	2	0.692308	1.0	0.291667	0.285714	0.344828	0.446080	1	1
6	2	0.538462	1.0	0.458333	0.000000	0.000000	0.259262	1	0
6	2	0.461538	1.0	0.416667	0.000000	0.206897	0.000000	1	3
0	2	0.615385	0.0	0.791667	0.000000	0.206897	0.322300	1	1
0	2	0.230769	0.0	0.291667	0.285714	0.206897	0.300905	1	3

Figure No. 44

Cluster frequency for K Means Cluster:

```
data.Clus_kmeans3.value_counts().sort_index()
```

```
0    1750
1    3820
2    2054
3    3636
Name: Clus_kmeans3, dtype: int64
```

Figure No. 45

Figure No. 45 shows
- Churn rate is high for cluster3 and low for cluster 1
- Churn rate is high for people with low *city_tier* and *Complain_ly* and high *Satisfaction score* given by customers of the account on service and customer care service provided by company, revenue growth percentage of the account.

4.3 Any other business insights

- The number of customers churned is less than customers not churned
- The churn rate is high for people with mobiles, who are married and with gender male, and who use debit cards and ewallet with regular spending