CAPSTONE PROJECT

FINAL REPORT

NAME: **ANISH DASGUPTA**

COURSE: **DSBA**

BATCH: **May’21**

SUBJECT**: E-Com Customer Churn**

TABLE OF CONTENTS:

DATA DICTIONARY:

Term	Description
AccountID	Account unique identifier
Churn	Account churn flag (Target variable)
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	Number of complaints 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 customers in the account have 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

OBJECTIVE:

The primary objective of this report is to provide the details of the predictive model built to determine the business implications of the presented problem statement. The insights provided in this report primarily analyze the problem under hand and attempts to provide an answer to the business problem by finding the customers who might churn from the service. The codes for deriving those insights are maintained separately.

Problem Statement:

An E Commerce company/DTH 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.

A churn prediction model needs to be developed for this company and provide business recommendations on the campaign. Campaign suggestion should be unique and clear. Since the campaign offer will go through the revenue assurance team review, if it is found that a lot of free (or subsidized) offers are being given way thereby making a loss to the company, the campaign might be disapproved.

Understanding business opportunity:

In today's world, with the advancement of new technology, it has become very convenient for the customers to shop for goods as well as entertainment. Similarly it has become very easy for the companies to cater to the customers' needs through their service portals and products. However, with convenience comes the problem of high market competition, where the competitors try to disrupt the system by their unique features/products thus pulling the customers away.

With the use of data analytics however, we can target the customers who show the signs of churning by studying their behavioral data. This provides us with the opportunity to protect/safeguard the company's interests which lie with the customers. In addition, we also have the opportunity to understand the reasons behind the customer churning, which will help us to improve the service through marketing campaigns and therefore expand the customer base further thus increasing company growth and sales.

Understanding social opportunity:

Entertainment binds every part of the world. For developing nations such as India Pakistan, the rural villages have had penetration of internet faster than books, media and other conventional methods of communication. We can use this project to aim for education of the masses in a faster and more efficient way.

Also, if competitors belong from a different country, it poses a social security risk for the country as lot of cases of data privacy breaches have occurred in our country where foreign companies have allegedly sold the data of our people to other institutions which may cause national level social and political threats.

Data Report:

Overview of Data:

The data has been provided in the form of an Excel sheet which contains the data as well as the meta-data (data about data). We see that the data has 11260 rows and 19 columns.

We can observe that the 'Churn' field is the target variable. On observing the data types of the columns, we can see that 14 fields are int/float type and 5 fields are object type. Out of the 18 predictor variables, we have only 17 useful variables since 'AccountID' is a unique identifier hence does not pose as a strong predictor.

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

Tbl1: Visual of Data:

We see that the columns provided in the data tells us the about each customer on their sales generating behavior as well as the attributes about the customer and his/her uses. The data presented is in the form monthly as well as year-on-year data.

```
 #   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
```

Tbl2: Info of data

We can find that some Integer/float fields are depicted as object (eg.Tenure,Account_user_count,rev_per_month,rev_growth_yoy,coupon_used_for_payment,Day_Since_CC_connect,cashback).

Description of Data:

	count	unique	top	freq	mean	std	min	25%	50%	75%	max
AccountID	11260	NaN	NaN	NaN	25629.5	3250.63	20000	22814.8	25629.5	28444.2	31259
Churn	11260	NaN	NaN	NaN	0.168384	0.374223	0	0	0	0	1
Tenure	11158	38	1	1351	NaN	NaN	NaN	NaN	NaN	NaN	NaN
City_Tier	11148	NaN	NaN	NaN	1.65393	0.915015	1	1	1	3	3
CC_Contacted_LY	11158	NaN	NaN	NaN	17.8671	8.85327	4	11	16	23	132
Payment	11151	5	Debit Card	4587	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Gender	11152	4	Male	6328	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Service_Score	11162	NaN	NaN	NaN	2.90253	0.725584	0	2	3	3	5
Account_user_count	11148	7	4	4569	NaN	NaN	NaN	NaN	NaN	NaN	NaN
account_segment	11163	7	Super	4062	NaN	NaN	NaN	NaN	NaN	NaN	NaN
CC_Agent_Score	11144	NaN	NaN	NaN	3.06649	1.37977	1	2	3	4	5
Marital_Status	11048	3	Married	5860	NaN	NaN	NaN	NaN	NaN	NaN	NaN
rev_per_month	11158	59	3	1746	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Complain_ly	10903	NaN	NaN	NaN	0.285334	0.451594	0	0	0	1	1
rev_growth_yoy	11260	20	14	1524	NaN	NaN	NaN	NaN	NaN	NaN	NaN
coupon_used_for_payment	11260	20	1	4373	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Day_Since_CC_connect	10903	24	3	1816	NaN	NaN	NaN	NaN	NaN	NaN	NaN
cashback	10789	5693	155.62	10	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Login_device	11039	3	Mobile	7482	NaN	NaN	NaN	NaN	NaN	NaN	NaN

Tbl3: Describing data

From the above table, we see that lot of descriptive statistic like mean, percentile are not provided for many Int/float fields, since they contain some special characters which miscast the field as string or object type. Hence this calls for a data cleanup.

Exploratory Data Analysis:

Data Cleanup:

```
array([4, 0, 2, 13, 11, '#', 9, 99, 19, 20, 14, 8, 26, 18, 5, 30, 7, 1,
       23, 3, 29, 6, 28, 24, 25, 16, 10, 15, 22, nan, 27, 12, 21, 17, 50,
       60, 31, 51, 61], dtype=object)
```

Fig1: Tenure data unique values

The above figure shows the unique values of the tenure data given to us. We observe that there is presence of special characters because of which this Int64 data type field is miscast as Object data type. Hence for all the fields which are miscast, data cleanup is done so as to showcase the correct data type.

```
array([ 4.,  0.,  2., 13., 11., nan,  9., 99., 19., 20., 14.,  8., 26.,
       18.,  5., 30.,  7.,  1., 23.,  3., 29.,  6., 28., 24., 25., 16.,
       10., 15., 22., 27., 12., 21., 17., 50., 60., 31., 51., 61.])
```

Fig2: Tenure data after cleanup

In the above figure we see that the data cleanup is performed in the ‘Tenure’ data, removing the special characters.

Similar measures are performed for the rest of the miscast fields.

	count	unique	top	freq	mean	std	min	25%	50%	75%	max
AccountID	11260	NaN	NaN	NaN	25629.5	3250.63	20000	22814.8	25629.5	28444.2	31259
Churn	11260	NaN	NaN	NaN	0.168384	0.374223	0	0	0	0	1
Tenure	11042	NaN	NaN	NaN	11.0251	12.8798	0	2	9	16	99
City_Tier	11148	NaN	NaN	NaN	1.65393	0.915015	1	1	1	3	3
CC_Contacted_LY	11158	NaN	NaN	NaN	17.8671	8.85327	4	11	16	23	132
Payment	11151	5	Debit Card	4587	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Gender	11152	4	Male	6328	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Service_Score	11162	NaN	NaN	NaN	2.90253	0.725584	0	2	3	3	5
Account_user_count	10816	NaN	NaN	NaN	3.69286	1.02298	1	3	4	4	6
account_segment	11163	7	Super	4062	NaN	NaN	NaN	NaN	NaN	NaN	NaN
CC_Agent_Score	11144	NaN	NaN	NaN	3.06649	1.37977	1	2	3	4	5
Marital_Status	11048	3	Married	5860	NaN	NaN	NaN	NaN	NaN	NaN	NaN
rev_per_month	10469	NaN	NaN	NaN	6.36259	11.9097	1	3	5	7	140
Complain_ly	10903	NaN	NaN	NaN	0.285334	0.451594	0	0	0	1	1
rev_growth_yoy	11257	NaN	NaN	NaN	16.1934	3.75772	4	13	15	19	28
coupon_used_for_payment	11257	NaN	NaN	NaN	1.79062	1.96955	0	1	1	2	16
Day_Since_CC_connect	10902	NaN	NaN	NaN	4.63319	3.69764	0	2	3	8	47
cashback	10787	NaN	NaN	NaN	196.236	178.661	0	147.21	165.25	200.01	1997
Login_device	11039	3	Mobile	7482	NaN	NaN	NaN	NaN	NaN	NaN	NaN

Tbl4: Data description after cleanup

After the cleanup action on the integer/float fields, we see that the data type of the fields are correctly cast and the descriptive details also showcase the statistical properties of the fields.

```
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
```

Tbl5: Data types after cleanup

Similarly, the similar measures are repeated for the categorical fields as well where there might be possibilities that a category is represented in different ways as shown in the below figure.

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

Fig3: Data cleanup of categorical variable ‘account_segment’

Missing Value Treatment:

We can observe that our data has null values, which need to be treated since the predictive models cannot perform efficiently with null data.

AccountID	0
Churn	0
Tenure	218
City_Tier	112
CC_Contacted_LY	102
Payment	109
Gender	108
Service_Score	98
Account_user_count	444
account_segment	97
CC_Agent_Score	116
Marital_Status	212
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	221
dtype: int64	

Tbl6: Total null values per column

We can find that most of the columns have null values but the number of null values is very less compared to the total number of rows(<10%). Hence we do not need to drop any column, however we can impute them.

As a mode of imputation, ideally median is used for the numerical variables whereas mode (most frequently used) is used for the categorical variables. The reason why mean is not used for numerical variables is because of the possible presence of outliers in the fields which affects the mean in a negative way and distorts the data.

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

Tbl7: Total null values per column after missing value treatment

After treatment of the missing values, we check whether the data contains any duplicate records or not. The data is free from duplicate values.

Removal of Unwanted Variables:

Since now we have our data preprocessed and cleaned, we move on to removing the unwanted columns.

As a part of unwanted variable, we see that the field ‘AccountID’ acts as a very poor predictor variable since each value is distinct for the field which does not help us understand any pattern in the data. Hence we drop the field from the table.

	Churn	Tenure	City_Tier	CC_Contacted_LY	Service_Score	Account_user_count	CC_Agent_Score	rev_per_month	Complain_ly	rev_growth_yoy	coupon_us
0	1	4.0	3.0	6.0	3.0	3.0	2.0	9.0	1.0	11.0	
1	1	0.0	1.0	8.0	3.0	4.0	3.0	7.0	1.0	15.0	
2	1	0.0	1.0	30.0	2.0	4.0	3.0	6.0	1.0	14.0	
3	1	0.0	3.0	15.0	2.0	4.0	5.0	8.0	0.0	23.0	
4	1	0.0	1.0	12.0	2.0	3.0	5.0	3.0	0.0	11.0	

Tbl8: Final dataset after preprocessing

Univariate Analysis:

We move over to plotting the box-plot for outlier checks and distribution curve for data visualization for each field to understand how the data looks. The name of the variable has been provided below each plot.

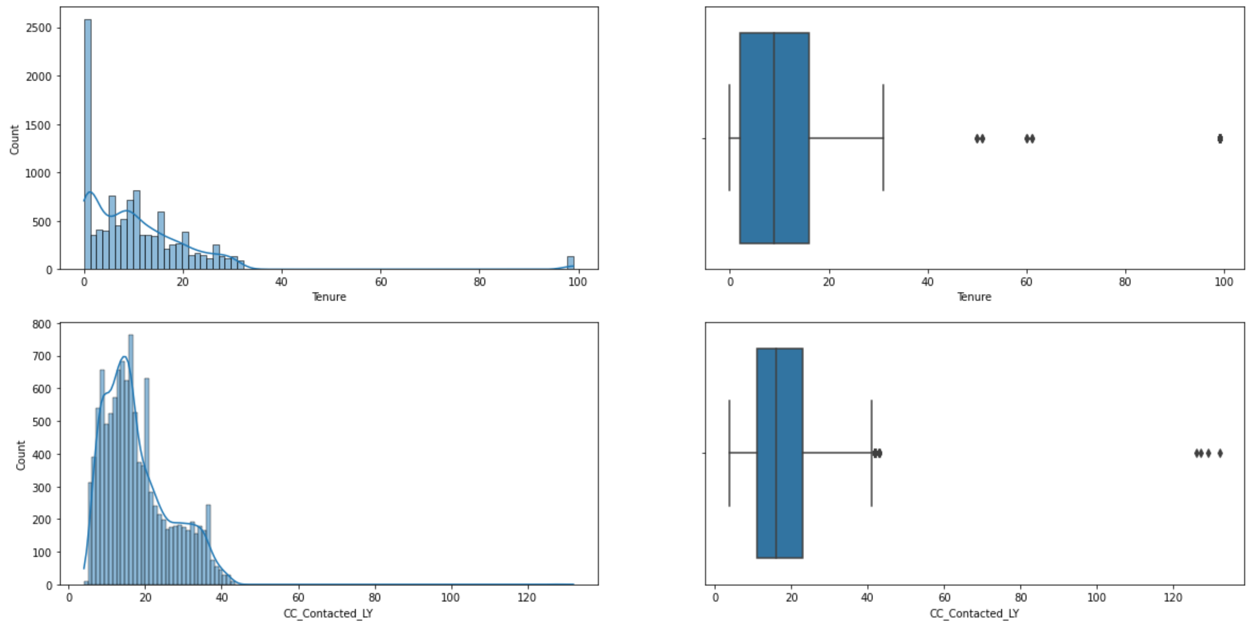


Fig4: Tenure,CC_Contacted_LY curve

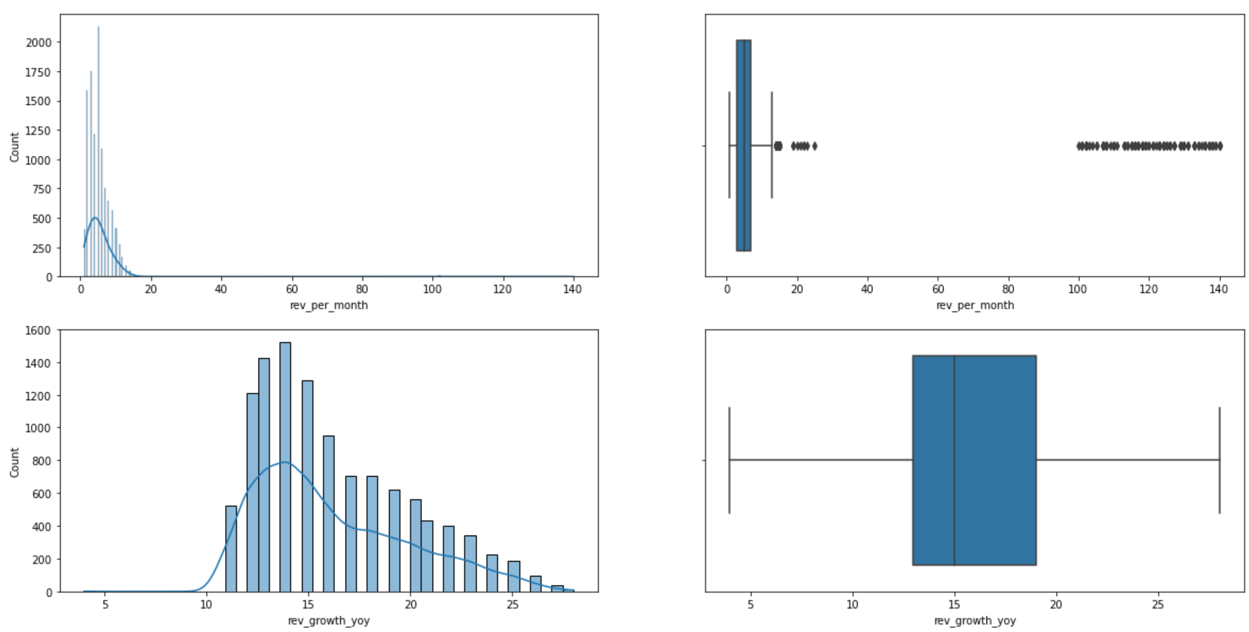


Fig5: rev_per_month,rev_growth_yoy curve

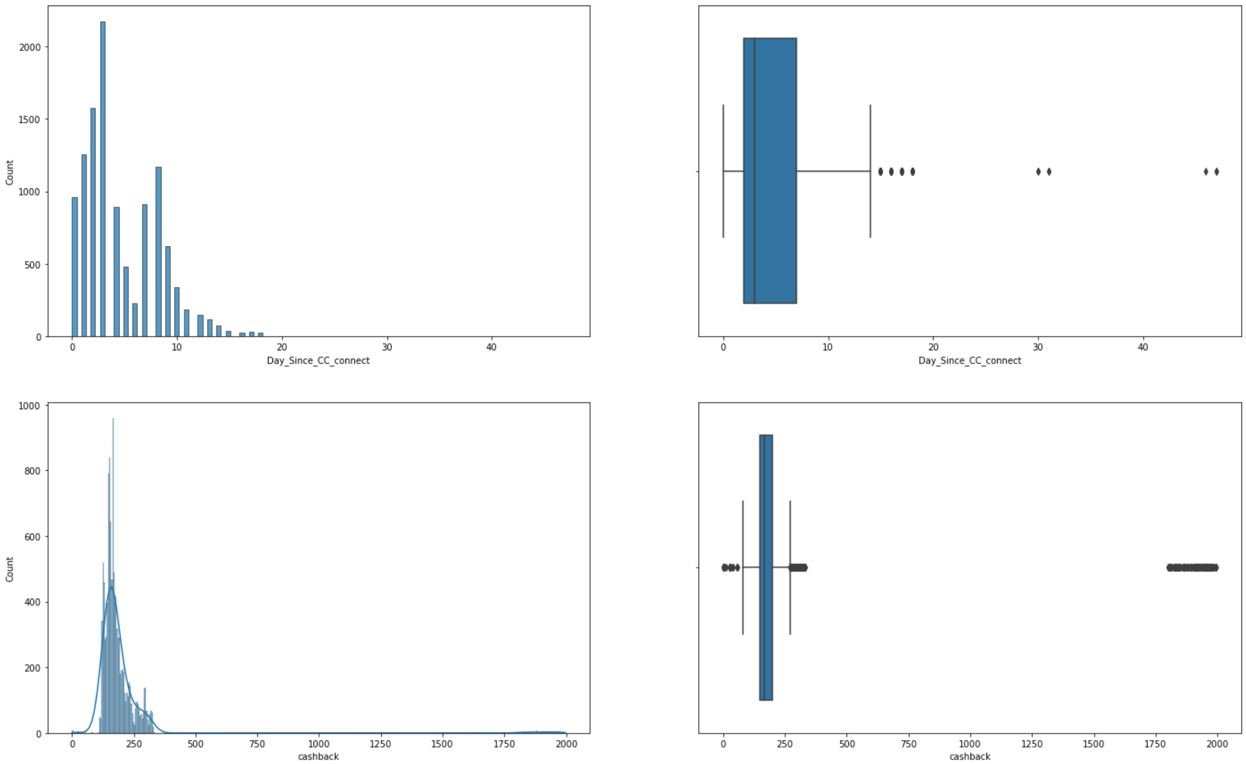


Fig6: Day_Since_cc_connect,cashback curve

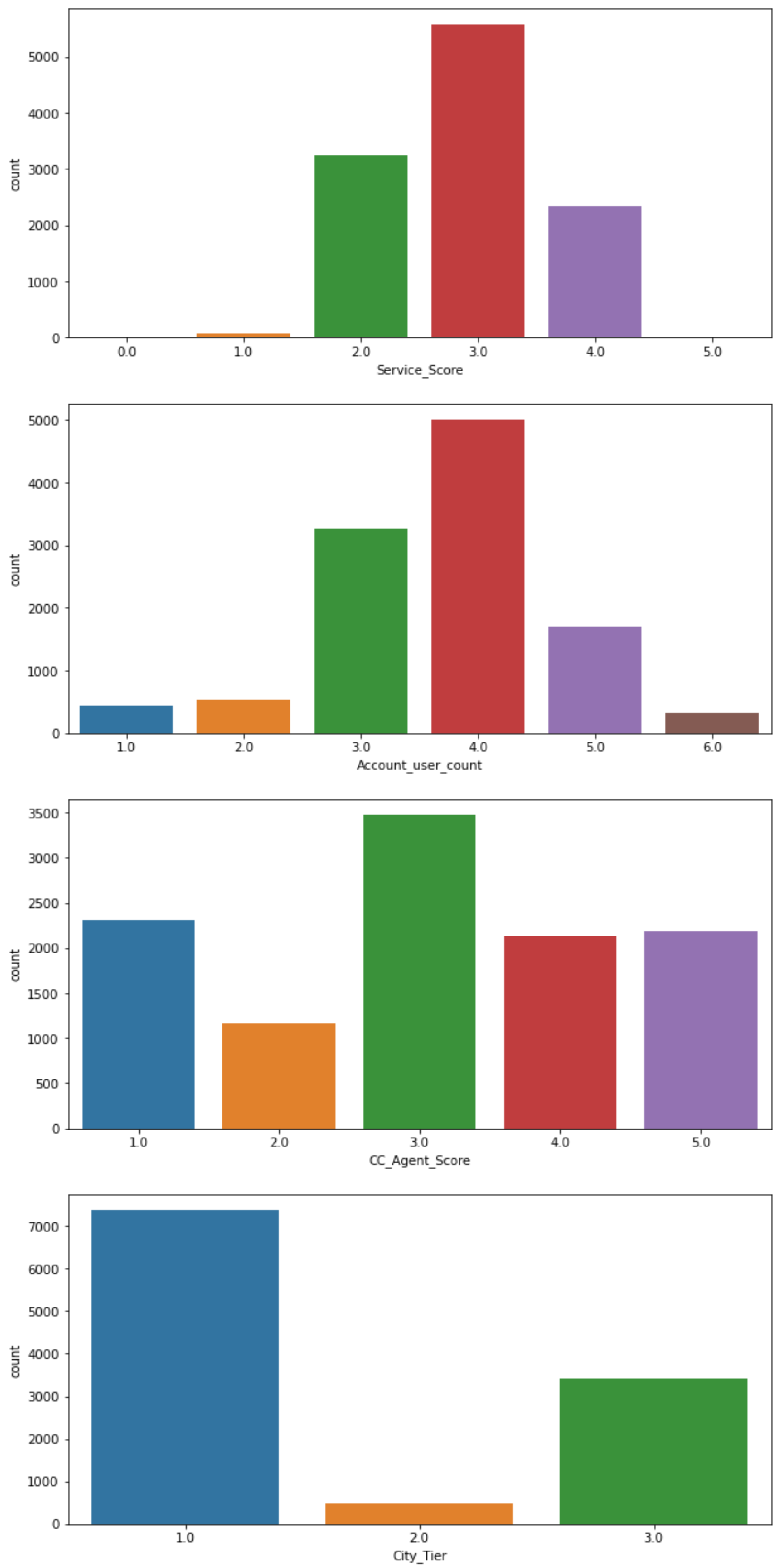
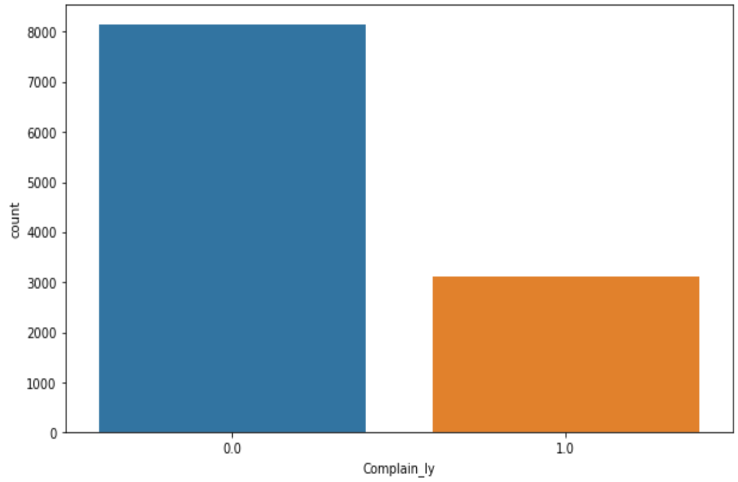
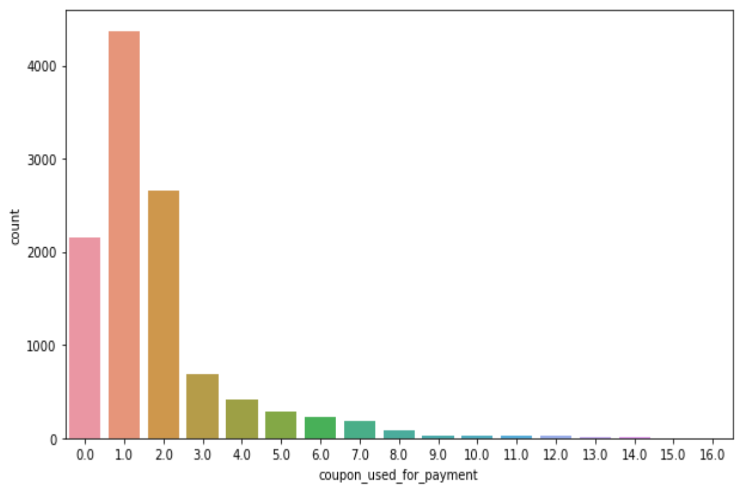


Fig7: Service_score,Account_user_count,CC_agent_score,City_tier curve





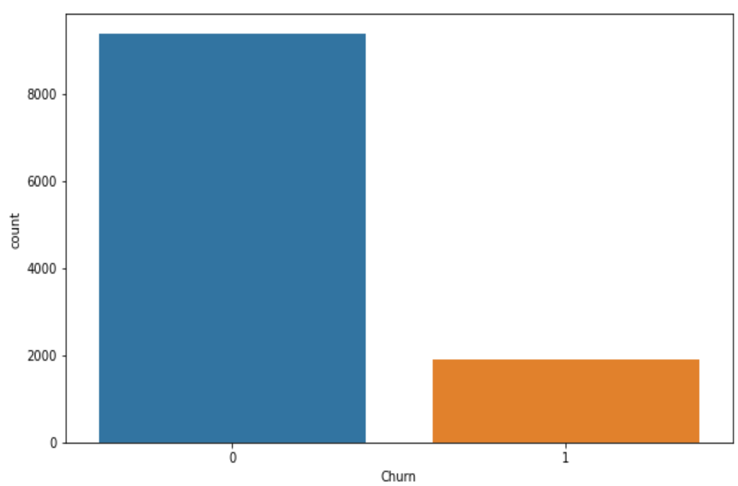


Fig8:Complain_ly, coupon_used_for_payment, Churn curve

Insights from Univariate Analysis:

1. From the distribution curves and the box-plots for ‘Tenure’,’CC_Contacted_LY’,’rev_per_monthl’,’rev_growth_yoy’,’Day_since_cc_connect’ and ’cashback’, we see that all of the mentioned are right tailed. This means that the above mentioned columns having heavy outliers to their right side i.e. majority outliers are present outside the upper range.

2. The box-plot for ‘cashback’ shows that this is the only field which has an outlier below the lower range.

3. The ‘rev_per_month’ and ‘cashback’ have very high density outliers whereas the rest fields either have minimal or no outliers.

4. From the count plot of ‘Account_user_count’, we understand that majority of the accounts have multiple linked using that particular account. The number of secondary users is maximum for 4 users to an account.

5. From the ‘Service score’, we see that most of the customers score intermediate to excellent scores. Very few counts are there which give very low rating, hence meaning that the service is decent.

6. From the plot of ‘CC_Agent_Score’, we see that the low/bad scores that agents provide for the customers seem unusually high.

7. The majority of the users belong from Tier 1 cities. The tier 2 cities have the lowest number of users.

8. The number of users who put up a complaint last year is almost 40% that of the customers who did not have any complaints.

9. The majority number of customers who use coupons for payment use 1-4 coupons in a given time period. Very few users use coupons>10.

10. Majority number of customers connect with the Customer care once in a period of 4-10 days. Very few customers are present who haven't contacted with the customer care for over 20 days.

Multivariate Analysis:

We have used a pair-plot and a heat map to understand the correlation of different attributes with each other.

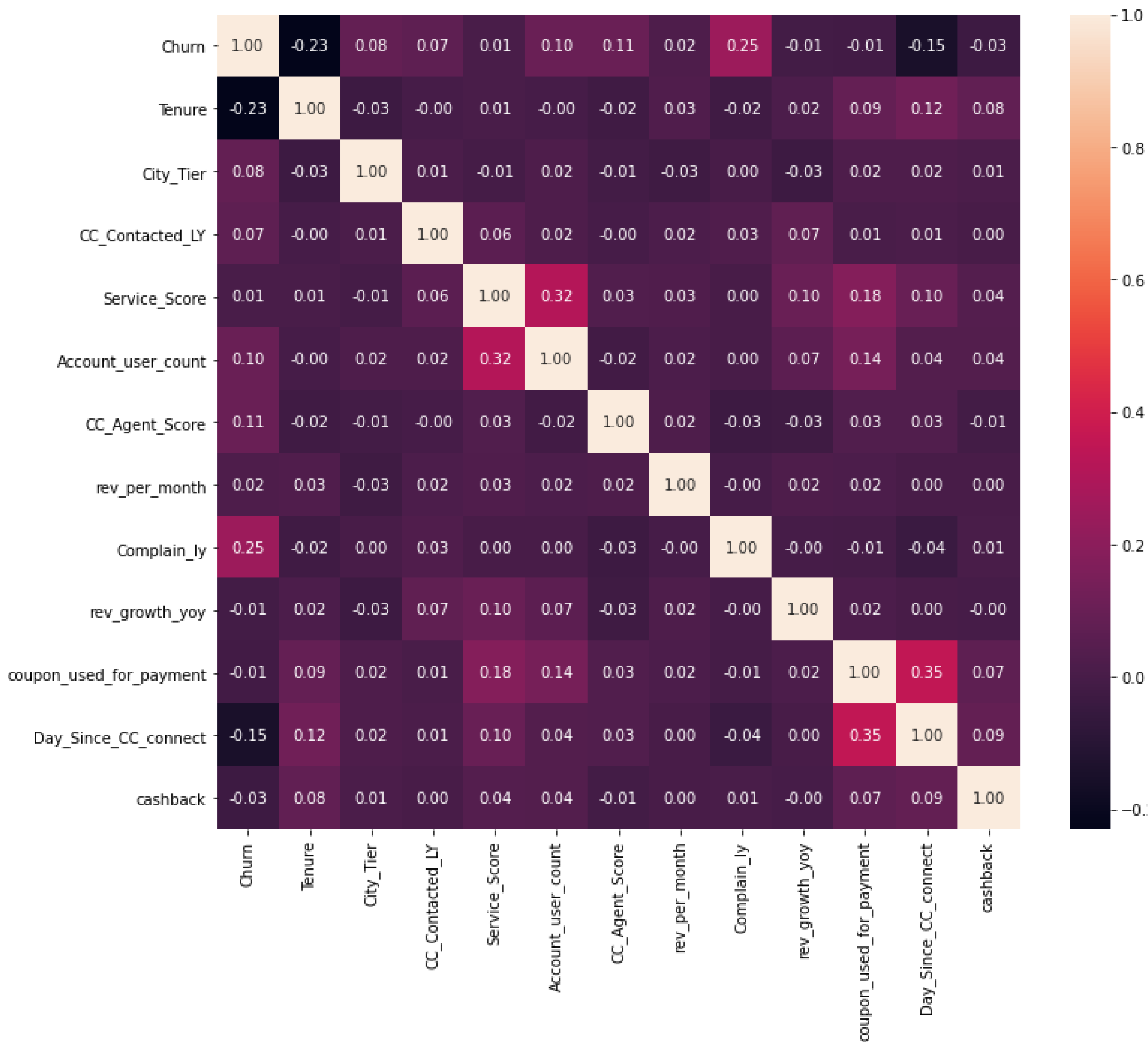


Fig9: Heatmap of data

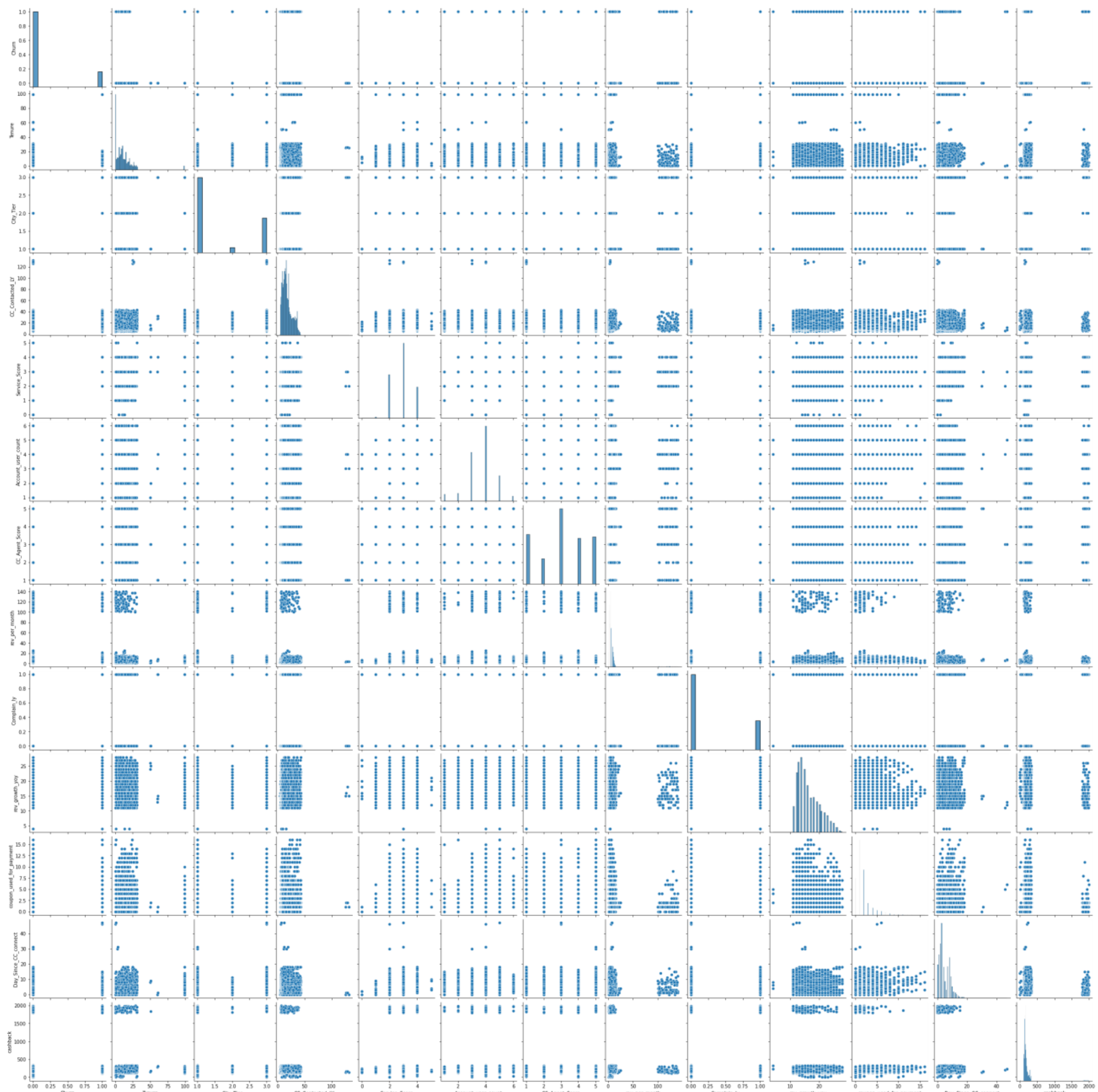


Fig10: Pair-plot of data

Insights from Multivariate Analysis:

1. We see that the correlation among the features is very low, the highest being 0.35 between the last connect with customer and coupons used for payment.

2. We see that there is a negative correlation of account tenure to the churn.

3. Account user count has high correlation with the service score meaning as the number of secondary customers increase, the service score also increases.

4. We see moderate correlation of the Customers who complained last year to the customers who churned.

5. The day since last customer care connect has negative correlation with Churn.

Business Insights from EDA:

1. Data Imbalance:

The number of customers who churn constitute of ~17% whereas the customers who do not churn are about ~83%. We see that the ratio of churners is nominal since no business targets to have the churner ratio high.

From the model building perspective, we would be using the raw data for building one model, as well as the Oversampling/Undersampling techniques to change the ratio slightly and build a model. We will then test the accuracies for both the models and see which performs better.

2. We see that the number of agents who score customers are unusually high for the low/bad scores, hinting about the customer care service system faultiness. It can also mean that we are not targeting the correct customers since they are showcasing bad behavior.

3. We saw that the business does not have many users in the tier 2 cities, hinting that some marketing campaign would have to be designed for the tier 2 cities.

4. We saw that the tenure has a negative correlation to churn meaning that the long time loyal customers remain loyal and the churners belong from the new-average tenure based.

5. The last day customer connect has a negative correlation to churn meaning that the customers are actually happy with the customer care service and not un interested.

6. Customers who complained last year are also churning showing that the issues have not been solved to their satisfaction.

Data Pre-Processing:

Outlier Treatment:

We see that our data has high number of outliers. However since the ratio of outliers to the total number of data is less than 10%, it is a good practice to remove the outliers so that the models do not get biased due to high weights.

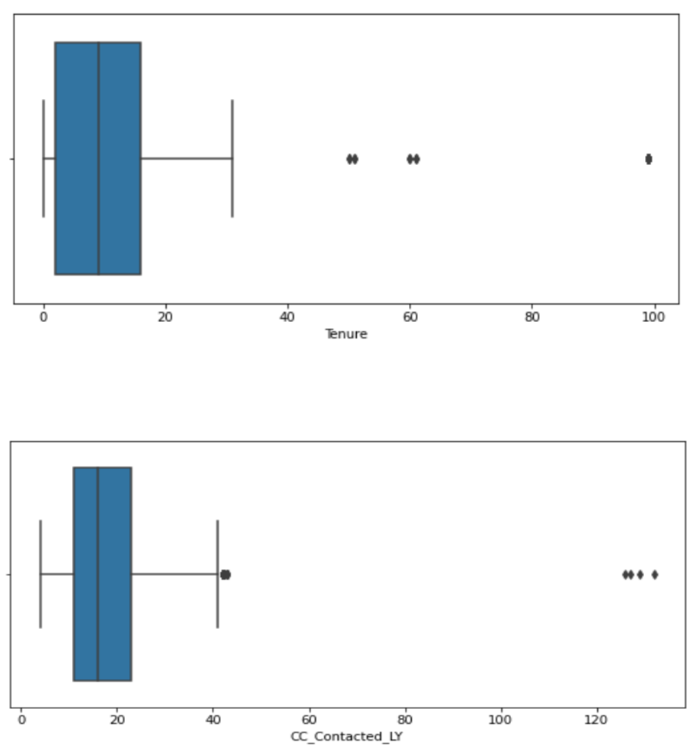
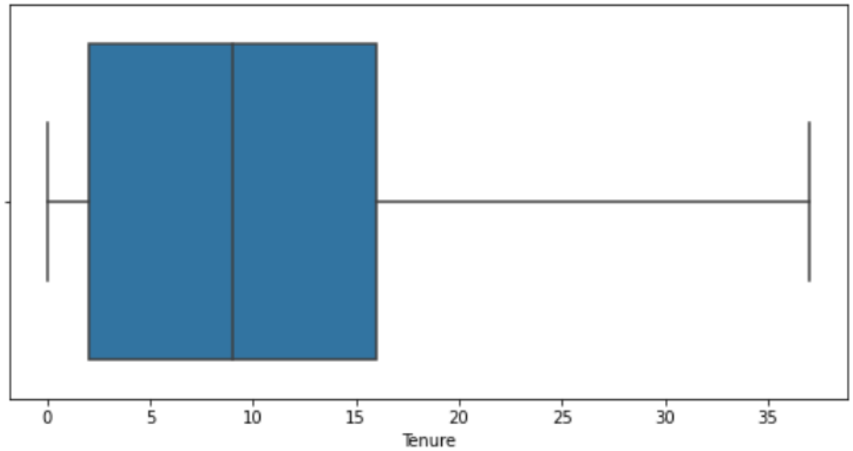


Fig11: Feature box-plot before outlier



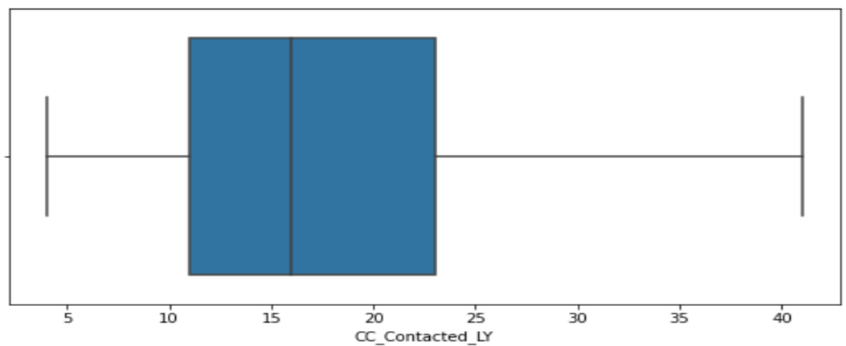


Fig12: Feature box-plot after outlier treatment

From the above figures we see that the outliers have been treated for all the features, making the data model ready for the next processing step.

Encoding Categorical Features (Variable Transformation):

```
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   Churn                   11260 non-null  int64
 1   Tenure                  11260 non-null  float64
 2   City_Tier               11260 non-null  float64
 3   CC_Contacted_LY         11260 non-null  float64
 4   Service_Score           11260 non-null  float64
 5   Account_user_count      11260 non-null  float64
 6   CC_Agent_Score          11260 non-null  float64
 7   rev_per_month           11260 non-null  float64
 8   Complain_ly             11260 non-null  float64
 9   rev_growth_yoy          11260 non-null  float64
 10  coupon_used_for_payment 11260 non-null  float64
 11  Day_Since_CC_connect    11260 non-null  float64
 12  cashback                11260 non-null  float64
 13  Payment                 11260 non-null  object
 14  Gender                  11260 non-null  object
 15  account_segment         11260 non-null  object
 16  Marital_Status          11260 non-null  object
 17  Login_device            11260 non-null  object
```

Tbl9: Data Info before encoding

The above table shows the features and their corresponding data types. We can see that the features;
'City_Tier','Service_Score','CC_Agent_Score','Account_user_count','Complain_ly','Payment', 'Gender','account_segment','Marital_Status','Login_device' are categorical in nature of which 5 are of 'object' data type.
Since Python cannot interpret string values, we need to encode these values into numerics so that Python can consume for model building:

```
['Debit Card' 'UPI' 'Credit Card' 'Cash on Delivery' 'E wallet']
['Female' 'Male']
['Super' 'Regular Plus' 'Regular' 'HNI' 'Super Plus']
['Single' 'Divorced' 'Married']
['Mobile' 'Computer' 'Unknown']
```

Fig13: Object data type Categorical feature values

#	Column	Non-Null Count	Dtype
---	------	--------------	-----
0	Churn	11260 non-null	int64
1	Tenure	11260 non-null	float64
2	City_Tier	11260 non-null	float64
3	CC_Contacted_LY	11260 non-null	float64
4	Service_Score	11260 non-null	float64
5	Account_user_count	11260 non-null	float64
6	CC_Agent_Score	11260 non-null	float64
7	rev_per_month	11260 non-null	float64
8	Complain_ly	11260 non-null	float64
9	rev_growth_yoy	11260 non-null	float64
10	coupon_used_for_payment	11260 non-null	float64
11	Day_Since_CC_connect	11260 non-null	float64
12	cashback	11260 non-null	float64
13	Payment	11260 non-null	int8
14	Gender	11260 non-null	int8
15	account_segment	11260 non-null	int64
16	Marital_Status	11260 non-null	int8
17	Login_device	11260 non-null	int8

Tbl10: Data Info after encoding

From the unique values of the features, we can observe that 'Payment','Gender','Marital_Status' and 'Login_device' are having nominal encoding whereas 'account_segment' is having ordinal encoding.

```
['Super' 'Regular Plus' 'Regular' 'HNI' 'Super Plus']

array([3, 2, 1, 5, 4], dtype=int64)
```

Fig14: 'account_segment' values before and after encoding

After this process, we finally have our final data ready to be used for model building.

<u>Predictive Model Building:</u>

Models to be used:

The problem statement talks about customers who are churning from the company services, which helps us to understand that the model to be used for this case study has to be a binary classification model.

For building the highest optimized predictive model, we have built 4 kinds of different models for classification purpose, namely Logistic Regression (which will work as our base model), Decision tree classifier, Random Forest classifier (which will be our ensemble model) and Artificial Neural Network classifiers.

For understanding the performance of the models, we have used **Accuracy**, **Precision, Recall and f1-score** as our performance metrics.

We have first split our data into train and test set. Keeping the test data untouched, we have trained our models with the train data and in the below table, we can see the different performance metric values.

Train Data metrics:

	Accuracy	Precision	Recall	f1-score
Logistic Regression	0.873382	0.803730	0.691298	0.726742
Decision Tree Classifier	1.000000	1.000000	1.000000	1.000000
Random Forest Classifier	1.000000	1.000000	1.000000	1.000000
ANN Classifier	1.000000	0.415377	0.500000	0.453777

Tbl11: Model Performance: Train Data

Test Data metrics:

	Accuracy	Precision	Recall	f1-score
Logistic Regression	0.883659	0.824565	0.710174	0.747874
Decision Tree Classifier	1.000000	1.000000	1.000000	1.000000
Random Forest Classifier	1.000000	1.000000	1.000000	1.000000
ANN Classifier	1.000000	0.570027	0.564408	0.302065

Tbl12: Model Performance: Test Data

We see that our designed models for Decision Tree, Random Forest and Artificial Neural Network is highly over fit, which would imply that these models, although have idealistic scores in train environment, will fail in the test environment.

We can see that the difference between the train and test metric values are also extremely large pointing to the fact that the models are over fit.

Hence to address this issue, we need to use Hyper tuning methods to reinforce our models for better metric values.

Model hyper tuning:

To fine tune our model we have used **Grid Search Cross-Validation** techniques to reduce the over fit of the model.

To determine the grid search parameter values, we have started off the modeling with Decision Tree. This model generates the tree structure determining the classification power of the model. On observing the tree, we see that the branches are overgrown and suggest that we require pruning techniques for the tree.

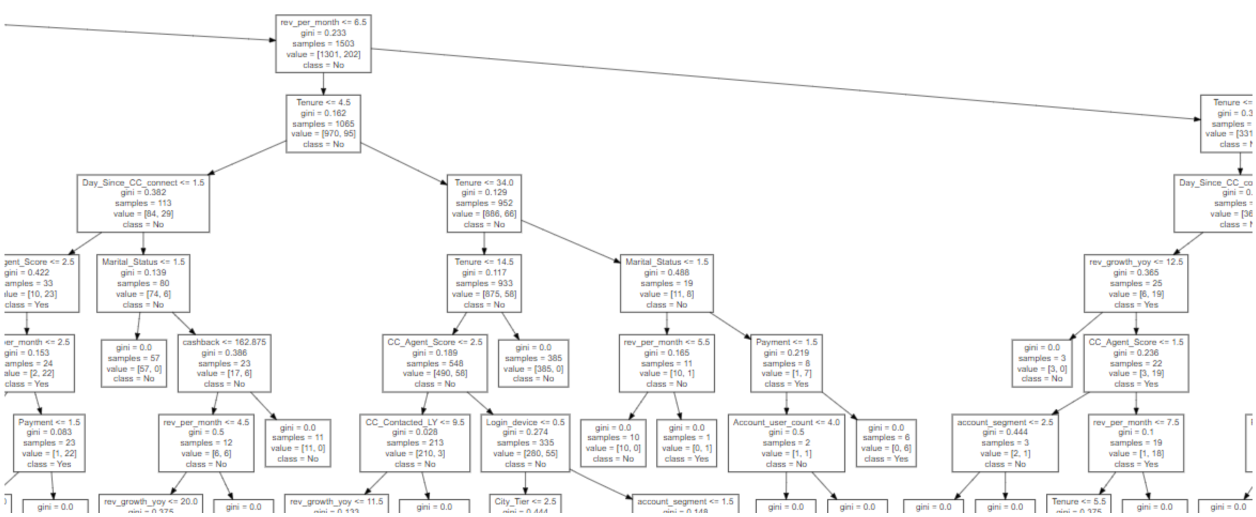


Fig15: Decision Tree

Reducing the max_depth, min_samples_leaf(1-2% of data) and min_samples_split(3 time the min_samples_leaf)size of the leaves, we prune the tree.

On pruning the tree, we test the metrics and observe that the over fit of model has decreased significantly. Based on the values obtained from graphical visualisation, we build our grid search model using values near to the observed values, to find the best parameters for the model.

```
GridSearchCV(cv=3, estimator=DecisionTreeClassifier(random_state=123),
             param_grid={'max_depth': [6, 8, 10, 12],
                         'min_samples_leaf': [50, 100, 200, 225, 300, 350],
                         'min_samples_split': [150, 300, 600, 675, 900, 1250]})
```

Fig16: GridSearchCV: Decision Tree

Similarly, for Random Forest and Artificial Neural Network, we build the grid search models and find the best parameters for fine tuning the existing model.

```
GridSearchCV(cv=3, estimator=RandomForestClassifier(),
             param_grid={'max_depth': [6, 8, 10], 'max_features': [3, 4, 5],
                         'min_samples_leaf': [50, 100, 150, 200],
                         'n_estimators': [200, 300, 500]})
```

Fig17: GridSearchCV: Random Forest

```
GridSearchCV(cv=3, estimator=MLPClassifier(max_iter=2000, random_state=123),
             param_grid={'activation': ['tanh', 'relu'],
                         'hidden_layer_sizes': [100, 500, (100, 100)],
                         'solver': ['sgd', 'adam']})
```

Fig18: GridSearchCV: ANN

After obtaining the best parameters for fine tuning our respective models, we rebuild the model again with the new optimised parameters and check if the tuning helps us obtain better metrics.

Train Data Metrics:

	Accuracy	Precision	Recall	f1-score
Logistic Regression	0.873382	0.803730	0.691298	0.726742
Decision Tree Classifier	0.895712	0.840576	0.758161	0.790021
Random Forest Classifier	0.902182	0.865449	0.759070	0.797812
ANN Classifier	0.900025	0.880559	0.737478	0.783546

Tbl4: Model Performance: Train GridSearchCV model

Test Data Metrics:

	Accuracy	Precision	Recall	f1-score
Logistic Regression	0.883659	0.824565	0.710174	0.747874
Decision Tree Classifier	0.887803	0.803524	0.774615	0.787794
Random Forest Classifier	0.886619	0.864258	0.689874	0.736141
ANN Classifier	0.875962	0.861684	0.650012	0.692576

Tbl14: Model Performance: Test GridSearchCV model

From the above comparison table, we find that when it comes to accuracy, precision,recall and f1-score, **Random Forest Classifier** performs the best and showcases the highest values on most of the metrics. Hence we finalise Random Forest for our final modelling.

SMOTE Data Modeling:

We see that there is a minor class imbalance in the target variable for the two classes. Hence we build the model on the SMOTE data and see if the re-sampling technique helps us increases the precision metrics for the models.

```
0    0.831616
1    0.168384
Name: Churn, dtype: float64
```

Tbl15: Class Imbalance

```
1    6548
0    6548
Name: Churn, dtype: int64
```

Tbl16: Class Imbalance treated using SMOTE

Train Data metrics:

	Accuracy	Precision	Recall	f1-score
Logistic Regression	0.801542	0.802181	0.801542	0.801438
Decision Tree Classifier	0.873167	0.873499	0.873167	0.873139
Random Forest Classifier	0.909896	0.910343	0.909896	0.909872
ANN Classifier	0.934331	0.934428	0.934331	0.934327

Tbl17: Model Performance: Train SMOTE Modeling

Test Data metrics:

	Accuracy	Precision	Recall	f1-score
Logistic Regression	0.883659	0.824565	0.710174	0.747874
Decision Tree Classifier	0.887803	0.803524	0.774615	0.787794
Random Forest Classifier	0.887211	0.864018	0.692365	0.738606
ANN Classifier	0.875962	0.861684	0.650012	0.692576

Tbl18: Model Performance: Test SMOTE Modeling

We see that for the SMOTE data, we have good metric values for the models. However, when we check the metrics of the models for test set, we see that the models **underperform** for test data. Hence SMOTE modelling causes slight overfit. Hence we finalise our model with Hyper tuned model and drop the SMOTE modelling.

Model Validation:

We finalize our model as the **Random Forest Classifier**. We see that the Decision Tree Classifier model is very close to Random Forest, also showing better consistency between the train and the test data metrics.

However, we have taken the metric importance in the order of Precision, Recall, f1-score and Accuracy. Random Forest, being an **ensemble** technique performs slightly better than Decision Tree when it comes to numbers, hence being the final model of our interest.

Probability of Customer Churn:

in_ly	rev_growth_yoy	coupon_used_for_payment	Day_Since_CC_connect	cashback	Payment	Gender	account_segment	Marital_Status	Login_device	Prob of Churn
1.0	11.0	1.0	5.0	159.93	2	0	3	2	1	0.248073
1.0	15.0	0.0	0.0	120.90	4	1	2	2	1	0.803917
1.0	14.0	0.0	3.0	165.25	2	1	2	2	1	0.694853
0.0	23.0	0.0	3.0	134.07	2	1	3	2	1	0.571379
0.0	11.0	1.0	3.0	129.60	1	1	2	2	1	0.395475

Tbl19: Customer Churn Probability

We finalise Random Forest for our modelling since it has the highest train-test score among the rest three and least difference between train and test rmse. Using the random forest, we predict the probabilities of each customer churning.

Using these probabilities, we can determine which customers would churn. Given their attributes, we can target our campaign according to the customer classes. We take the probability data and append it to the excel data, mapping every probability to the customer.

Day_Since_CC_connect	cashback	Payment	Gender	account_segment	Marital_Status	Login_device	Prob of Churn
5	159.93	2	0	3	2	1	0.248073361
0	120.9	4	1	2	2	1	0.803916628
3	165.25	2	1	2	2	1	0.694852592
3	134.07	2	1	3	2	1	0.571378808
3	129.6	1	1	2	2	1	0.395474852
7	139.19	2	0	2	2	0	0.729841709

Tbl20: Churn Probability Excel data

Model Interpretation:

We can sort the data in descending order in the excel sheet. This is going to give us the highest probability churn customers.

To study the attributes for these customers and find the common grounds for them, we use the ‘feature_importance_’ function of the random forest model. From this function, we understand that the following features are the most influential while classifiying the model:

The 'Tenure','Complain_ly',’Day_Since_CC_connect’ , 'rev_per_month','cashback','account_segment',’Marital_Status’ have the **highest relevances**.

We see that the fields 'Login_device','Gender','coupon_used_for_payment','Service score' have the **least importances** in the modelling.

Business Insights from Modelling:

1. To gain useful business insight, we target the customers who have the highest probability of churning, and also who have the tendency to generate the highest revenue/profits/sales.

2. We observe that the customers having the lowest tenure (**less than 1 year**) are having the highest churn probability and vice versa.

3. Customers who complained last year about the service have also extremely high churning rate **(>75%).**

4. The frequency of high churning customers having contacted the customer care very recently is also very high.

5. The top churners also show a unique characteristic of being a ‘Regular Plus’ account segment holders.

6. The customers belonging to the ‘Singles’ category of marital status also show high tendency to churn.

7. We saw that the business does not have many users in the tier 2 cities.

Campaign Design:

Mentioned below are few of the campaign agendas which can be addressed to decrease customer churn rate.

1. Since low tenure customers are churning, the company can increase the frequency of marketing advertisements via telesales/SMS alerts/emails to the newly joined customers, making them aware of new offers/discounts for their accounts.

2. Customers lodging complaints last year as well as customers who have contacted customer care recently also show high churning. The company needs to reform its customer care procedures. It needs to review the customer care agents and monitor their efficiency and affectivity while addressing a distressed customer. The refund policies may be reviewed and leniency should be brought to the customer issues regarding refund complaints.

3. Company needs to add more freely available content to 'Regular Plus' account segment since these customers show tendency of using low price account segment but may be wanting more content for their account price. Cheaper channel packs can be made available for the regular plus account holders. A customizable pack can also be designed for the system where customer can select the channels of his choice rather than being given a pre-set channel pack.

4. 'Singles' category customer also show high churn since they might be interested in channels which showcase them series/movies more relatable to their preference. Channel advertisements for romantic comedies, action adventure can be displayed for their accounts in a more frequent manner.

5. Tier 2 cities do not have many customers, which can be addressed by marketing campaigns for the company DTH services for these areas of the country. Banners and posters for the company services need to be used in tier 2 cities to create more awareness about the services.