

Capstone Project



Project Notes -II

Table of Contents

Table of Contents	2
List of Figures	3
List of Tables	4
PART - I	
1. Introduction of the business problem	6
Defining problem statement	6
Need of the study/project	6
Understanding business/social opportunity	6
2. Data Report	7
Understanding how data was collected in terms of time, frequency and methodology	6
Visual inspection of data (rows, columns, descriptive details)	6
Understanding of attributes (variable info, renaming if required)	12
3. Exploratory Data Analysis	13
Univariate analysis	13
Bivariate analysis (relationship between different variables , correlations)	21
Removal of unwanted variables	31
Missing Value treatment	31
Outlier treatment	32
Variable transformation	33
Addition of new variables	35
4. Business insights from EDA	35
Is the data unbalanced? If so, what can be done? Please explain in the context of the business	35
Any business insights using clustering	37
Any other business insights	39
PART - II	
1). Model building and interpretation.	42
a. Build various models (You can choose to build models for either or all of descriptive, predictive or prescriptive purposes)	44
b. Test your predictive model against the test set using various appropriate performance metrics	44
c. Interpretation of the model(s)	44
2). Model Tuning and business implication	52
a. Ensemble modelling, wherever applicable	52
b. Any other model tuning measures(if applicable)	52
c. Interpretation of the most optimum model and its implication on the business	64

List Of Tables

Table	Page No
Table 1.1 Sample of the dataset	6
Table 1.2 Information of the dataset	7
Table 1.3 Unique values in categorical variables	8
Table 1.4 Unique values in Numerical variables	9
Table 1.5 Information of the dataset post cleaning	10
Table 1.6 Descriptive Statistics of the Numerical Variable	10
Table 1.7 Descriptive Statistics of the Categorical Variable	11
Table 1.8 Data Descriptive of Variable	12
Table 1.9 Missing values in dataset	32
Table 1.10 Encoded categorical variables	34
Table 1.11 Scaled variables	34
Table 1.12 Silhouette Score	36
Table 1.13 Cluster Profiles	38
Table 1.14 Sample of the dataset	41
Table 1.15 Classification report Decision Tree	43
Table 1.16 Classification report Random Forest	44
Table 1.17 Classification report LDA	46
Table 1.18 Classification report KNN	47
Table 1.19 Classification report Naïve Bayes	48
Table 1.20 Classification report ANN	50
Table 1.21 Classification report Tuned LD	51
Table 1.22 Classification report Regularised Decision Tree	52
Table 1.23 Classification report Bagging	54
Table 1.24 Classification report Tuned Random Forest	55
Table 1.25 Classification report Adaboost	57
Table 1.26 Classification report XGBoost	58
Table 1.27 Classification report CatBoost	60
Table 1.28 Classification report SVM	61
Table 1.29 Model comparison	63

List Of Figures

Figure	Page No
fig 1.1 Target variable distribution	13
fig 1.2 Univariate Analysis- Tenure	14
fig 1.3 Univariate Analysis- CC_Contacted_LY	14
fig 1.4 Univariate Analysis- rev_pre_month	15
fig 1.5 Univariate Analysis- rev_growth_yoy	15
fig 1.6 Univariate Analysis- coupon_used_for_payment	16
fig 1.7 Univariate Analysis- Day_Since_CC_connect	16
fig 1.8 Univariate Analysis- Cashback	17
fig 1.9 Univariate Analysis- City_tier	17
fig 1.10 Univariate Analysis- Payment	18

fig 1.11 Univariate Analysis- Gender	18
fig 1.12 Univariate Analysis- Service_Score	18
fig 1.13 Univariate Analysis- Account_User_count	19
fig 1.14 Univariate Analysis- Account_Segment	19
fig 1.15 Univariate Analysis- CC_Agent_Score	20
fig 1.16 Univariate Analysis- Marital Status	20
fig 1.17 Univariate Analysis- Complaint_LY	20
fig 1.18 Univariate Analysis- Login_device	21
fig 1.19 Bivariate Analysis- City_Tier with Churn	21
fig 1.20 Bivariate Analysis- Payment with Churn	22
fig 1.21 Bivariate Analysis- Payment with Churn	22
fig 1.22 Bivariate Analysis- Service_Score with Churn	23
fig 1.23 Bivariate Analysis- Account_user with Churn	23
fig 1.24 Bivariate Analysis- Account Segment with Churn	24
fig 1.25 Bivariate Analysis- CC_Agent_Score with Churn	24
fig 1.26 Bivariate Analysis- Marital Status with Churn	25
fig 1.27 Bivariate Analysis- Complain_ly with Churn	25
fig 1.28 Bivariate Analysis- Login Device with Churn	26
fig 1.29 Bivariate Analysis- Tenure with Churn	26
fig 1.30 Bivariate Analysis- CC_Contacted_LY with Churn	27
fig 1.31 Bivariate Analysis- rev_per_month with Churn	27
fig 1.32 Bivariate Analysis- rev_growth_yoy with Churn	27
fig 1.33 Bivariate Analysis- coupon_used_for_payment with Churn	28
fig 1.34 Bivariate Analysis - Day_Since_CC_connect with Churn	28
fig 1.35 Bivariate Analysis- cashback with Churn	28
fig 1.36 Pairplot	29
fig 1.37 Correlation Matrix	30
fig 1.38 Missing value Treatment	31
fig 1.39 Outliers in the data	32
fig 1.40 Outliers after treatment	33
fig 1.41 Target Class imbalance	35
fig 1.42 distortion Score Elbow for KMeans Clustering	36
fig 1.43 Plot of the clusters	37
fig 1.44 Cluster distribution	37
Fig 1.46 Confusion Matrix Decision Tree	44
Fig 1.47 ROC-AUC Curve Decision Tree	44
Fig 1.48 Learning Curve Decision Tree	45
Fig 1.49 Feature importance Decision Tree	45
Fig 1.50 Confusion Matrix Random Forest	46
Fig 1.51 ROC-AUC Curve Random Forest	46
Fig 1.52 Learning Curve Random Forest	46
Fig 1.53 Feature importance Random Forest	46
Fig 1.54 Confusion Matrix LDA	47
Fig 1.55 ROC-AUC Curve LDA	47
Fig 1.56 Learning Curve LDA	48

Fig 1.57 Feature importance LDA	48
Fig 1.58 Confusion Matrix KNN	48
Fig 1.59 ROC-AUC Curve KNN	49
Fig 1.60 Learning Curve KNN	49
Fig 1.61 Confusion Matrix Naïve Bayes	50
Fig 1.62 ROC-AUC Curve Naïve Bayes	50
Fig 1.63 Learning Curve Naïve Bayes	50
Fig 1.64 Confusion Matrix ANN	51
Fig 1.65 ROC-AUC Curve ANN	51
Fig 1.66 Learning Curve ANN	51
Fig 1.67 Confusion Matrix Tuned LDA	52
Fig 1.68 ROC-AUC Curve Tuned LDA	52
Fig 1.69 Learning Curve Tuned LDA	53
Fig 1.70 Confusion Matrix Regularised Decision Tree	53
Fig 1.71 ROC-AUC Curve Regularised Decision Tree	54
Fig 1.72 Learning Curve Regularized Decision Tree	54
Fig 1.73 Feature importance Regularized Decision Tree	54
Fig 1.74 Confusion Matrix Bagging	55
Fig 1.75 ROC-AUC Curve Bagging	55
Fig 1.76 Learning Curve Bagging	55
Fig 1.77 Confusion Matrix Tuned Random Forest	56
Fig 1.78 ROC-AUC Curve Tuned Random Forest	57
Fig 1.79 Learning Curve Tuned Random Forest	57
Fig 1.80 Feature importance Tuned Random Forest	57
Fig 1.81 Confusion Matrix Adaboost	58
Fig 1.82 ROC-AUC Curve Adaboost	58
Fig 1.83 Learning Curve Adaboost	58
Fig 1.84 Feature importance Adaboost	58
Fig 1.85 Confusion Matrix XGBoost	59
Fig 1.86 ROC-AUC Curve XGBoost	60
Fig 1.87 Learning Curve XGBoost	60
Fig 1.88 Feature importance XGBoost	60
Fig 1.89 Confusion Matrix CatBoost	61
Fig 1.90 ROC-AUC Curve CatBoost	61
Fig 1.91 Learning Curve CatBoost	62
Fig 1.92 Feature importance CatBoost	62
Fig 1.93 Confusion Matrix SVM	63
Fig 1.94 ROC-AUC Curve SVM	63
Fig 1.95 Learning Curve SVM	63
Fig 1.96 ROC-AUC Curve Train data	64
Fig 1.97 ROC-AUC Curve Test data	64

CUSTOMER CHURN PROJECT NOTES -I

1. INTRODUCTION OF THE BUSINESS PROBLEM

PROBLEM STATEMENT:

The problem at hand is developing a churn prediction model for a DTH service provider. The primary challenge faced by the company is retaining existing customers in a highly competitive market. Account churn, where one account can encompass multiple customers, poses a significant threat, as the loss of a single account can result in the attrition of several customers. The problem is to create a predictive model to identify potential churners and design segmented offers to retain them.

WHY IS IT NEEDED?:

The need for this study or project arises from the company's requirement to address the critical issue of customer churn. Retaining customers is pivotal, and the company aims to achieve this by creating a model that can predict churn and provide tailored offers to potential churners. In this context, the project is essential for enhancing customer retention, staying competitive in the market, exploring revenue growth opportunities, and elevating the overall customer experience.

UNDERSTANDING BUSINESS OPPORTUNITY

The business opportunity lies in devising a sophisticated churn prediction model and formulating distinct campaign recommendations to combat churn effectively.

The potential benefits include strengthening customer loyalty, improving market competitiveness, unlocking revenue growth, and delivering an exceptional customer experience.

The project represents a unique business opportunity to align recommendations with strategic objectives, ensuring they don't inadvertently lead to losses for the company. It underscores the importance of creating offers that are both enticing to customers and financially viable for the business, presenting a balance between customer satisfaction and revenue assurance.

2. DATA REPORT:

a) Understanding how data was collected in terms of time, frequency and methodology

The data source and context are not explicitly detailed. It appears to be derived from a DTH (or Cable) service provider, potentially originating from historical records. The data column description indicates it as average of previous 12 months at any point in time. However, geographical coverage and data duration (Date/year) remain unspecified, so we make the assumption that it pertains to the Indian States, with monetary values expressed in Rupees, this assumption is taken basis the payment methods shown as UPI in data.

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

About the dataset:

The number of rows are **11260**

The number of columns (variables) are **19**

Sample of the data

'First 5 samples:'																			
	AccountID	Churn	Tenure	City_Tier	CC_Contacted_LY	Payment	Gender	Service_Score	Account_user_count	account_segment	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	20000	1	4.0	3.0		6.0	Debit Card	Female	3.0	3.0	Super	2.0	Single	9.0	1.0	11.0	1.0	5.0 159.929993	Mobile
1	20001	1	0.0	1.0		8.0	UPI	Male	3.0	4.0	Regular Plus	3.0	Single	7.0	1.0	15.0	0.0	0.0 120.900002	Mobile
2	20002	1	0.0	1.0		30.0	Debit Card	Male	2.0	4.0	Regular Plus	3.0	Single	6.0	1.0	14.0	0.0	3.0 Nan	Mobile
3	20003	1	0.0	3.0		15.0	Debit Card	Male	2.0	4.0	Super	5.0	Single	8.0	0.0	23.0	0.0	3.0 134.070007	Mobile
4	20004	1	0.0	1.0		12.0	Credit Card	Male	2.0	3.0	Regular Plus	5.0	Single	3.0	0.0	11.0	1.0	3.0 129.600006	Mobile
'Last 5 samples:'																			
	AccountID	Churn	Tenure	City_Tier	CC_Contacted_LY	Payment	Gender	Service_Score	Account_user_count	account_segment	CC_Agent_Score	Marital_Status	rev_per_month	Complain_ly	rev_growth_yoy	coupon_used_for_payment	Day_Since_CC_connect	cashback	Login_device
11255	31255	0	10.0	1.0		34.0	Credit Card	Male	3.0	2.0	Super	1.0	Married	9.0	0.0	19.0	1.0	4.0 153.710007	Computer
11256	31256	0	13.0	1.0		19.0	Credit Card	Male	3.0	5.0	HNI	5.0	Married	7.0	0.0	16.0	1.0	8.0 226.910004	Mobile
11257	31257	0	1.0	1.0		14.0	Debit Card	Male	3.0	2.0	Super	4.0	Married	7.0	1.0	22.0	1.0	4.0 191.419998	Mobile
11258	31258	0	23.0	3.0		11.0	Credit Card	Male	4.0	5.0	Super	4.0	Married	7.0	0.0	16.0	2.0	9.0 179.899994	Computer
11259	31259	0	8.0	1.0		22.0	Credit Card	Male	3.0	2.0	Super	3.0	Married	5.0	0.0	13.0	2.0	3.0 175.039993	Mobile

Table 1.1 Sample of the dataset

Information about data

```

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)

```

Table 1.2 Information of the dataset

The initial data review suggests the need for data preprocessing, including handling missing values and potentially converting data types. Additionally, feature engineering and cleaning will be required to prepare the data for analysis. Below are the key details:

- **Data Size and Completeness:**
 - The dataset contains **11,260 rows** and **19 columns**.
 - There are missing values in several columns, indicated by the "Non-Null Count" being less than **11,260**.
- **Target Variable (Churn):**
 - The "Churn" column is binary, indicating whether an account churned or not. This is binary value, hence this is a **classification problem**
- **Categorical and Numerical Features:**
 - The dataset contains a mix of categorical and numerical features.
 - Several features are incorrectly labelled as type object or Numerical which will require data type conversion for analysis.
 - "City_Tier" is currently represented as a float but may need to be converted to a categorical data type.
 - "rev_per_month," "coupon_used_for_payment," "cashback" are tagged as categorical features, but they appear to be numeric. These columns need their data types corrected.
- **Missing Values:**
 - "Tenure" and "City_Tier" columns contain missing values, requiring data imputation.
 - "CC_Contacted_LY," "Payment," "Gender," "Marital_Status," "Account_user_count," "account_segment," "Complain_ly," "Day_Since_CC_connect," "rev_per_month," "coupon_used_for_payment," "cashback," and "Login_device" columns also have missing values that may need data imputation or cleaning.

Data Cleaning & Fixing Data Types

We'll start with the Categorical columns, wherein we will identify unique values in each and fix the values wherever we find data issues.

Categorical

- Lots of Numeric values seems to be tagged as Object, we checked their unique values & found unexpected values as shown below:

```

Value counts for column 'Tenure':
[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]

Value counts for column 'Payment':
['Debit Card' 'UPI' 'Credit Card' 'Cash on Delivery' 'E wallet' nan]

Value counts for column 'Gender':
['Female' 'Male' 'F' nan 'M']

Value counts for column 'Account_user_count':
[3 4 nan 5 2 '@' 1 6]

Value counts for column 'account_segment':
['Super' 'Regular Plus' 'Regular' 'HNI' 'Regular +' nan 'Super Plus'
 'Super +']

Value counts for column 'Marital_Status':
['Single' 'Divorced' 'Married' nan]

Value counts for column 'rev_per_month':
[9 7 6 8 3 2 4 10 1 5 '+' 130 nan 19 139 102 120 138 127 123 124 116 21
 126 134 113 114 108 140 133 129 107 118 11 105 20 119 121 137 110 22 101
 136 125 14 13 12 115 23 122 117 131 104 15 25 135 111 109 100 103]

Value counts for column 'rev_growth_yoy':
[11 15 14 23 22 16 12 13 17 18 24 19 20 21 25 26 '$' 4 27 28]

Value counts for column 'coupon_used_for_payment':
[1 0 4 2 9 6 11 7 12 10 5 3 13 15 8 '#' '$' 14 '*' 16]

Value counts for column 'Day_Since_CC_connect':
[5 0 3 7 2 1 8 6 4 15 nan 11 10 9 13 12 17 16 14 30 '$' 46 18 31 47]

Value counts for column 'cashback':
[159.93 120.9 nan ... 227.36 226.91 191.42]

Value counts for column 'Login_device':
['Mobile' 'Computer' '&&&' nan]

```

Table 1.3 Unique values in categorical variables

Column value counts include a mix of valid and invalid entries, potentially requiring data cleaning and standardization.:

NAN / Missing Values: Below columns have Nan present:

- Tenure, City_Tier, CC_Contacted_LY, Payment, Gender, Service_Score, Account_user_count, ac count_segment, CC_Agent_Score, Marital_Status, rev_per_month, Complain_ly, Day_Since_ CC_connect, cashback, Login_device

Unexpected Values:

- Tenure** (Unexpected values like '#', high numerical values)
- Account_user_count** (Unexpected value '@' alongside numerical counts)

- **Gender** (Inconsistent formatting with '**F**' and '**M**' alongside 'Female' and 'Male')
- **rev_per_month** (Unexpected values like '**+**', high numerical values)
- **rev_growth_yoy** (Unexpected values like '**\$**' mixed with numerical values)
- **coupon_used_for_payment** (Unexpected values like '**#**', '**\$**', and '*****' with numerical value)
- **Day_Since_CC_connect** (Unexpected values like '**\$**' mixed with numerical values)
- **Login_device** (Unexpected value '**&&&**' alongside 'Mobile' and 'Computer')

Approach to fix: We fixed these unexpected values by **replacing them with nan(Null)**.
The nan will be treated further using appropriate methods.

Numerical Columns

```

Value counts for column 'Tenure':
[ 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. ]

Value counts for column 'City_Tier':
[ 3.  1.  nan  2. ]

Value counts for column 'CC_Contacted_LY':
[ 6.  8.  30.  15.  12.  22.  11.  9.  31.  18.  13.  20.  29.  28.
 26.  14.  10.  25.  27.  17.  23.  33.  19.  35.  24.  16.  32.  21.
 nan  34.  5.  4.  126.  7.  36.  127.  42.  38.  37.  39.  40.  41.
 132.  43.  129. ]

Value counts for column 'Service_Score':
[ 3.  2.  1.  nan  0.  4.  5. ]

Value counts for column 'Account_user_count':
[ 3.  4.  nan  5.  2.  1.  6. ]

Value counts for column 'CC_Agent_Score':
[ 2.  3.  5.  4.  nan  1. ]

Value counts for column 'rev_per_month':
[ 9.  7.  6.  8.  3.  2.  4.  10.  1.  5.  nan  130.  19.  139.
 102.  120.  138.  127.  123.  124.  116.  21.  126.  134.  113.  114.  108.  140.
 133.  129.  107.  118.  11.  105.  20.  119.  121.  137.  110.  22.  101.  136.
 125.  14.  13.  12.  115.  23.  122.  117.  131.  104.  15.  25.  135.  111.
 109.  100.  103. ]

Value counts for column 'Complain_ly':
[ 1.  0.  nan]

Value counts for column 'rev_growth_yoy':
[11.  15.  14.  23.  22.  16.  12.  13.  17.  18.  24.  19.  20.  21.  25.  26.  nan  4.
 27.  28. ]

Value counts for column 'coupon_used_for_payment':
[ 1.  0.  4.  2.  9.  6.  11.  7.  12.  10.  5.  3.  13.  15.  8.  nan  14.  16. ]

Value counts for column 'Day_Since_CC_connect':
[ 5.  0.  3.  7.  2.  1.  8.  6.  4.  15.  nan  11.  10.  9.  13.  12.  17.  16.
 14.  30.  46.  18.  31.  47. ]

```

Table 1.4 Unique values in Numerical variables

- We did not find any unexpected or negative values in Numerical columns except those which were tagged as Object and **NAN values**
- There were many columns that were tagged as Numerical, we converted below Columns to categorical.

City_Tier
 Complain_ly
 Service_Score
 Account_user_count
 CC_Agent_Score

Post the data cleaning & datatype transformation, here is the **updated data types**:

RangeIndex: 11260 entries, 0 to 11259			
Data columns (total 19 columns):			
#	Column	Non-Null Count	Dtype
0	AccountID	11260 non-null	object
1	Churn	11260 non-null	int64
2	Tenure	11042 non-null	float64
3	City_Tier	11260 non-null	object
4	CC_Contacted_LY	11158 non-null	float64
5	Payment	11151 non-null	object
6	Gender	11152 non-null	object
7	Service_Score	11260 non-null	object
8	Account_user_count	11260 non-null	object
9	account_segment	11163 non-null	object
10	CC_Agent_Score	11260 non-null	object
11	Marital_Status	11048 non-null	object
12	rev_per_month	10469 non-null	float64
13	Complain_ly	11260 non-null	object
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	float32
18	Login_device	10500 non-null	object
dtypes: float32(1), float64(6), int64(1), object(11)			

Table 1.5 Information of the dataset post cleaning

Now, the data type has been fixed. We have 8 Numerical & 11 Categorical column

Statistical Summary of dataset

Numerical columns

	count	mean	std	min	25%	50%	75%	max
Churn	11260.0	0.168384	0.374223	0.0	0.000000	0.00	0.000000	1.0
Tenure	11042.0	11.025086	12.879782	0.0	2.000000	9.00	16.000000	99.0
CC_Contacted_LY	11158.0	17.867091	8.853269	4.0	11.000000	16.00	23.000000	132.0
rev_per_month	10469.0	6.362594	11.909686	1.0	3.000000	5.00	7.000000	140.0
rev_growth_yoy	11257.0	16.193391	3.757721	4.0	13.000000	15.00	19.000000	28.0
coupon_used_for_payment	11257.0	1.790619	1.969551	0.0	1.000000	1.00	2.000000	16.0
Day_Since_CC_connect	10902.0	4.633187	3.697637	0.0	2.000000	3.00	8.000000	47.0
cashback	10787.0	196.236343	178.660522	0.0	147.210007	165.25	200.009995	1997.0

Table 1.6 Descriptive Statistics of the Numerical Variable

- The average **tenure** of accounts is approximately 11 months, with a wide range from 0 to 99 months.
- On average, **customer care connect** in last year was approximately 17.87 times in the last year, with a range from 4 to 132 times.
- The average **monthly revenue generated** by accounts is 6.36, with a wide range from 1.0 to 140.0.
- The accounts show an average **year-over-year revenue growth** of approximately 16.19%, with values ranging from 4.0% to 28.0%.
- On average, accounts use approximately 1.79 **coupons for payments**, with a range from 0 to 16 coupons.
- On average, accounts have not **contacted customer care** for approximately 4.63 days, with a range from 0 to 47 days
- The average **monthly cashback** generated by accounts is approximately 196.24, with a range from 0.0 to 1997.0.

Categorical columns

	count	unique	top	freq
AccountID	11260	11260	20000	1
City_Tier	11260	4	1.0	7263
Payment	11151	5	Debit Card	4587
Gender	11152	2	Male	6704
Service_Score	11260	7	3.0	5490
Account_user_count	11260	7	4.0	4569
account_segment	11163	5	Regular Plus	4124
CC_Agent_Score	11260	6	3.0	3360
Marital_Status	11048	3	Married	5860
Complain_ly	11260	3	0.0	7792
Login_device	10500	2	Mobile	7482

Table 1.7 Descriptive Statistics of the Categorical Variable

- AccountID:** All 11260 entries are unique.
- City_Tier:** There are 4 unique city tiers. The most frequent city tier is 1.0, which appears 7263 times.
- Payment:** There are 5 unique payment methods. The most frequent payment method is "Debit Card," occurring 4587 times.
- Gender:** There are 2 unique genders. "Male" is the most frequent gender, with 6704 occurrences.

- Service_Score:** There are 7 unique service scores. The most frequent service score is 3.0, appearing 5490 times.
- Account User count-** Most of the accounts has user count of 4 (4569), with a minimum of 1 and a maximum of 6 users.
- account_segment:** There are 5 unique account segments. "Regular Plus" is the most frequent account segment, occurring 4124 times.
- CC_Agent_Score:** There are 6 unique customer care agent scores. The most frequent agent score is 3.0, with 3360 occurrences.
- Marital_Status:** There are 3 unique marital statuses. "Married" is the most frequent status, occurring 5860 times.
- Complain_ly:** There are 3 unique values. The most frequent value is 0.0, appearing 7792 times.
- Login_device:** There are 2 unique login devices. "Mobile" is the most frequent login device, occurring 7482 times.

Please note: the above descriptive summary is of original dataset post cleaning, inclusive of nan & outliers, hence count may vary post treatment.

Duplicate Values

Total **duplicates** in the data are: **0**

c) Understanding of attributes

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 the last 12 months
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 the 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 the 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 have 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 months)

coupon_used_l12m	How many times customers have used coupons to do the payment in the 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 the last 12 months
Login_device	Preferred login device of the customers in the account

Table 1.8 Data Descriptive of Variable

- There is **no renaming of columns needed** here.

3. Exploratory Data Analysis

a) Univariate analysis

Target Variable

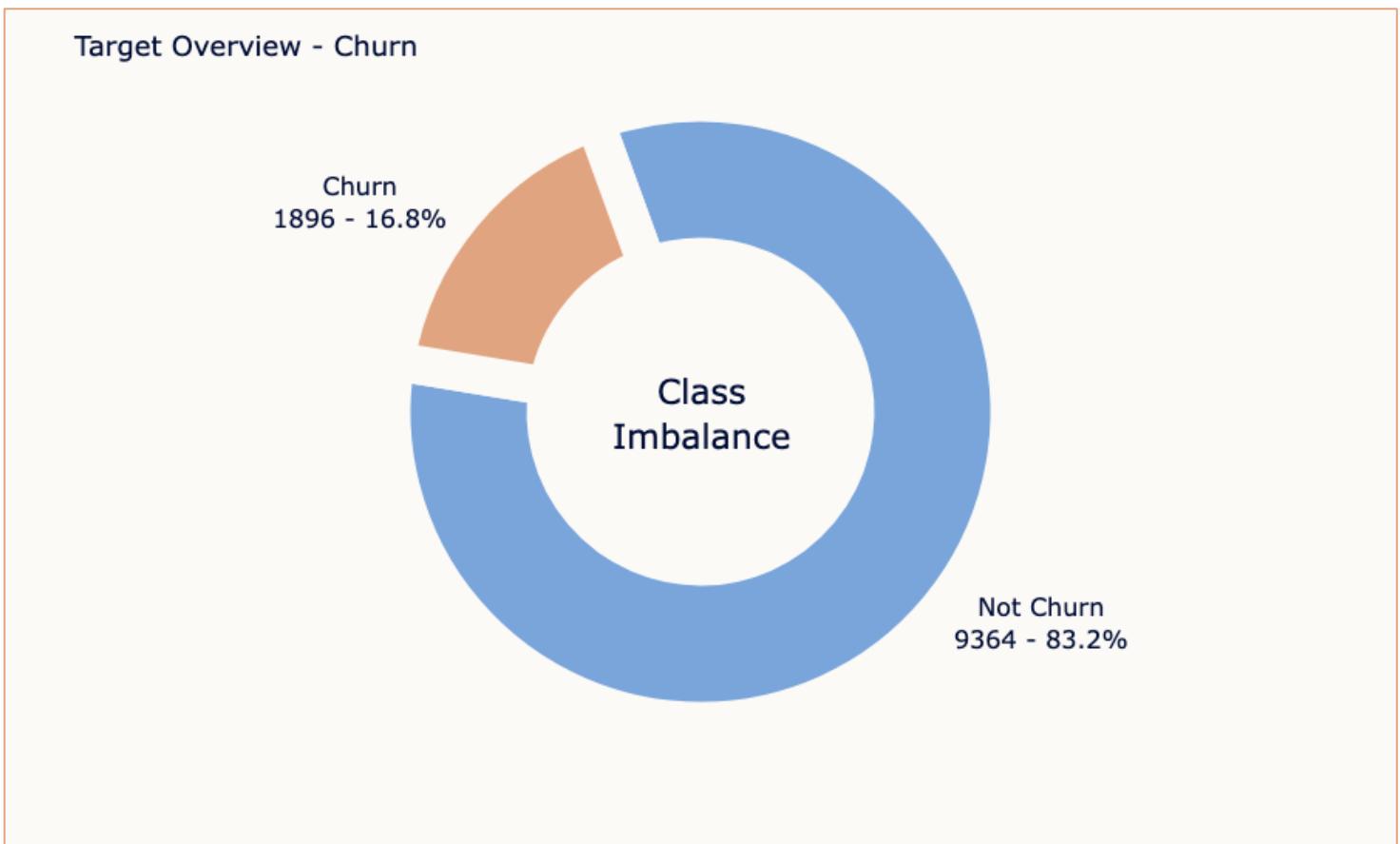


fig 1.1 Target variable distribution

- Churn Variable has 2 unique values , **1 which stands for Churned customer and 0 for non churned customer.**
- We can clearly see the class Imbalance here. only 16.8% of the customers are churned.

Numerical Variables

Tenure:

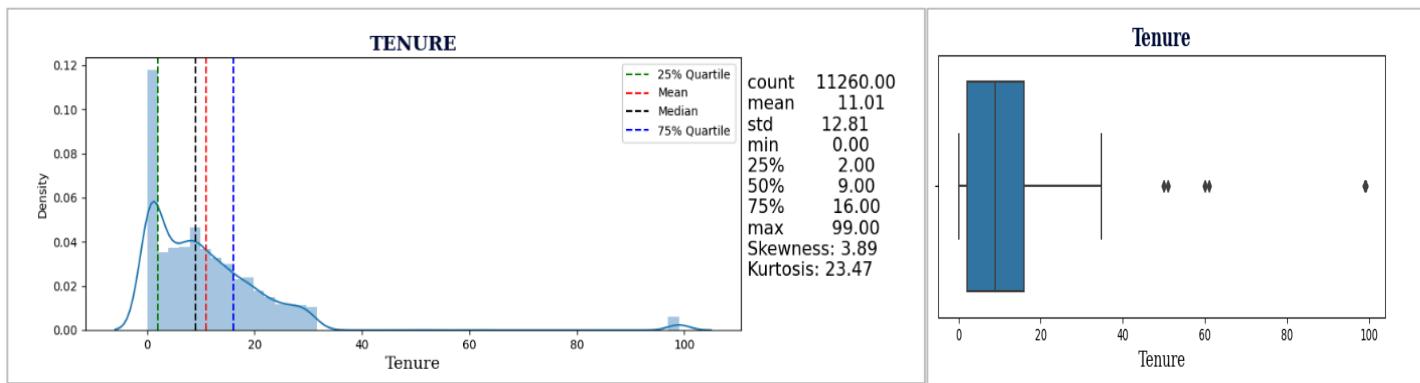


fig 1.2 Univariate Analysis- Tenure

- The tenure values have a wide range from 0 to 99 months, indicating variability in customer loyalty.
- The distribution is positively skewed (skewness 3.90), suggesting that more customers have shorter tenure.
- There are significant outliers on the higher end (kurtosis 23.37), which might represent long-term loyal customers or data anomalies.

CC_Contacted_LY:

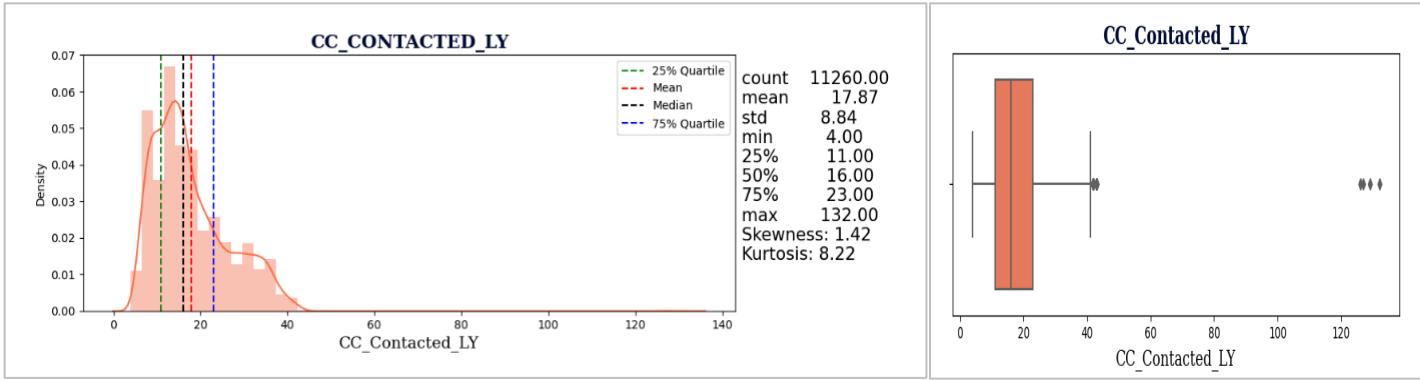


fig 1.3 Univariate Analysis- CC_Contacted_LY

- The number of CC contacts in the last year ranges from 4 to 132, indicating variation in customer interactions.
- The distribution is moderately positively skewed (skewness 1.42), suggesting that most customers had relatively fewer contacts.
- There are some outliers with a higher number of contacts (kurtosis 8.23), indicating potential high engagement customers or data anomalies.

rev_per_month:

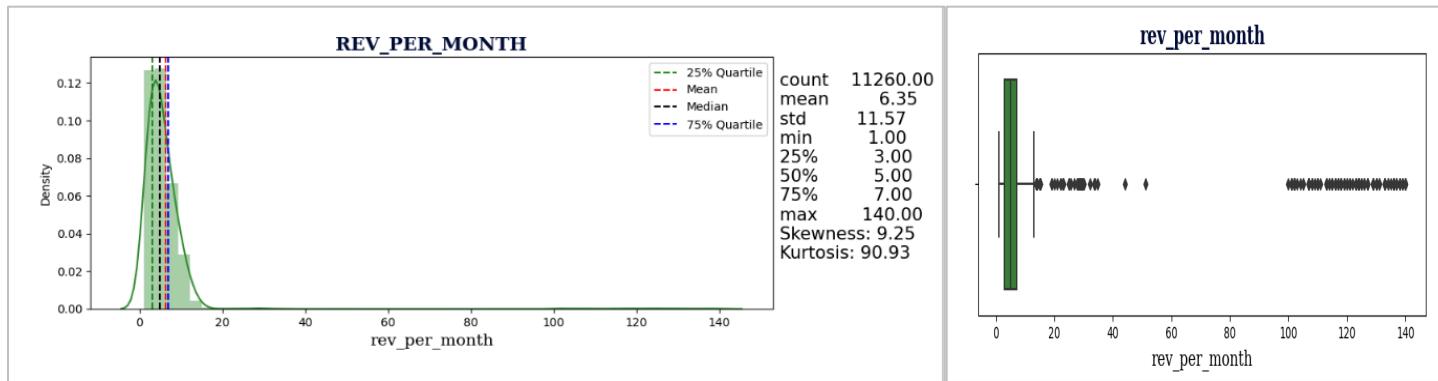


fig 1.4 Univariate Analysis- rev_per_month

- Monthly revenue varies significantly, with values ranging from 1 to 140.
- The distribution is highly positively skewed (skewness 9.09), indicating that most customers generate lower revenue.
- There are many extreme outliers (kurtosis 86.96), potentially representing high-revenue customers or data anomalies.

rev_growth_yoy:

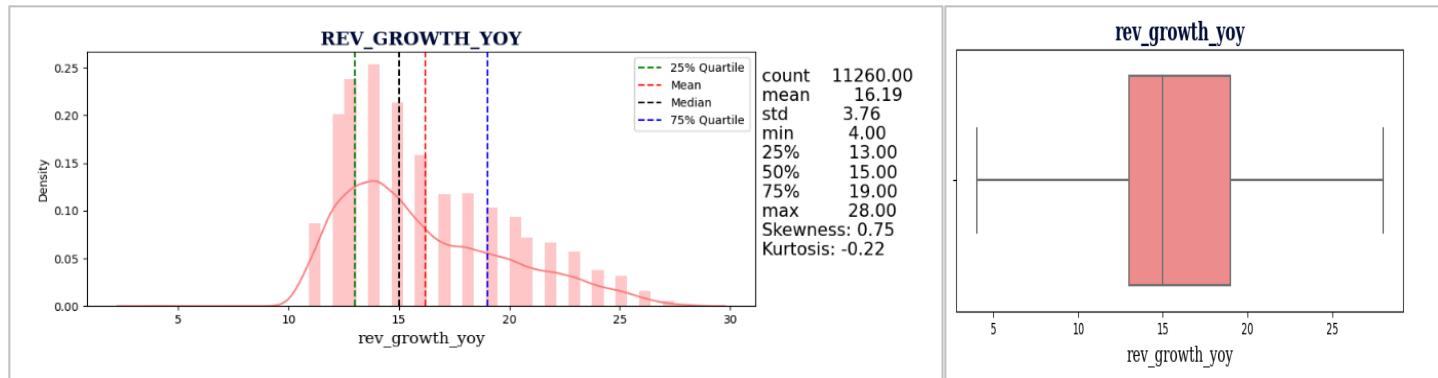


fig 1.5 Univariate Analysis- rev_growth_yoy

- Year-over-year revenue growth percentages range from 4% to 28%.
- The distribution is moderately positively skewed (skewness 0.75), suggesting steady growth rates.
- The kurtosis value is close to zero (kurtosis -0.22), indicating a distribution with lighter tails and fewer outliers.

coupon_used_for_payment:

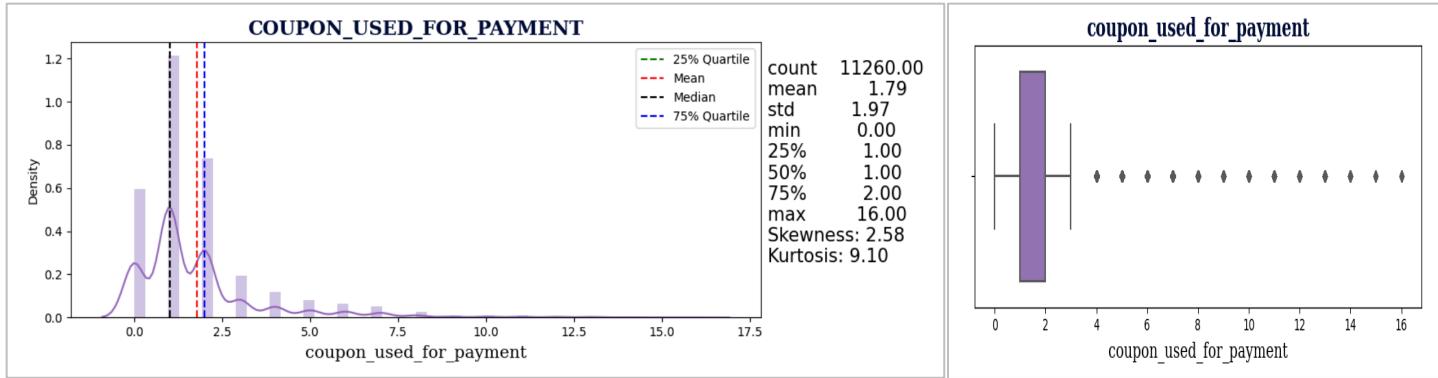


fig 1.6 Univariate Analysis- coupon_used_for_payment

- Most customers use 1 to 2 coupons for payments, with a range of 0 to 16.
- The distribution is positively skewed (skewness 2.58), indicating that a significant portion of customers uses coupons for payment.
- The kurtosis value is relatively high (kurtosis 9.10), suggesting a distribution with some outliers related to coupon usage.

Day_Since_CC_connect:

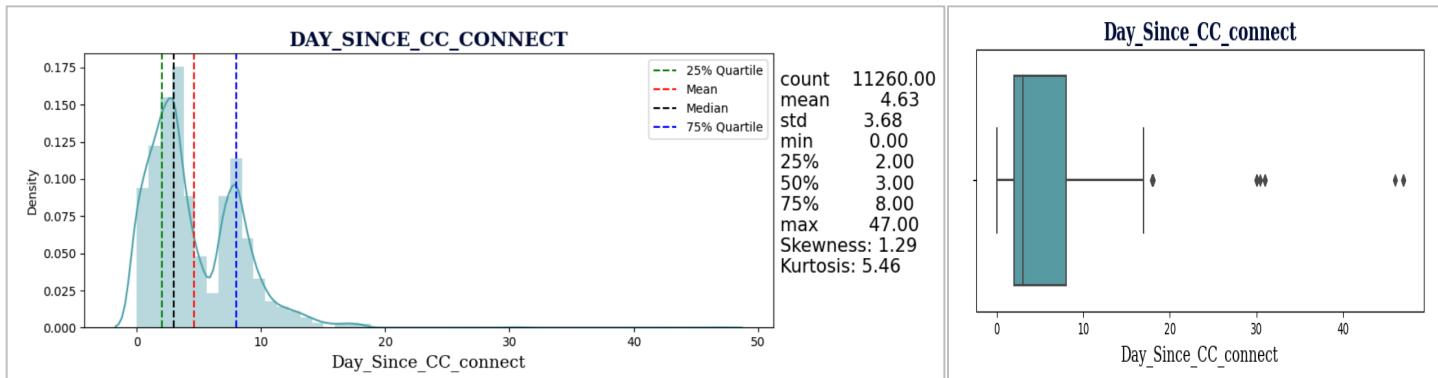


fig 1.7 Univariate Analysis- Day_Since_CC_connect

- The number of days since the last customer care connect spans from 0 to 47 days.
- The distribution is positively skewed (skewness 1.27), indicating that most customers recently connected. Around 50% of the user have not connected since 3 days or more.
- The kurtosis value is 5.33, suggesting a distribution with some customers having longer periods since the last connect.

cashback:

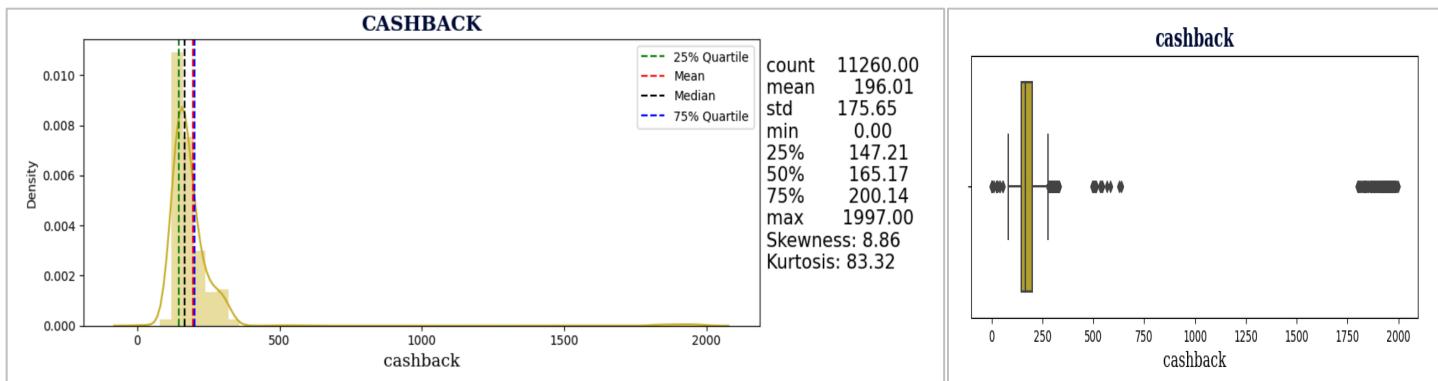


fig 1.8 Univariate Analysis- Cashback

- **Cashback** values vary widely from **0 to 1997**.
- The **distribution** is highly **positively skewed** (skewness 8.77), indicating that **most customers receive lower cashback amounts**.
- There are numerous **extreme outliers** (kurtosis 81.11), which could represent customers with **exceptionally high cashback or anomalies in the data**.

Categorical Variables

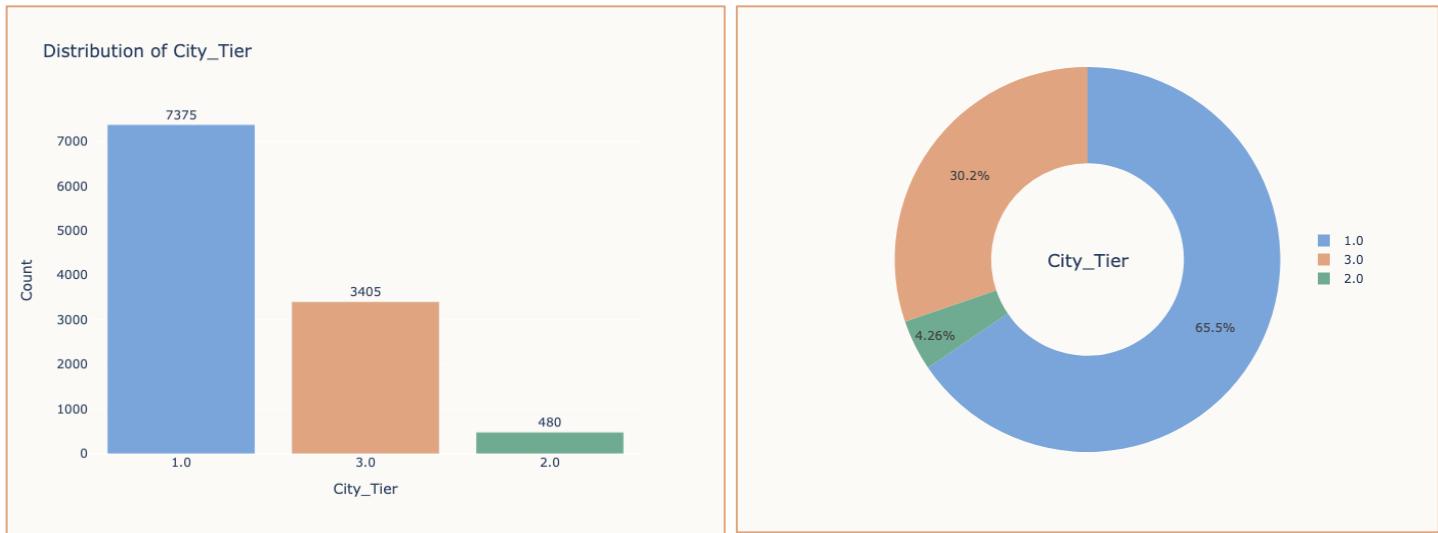


fig 1.9 Univariate Analysis- City_tier

- Approximately **65.50%** of customers are in **City Tier 1** while around **30.24%** of customers are in **City Tier 3**.
- The customers decreases with tiers, least customers are from lower tier cities.

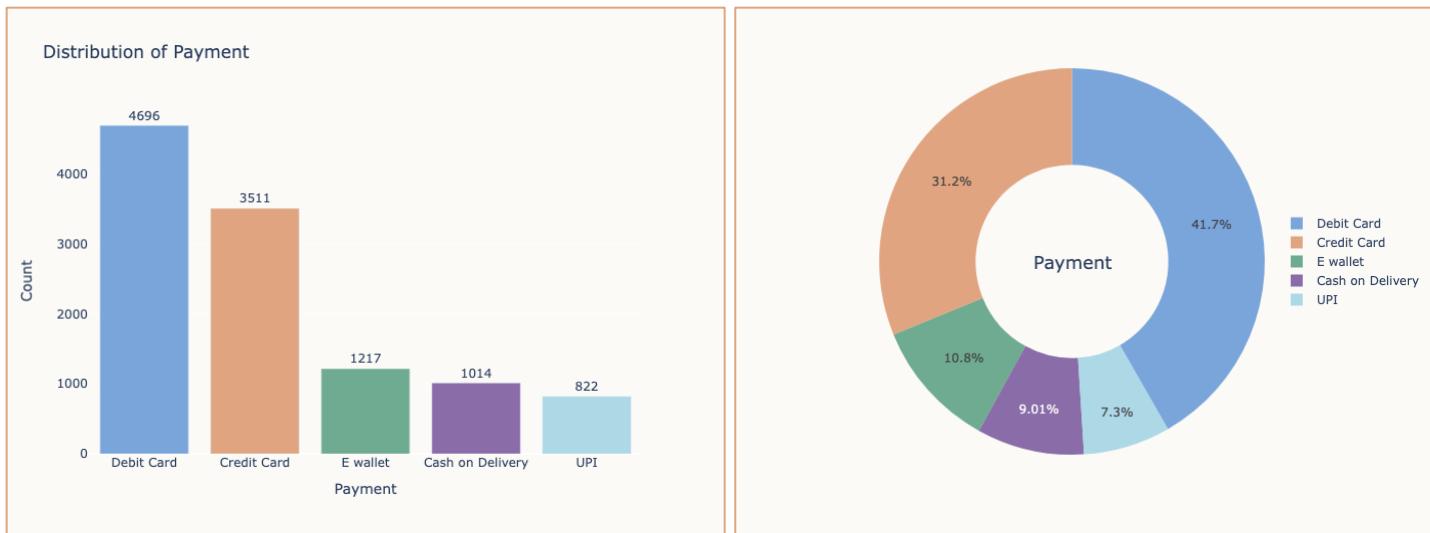


fig 1.10 Univariate Analysis- Payment

- The data suggests that most customers prefer "Debit Card" (41.71%) and "Credit Card" (31.18%) as payment method

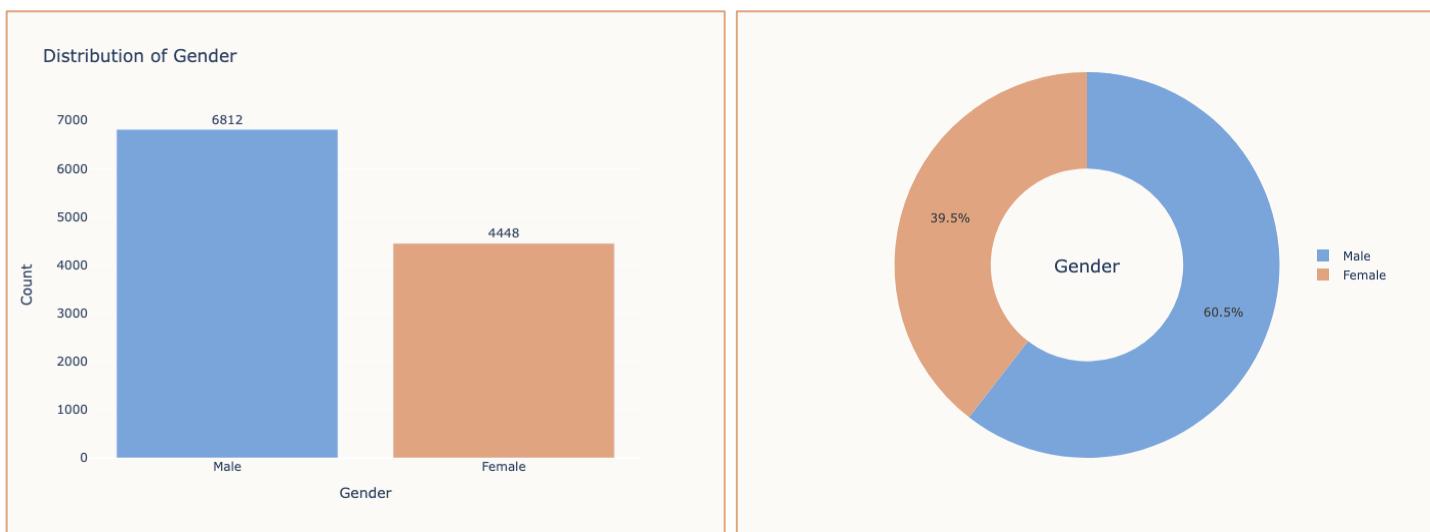


fig 1.11 Univariate Analysis- Gender

- 60.50% of customers are male (Majority)

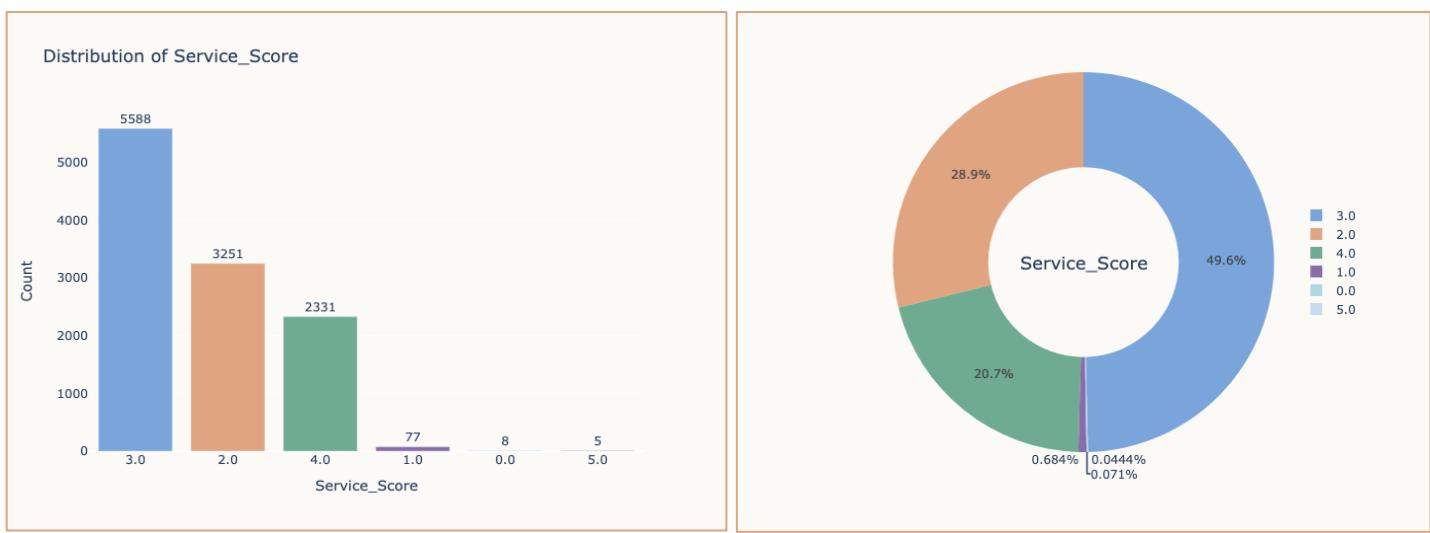


fig 1.12 Univariate Analysis- Service_Score

- A significant portion of customers have "**Service_Score**" of **3.0 (49.63%)** suggesting customers are relatively **neutral with the services**.
- There are only **5 (0.071%) users** who are **highly satisfied & 20.7% just satisfied**, which indicates **need of improvement in services**.

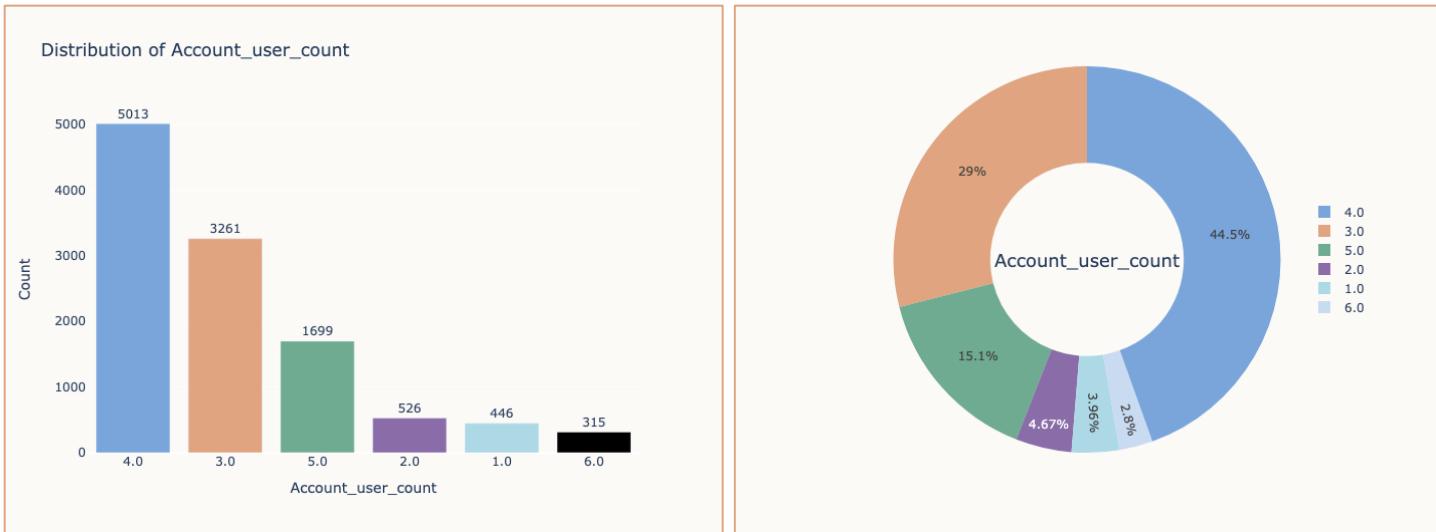


fig 1.13 Univariate Analysis- Account_User_Count

- Most customers have **3 to 4 account users**, with a range of 1 to 6 users.

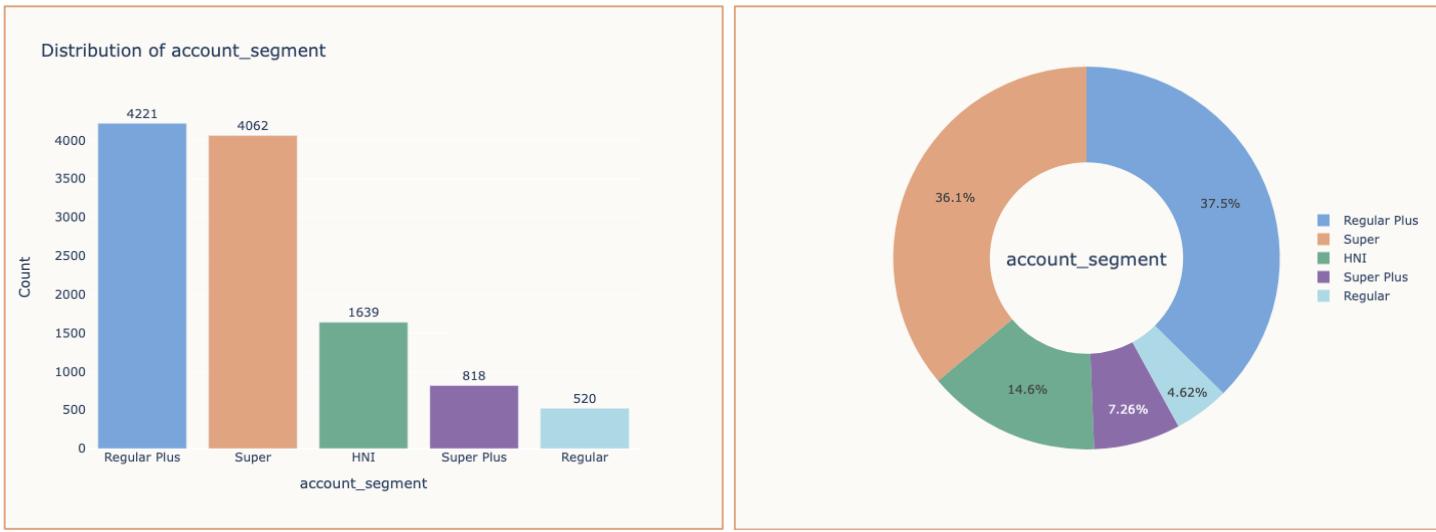


fig 1.14 Univariate Analysis- Account_Segment

- **"Regular Plus" and "Super"** are the most common account segments, making up **37.49% and 36.07%** of the customer base, respectively.

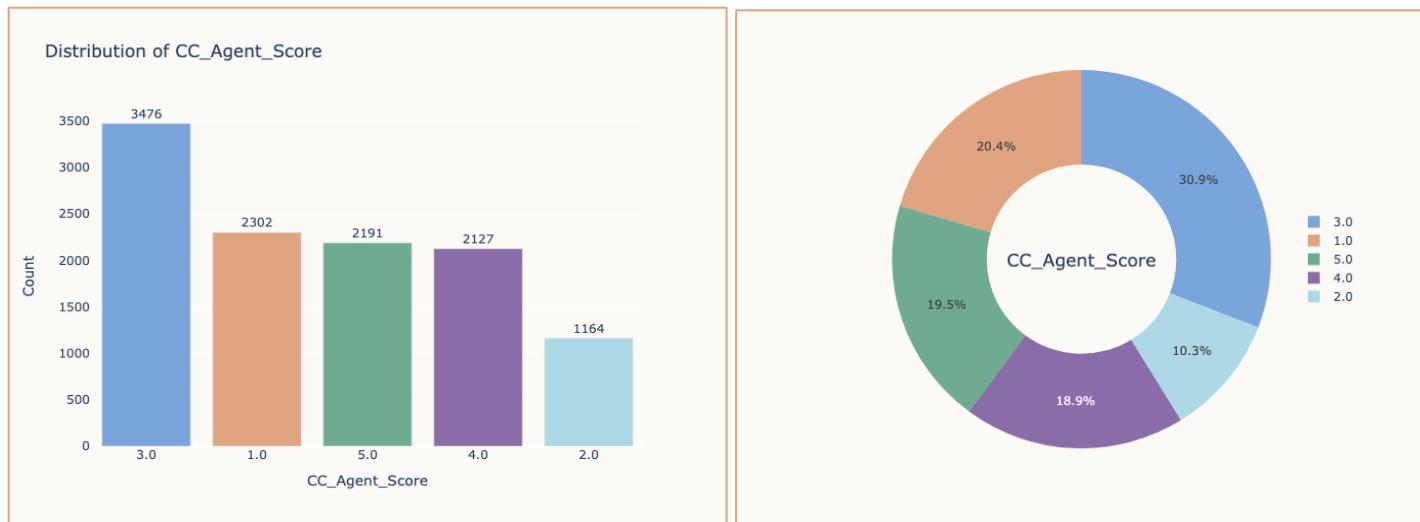


fig 1.15 Univariate Analysis- CC_Agent_Score

- **38.35% having customer care score 4 & 5** suggesting a ratio of **Satisfied users with customer care** while **30.78% having score 1 & 2** suggesting **relatively high dissatisfaction**

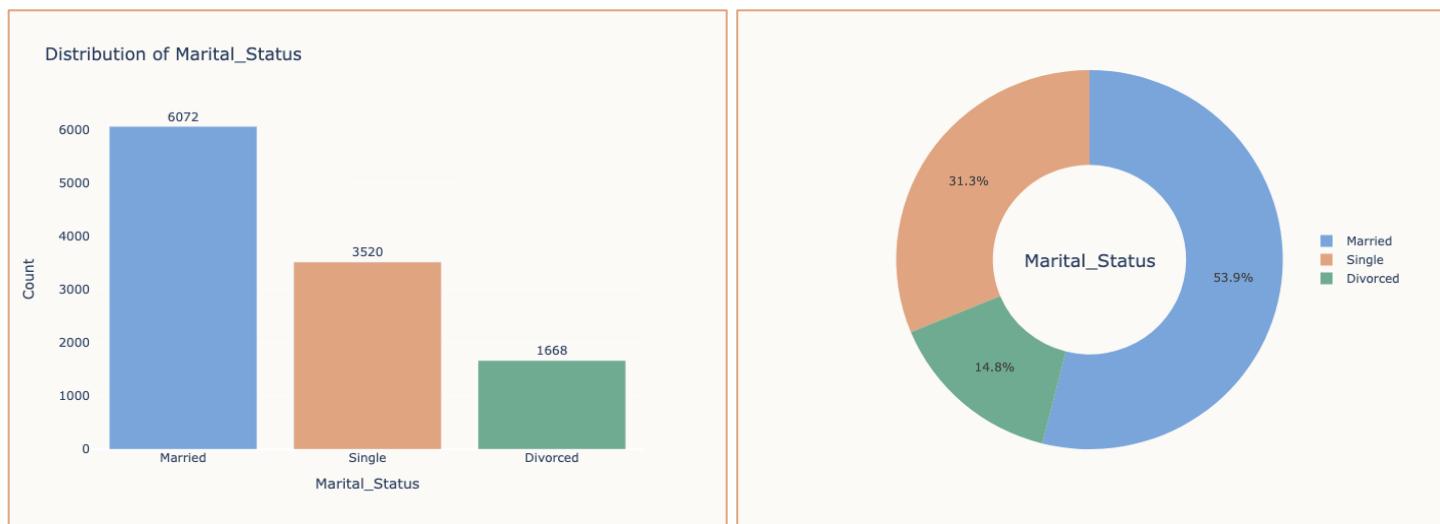


fig 1.16 Univariate Analysis- Marital Status

- **53.9% of the subscribers are married** while **31.3% are single.**

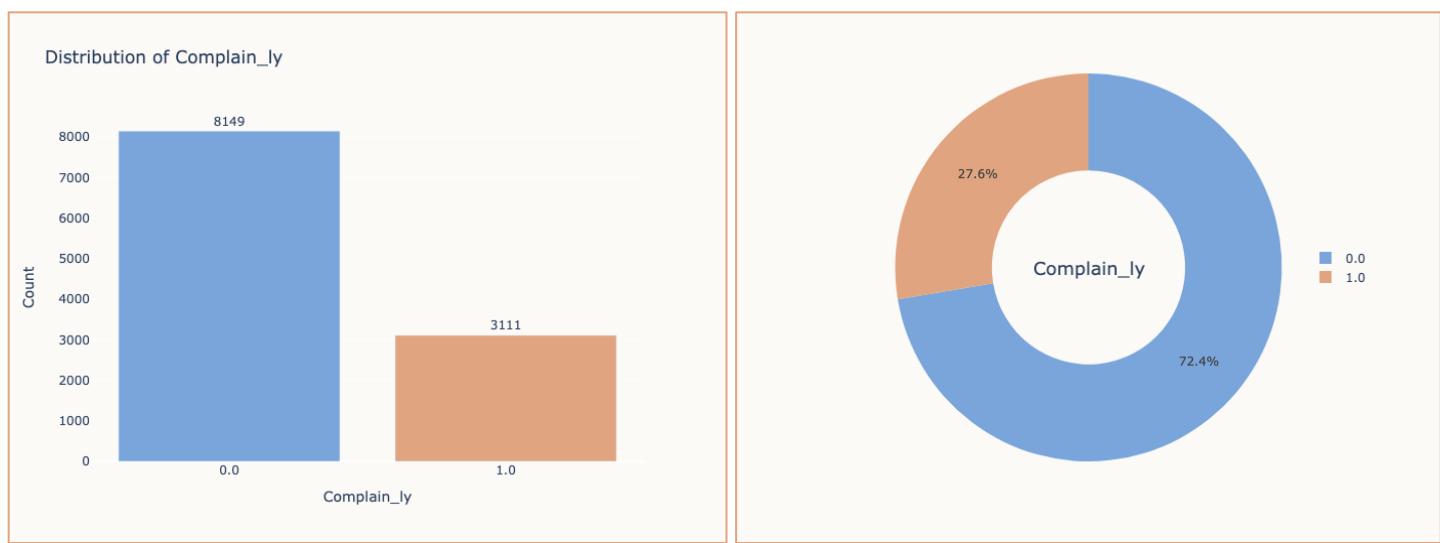
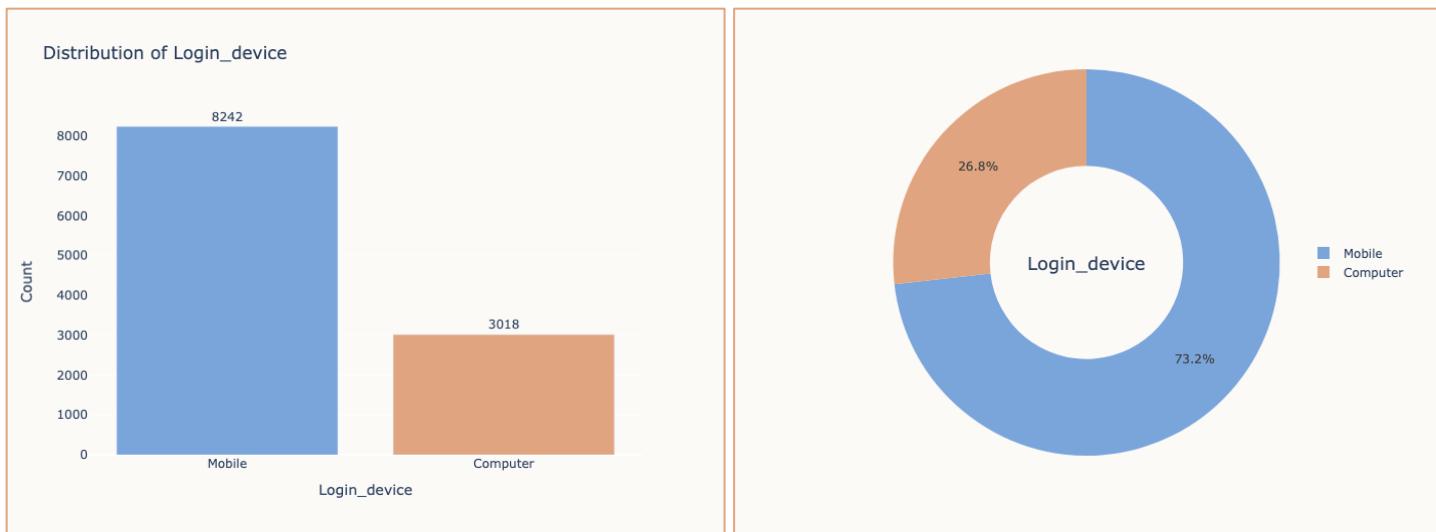


fig 1.17 Univariate Analysis- Complain_LY

- Most customers (**72.37%**) did not file any complaints in the last year which indicates a **good sign of less customers facing issues**.

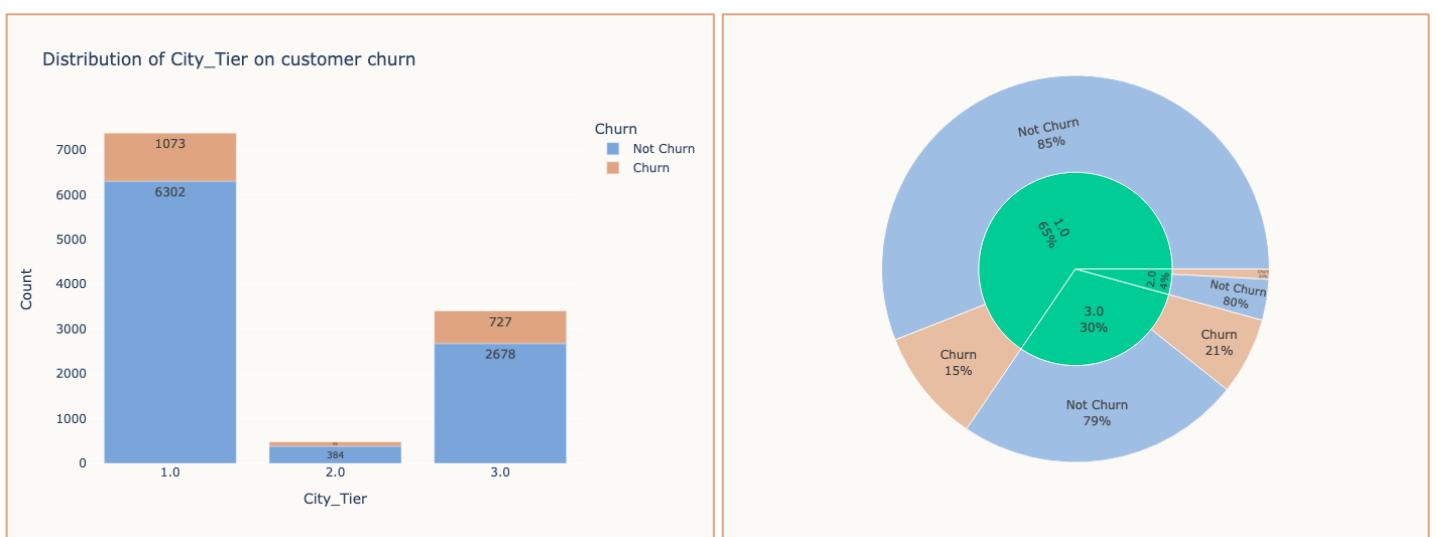


- A **large portion** of customers use **mobile devices (73.20%) for login**.

b) Bivariate analysis

Categorical Columns with Churn

City_Tier:



- City_Tier 1** has the **highest customer base**.
- Churn rate is relatively higher in **City_Tier 3 (21.3%)** compared to other City_Tiers.
- This might be either due to higher cost for tier 3 cities or availability of regional/local contents.

Payment:

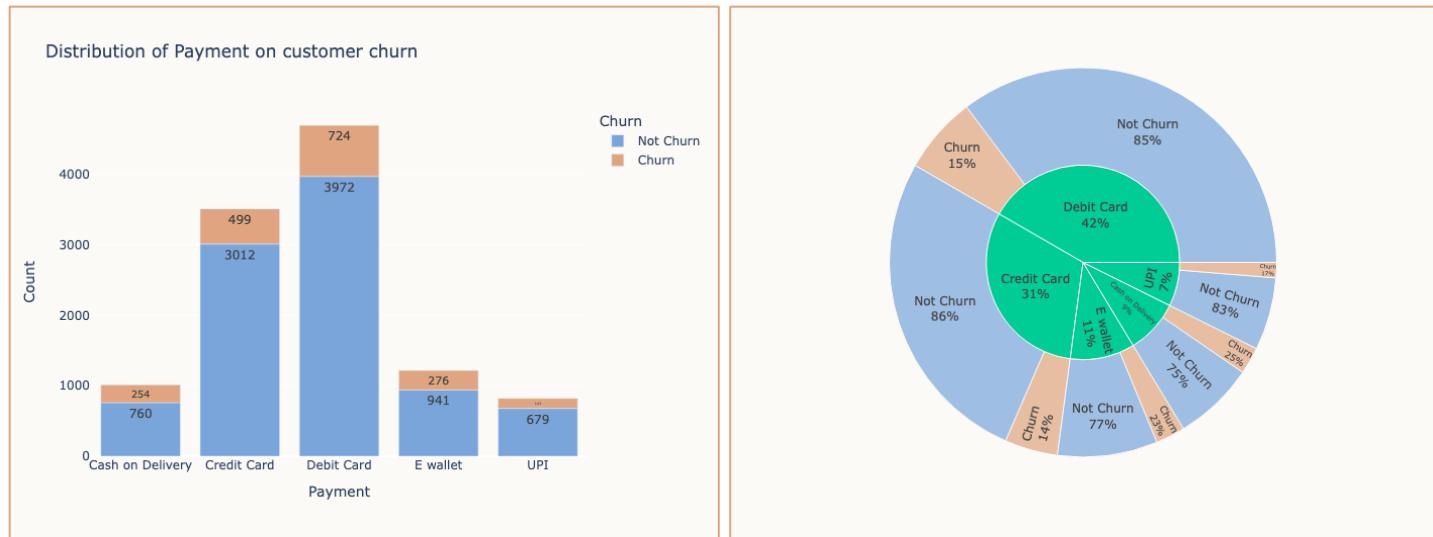


fig 1.20 Bivariate Analysis- Payment with Churn

- Customers who use 'Cash on Delivery' as their payment method have the highest churn rate (25.05%).
- 'Credit Card' and 'Debit Card' users have lower churn rates compared to 'Cash on Delivery' users.

Gender:

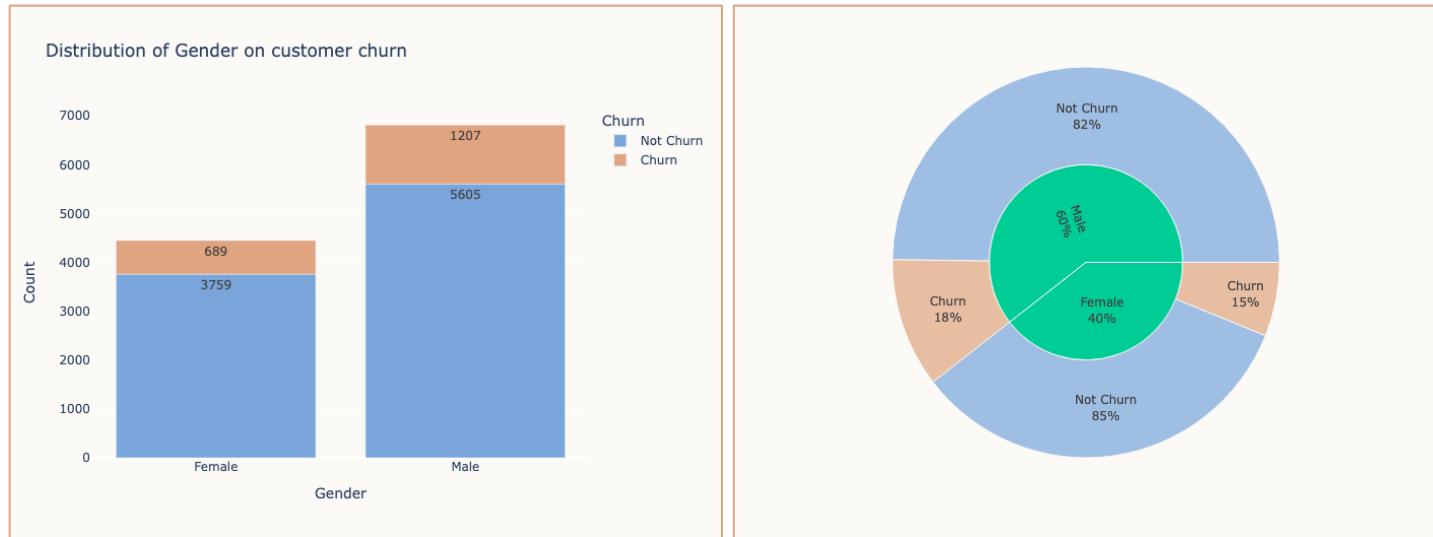


fig 1.21 Bivariate Analysis- Payment with Churn

- Churn is slightly **higher among 'Male'** customers (~18%) compared to '**Female**' customers (15%).

Service_Score

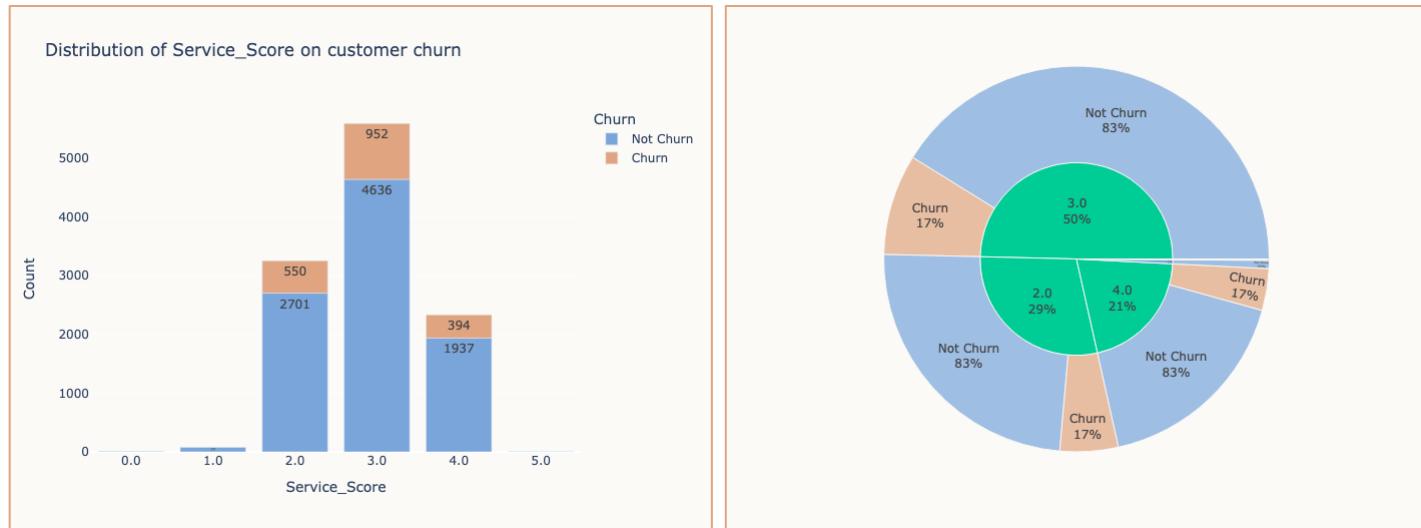


fig 1.22 Bivariate Analysis- Service_Score with Churn

- A significant portion of customers have "**Service_Score** of **3.0 (49.6%)** suggesting customers are relatively **neutral with the services**.
- There are only **5 (0.071%) users** who are **highly satisfied** have 0 churn rate, while customers with score 0 and 1 also have 0% churn.
- On the other hand, customers with a '**Service_Score**' of **2 and 3 & 4** have almost similar **churn rates (~17%)**, indicates that the Churn rate is constant irrespective Service score.

Account_User:

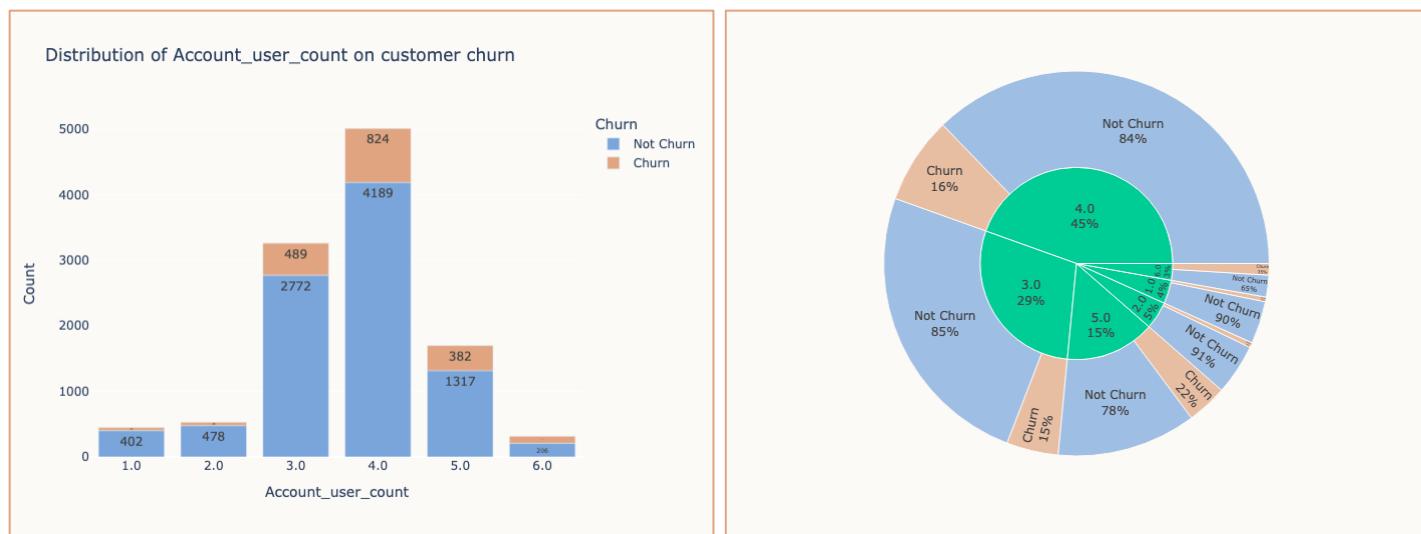


fig 1.23 Bivariate Analysis- Account_user with Churn

- Most customers have 'Account_user_count' of **3 or 4**.
- Churn rate **increases** as '**Account_user_count**' **increases**.
- Customers with 'Account_user_count' of **6** have the **highest churn rate (34%)**, followed by **5 having chrun rate of 22%**.

Account Segment:

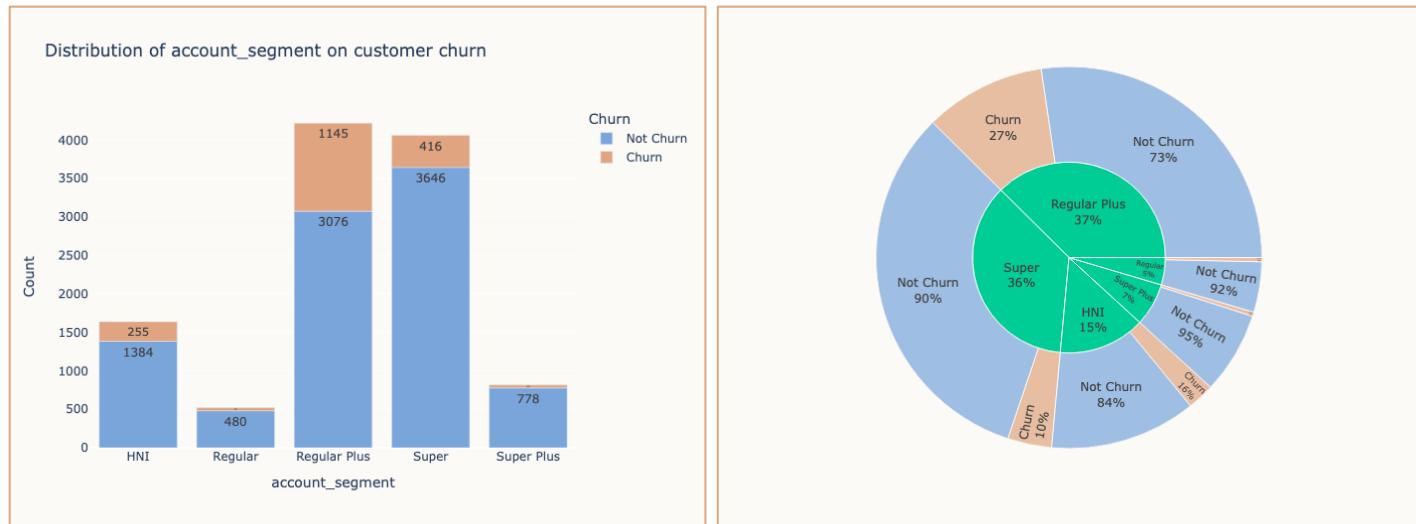


fig 1.24 Bivariate Analysis- Account Segment with Churn

- **'Regular Plus'** account segment has the **highest churn rate (27.13%)**.
- **'Regular', 'Super Plus', 'Super'** segments have the **lowest churn rates**.

CC_Agent_Score:

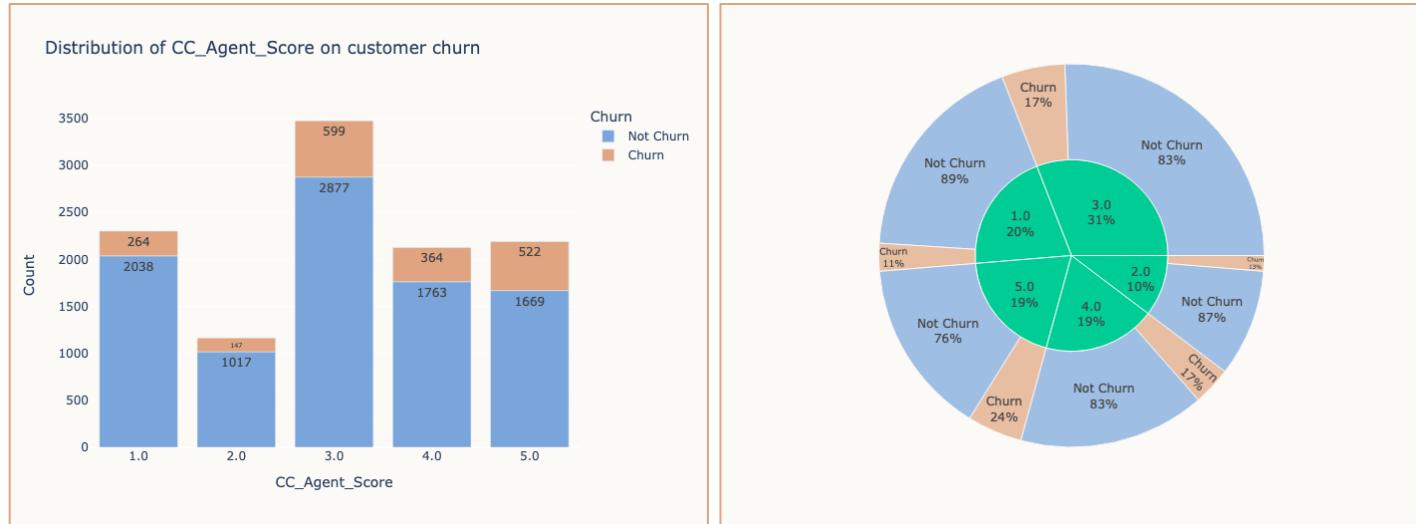


fig 1.25 Bivariate Analysis- CC_Agent_Score with Churn

- Customers with '**CC_Agent_Score**' of **5.0** have the **highest churn rate (24%)**.
- '**CC_Agent_Score**' of **1.0** and **2.0** have the **lowest churn rates (11% and 13% respectively)**.
- **As 'CC_Agent_Score' increases, the churn rate also increases**, which is quite interesting to have a deep dive on why customer with higher Customer Care Agent score tends to churn which ideally should be opposite i.e. with higher CC agent score, less customers to churn.

Marital Status:

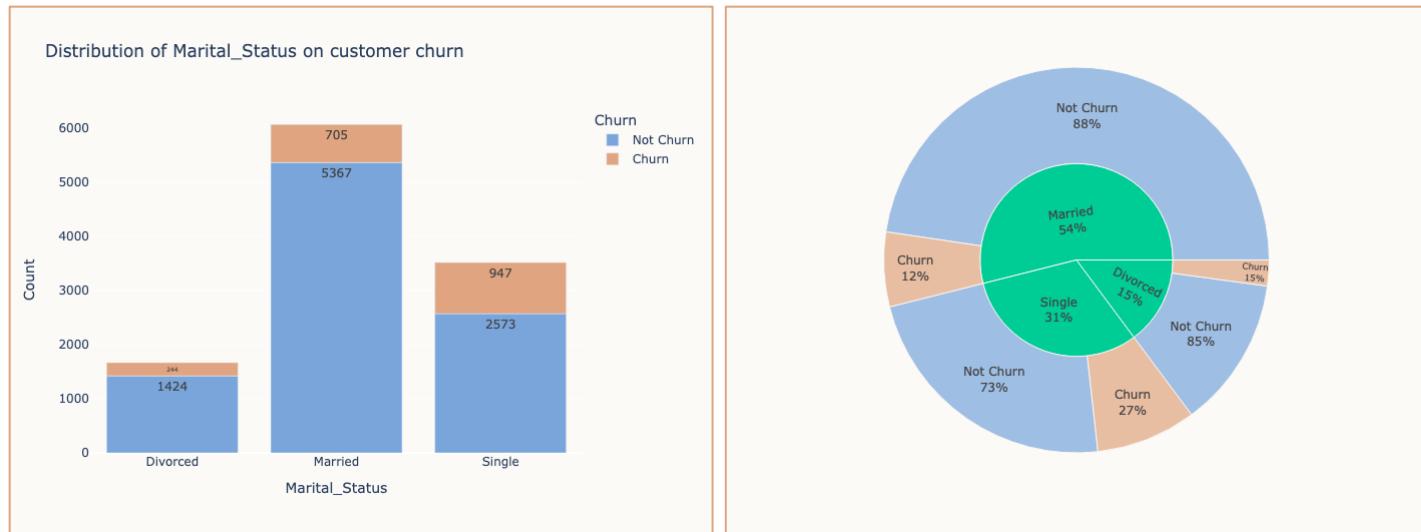


fig 1.26 Bivariate Analysis- Marital Status with Churn

- Customers with '**Single**' marital status have the **highest churn rate (27%)**, which generally represents a younger population, might be due to unavailability of engaging contents or plans for younger/singles.
- '**Married**' customers have a **lower churn rate (12%)**.
- '**Divorced**' customers have a **moderate churn rate (15%)**.

Complain_ly:

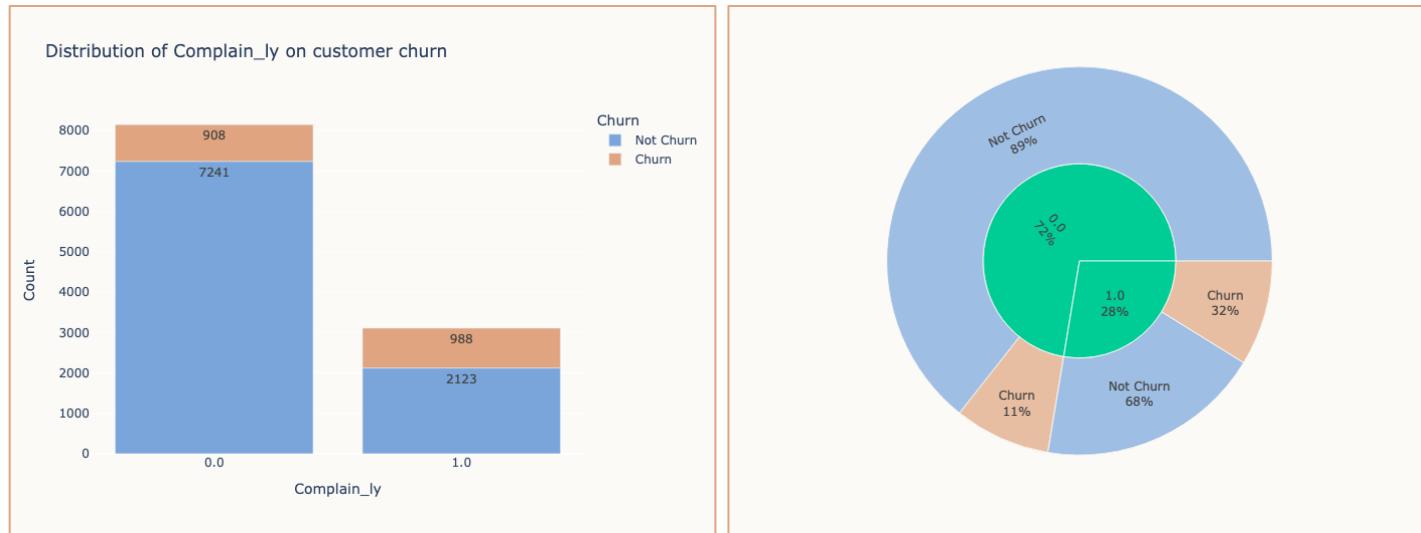


fig 1.27 Bivariate Analysis- Complain_ly with Churn

- Customers who have **complained in the last year** ('Complain_ly' = 1.0) have a significantly **higher churn rate (31.76%)**.
- Customers who **did not complain** ('Complain_ly' = 0.0) have a **lower churn rate (11.14%)**.

- This indicates that the **issues with services leads to higher churn**, hence the complaints to be looked further to know the root cause & eliminate to reduce churn.

Login Device:

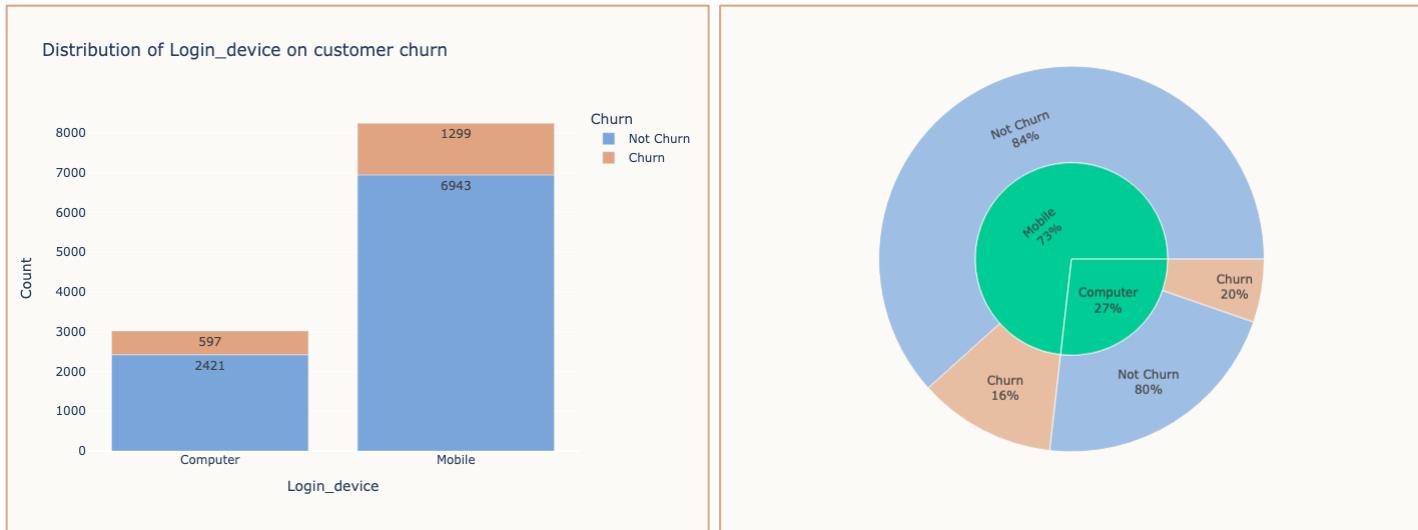


fig 1.28 Bivariate Analysis- Login Device with Churn

- Customers who use '**Computer**' as their **login device** have a **higher churn rate (19.78%)**.
- Customers who use '**Mobile**' devices have a relatively **lower churn rate (15.76%)**.
- This might indicate that users who prefer **login on the go or ease of login** tends to **churn lower**, the **company can promote app/mobile based logins to reduce churn**.

Bivariate Analysis

Distribution of Numerical Columns with Churn

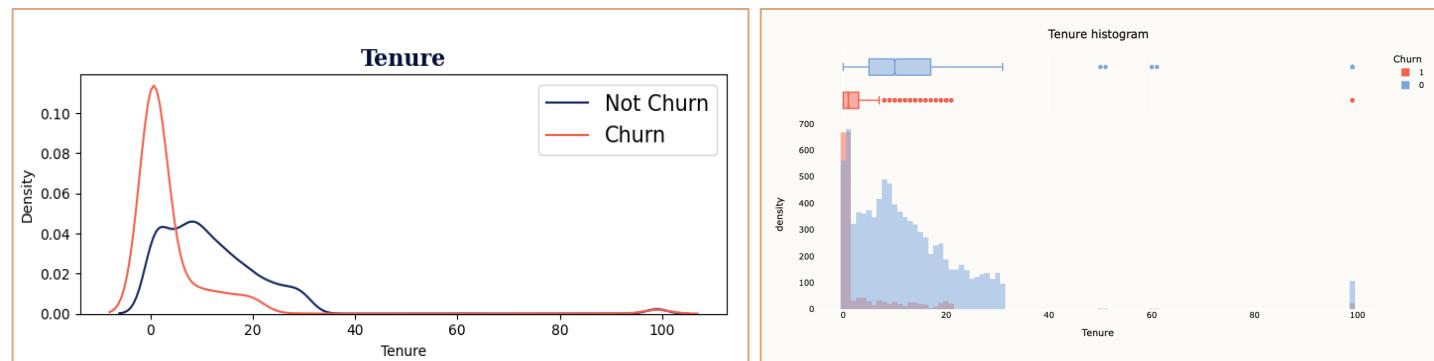


fig 1.29 Bivariate Analysis- Tenure with Churn

Tenure:

- Customers with **lower tenure (recent customers)** have a **higher churn rate**.
- Customers with **longer tenure (loyal customers)** have a **lower churn rate**.

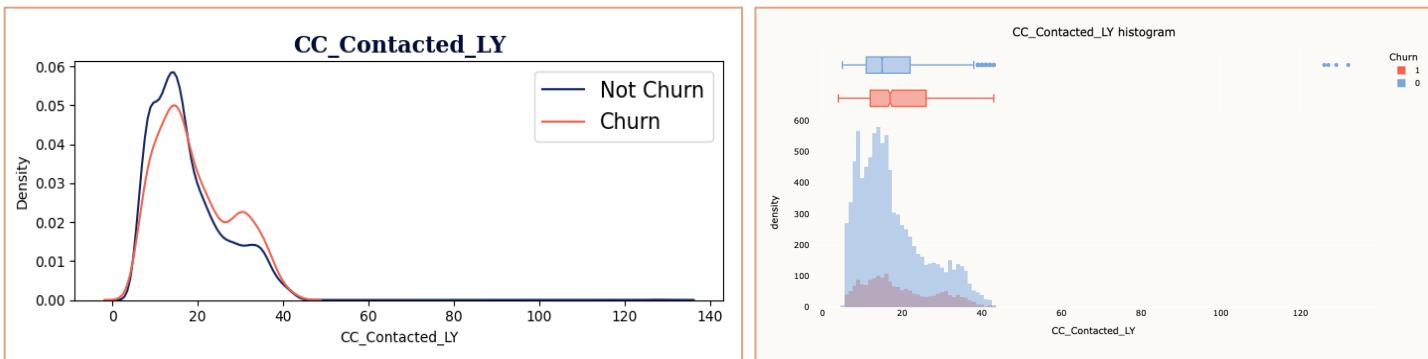


fig 1.30 Bivariate Analysis- CC_Contacted_LY with Churn

CC_Contacted_LY:

- The **median connects** in last year is **higher** for customer who **churned** as compared to those who didn't.

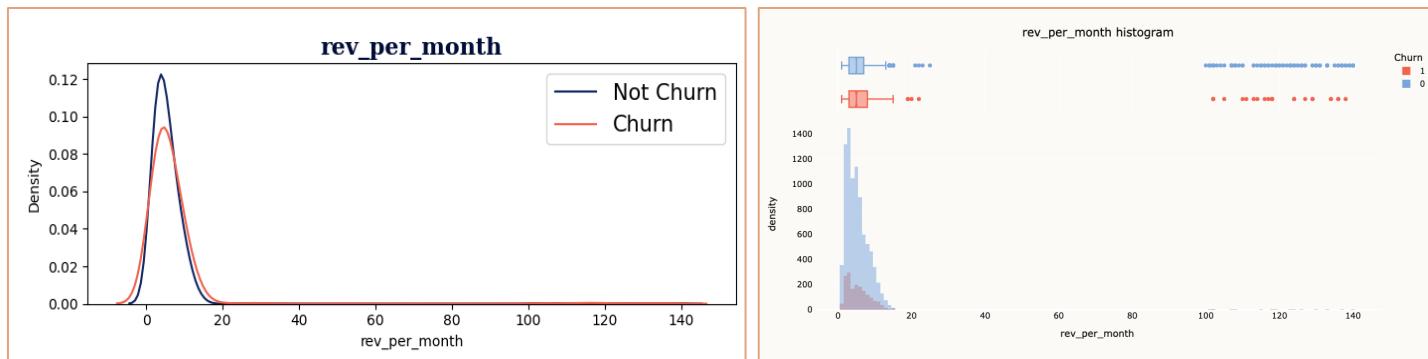


fig 1.31 Bivariate Analysis- rev_per_month with Churn

rev_per_month:

- Customers with **lower monthly revenue** tend to have a **higher churn rate**.
- Customers with **higher monthly revenue** have a **lower churn rate**.

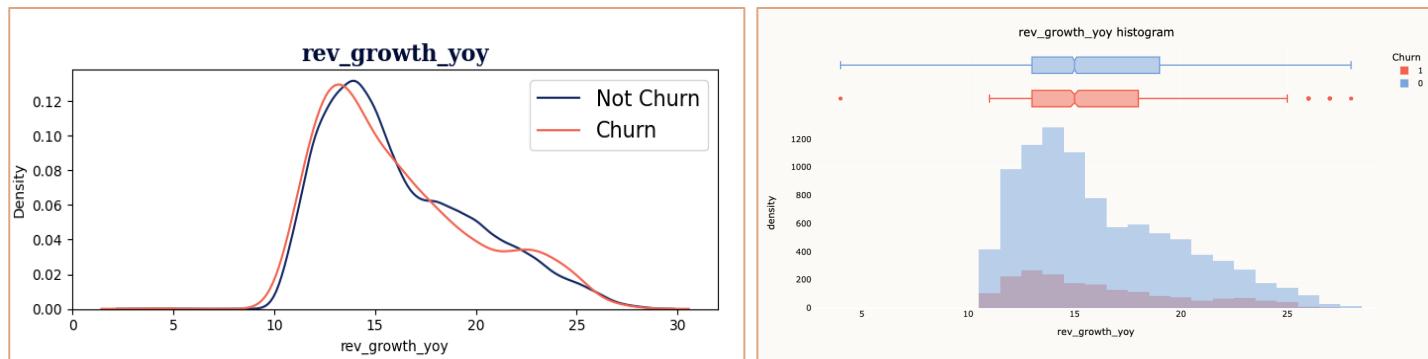


fig 1.32 Bivariate Analysis- rev_growth_yoy with Churn

rev_growth_yoy:

- Customers with **lower year-over-year revenue growth** tend to have a **higher churn rate**.
- Customers with **higher year-over-year revenue growth** have a **lower churn rate**.

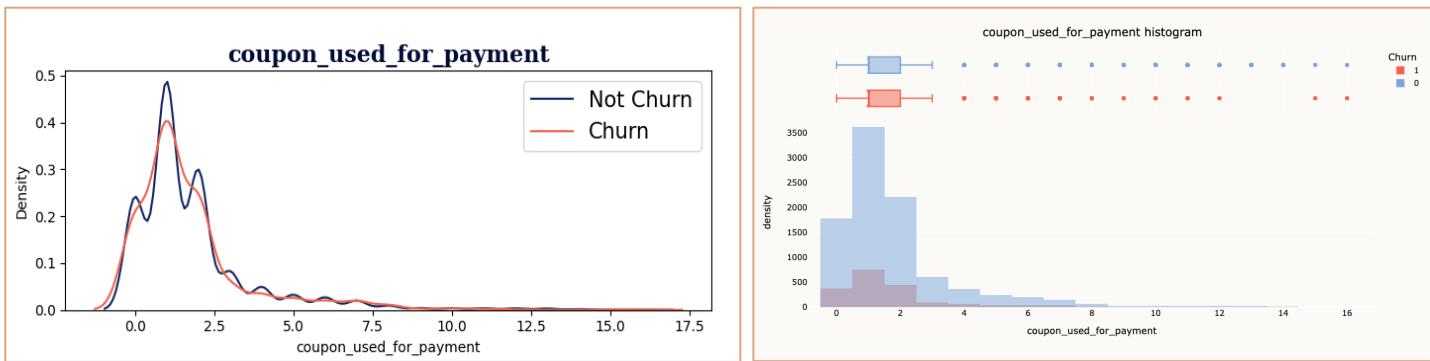


fig 1.33 Bivariate Analysis- coupon_used_for_payment with Churn

coupon_used_for_payment:

- Customers who use **fewer coupons** for payment tend to have a **higher churn rate**.
- Customers who use **more coupons** for payment have a **lower churn rate**.

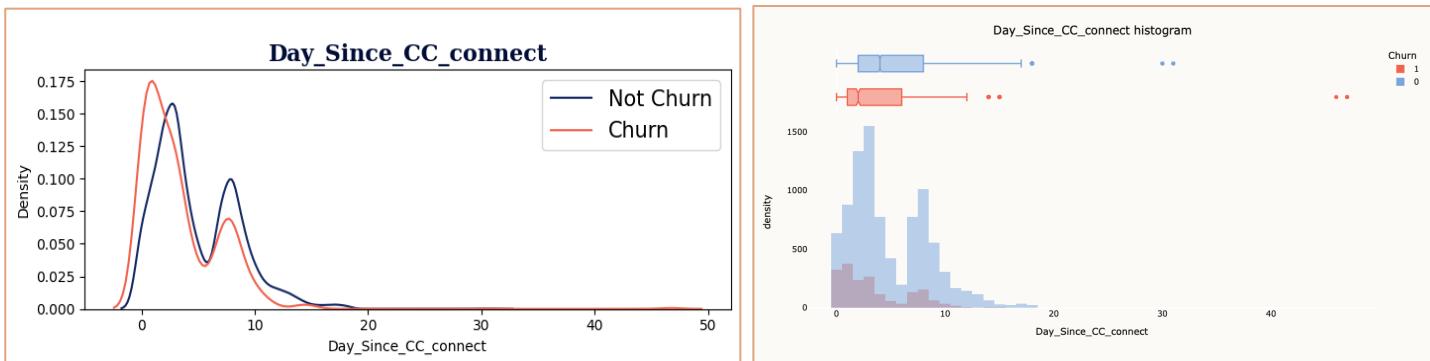


fig 1.34 Bivariate Analysis - Day_Since_CC_connect with Churn

Day_Since_CC_connect:

- Customers with a **shorter time since the last contact** with customer care tend to **have a higher churn rate**.
- Customers with a **longer time since the last contact** with customer care have a **lower churn rate**.

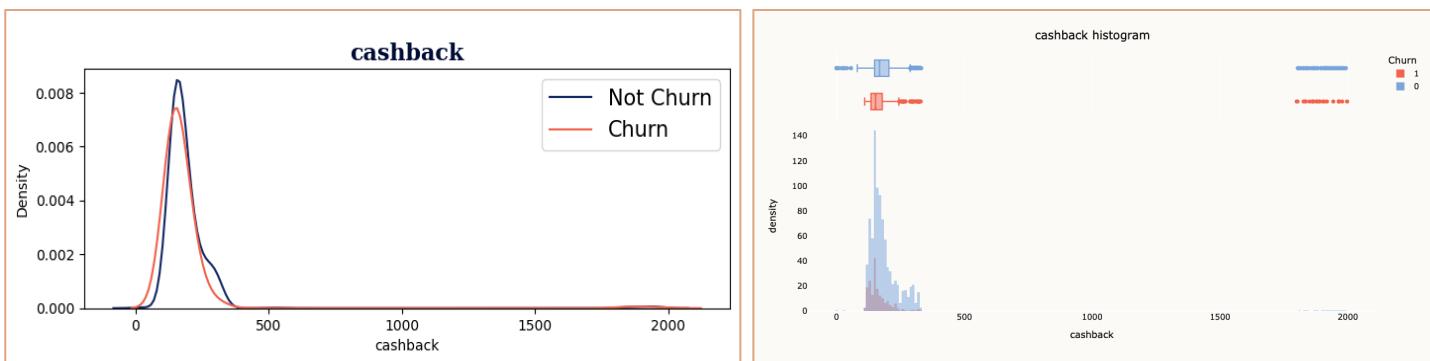


fig 1.35 Bivariate Analysis- cashback with Churn

cashback:

- Customers with **lower cashback amounts** tend to have a **higher churn rate**.
- Customers with **higher cashback amounts** have a **lower churn rate**.

Pairplot

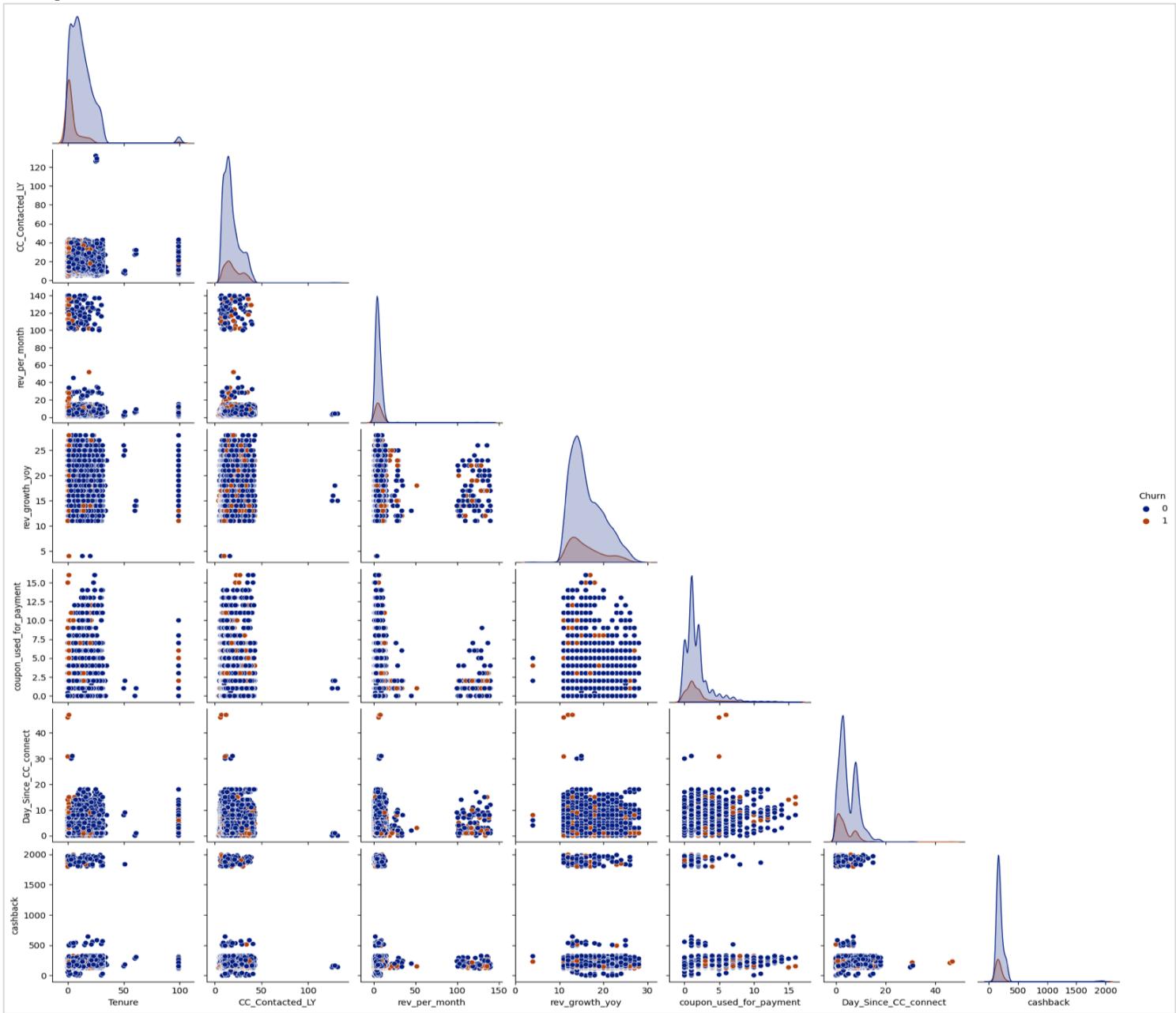


fig 1.36 Pairplot

Correlation Heatmap

Correlation Matrix								
Tenure -	1.00	-0.01	0.03	0.02	0.09	0.12	0.08	-0.23
CC_Contacted_LY -	-0.01	1.00	0.02	0.07	0.01	0.01	0.00	0.07
rev_per_month -	0.03	0.02	1.00	0.03	0.02	-0.00	0.00	0.02
rev_growth_yoy -	0.02	0.07	0.03	1.00	0.02	-0.00	-0.00	-0.01
coupon_used_for_payment -	0.09	0.01	0.02	0.02	1.00	0.36	0.07	-0.01
Day_Since_CC_connect -	0.12	0.01	-0.00	-0.00	0.36	1.00	0.09	-0.15
cashback -	0.08	0.00	0.00	-0.00	0.07	0.09	1.00	-0.03
Churn -	-0.23	0.07	0.02	-0.01	-0.01	-0.15	-0.03	1.00
	Tenure	CC Contacted LY	rev per month	rev growth yoy	coupon used for payment	Day Since CC connect	cashback	Churn

fig 1.37 Correlation Matrix

- Tenure, the time since the last contact with customer care (Day_Since_CC_connect), and cashback appear to have the most significant impact on churn. Customers with longer tenure, recent contact with customer care, and higher cashback are less likely to churn.
- Other variables, such as CC_Contacted_LY, Account_user_count, and usage of coupons for payment, have weaker correlations with churn and may have a less pronounced impact.
- Monthly revenue and year-over-year revenue growth do not have strong correlations with churn, indicating that these factors may not be the primary drivers of customer churn.

1. Positive Relationships:

- Customers with longer tenure tend to generate higher monthly revenue and have been using their credit cards for a longer period.
- There is a strong positive relationship between monthly revenue and cashback, indicating that customers who generate more revenue receive higher cashback.
- In higher city tiers, customers are more likely to use E-wallets for payments.

2. Negative Relationships:

- Longer-tenured customers are less likely to belong to the "Regular Plus" account segment and are slightly less likely to be married.
- Customers in higher city tiers are less likely to be in the "Regular Plus" account segment.
- Customers receiving higher cashback are less likely to be in the "Regular Plus" account segment.

3. Low Relationships:

- Gender does not strongly correlate with most variables, suggesting that it may not significantly impact other customer behaviors.
- Contacting customer care in the last year does not show strong correlations with other variables, indicating that this activity might not strongly influence other aspects of customer behaviour.

a) Removal of unwanted variables

For further analysis, we will drop the column **AccountID** as this is only for unique account identification assigned by system and have no significance in analysis.

b) Missing Value treatment

The total number of missing values are **4361**, which is **2.04%** of total data.

Tenure has **218** missing values, which is **1.94%** of the column.

City_Tier has **112** missing values, which is **0.99%** of the column.

CC_Contacted_LY has **102** missing values, which is **0.91%** of the column.

Payment has **109** missing values, which is **0.97%** of the column.

Gender has **108** missing values, which is **0.96%** of the column.

Service_Score has **98** missing values, which is **0.87%** of the column.

Account_user_count has **444** missing values, which is **3.94%** of the column.

account_segment has **97** missing values, which is **0.86%** of the column.

CC_Agent_Score has **116** missing values, which is **1.03%** of the column.

Marital_Status has **212** missing values, which is **1.88%** of the column.

rev_per_month has **791** missing values, which is **7.02%** of the column.

Complain_ly has **357** missing values, which is **3.17%** of the column.

rev_growth_yoy has **3** missing values, which is **0.03%** of the column.

coupon_used_for_payment has **3** missing values, which is **0.03%** of the column.

Day_Since_CC_connect has **358** missing values, which is **3.18%** of the column.

cashback has **473** missing values, which is **4.20%** of the column.

Login_device has **760** missing values, which is **6.75%** of the column.

Table 1.9 Missing values in dataset

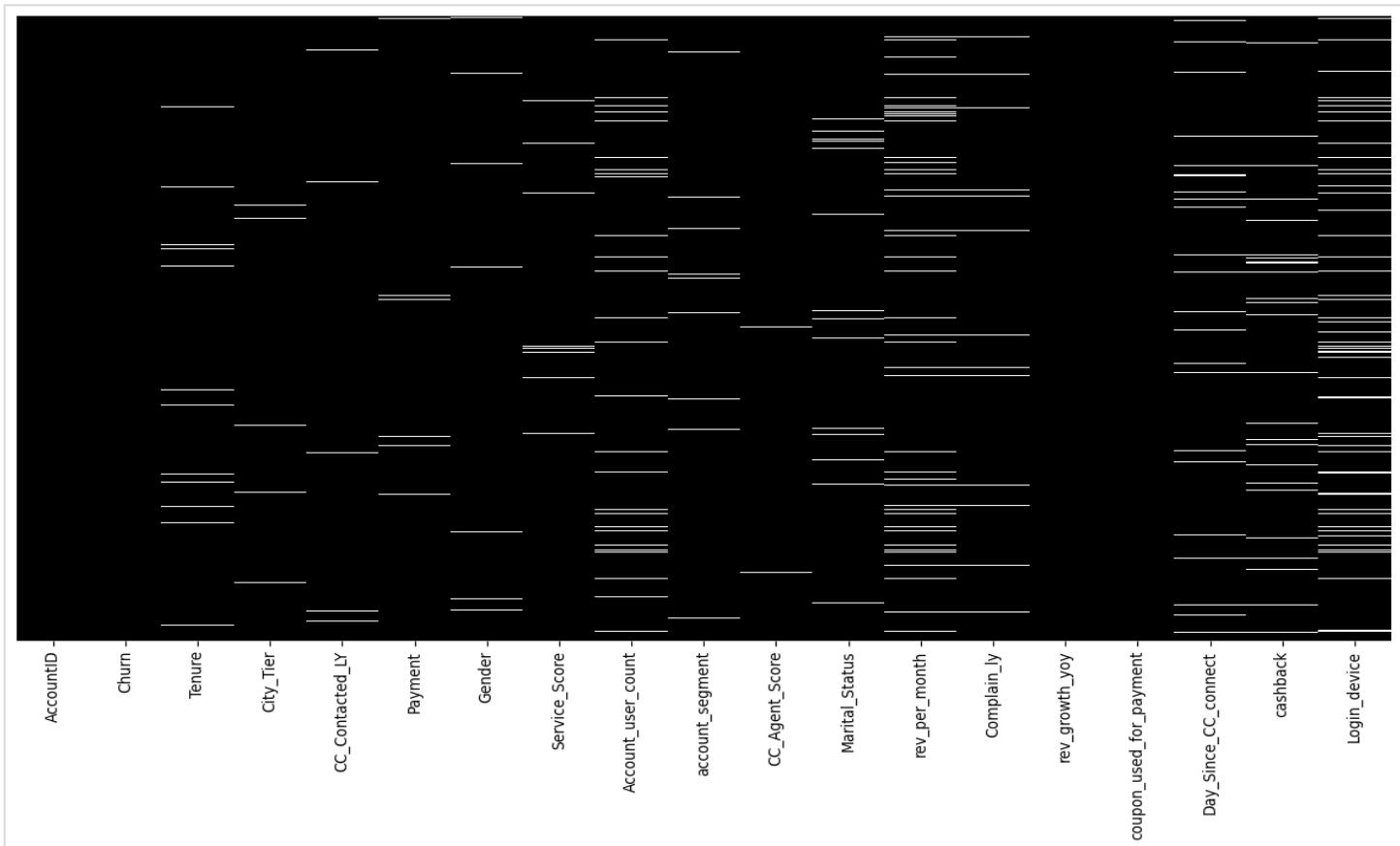


fig 1.38 Missing value Treatment

Imputing Missing Values

- We imputed all missing in categorical using Mode
- For all Numeric, we will use KNN imputer

Data after missing value Treatment:-

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
Churn	0

Table 1.10 Missing values post treatment

d) Outlier treatment

Outliers in the data:

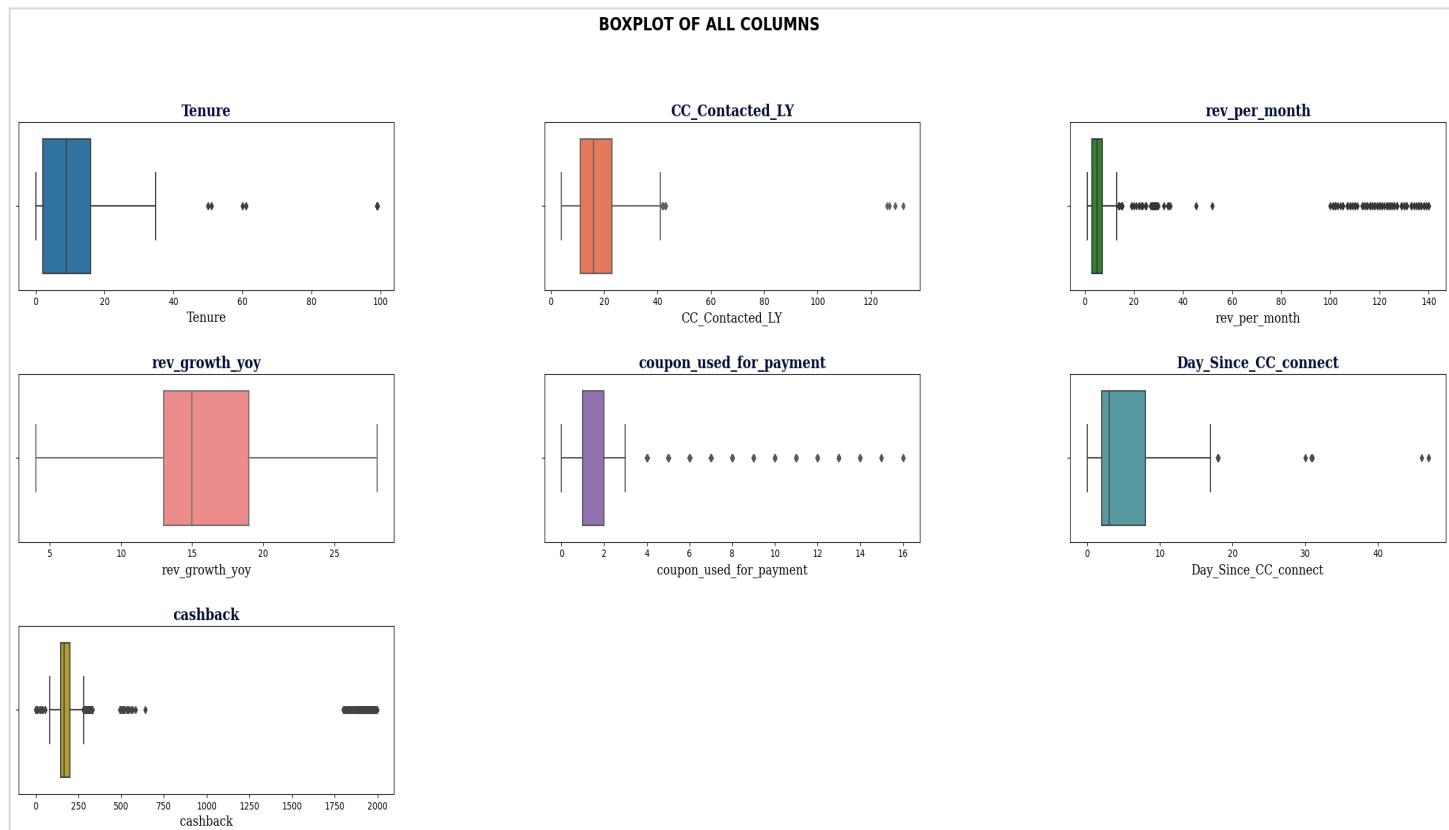


fig 1.39 Outliers in the data

We can see that there are outliers present in the dataset. We have taken the values beyond 1st & 99th %tile as the outliers. Below are the outliers in each variable

- o CC_Contacted_LY - **80**
- o rev_per_month - **113**
- o rev_growth_yoy - **52**
- o coupon_used_for_payment- **98**
- o Day_Since_CC_connect- **94**
- o Cashback- **226**

Treatment Approach: We will cap the outliers to 1st & 99th %tile. Since our data was already scaled & treated, we will do the outlier treatment on scaled data. The result of Outlier treatment on Original as well as on Scaled data is same, hence we are considering scaled data for ease of further analysis in model building

Data post Outlier Treatment

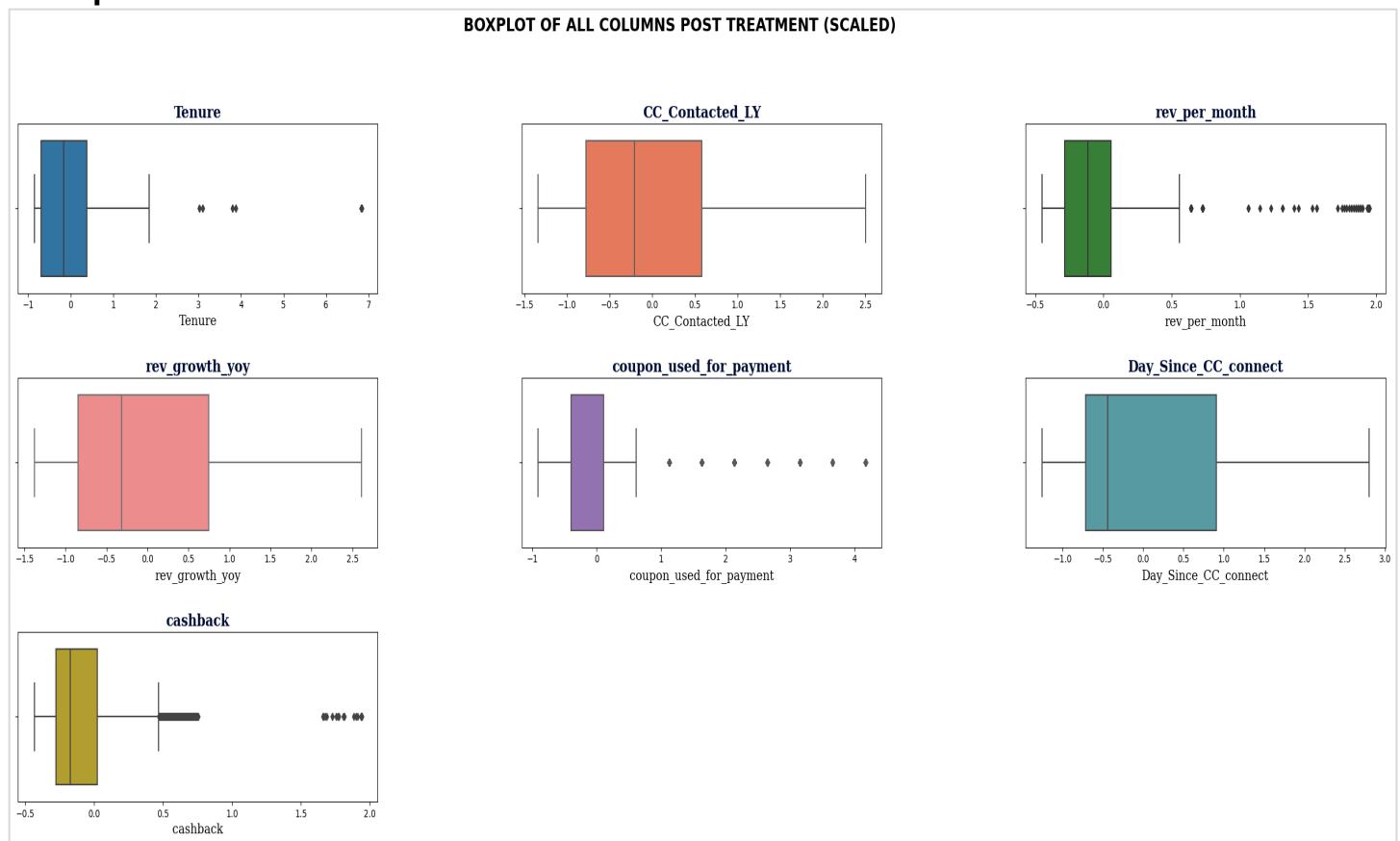


fig 1.40 Outliers after treatment

e) Variable transformation

Datatypes correction:

Categorical columns: rev_per_month", "rev_growth_yoy", "coupon_used_for_payment", "Day_Since_CC_connect ", "Cashback" were transformed to their correct datatypes after fixing of bad data earlier.

Numeric columns: AccountID , Churn, City_Tier, Complain_ly , CC_Agent_Score. Were converted to object since these were Ordinal/Categorical Variables. We dropped AccountID.

Encoding the categorical variables

The categorical variables need to be encoded before feeding into the KNN imputer, Clustering algorithms & machine learning models. We used **Label encoder** to encode all the categorical columns here.

Tenure	City_Tier	CC_Contacted_LY	Payment	Gender	Service_Score	Account_user_count	account_segment	CC_Agent_Score	Marital_Status	r
4.0	2	6.0	2	0	3	2	3	1	2	
0.0	0	8.0	4	1	3	3	2	2	2	
0.0	0	30.0	2	1	2	3	2	2	2	
0.0	2	15.0	2	1	2	3	3	4	2	
0.0	0	12.0	1	1	2	2	2	4	2	

rev_per_month	Complain_ly	rev_growth_yoy	coupon_used_for_payment	Day_Since_CC_connect	cashback	Login_device	
9.0	1	11.0		1.0	5.0	159.929993	1
7.0	1	15.0		0.0	0.0	120.900002	1
6.0	1	14.0		0.0	3.0	NaN	1
8.0	0	23.0		0.0	3.0	134.070007	1
3.0	0	11.0		1.0	3.0	129.600006	1

Table 1.10 Encoded categorical variables

Feature Scaling:

We scaled the numerical variables in the dataset using **Standard Scaler**. The dataset after scaling is:

Tenure	City_Tier	CC_Contacted_LY	Payment	Gender	Service_Score	Account_user_count	account_segment	CC_Agent_Score	Marital_Status	
-0.545460	2	-1.340479	2	0	3	2	3	1	2	
-0.856038	0	-1.114564	4	1	3	3	2	2	2	
-0.856038	0	1.370505	2	1	2	3	2	2	2	
-0.856038	2	-0.323860	2	1	2	3	3	4	2	
-0.856038	0	-0.662733	1	1	2	2	2	4	2	

rev_per_month	Complain_ly	rev_growth_yoy	coupon_used_for_payment	Day_Since_CC_connect	cashback	Login_device	Churn
0.221461	1	-1.382120		-0.401439	0.099207	-0.203224	1 1
0.053522	1	-0.317598		-0.909191	-1.253070	-0.421693	1 1
-0.030447	1	-0.583728		-0.909191	-0.441704	NaN	1 1
0.137492	0	1.811447		-0.909191	-0.441704	-0.347974	1 1
-0.282355	0	-1.382120		-0.401439	-0.441704	-0.372995	1 1

Table 1.11 Scaled variables

f) Addition of new variables

- **Clusters:** Variable will be formed after the end of clustering process.
- No other new variable was added in the dataset

4. BUSINESS INSIGHTS FROM EDA

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

From the below target variable plot, we can clearly see the **class Imbalance here**. only 16.8% of the customers are churned.

Churn rates vary significantly across different categorical variables, with some categories having higher churn rates than others. For instance, certain city tiers, payment methods, gender, account segments, and marital status categories have notably higher churn rates compared to their counterparts.

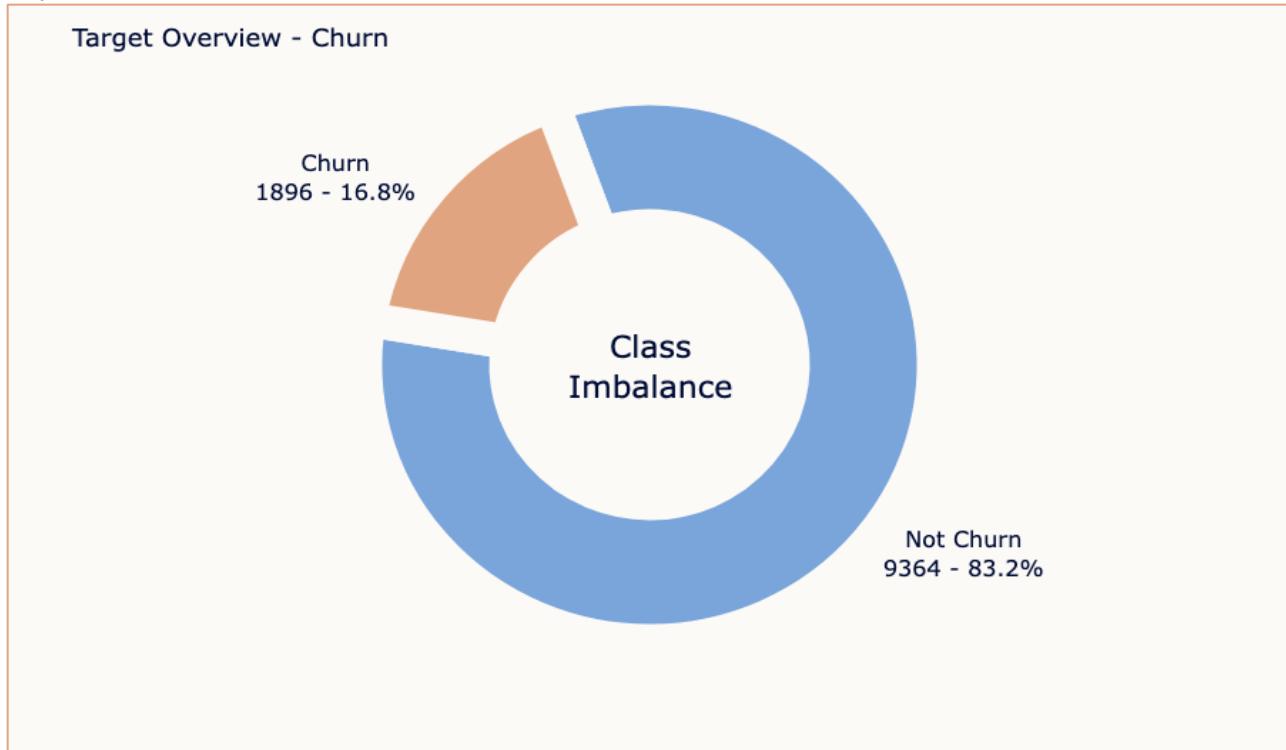


fig 1.41 Target Class imbalance

To address this issue in the context of the DTH service provider, the following actions can be taken:

Collect More Data: Collecting more data, especially from the minority class, can help improve the balance in the dataset. Engaging with more customers who have churned to gather their feedback and understanding their pain points can be valuable.

Resampling: If more data is unavailable, to balance the dataset, we can employ resampling techniques such as SMOTE. This can involve oversampling the minority class (Churn) by generating

synthetic samples or under sampling the majority class (Not Churn). Both methods can help balance the class distribution and improve model performance.

b) Any business insights using clustering

KMeans Clustering Approach:

We used k-means clustering which is an unsupervised clustering technique for segmenting the similar type of matches in k numbers of clusters.

We checked **2 methods** to find the optimum clusters.

Method 1: First, we scaled the data & applied PCA with 3 components wherein we got the k clusters as 4 using elbow visualizer plot. However, since we only took 3 principle components from PCA that didn't capture much variance hence we considered it as not sufficient to proceed further & did not go ahead with it.

Method 2: In this, we took the scaled data plotted the distortion score Elbow and got k=5.

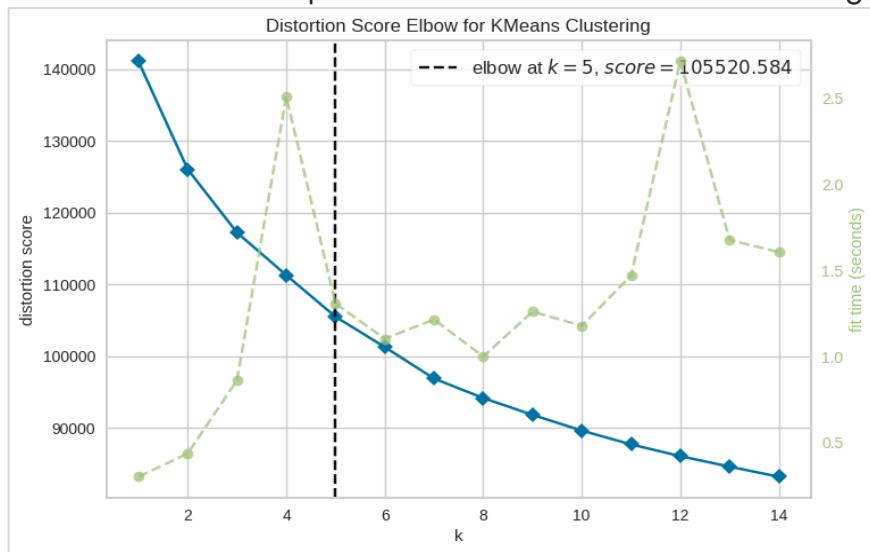


fig 1.42 distortion Score Elbow for KMeans Clustering

Let's confirm value of k among these using silhouette score. We'll check on 2 to 8

```

k = 2
Silhouette Score = 0.13459603536413353
Minimum Sil Width = 0.012092665037392987

k = 3
Silhouette Score = 0.1274244057273217
Minimum Sil Width = -0.0608394598976085

k = 4
Silhouette Score = 0.10234412335716576
Minimum Sil Width = -0.0823392577919645

k = 5
Silhouette Score = 0.09681396366813581
Minimum Sil Width = -0.05970787597370444

k = 6
Silhouette Score = 0.11060016505956112
Minimum Sil Width = -0.11400279732332869

k = 7
Silhouette Score = 0.09244997648418969
Minimum Sil Width = -0.12633030221423378

k = 8
Silhouette Score = 0.09766469426462024
Minimum Sil Width = -0.12475225167650357

```

Table 1.12 silhouette score

- **Silhouette Score close to 1** => Clusters are well separated from each other.

- **Minimum Sil-Width = +ve** => No Data points are mapped incorrectly to their cluster

In this case, the other clusters don't well differentiate because 2 groups out of a set of data points are obvious.

For k = 2, we have the highest **Silhouette Score (0.134)** among the options. The **Minimum Silhouette Width** is also **positive (0.012)**, indicating that there is a reasonable separation between the clusters.

Taking 6 Clusters basis previous elbow method shows that increasing k post 6 found the point where adding more clusters provides diminishing returns in terms of reducing the inertia. However, having 6 cluster might become difficult in interpretability and practical implications of the resulting clusters.

We have gone ahead with k=2 for easier interpretation

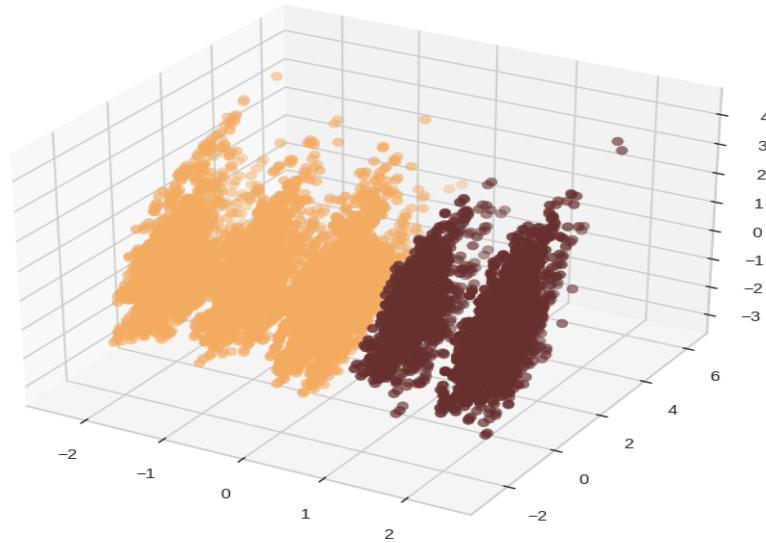


fig 1.43 Plot of the clusters

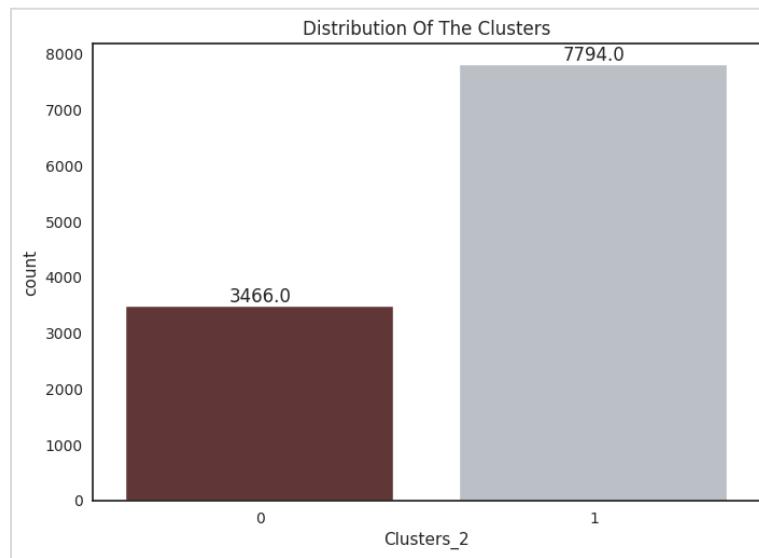


fig 1.44 Cluster distribution

Clusters_2	0	1
Tenure	11.23	10.91
City_Tier	0.68	0.63
CC_Contacted_LY	17.68	17.88
Payment	1.81	1.74
Gender	0.63	0.59
Service_Score	2.88	2.91
Account_user_count	2.73	2.69
account_segment	2.20	2.16
CC_Agent_Score	0.34	2.84
Marital_Status	1.18	1.16
rev_per_month	5.27	5.59
Complain_ly	0.30	0.26
rev_growth_yoy	16.41	16.09
coupon_used_for_payment	1.73	1.78
Day_Since_CC_connect	4.56	4.61
cashback	183.69	183.05
Login_device	0.73	0.73
freq	3466.00	7794.00

Table 1.13 Cluster Profiles

Cluster 0:

- Customers in Cluster 0 have a slightly longer tenure of approximately 11.23 months, indicating loyal relationships.
- They exhibit moderate service satisfaction with a service score of 2.88 but generate significant monthly revenue of approximately 5.27 units.
- Cluster 0 has lower customer care agent scores (0.34) and a higher percentage of married customers (approximately 1.18), which suggests an area for improvement.
- These customers file relatively fewer complaints, with a complaint rate of about 0.30, indicating overall satisfaction.

Cluster 1:

- Customers in Cluster 1 have a slightly shorter tenure of approximately 10.91 months but higher customer care agent scores (2.84).
- They generate slightly higher monthly revenue (approximately 5.59 units) and have a lower number of complaints (around 0.26).
- The majority of customers in this cluster are male (approximately 0.63) and unmarried (around 1.16).
- Cluster 1 represents a larger customer base with significantly more frequent interactions (7794 times).

Insights from Clusters

- **Tenure Variation:** The average tenure of customers in both clusters is quite close, with Cluster 0 having slightly higher tenure compared to Cluster 1.
- **Contact Frequency:** Cluster 1 has a slightly higher average contact frequency in the last year compared to Cluster 0, suggesting that Cluster 1 customers may have more interactions with the company.
- **Gender Balance:** Both clusters have a balanced distribution of gender, and there is no gender bias in either group.
- **Service Satisfaction:** Cluster 1 has a slightly higher average service score, indicating higher levels of satisfaction with the service offered by the company.
- **Account User Count:** Customers in Cluster 0 have a slightly higher average account user count compared to Cluster 1.
- **CC Agent Score:** Cluster 1 stands out with a significantly higher average agent score compared to Cluster 0, indicating more positive interactions with customer care agents.
- **Revenue per Month:** Cluster 1 exhibits a slightly higher average monthly revenue per customer compared to Cluster 0, indicating that Cluster 1 customers may be more profitable.
- **Interaction Frequency:** Cluster 1 has a significantly higher average interaction frequency compared to Cluster 0, suggesting that these customers are more engaged and active.

Cluster 1 may represent the **company's most loyal and engaged customers**, while **Cluster 0** could be **targeted for improvements in services and addressing complaints**.

c) Any other business insights

Executive Summary

1. Data Collection and Pre-processing:

- The dataset appears to be derived from a DTH (Direct-To-Home) service provider, containing records over an unspecified period, possibly in Indian states with payments in Rupees via UPI.
- Data pre-processing was performed to address missing values, correct data types, and standardize data. Anomalies and unexpected values were resolved.

2. Exploratory Data Analysis:

- The dataset consists of **11,260 rows** and **19 columns** with a **binary target variable, "Churn,"** representing **customer attrition**.
- Key findings include **class imbalance** with **only 16.8%** of customers **marked as churned**.
- **Numerical features** exhibit **varying distributions**, such as **positively skewed tenure**, **moderate-skewed revenue**, and **highly skewed cashback**.
- **Categorical variables** like "**City_Tier**" show that the **majority of customers are in Tier 1 cities**, and **payment methods like "Debit Card" and "Credit Card" are popular**.
- Analysis of categorical variables indicated relationships between features and churn, highlighting the impact of "**City_Tier**," "**Payment**," and "**Complain_ly**" on customer attrition.

- In numerical features, **lower tenure, higher CC contacts in the last year, lower monthly revenue, and higher cashback amounts** were associated with **higher churn rates**.

3. Data Preparation:

- Unwanted variable **Account id was removed**
- **Missing values** were **imputed using Mode for categorical & KNN for numerical**, with **2.04%** of the data containing missing values.
- **Outliers were treated** by **capping** values at the **1st and 99th percentiles**.
- **Label Encoding** was done **for categorical** and **Standard scaling** was applied to the **numerical variables** in the dataset.

Business Insights:

1. Churn Analysis:

- The **class imbalance**, with only **16.8% churned** customers, indicates the **need to focus on retaining existing customers** to maintain business stability.

2. Demographics and Churn:

- Customers from **City_Tier 3** have a **higher churn rate**, suggesting that **tailored offerings or incentives** may be **needed to retain** these customers.
- **Churn rates** differ based on **payment methods** and **gender**. Efforts should be made to engage male customers more effectively.

3. Service and Customer Care:

- The "**Service_Score**" analysis shows that higher scores don't necessarily lead to lower churn. **Service quality may need improvement**.
- **Customers who complained in the last year** had a **significantly higher churn rate**, indicating a **need to address their issues promptly**.

4. Engagement and Loyalty:

- **Increasing the number of account users** leads to a **higher churn rate**. **Strategies** for **retaining multi-user accounts** should be explored.
- **Encouraging customers to use mobile devices for logins** may reduce churn, as **mobile** users exhibit a **lower attrition rate**.

Recommendations:

- **Balancing Churn Rate:** Implement strategies to retain existing customers, including offering personalized deals and promotions.
- **Customized Service Quality:** Enhance services for specific segments, like "**Regular Plus**" customers, who have a **higher churn rate**. **Address service quality issues** identified in the data.
- **Prompt Complaint Resolution:** Improve customer care and promptly **address customer complaints** to prevent churn.
- **Multi-User Account Strategies:** Create incentives or tailored offerings for multi-user accounts to reduce attrition.
- **Mobile Login Promotion:** Promote mobile login to lower churn by making it **more convenient** for customers.

PROJECT NOTES -II

MODEL BUILDING

APPROACH

The dataset post pre-processing contains 11260 rows and 18 columns.

Tenure	City_Tier	CC_Contacted_LY	Payment	Gender	Service_Score	Account_user_count	account_segment	CC_Agent_Score	Marital_Status	rev_per_month	Complain_ly	rev_growth_yoy	coupon_used_for_payment	Day_Since_CC_connect	cashback	Login_device	Churn	
0	-0.545460	2.0	-1.340479	2.0	0.0	3.0	2.0	3.0	1.0	2.0	0.221461	1.0	-1.382120	-0.401439	0.099207	-0.203224	1.0	1.0
1	-0.856038	0.0	-1.114564	4.0	1.0	3.0	3.0	2.0	2.0	2.0	0.053522	1.0	-0.317598	-0.909191	-1.253070	-0.421693	1.0	1.0
2	-0.856038	0.0	1.370505	2.0	1.0	2.0	3.0	2.0	2.0	2.0	-0.030447	1.0	-0.583728	-0.909191	-0.441704	-0.381940	1.0	1.0
3	-0.856038	2.0	-0.323860	2.0	1.0	2.0	3.0	3.0	4.0	2.0	0.137492	0.0	1.811447	-0.909191	-0.441704	-0.347974	1.0	1.0
4	-0.856038	0.0	-0.662733	1.0	1.0	2.0	2.0	2.0	4.0	2.0	-0.282355	0.0	-1.382120	-0.401439	-0.441704	-0.372995	1.0	1.0

Table I.14 Sample of the dataset

Step 1: Segregate dependent variable “Churn” from the independent variables.

We assigned all predictors to X & response on y variable before split.

Step 2: Split the data into train & Test set

We split the data with 20% of the records going to the test set & 80% to train, random state of 42 was selected. Dataset shape post-split for predictors.

Shape of Train dataset: (9008, 17)

Shape of Test dataset: (2252, 17)

Step 3: Model(s) Building

Model Selection

In total **14 models** were built and their performance was evaluated to find the best fitting model. We built the following models on the dataset & evaluate the performance:

1. Decision Tree
2. Random Forest
3. Linear Discriminant Analysis (LDA)
4. k-Nearest Neighbours(KNN)
5. Naive Bayes
6. Neural Network (ANN)
7. SVM

Of these models, the models listed below were **ensembled and hyperparameter tuned** as well to obtain the optimum model.

1. Decision Tree
2. Linear Discriminant Analysis (LDA)
3. Random Forest
4. Bagging (base- Decision Tree)
5. Adaboost (base- Decision Tree)
6. Extreme Gradient Boost (XGB)
7. Categorical Boost(CatBoost)

8. Neural Network (ANN)
9. SVM

Step 4: Selection of appropriate Evaluation Matrix

In a churn prediction scenarios wherein the dataset is also imbalanced with 17% Churn cases, the **primary goal is to identify potential churners (Churn cases)**. Missing even a small fraction of true churners can be costly for the business, as retaining these customers is often more cost-effective than acquiring new ones. Therefore, **prioritizing recall**, or other metrics that consider true positive rates, is a suitable approach for your imbalanced churn prediction dataset

This ensures that your model is effective in capturing Churn cases, which aligns with the business objective of reducing customer attrition. However, it's also essential to consider **other metrics like precision, F1 score, and AUC-ROC in combination with recall** to get a comprehensive evaluation of the model's performance.

Step 4: Model Tuning

A combination of ensemble methods and hyperparameter tuning can lead to highly accurate and robust machine learning models.

Ensemble Methods:

Ensemble methods involve combining multiple machine learning models to improve overall predictive performance. The key idea is to leverage the strengths of different models and overcome their individual weaknesses

Hyperparameter Tuning:

Grid Search, also known as parameter sweeping, is one of the most basic and traditional methods of hyperparametric optimization.

Hyperparameters are parameters that are not learned from the data but need to be set before training a machine learning model. Hyperparameter tuning is the process of finding the best combination of hyperparameters to optimize a model's performance.

Data Imbalance

While the **dataset is imbalanced we have not oversampled the minority class to avoid synthetic data**, since the models were able to perform fairly good without need to introducing oversampled data. However, this is also one way to tune the model for better results when data is imbalanced.

Step 5: Model(s) Comparison & Optimum Model.

Let's go through all the models one by one

Decision Tree Model

A decision tree, as the term suggests, uses a tree-like model to make predictions.

- A decision tree splits data into multiple sets of data. Each of these sets is then further split into subsets to arrive at a decision
- It can be used to model highly non-linear data.
- Decision trees are non-parametric means no specific data distribution is necessary. Decision trees easily handle continuous and categorical variables.

A criterion of Gini is selected

Train data Classification Report					Test data Classification Report				
	precision	recall	f1-score	support		precision	recall	f1-score	support
0.0	1.00	1.00	1.00	7508	0.0	0.96	0.97	0.97	1856
1.0	1.00	1.00	1.00	1500	1.0	0.84	0.84	0.84	396
accuracy			1.00	9008	accuracy			0.94	2252
macro avg	1.00	1.00	1.00	9008	macro avg	0.90	0.90	0.90	2252
weighted avg	1.00	1.00	1.00	9008	weighted avg	0.94	0.94	0.94	2252

Table 1.15 Classification report Decision Tree

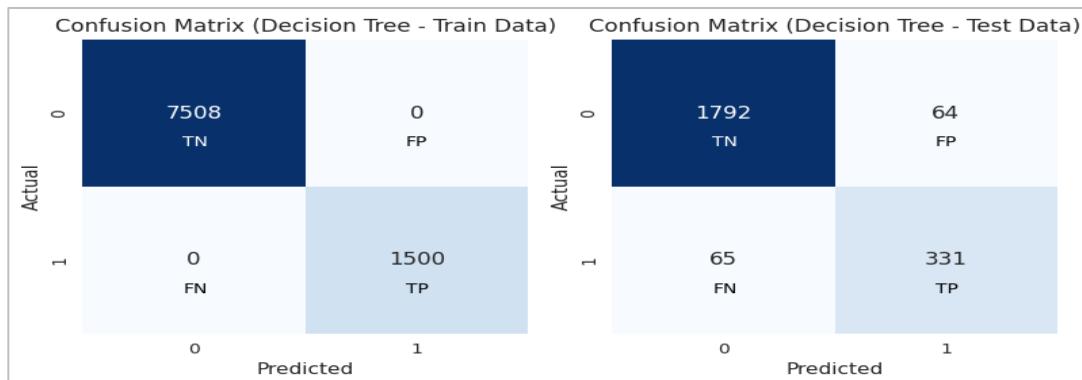


Fig 1.46 Confusion Matrix Decision Tree

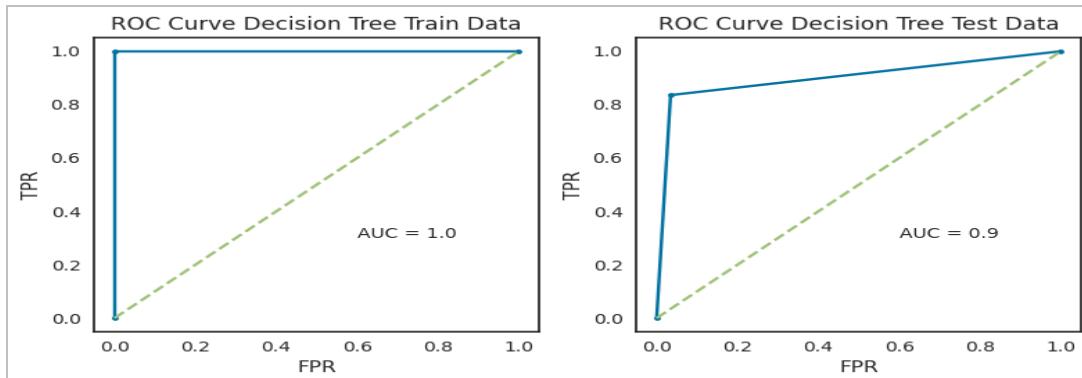


Fig 1.47 ROC-AUC Curve Decision Tree

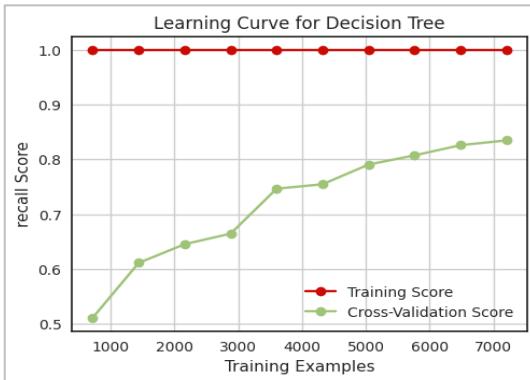


Fig 1.48 Learning Curve Decision Tree

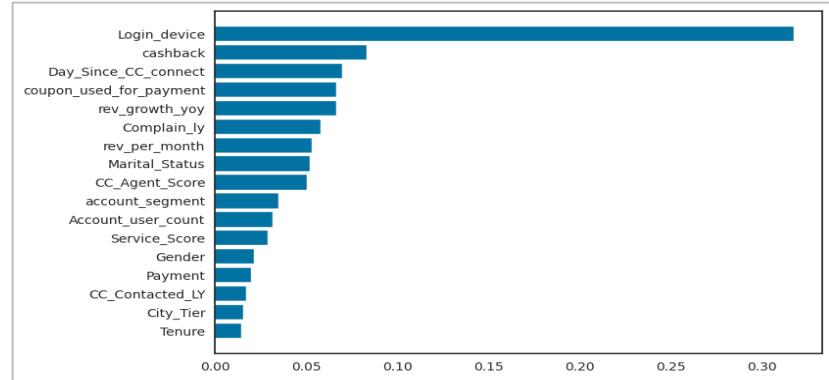


Fig 1.49 Feature importance Decision Tree

Insights

- The model is able to identify a fair proportion of the actual positive cases (**84% recall for the minority class**). This is important for applications where it is critical to avoid false negatives.
- The **model is overfitting the training data**. This is evident from the relatively large difference between the training accuracy and the cross-validation score.
- The **model may not generalize well to new data**. This is evident from the plateau in the learning curve.
- Regularization can help to prevent the model from overfitting the training data. Or Pruning the tree can help to reduce the complexity of the tree and prevent overfitting

Random Forest Model

Random Forest is a powerful ensemble learning technique, often regarded as an extension of decision trees. It operates through the following steps:-

- **Random Sampling:** Random Forest selects random subsets of data from the given dataset.
- **Decision Trees:** For each of these data subsets, it constructs a decision tree, making predictions on each subset.
- **Voting:** Random Forest then aggregates the predictions from all decision trees through a voting mechanism.
- **Majority Rule:** The prediction result with the highest number of votes becomes the final prediction.

Train data Classification Report					Test data Classification Report					
	precision	recall	f1-score	support		precision	recall	f1-score	support	
0.0	1.00	1.00	1.00	7508		0.0	0.98	0.99	0.98	1856
1.0	1.00	1.00	1.00	1500		1.0	0.96	0.89	0.92	396
accuracy			1.00	9008	accuracy			0.97	2252	
macro avg	1.00	1.00	1.00	9008	macro avg	0.97	0.94	0.95	2252	
weighted avg	1.00	1.00	1.00	9008	weighted avg	0.97	0.97	0.97	2252	

Table 1.16 Classification report Random Forest

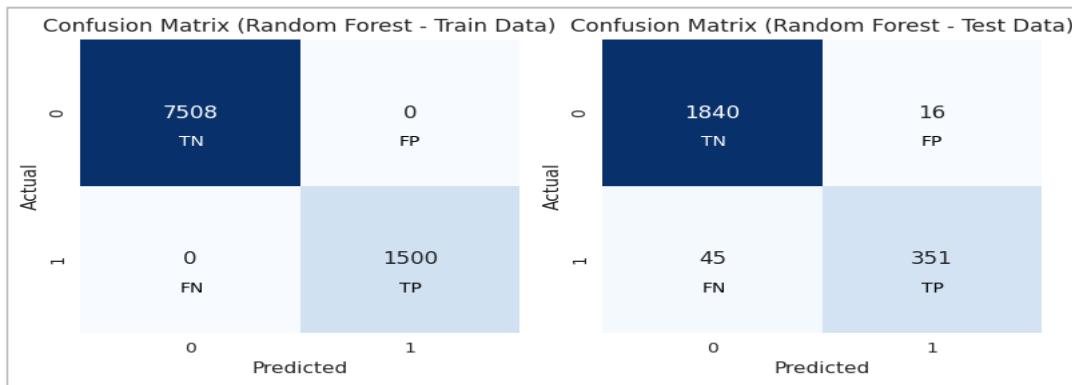


Fig 1.50 Confusion Matrix Random Forest

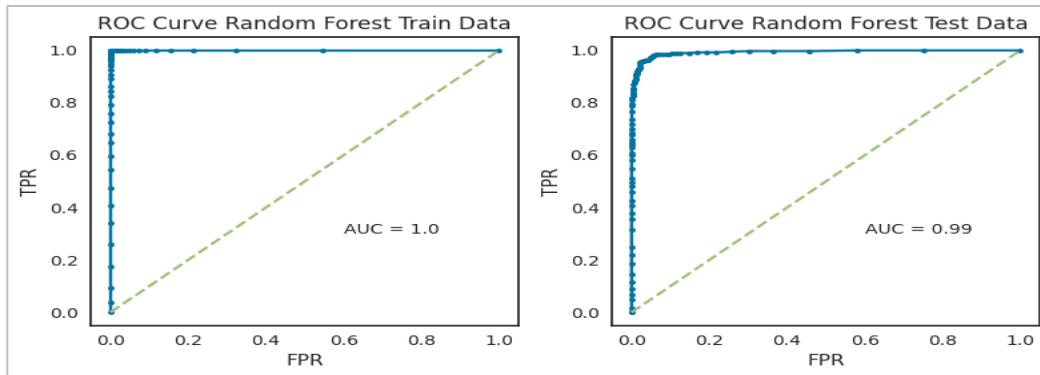


Fig 1.51 ROC-AUC Curve Random Forest

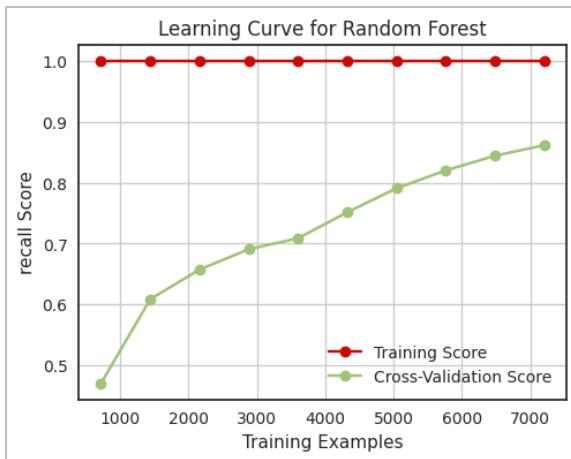


Fig 1.52 Learning Curve Random Forest

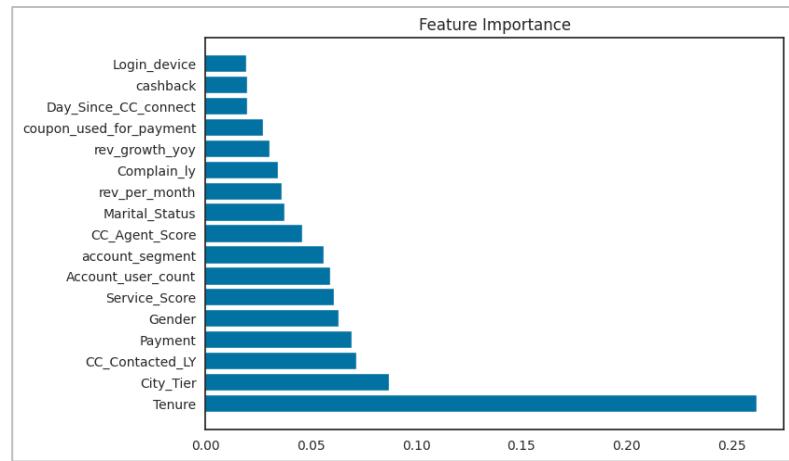


Fig 1.53 Feature importance Random Forest

Insights

- The random forest model is performing well on the test set, with a **high recall for the minority class and a low level of overfitting**.
- The model is **able to identify a good proportion of the actual positive cases (89% recall for the minority class)**. This is important for applications where it is critical to avoid false negatives.
- The **model is slightly overfitting the training data**. The overfitting is not severe and the model is still able to generalize well to new data.
- However, **we will tune the model to reduce overfitting** & see if it improves recall

Linear Discriminant Analysis (LDA)

Linear Discriminant Analysis predicts the class in dependent variable using a linear combination of independent variables.

LDA projects the features in higher-dimensional space onto a lower dimensional space. LDA then searches for a linear combination of independent variables (line, plane, or hyperplane) that best separates the classes of the dependent variable.

Assumptions of LDA:

- Each feature in the dataset is a gaussian distribution
- Each feature has the same variance, the value of each feature varies around the mean with the same amount on average.
- Each feature is assumed to be randomly sampled.
- Lack of multicollinearity in independent features. Increase in correlations between independent features and the power of prediction decreases.

Train data Classification Report					Test data Classification Report				
	precision	recall	f1-score	support		precision	recall	f1-score	support
0.0	0.88	0.98	0.92	7508	0.0	0.87	0.98	0.92	1856
1.0	0.74	0.30	0.43	1500	1.0	0.76	0.33	0.46	396
accuracy			0.87	9008	accuracy			0.86	2252
macro avg	0.81	0.64	0.68	9008	macro avg	0.82	0.65	0.69	2252
weighted avg	0.85	0.87	0.84	9008	weighted avg	0.85	0.86	0.84	2252

Table 1.17 Classification report LDA

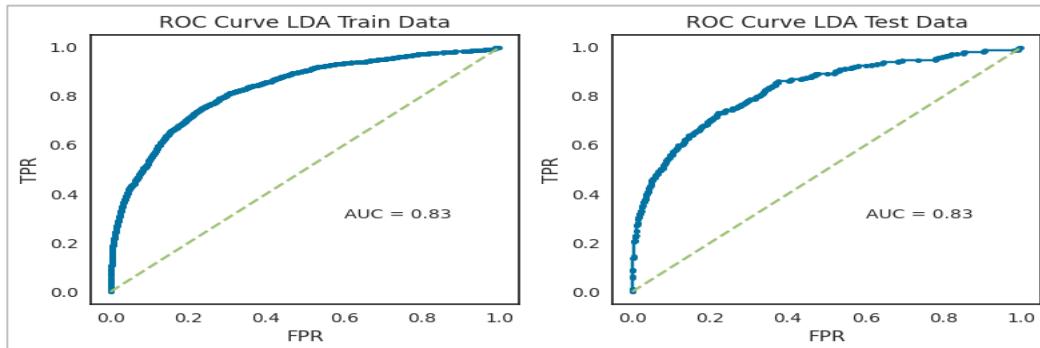
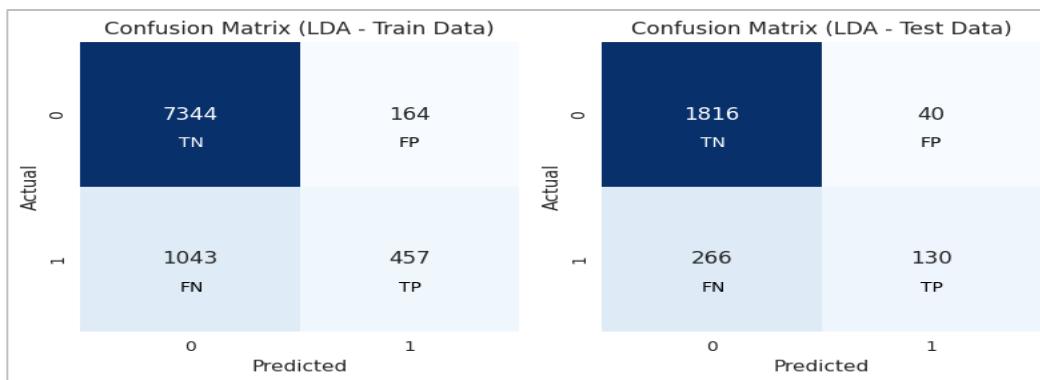


Fig 1.55 ROC-AUC Curve LDA

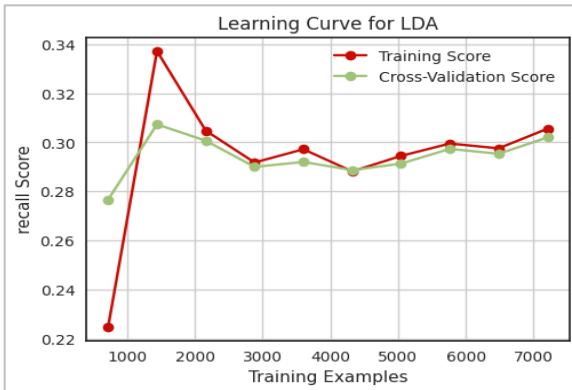


Fig 1.56 Learning Curve LDA

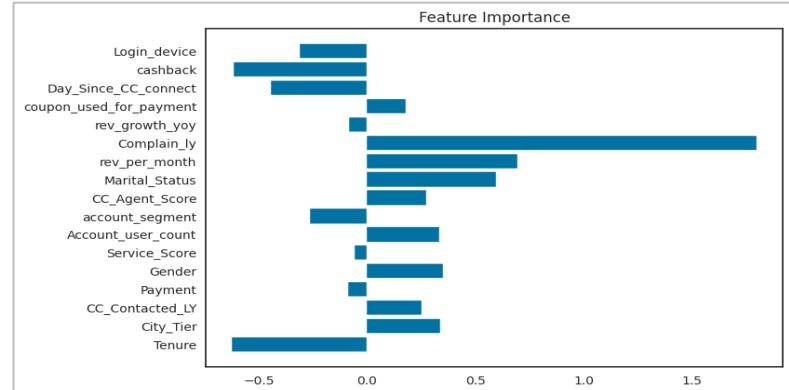


Fig 1.57 Feature importance LDA

Insights:

- The model is **only able to correctly predict 33% of the churn customers**. This is a **very low recall score**, and it is not good enough for churn prediction
- The LDA Model is **overfitting the training data**.
- **Not able to generalize well to unseen data** hence **not a good model** for predicting churn.

K-Nearest Neighbors (KNN)

The data-point in KNN is classified on the basis of its **k Nearest Neighbors**, followed by the **majority vote of those nearest neighbors**; a query point is assigned the data class which has the most representatives within the nearest neighbors of the point

K-NN is a non-parametric algorithm, which means it does not make any assumptions on underlying data

Train data Classification Report					Test data Classification Report				
	precision	recall	f1-score	support		precision	recall	f1-score	support
0.0	0.99	1.00	1.00	7508	0.0	0.98	0.99	0.98	1856
1.0	0.99	0.97	0.98	1500	1.0	0.96	0.89	0.92	396
accuracy			0.99	9008	accuracy			0.97	2252
macro avg	0.99	0.99	0.99	9008	macro avg	0.97	0.94	0.95	2252
weighted avg	0.99	0.99	0.99	9008	weighted avg	0.97	0.97	0.97	2252

Table 1.18 Classification report KNN

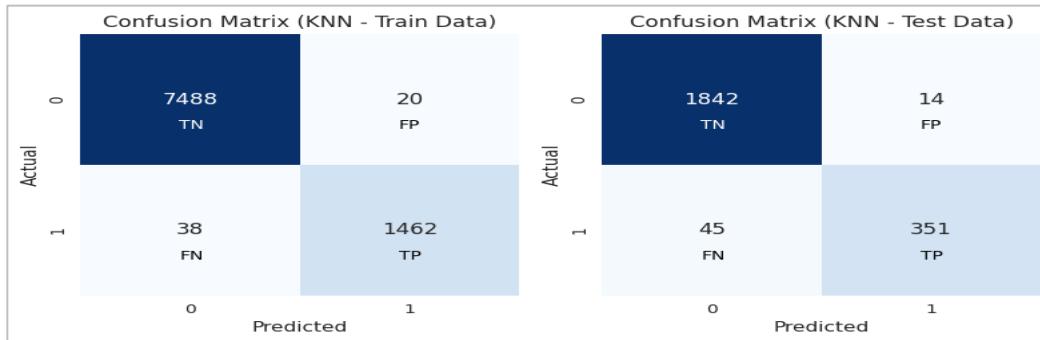


Fig 1.58 Confusion Matrix KNN

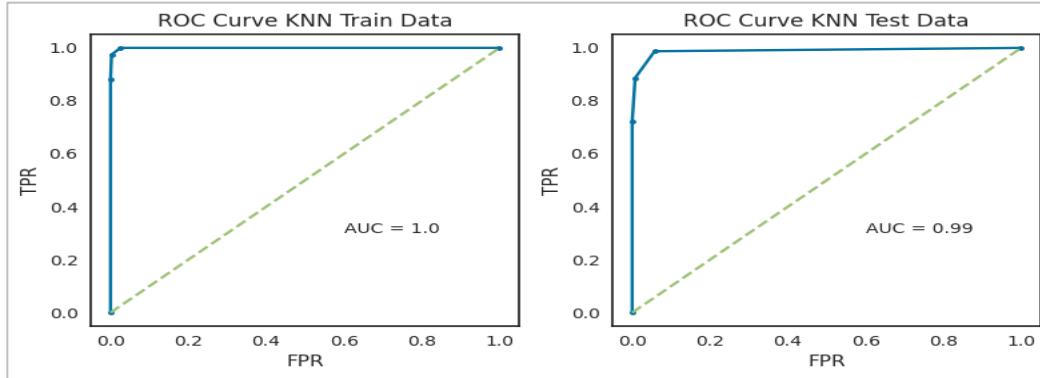


Fig 1.59 ROC-AUC Curve KNN

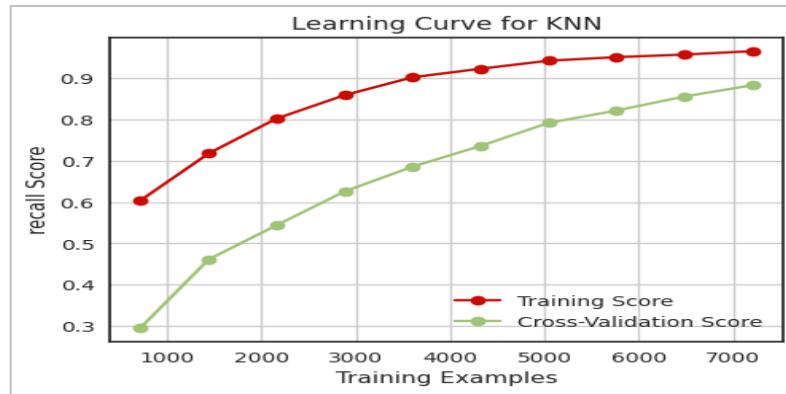


Fig 1.60 Learning Curve KNN

Insights

- The K-Nearest Neighbors (KNN) model exhibits **overall good performance on both the training and test data.**
- It achieves high precision, recall, and F1-score, particularly for non-churn cases (0.0) while drops slightly on test data. The **recall on train set is 0.97** while **0.89 on test set.**
- The learning curve shows that the training accuracy and cross-validation accuracy both converge quickly. The **model slightly overfits**

Naïve Bayes

Naïve Bayes Classifier can be trained easily and fast and can be used as benchmark model. It requires a strong assumption of independent predictors, so when the model has a bad performance, the reason leading to that may be the dependence between predictors.

Train data Classification Report					Test data Classification Report					
	precision	recall	f1-score	support		precision	recall	f1-score	support	
0.0	0.86	0.98	0.91	7508		0.0	0.85	0.98	0.91	1856
1.0	0.61	0.19	0.29	1500		1.0	0.67	0.21	0.32	396
accuracy			0.84	9008	accuracy			0.84	2252	
macro avg	0.73	0.58	0.60	9008	macro avg	0.76	0.59	0.61	2252	
weighted avg	0.82	0.84	0.81	9008	weighted avg	0.82	0.84	0.81	2252	

Table 1.19 Classification report Naïve Bayes

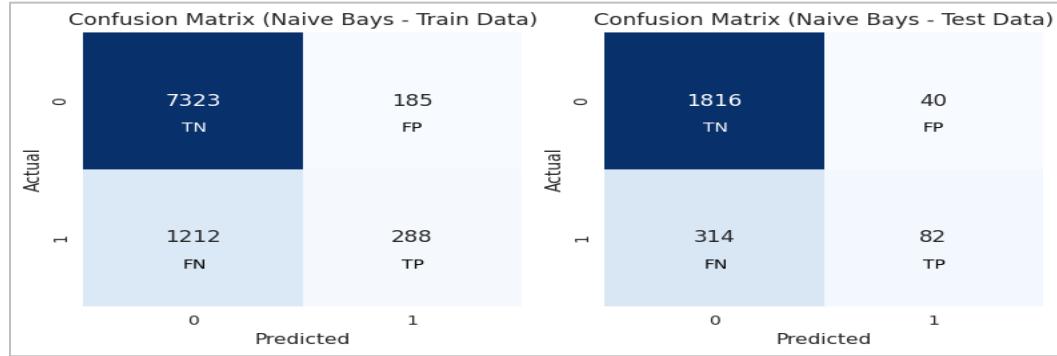


Fig 1.61 Confusion Matrix Naïve Bayes

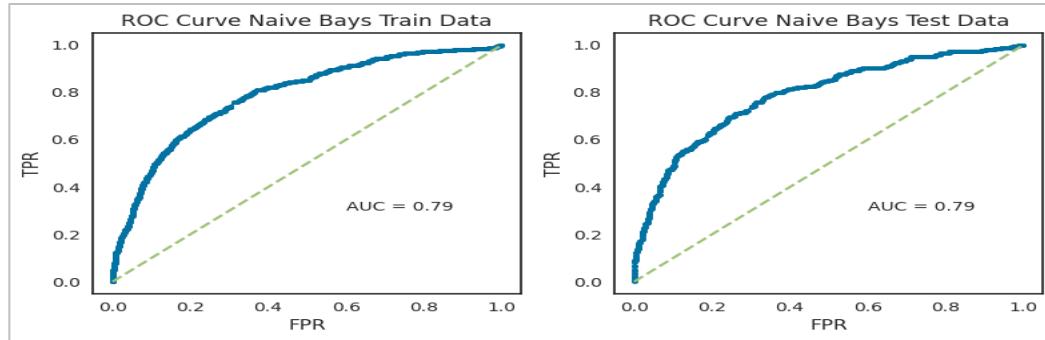


Fig 1.62 ROC-AUC Curve Naïve Bayes

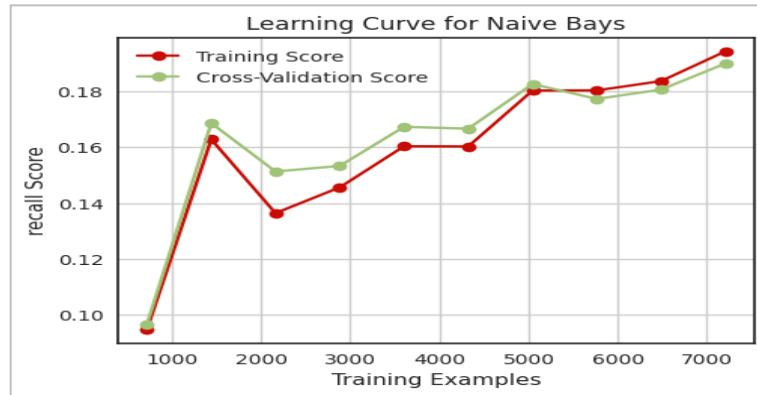


Fig 1.63 Learning Curve Naïve Bayes

Insights

- The Naive Bayes model in the classification report is **not a good** model for predicting churn. It has a **low recall score on the train & test data (0.19 & 0.21 respectively)**
- The **model is overfitting** the training data, it is **not likely to generalize well to unseen data**.
- The above **model is not a good model for predicting churn**.

Neural Network (ANN)

ANNs are nonlinear statistical models which display a complex relationship between the inputs and outputs to discover a new pattern

Artificial Neural network is typically organized in layers. Layers are being made up of many interconnected ‘nodes’ which contain an ‘activation function’.

Train data Classification Report					Test data Classification Report					
	precision	recall	f1-score	support		precision	recall	f1-score	support	
0.0	0.97	1.00	0.98	7508		0.0	0.95	0.99	0.97	1856
1.0	0.99	0.83	0.90	1500		1.0	0.93	0.74	0.82	396
accuracy			0.97	9008	accuracy			0.94	2252	
macro avg	0.98	0.91	0.94	9008	macro avg	0.94	0.86	0.89	2252	
weighted avg	0.97	0.97	0.97	9008	weighted avg	0.94	0.94	0.94	2252	

Table 1.20 Classification report ANN

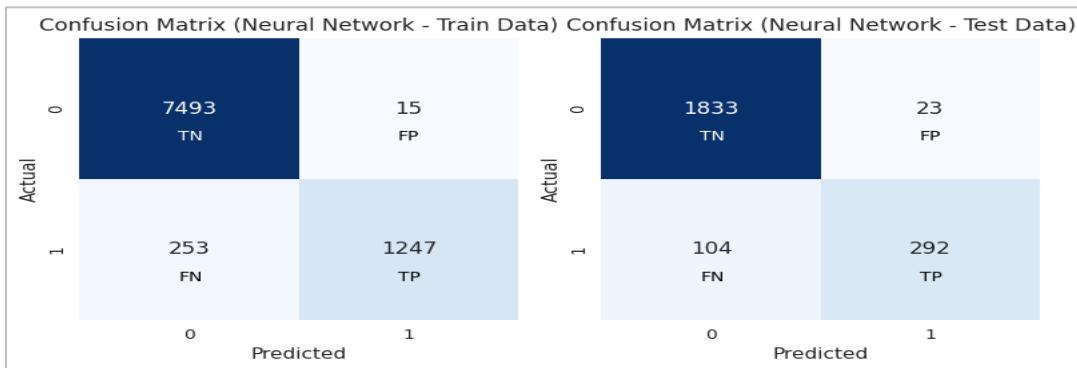


Fig 1.64 Confusion Matrix ANN

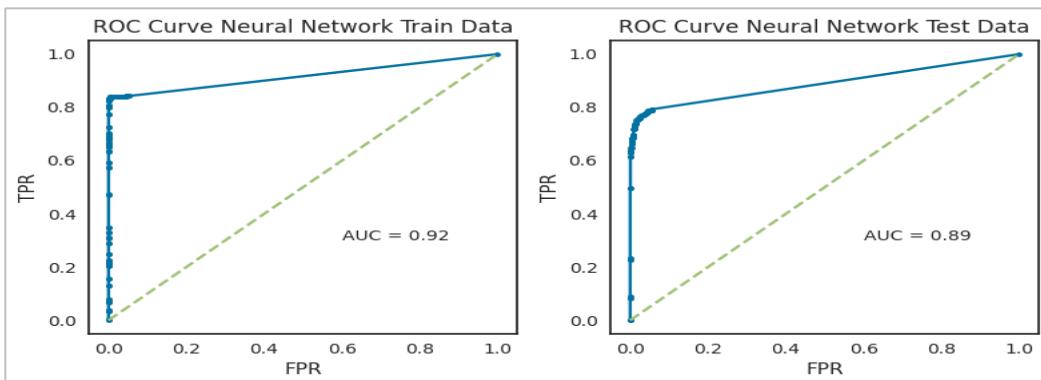


Fig 1.65 ROC-AUC Curve ANN

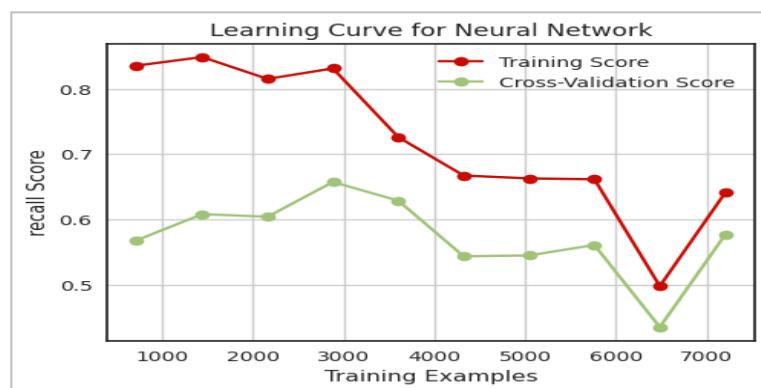


Fig 1.66 Learning Curve ANN

Insights

- The ANN model has a comparatively lower recall score of **0.74** on the test data for the minority class (churn) than few of previous models.
- The **model is overfitting** the training data and may not be able to generalize well to new data
- The model **may not be able to generalize well to new data**.

Model Tuning

Ensemble methods & Hyperparameter Tuning

Since most of the models are either overfitting or not giving a good recall score, we tried using Model Tuning to improve the performance.

A combination of ensemble methods and hyperparameter tuning can lead to highly accurate and robust machine learning models.

Tuned LDA

The LDA model was tuned and the best parameters came as: `{'solver': 'lsqr', 'tol': 0.1}`

Train data Classification Report					Test data Classification Report				
	precision	recall	f1-score	support		precision	recall	f1-score	support
0.0	0.88	0.98	0.92	7508	0.0	0.87	0.98	0.92	1856
1.0	0.74	0.30	0.43	1500	1.0	0.76	0.33	0.46	396
accuracy			0.87	9008	accuracy			0.86	2252
macro avg	0.81	0.64	0.68	9008	macro avg	0.82	0.65	0.69	2252
weighted avg	0.85	0.87	0.84	9008	weighted avg	0.85	0.86	0.84	2252

Table 1.21 Classification report Tuned LDA

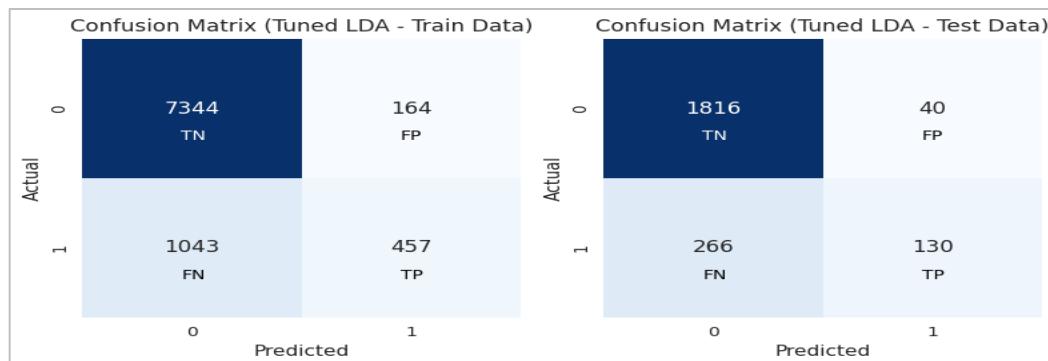


Fig 1.67 Confusion Matrix Tuned LDA

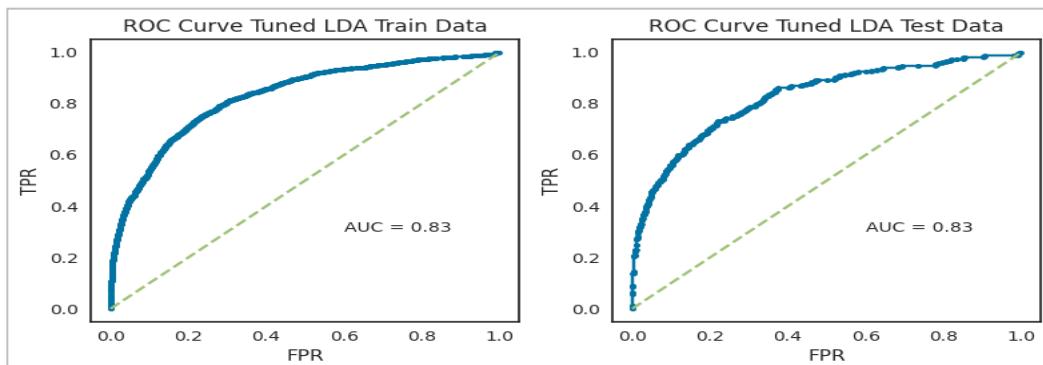


Fig 1.68 ROC-AUC Curve Tuned LDA

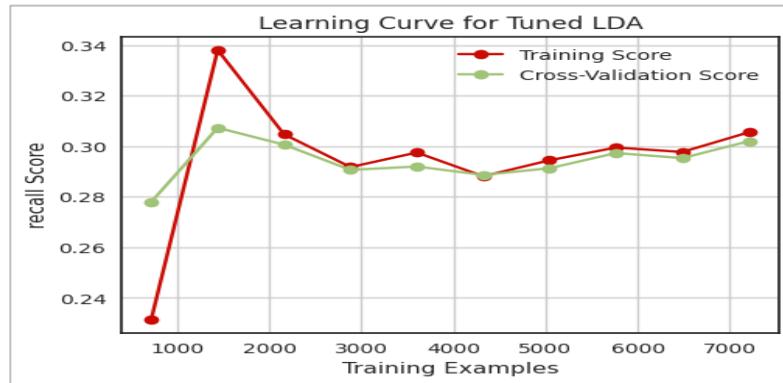


Fig 1.69 Learning Curve Tuned LDA

Insights

- The Tuned LDA model **still has a poor recall score of 0.33** on the test data for the minority class (churn).
- The cross-validation scores are also very low, with a mean score of 0.302, which confirms that the **model is overfitting the training data**.
- The area under curve of 0.83 on both the training and test data is misleading in this case, as it is not a reliable measure of model performance when the class is imbalanced.
- There is **no improvement in LDA post tuning**. This might be due to imbalance dataset, since the tree models shows better performance, we will hyper tune /ensemble them. If none of them works well, we will try balancing data by oversampling data as last resort.

Regularised Decision Tree

The Decision Tree model was regularized and the best parameters came as :

Best max_depth: 12

Best min_samples_split: 2

Best min_samples_leaf: 1

Train data Classification Report					Test data Classification Report				
	precision	recall	f1-score	support		precision	recall	f1-score	support
0.0	0.99	1.00	0.99	7508	0.0	0.95	0.97	0.96	1856
1.0	0.98	0.94	0.96	1500	1.0	0.84	0.78	0.81	396
accuracy			0.99	9008	accuracy			0.94	2252
macro avg	0.98	0.97	0.98	9008	macro avg	0.90	0.88	0.89	2252
weighted avg	0.99	0.99	0.99	9008	weighted avg	0.94	0.94	0.94	2252

Table 1.22 Classification report Regularised Decision Tree

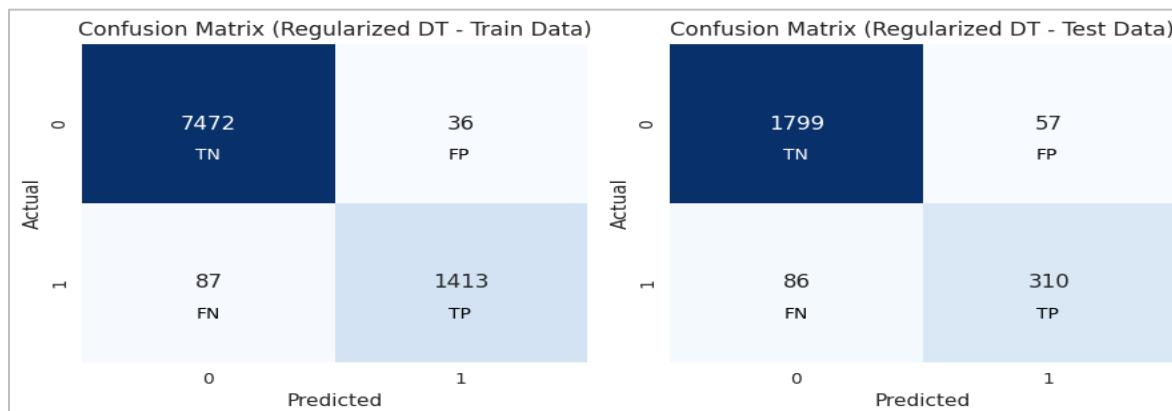


Fig 1.70 Confusion Matrix Regularised Decision Tree

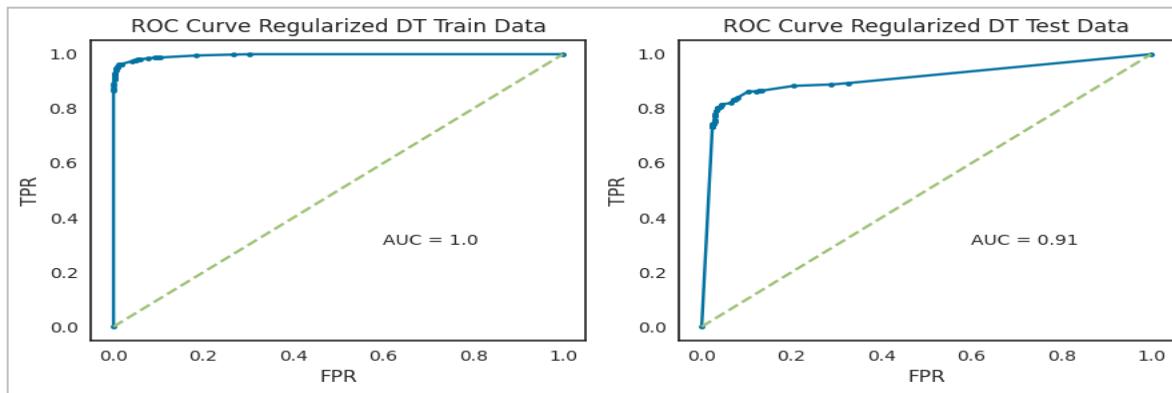


Fig 1.71 ROC-AUC Curve Regularised Decision Tree

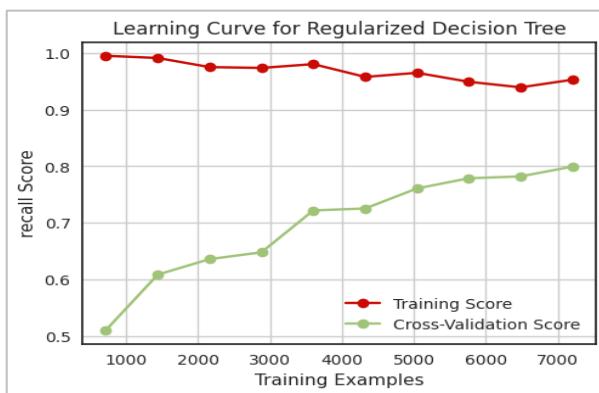


Fig 1.72 Learning Curve Regularized Decision Tree

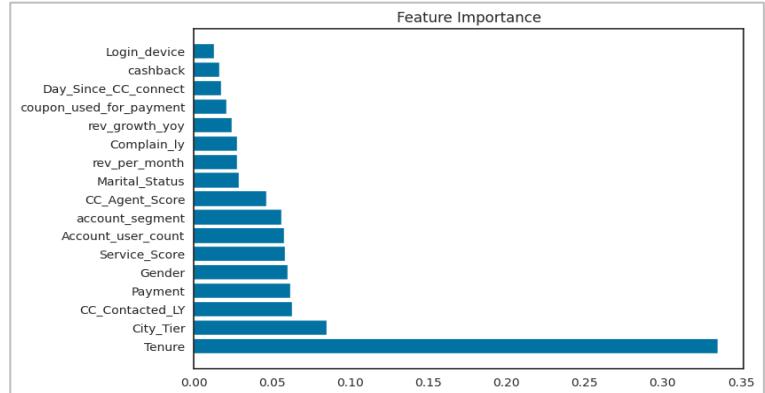


Fig 1.73 Feature importance Regularized Decision Tree

Insights

- The model is **able to identify a moderate proportion of the actual positive cases** (78% recall for the minority class).
- The **model is overfitting the training data**. This is evident from the large gap between the cross-validation scores and the training accuracy.
- The model **may not generalize well to new data**. This is evident from the plateau in the learning curve.
- Overall, the **regularized decision tree model is performing moderately** on the test set, with a **moderate recall for the minority class** and **overfitting**.

Bagging

Bagging also known as bootstrap aggregation is an ensemble technique that trains multiple complex models (overfit models) in parallel and tries to smooth out their predictions by aggregating.

For Bagging, We have used the regularized decision Tree as base estimator.

Train data Classification Report					Test data Classification Report				
	precision	recall	f1-score	support		precision	recall	f1-score	support
0.0	0.99	1.00	1.00	7508		0.0	0.96	0.99	0.98
1.0	1.00	0.96	0.98	1500		1.0	0.93	0.83	0.88
accuracy			0.99	9008	accuracy			0.96	2252
macro avg	0.99	0.98	0.99	9008	macro avg	0.95	0.91	0.93	2252
weighted avg	0.99	0.99	0.99	9008	weighted avg	0.96	0.96	0.96	2252

Table 1.23 Classification report Bagging

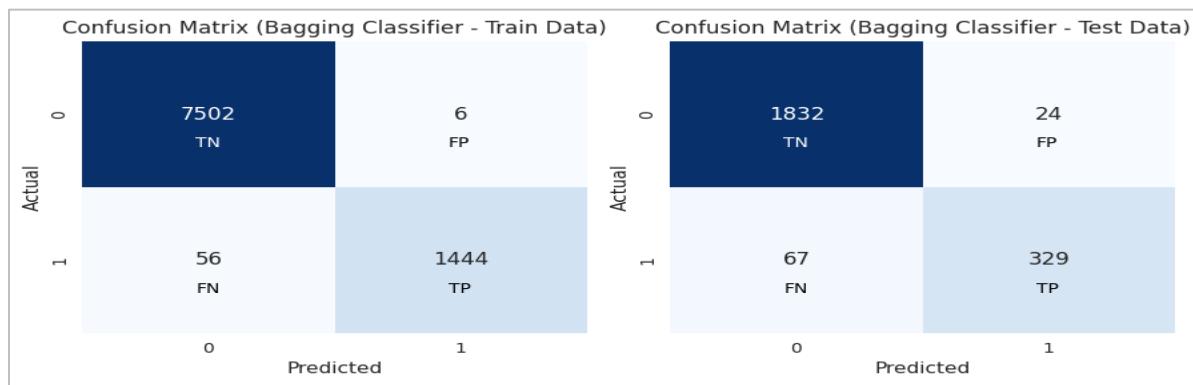


Fig 1.74 Confusion Matrix Bagging

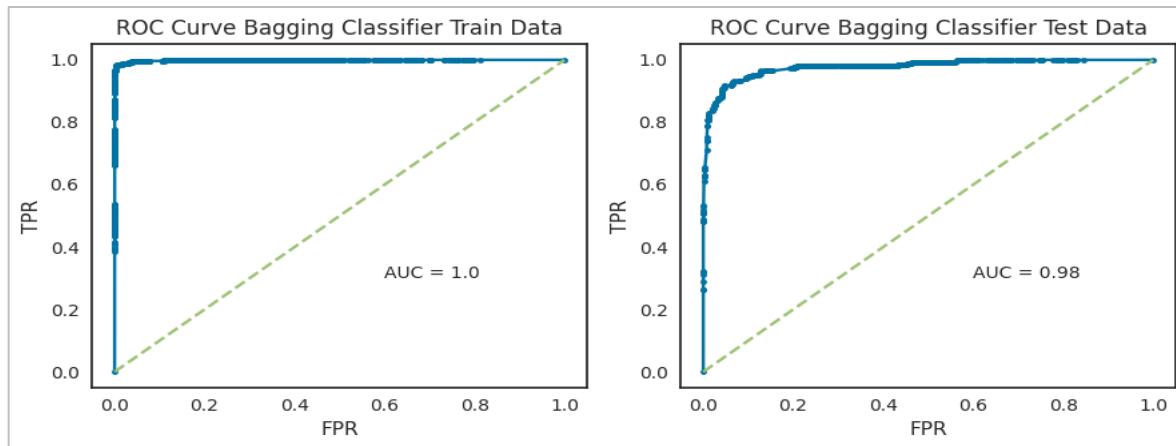


Fig 1.75 ROC-AUC Curve Bagging

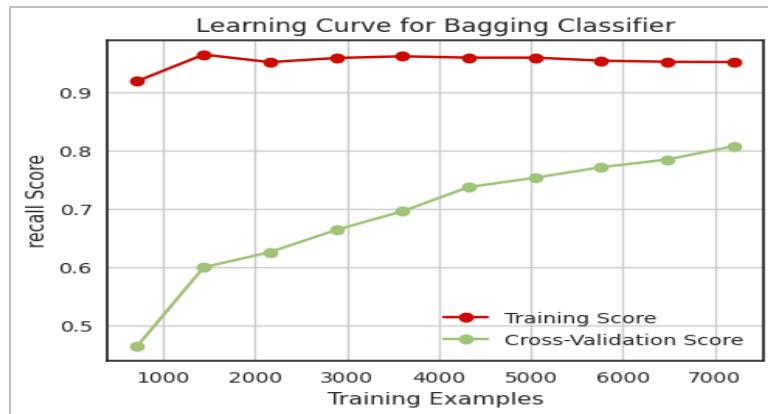


Fig 1.76 Learning Curve Bagging

Insights

- The model is **able to identify a good proportion of the actual positive cases (83% recall for the minority class)**. This may be sufficient for some applications, but it may not be sufficient for applications where it is critical to avoid false negatives.
- The **model is overfitting the training data**. This is evident from the large gap between the cross-validation scores and the training accuracy.
- The model **may not generalize well to new data**. This is evident from the plateau in the learning curve.
- Overall, the **bagging classifier is performing moderately on the test set**, with a **fair recall** for the minority class and **overfitting**.

Random Forest with Tuning

We built a random forest classifier, & search for the best fit model, using Grid Search Cross Validation Technique with the following values:

```
'max_depth': [4, 5, 6, 7]
'max_features': [1, 2, 3]
'min_samples_leaf': [1, 2, 3]
'min_samples_split': [6, 8, 10]
'n_estimators': [101, 301, 501]
```

We obtain the **best fit model** with the parameters:

```
'class_weight': 'balanced'
'max_depth': 7
'max_features': 3
'min_samples_leaf': 1
'min_samples_split': 8
'n_estimators': 301
```

The best fit model parameters are then used for creating a random forest classifier model & predicting the train & test labels based on the independent variable values. The performance of the model on the train & test set are as follows:

Train data Classification Report					Test data Classification Report				
	precision	recall	f1-score	support		precision	recall	f1-score	support
0.0	0.97	0.90	0.94	7508	0.0	0.96	0.90	0.93	1856
1.0	0.64	0.88	0.74	1500	1.0	0.63	0.84	0.72	396
accuracy			0.90	9008	accuracy			0.89	2252
macro avg	0.81	0.89	0.84	9008	macro avg	0.80	0.87	0.83	2252
weighted avg	0.92	0.90	0.90	9008	weighted avg	0.91	0.89	0.89	2252

Table I.24 Classification report Tuned Random Forest

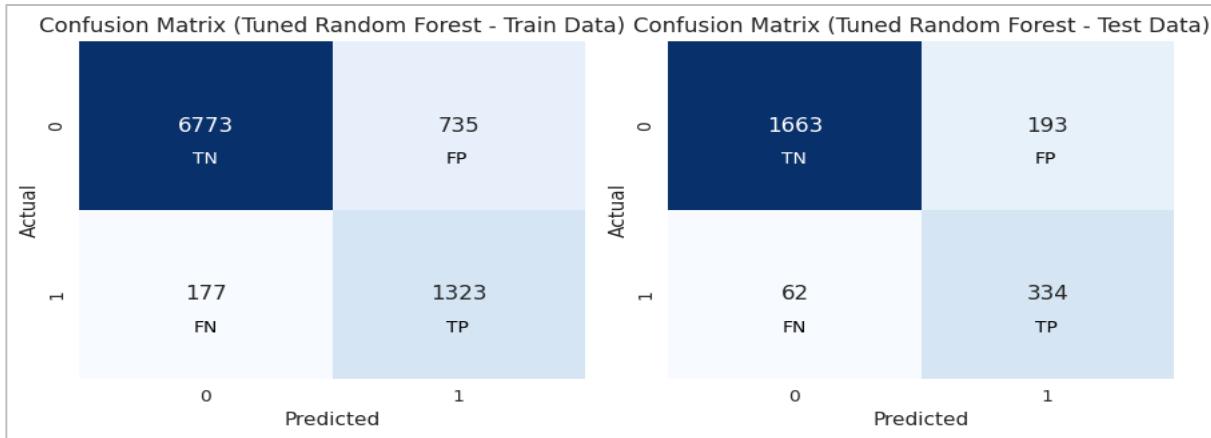


Fig 1.77 Confusion Matrix Tuned Random Forest

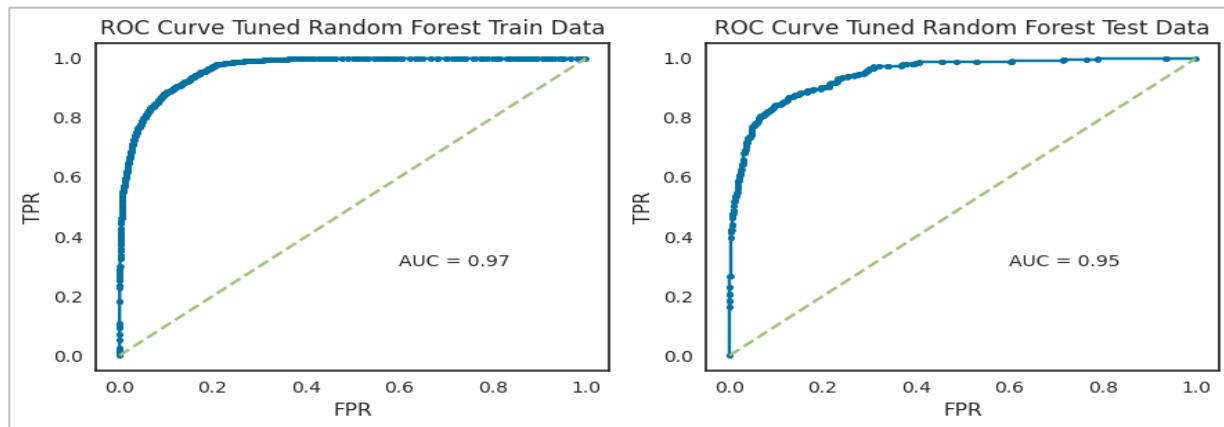


Fig 1.78 ROC-AUC Curve Tuned Random Forest

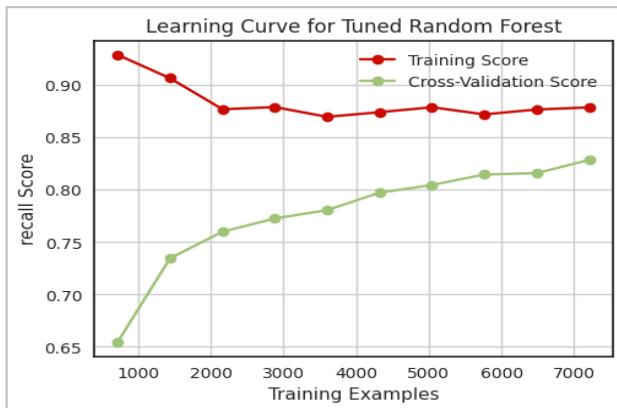


Fig 1.79 Learning Curve Tuned Random Forest

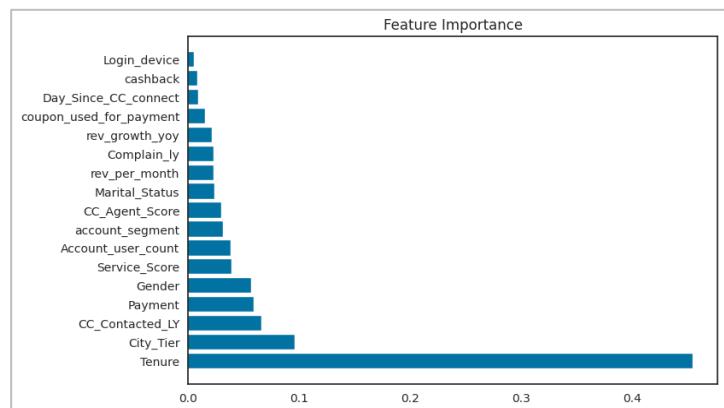


Fig 1.80 Feature importance Tuned Random Forest

Insights

- The model is able to identify a **good proportion of the actual positive cases (88% recall for the minority class)**. This may be sufficient for some applications, but it may not be sufficient for applications where it is critical to avoid false negatives.
- The model is **slightly overfitting the training data**. However, the overfitting is not severe and the model is still able to generalize well to new data.
- The most important Features are **Tenure, City Tier and CC_Contacted_LY** for the model

Adaboost

Boosting trains a large number of simple models (weak/ underfitting models) in sequence & combines them into a single strong model. The Training data remains the same during the creation of successive models only weights are added to the misclassified datapoints of the previous model & reduced for the correctly classified datapoints of the previous model.

We build an **adaptive boosting classifier** using the **regularized decision as base estimator** & search for the appropriate number of simple models using the Grid Search Cross Validation Technique:

No. of estimators: 301, CV=10, Scoring = 'recall'

We then create an adaptive boosting classifier model & predict the train & test labels based on the independent variable values. The performance of the model on the train & test set are as follows:

Train data Classification Report					Test data Classification Report				
	precision	recall	f1-score	support		precision	recall	f1-score	support
0.0	1.00	1.00	1.00	7508	0.0	0.97	1.00	0.98	1856
1.0	1.00	1.00	1.00	1500	1.0	0.98	0.87	0.92	396
accuracy				1.00	9008				0.97
macro avg	1.00	1.00	1.00	9008	accuracy				2252
weighted avg	1.00	1.00	1.00	9008	macro avg	0.98	0.93	0.95	2252
					weighted avg	0.97	0.97	0.97	2252

Table 1.25 Classification report Adaboost

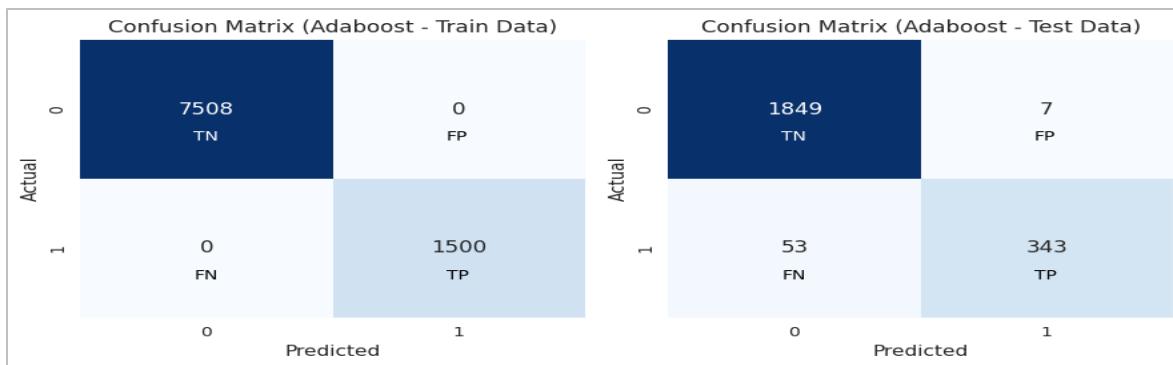


Fig 1.81 Confusion Matrix Adaboost

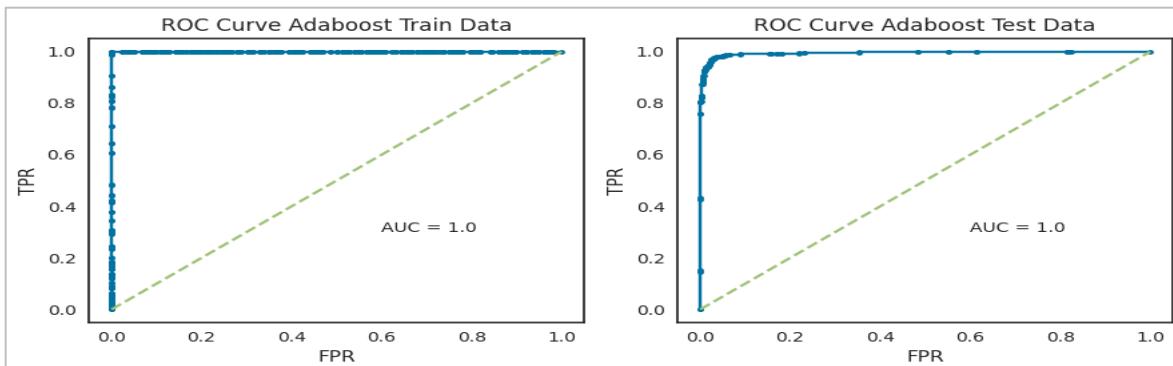


Fig 1.82 ROC-AUC Curve Adaboost

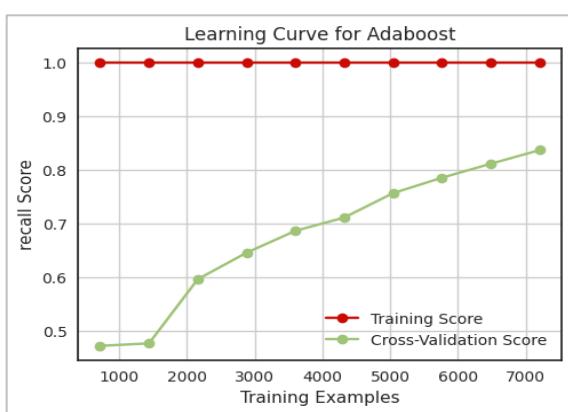


Fig 1.83 Learning Curve Adaboost

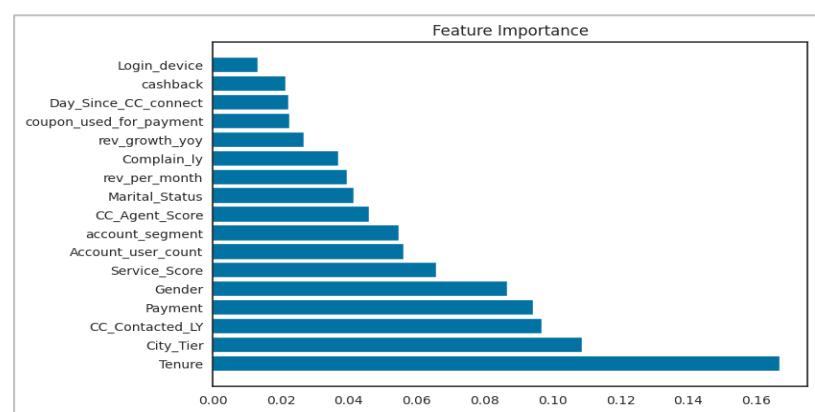


Fig 1.84 Feature importance Adaboost

Insights

- The model is able to identify a **good proportion of the actual positive cases (87% recall for the minority class)**.
- The model is **slightly overfitting the training data**. However, the overfitting is not severe and the model is still able to generalize well to new data.
- The AUC is 100% for both test & train, however, would not be the accurate representation

- The most important Features are **Tenue, City Tier and CC_Contacted_LY** for the model

XG Boost with Hyperparameter Tuning

Xtreme Gradient Boosting is an ensemble technique that uses an ensemble of decision trees and gradient boosting to make predictions. We build a XGBoost classifier & search for the best fit model, using Grid Search Cross Validation Technique with the following values:

```
'learning_rate': [0.2, 0.1, 0.05],
'max_depth': [6, 8, 10],
'colsample_bytree': [0.9, 0.8, 0.7],
'colsample_bynode': [1, 0.9],
'colsample_bylevel': [0.9, 0.8],
'scale_pos_weight': [0.2, 0.4],
'subsample': [0.9, 0.8, 0.7],
'gamma': [0, 0.5, 1],
```

The best fit parameters came as :

```
{'colsample_bylevel': 0.8, 'colsample_bynode': 1,
'colsample_bytree': 0.9, 'gamma': 0,
'learning_rate': 0.2, 'max_depth': 10,
'scale_pos_weight': 0.4, 'subsample': 0.9}
```

The best fit model parameters are then used for creating a XG Boost classifier model & predicting the train & test labels based on the independent variable values. The performance of the model on the train & test set are as follows:-

Train data Classification Report					Test data Classification Report				
	precision	recall	f1-score	support		precision	recall	f1-score	support
0.0	1.00	1.00	1.00	7508	0.0	0.97	0.99	0.98	1856
1.0	1.00	1.00	1.00	1500	1.0	0.96	0.87	0.91	396
accuracy			1.00	9008	accuracy			0.97	2252
macro avg	1.00	1.00	1.00	9008	macro avg	0.97	0.93	0.95	2252
weighted avg	1.00	1.00	1.00	9008	weighted avg	0.97	0.97	0.97	2252

Table 1.26 Classification report XGBoost

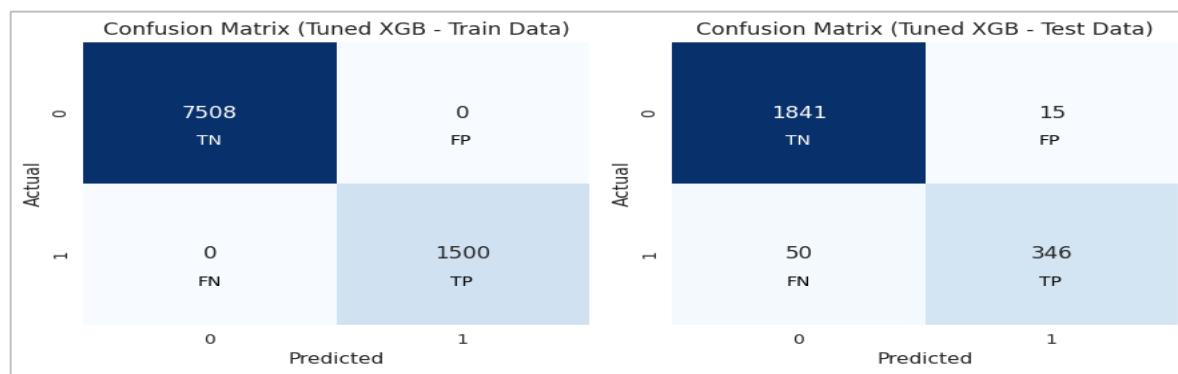


Fig 1.85 Confusion Matrix XGBoost

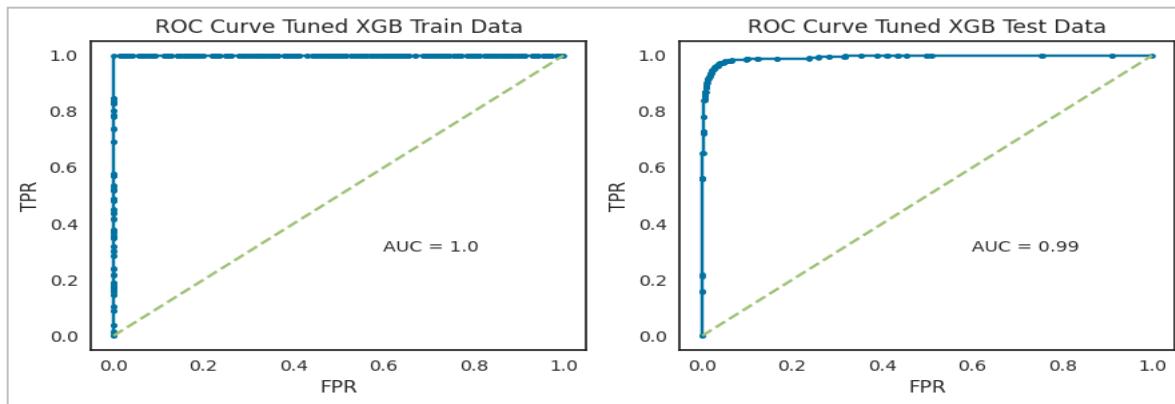


Fig 1.86 ROC-AUC Curve XGBoost

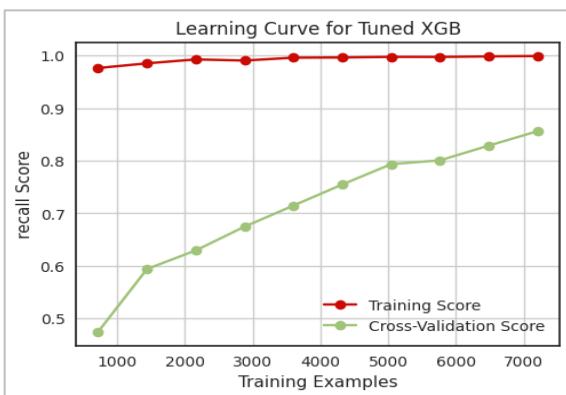


Fig 1.87 Learning Curve XGBoost

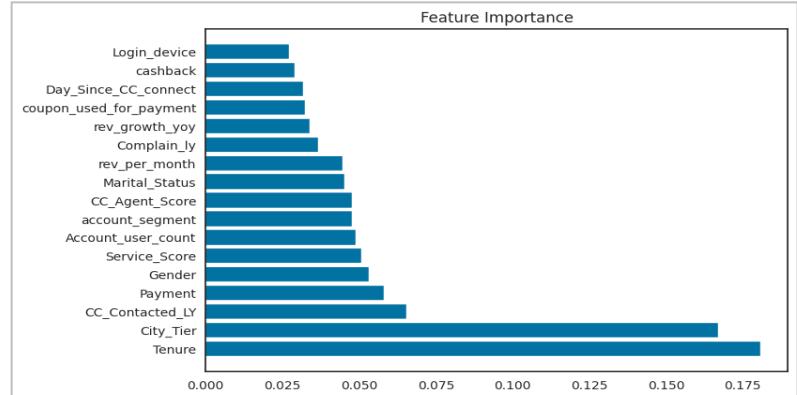


Fig 1.88 Feature importance XGBoost

Insights

- The model is **able to identify 87% of the actual positive cases (recall of minority class)**. This is a good result, but it could be improved.
- The **model is overfitting** the training data. This is evident from the cross-validation scores and the learning curve.
- The model is performing well on the test set, with an **accuracy of 97%**. This suggests that the **model is able to generalize to new data**.
- The most important Features are **Tenue, City Tier and CC_Contacted_LY** for the model
- There is room for improvement in the model performance, especially in terms of recall of minority class and overfitting.

CatBoost with Hyperparameter Tuning

Catboost (or Categorical Boosting) is a variant of gradient boosting that can handle both categorical and numerical features. It is especially powerful in two ways:

- It yields state-of-the-art results without extensive data training typically required by other machine learning methods, and
- Provides powerful out-of-the-box support for the more descriptive data formats that accompany many business problems.

While Catboost can work on categorical data without need of encoding, however, we used encoded data here. We build a CatBoost classifier & search for the best fit model, using Grid Search Cross Validation Technique with the following values:

'depth': [6, 8, 10],

```
'learning_rate': [0.1, 0.05, 0.01],
'iterations': [100, 200, 300],
'l2_leaf_reg': [3, 5, 7],
```

We obtain the **best fit** model with the parameters:

```
'depth': 10,
'iterations': 300,
'l2_leaf_reg': 3,
'learning_rate': 0.1}
```

The best fit model parameters are then used for creating a Cat Boost classifier model & predicting the train & test labels based on the independent variable values. The performance of the model on the train & test set are as follows:-

Train data Classification Report					Test data Classification Report				
	precision	recall	f1-score	support		precision	recall	f1-score	support
0.0	1.00	1.00	1.00	7508	0.0	0.99	1.00	0.99	1856
1.0	1.00	1.00	1.00	1500	1.0	0.98	0.94	0.96	396
accuracy			1.00	9008	accuracy			0.99	2252
macro avg	1.00	1.00	1.00	9008	macro avg	0.98	0.97	0.98	2252
weighted avg	1.00	1.00	1.00	9008	weighted avg	0.99	0.99	0.99	2252

Table 1.27 Classification report CatBoost

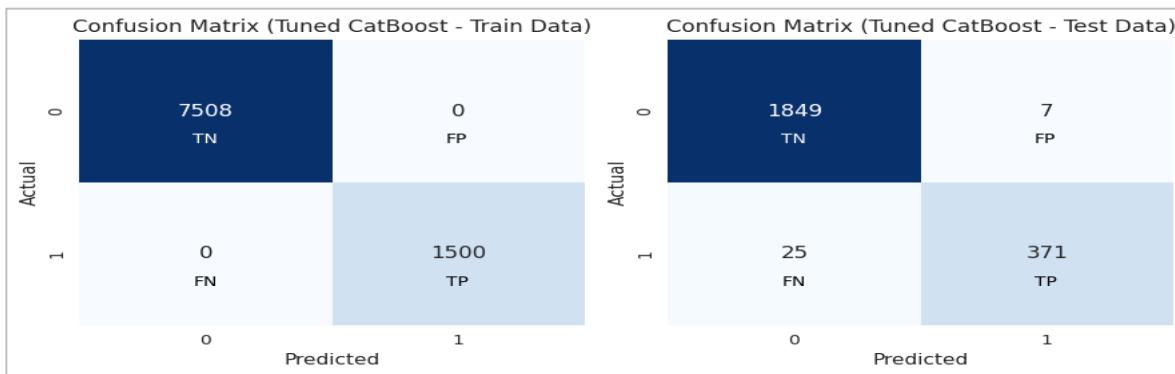


Fig 1.89 Confusion Matrix CatBoost

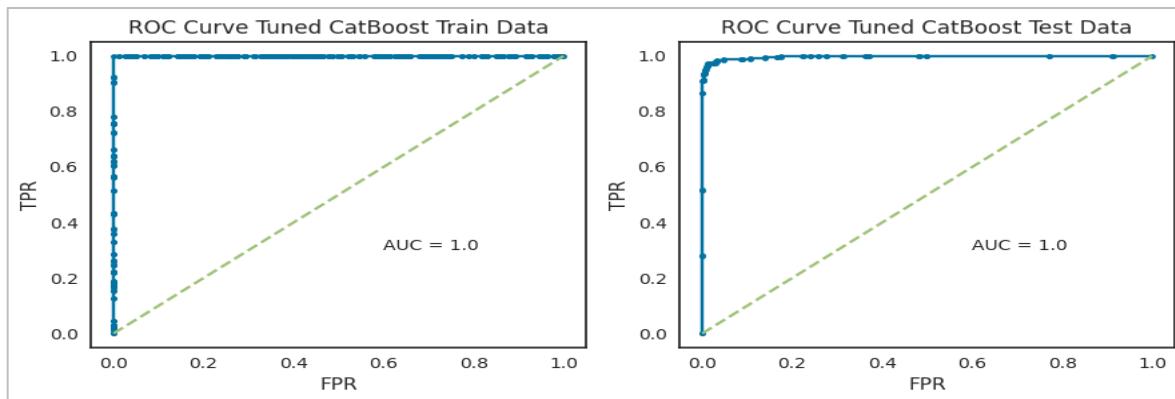


Fig 1.90 ROC-AUC Curve CatBoost

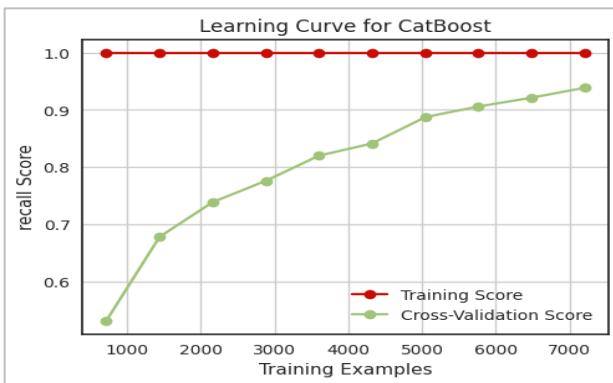


Fig 1.91 Learning Curve CatBoost

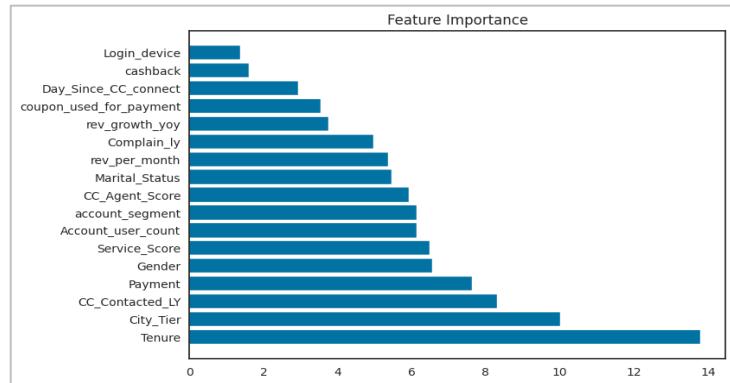


Fig 1.92 Feature importance CatBoost

Insights

- The model is **able to identify a large proportion of the actual positive cases (94% recall for the minority class)**. This is so far the best model results we are able to get.
- The model is **not severely overfitting the training data**. This is evident from the relatively small difference between the training accuracy and the cross-validation score.
- The **model is able to generalize well to new data**. This is evident from the high plateau in the learning curve.
- The most important Features are **Tenue, City Tier and CC_Contacted_LY** for the model

SVM Model with Tuning

SVM offers very high accuracy compared to other classifiers such as logistic regression, and decision trees. SVM constructs a hyperplane in multidimensional space to separate different classes. It generates optimal hyperplane in an iterative manner, which is used to minimize an error. The core idea of SVM is to find a maximum marginal hyperplane(MMH) that best divides the dataset into classes

We build a Support Vector classifier & search for the best fit model, using Grid Search Cross Validation Technique with the following values:

Regularisation Parameter (C): 0.1, 1, 10

Gamma: 0.01, 0.1, 1

Kernel: rbf, poly

Class Weight: balanced

We obtain the best fit model with the parameters:

```
'C': 10,
'class_weight':'balanced',
'gamma': 0.1,
'kernel': 'rbf'
```

The best fit model parameters are then used for creating a SVM classifier model & predicting the train & test labels based on the independent variable values. The performance of the model on the train & test set are as follows:-

Train data Classification Report					Test data Classification Report				
	precision	recall	f1-score	support		precision	recall	f1-score	support
0.0	1.00	0.99	0.99	7508	0.0	0.99	0.98	0.98	1856
1.0	0.95	1.00	0.97	1500	1.0	0.91	0.95	0.93	396
accuracy			0.99	9008	accuracy			0.97	2252
macro avg	0.97	0.99	0.98	9008	macro avg	0.95	0.97	0.96	2252
weighted avg	0.99	0.99	0.99	9008	weighted avg	0.97	0.97	0.97	2252

Table 1.28 Classification report SVM

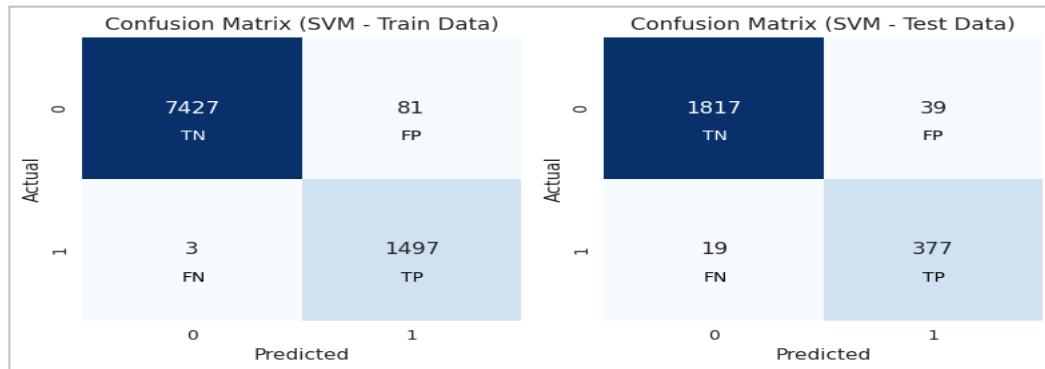


Fig 1.93 Confusion Matrix SVM

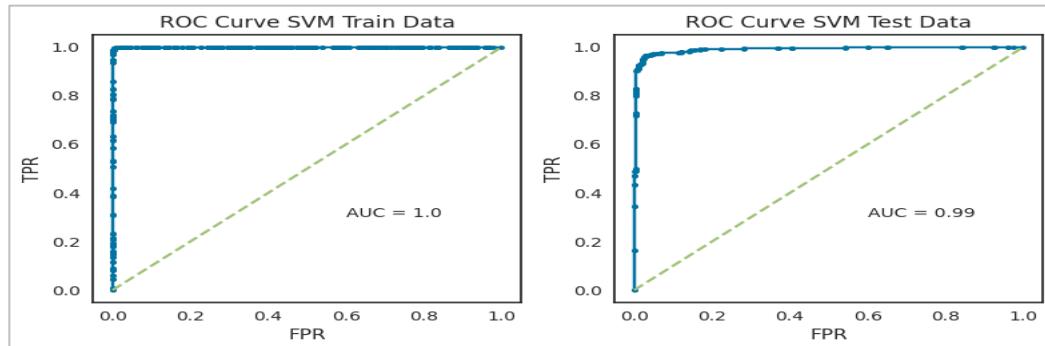


Fig 1.94 ROC-AUC Curve SVM

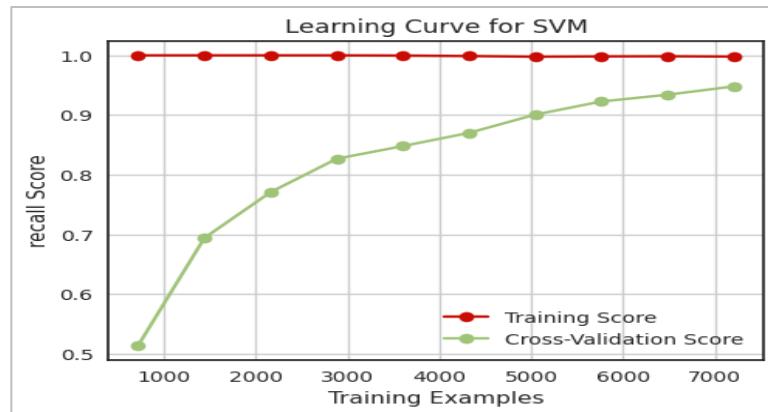


Fig 1.95 Learning Curve SVM

Insights

- The model is **able to identify a very high proportion of the actual positive cases (95% recall for the minority class)**. This is important for applications where it is critical to avoid false negatives.
- The model is **not overfitting the training data**. This is evident from the fact that the cross-validation scores are very close to the training accuracy (**mean cross-validation score is 94.8%**, which is very close to the **training accuracy of 99%**)
- The **model is able to generalize well to new data**. This is evident from the fact that the training score and the cross-validation score are both high and very close.
- Overall, **the SVM model is a very good choice for imbalanced classification** here. It is able to achieve a high recall for the minority class without overfitting the training data.

Model Comparison

Model	Majority Class(Not Churn)			Minority Class(Churn)			accuracy
	Precision	Recall	f1-score	Precision	Recall	f1-score	
SVM with Tuning	98.97%	97.90%	98.43%	90.63%	95.20%	92.86%	97.42%
CatBoost with Tuning	98.67%	99.62%	99.14%	98.15%	93.69%	95.87%	98.58%
Random Forest	97.61%	99.14%	98.37%	95.64%	88.64%	92.01%	97.29%
KNN	97.62%	99.25%	98.42%	96.16%	88.64%	92.25%	97.38%
XGBoost with Tuning	97.36%	99.19%	98.27%	95.84%	87.37%	91.41%	97.11%
Adaboost	97.21%	99.62%	98.40%	98.00%	86.62%	91.96%	97.34%
Tuned Random Forest	96.41%	89.60%	92.88%	63.38%	84.34%	72.37%	88.68%
Decision Tree	96.50%	96.55%	96.53%	83.80%	83.59%	83.69%	94.27%
Bagging Classifier	96.47%	98.71%	97.58%	93.20%	83.08%	87.85%	95.96%
Regularized Decision Tree	95.44%	96.93%	96.18%	84.47%	78.28%	81.26%	93.65%
Neural Network	94.63%	98.76%	96.65%	92.70%	73.74%	82.14%	94.36%
LDA	87.22%	97.84%	92.23%	76.47%	32.83%	45.94%	86.41%
Tuned LDA	87.22%	97.84%	92.23%	76.47%	32.83%	45.94%	86.41%
Naive bayes	85.26%	97.84%	91.12%	67.21%	20.71%	31.66%	84.28%

Table 1.29 Model comparison

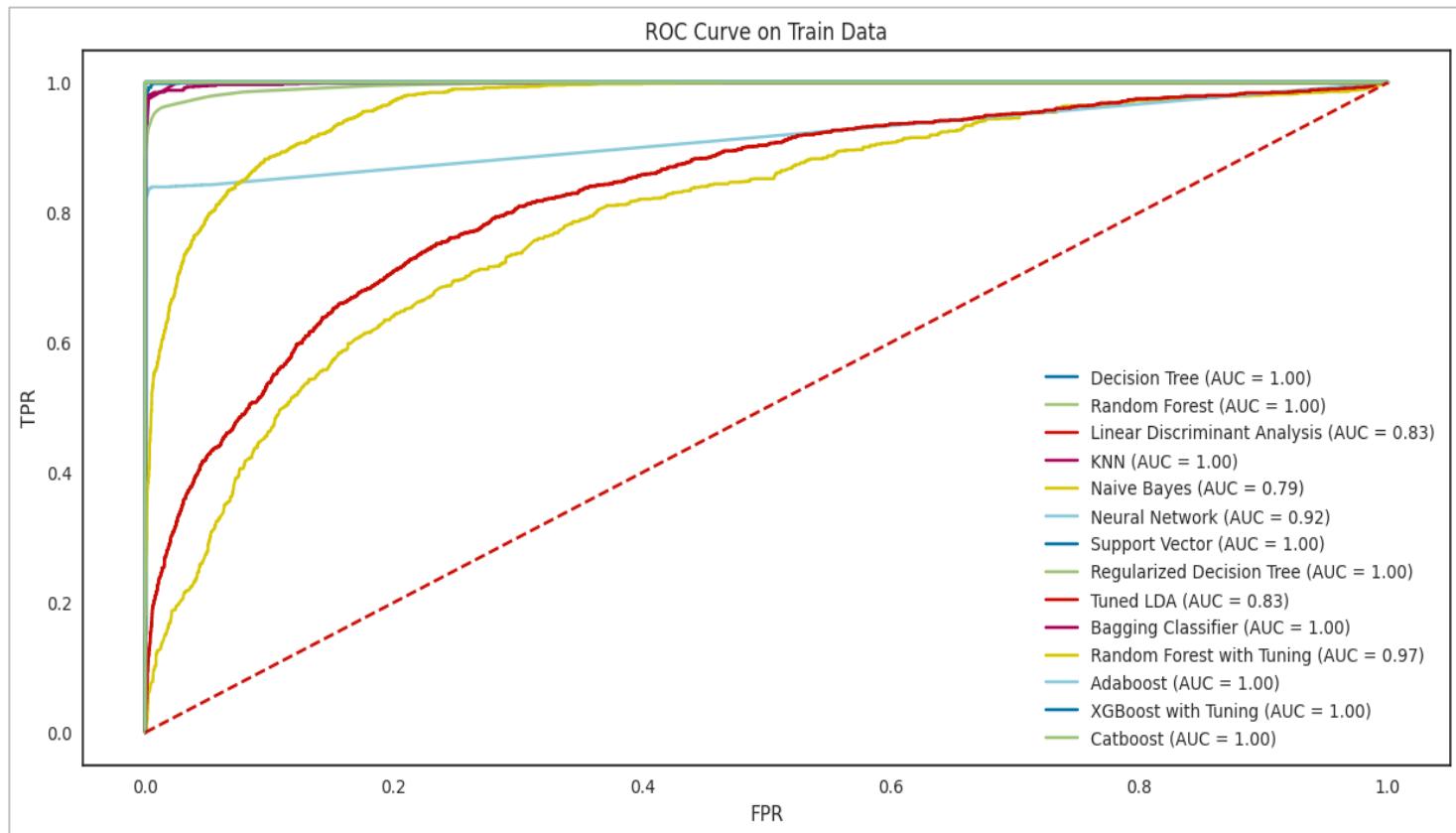


Fig 1.96 ROC-AUC Curve Train data

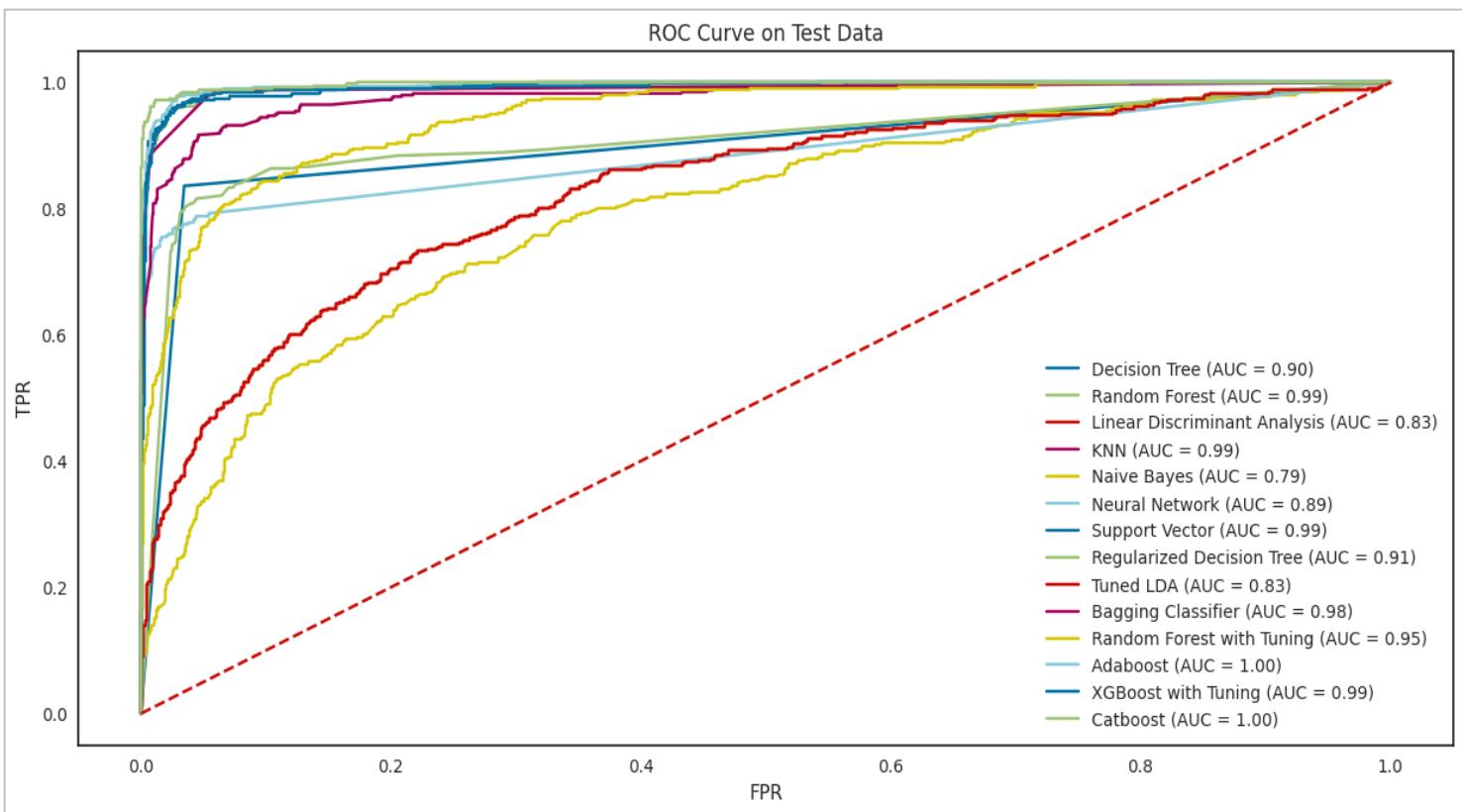


Fig 1.97 ROC-AUC Curve Test data

*Confusion matrix in Appendix for all models or individually against model above,

Model Comparison Insights

- "SVM," "CatBoost," exhibit **excellent performance** with high recall for Churn (minority class). They successfully identify customers at risk of churning, minimizing false negatives.
- SVM has the **highest recall** of the **minority class (95.20%)**, followed by **Catboost (93.69%)** and **Random Forest (88.63%)**.
- **Linear Discriminant Analysis (32.83%)** and **Naive Bayes (20.71%)** have the **lowest recall** of the minority class. These also exhibit high overfitting.
- In general, the **decision Tree, tuned models (Random Forest with Tuning and Tuned LDA)** have **lower recall** on the **minority class than the untuned models**. This is likely because the **tuned models are less overfit than the untuned models**. Further, these are more focused on maximizing overall accuracy, which can lead to sacrificing recall on the minority class
- The accuracy and other measuring parameter of a model can be further improved by trying various other combinations of hyperparameter. It is an iterative process.

Most Optimum Model & Business Implication

- If the business is looking for a model, which can help them to **target potential churning customers and thus helping them to reduce client acquisition cost than model with higher precision makes sense**. As in this correctness of predicting churners is more important. Lower the percentage means that the company lost money on the respective lead.
- If the business is looking to know **high number of churners irrespective of cost of acquisition than recall makes more sense**. Here they want to maximise prediction of employee opting for package.
- In the second case, the business should use the **SVM** model. It has the **highest recall (95.20%)**, meaning that it is **very good at identifying churn customers**. This is important because it allows the business to target their churn prevention efforts to the customers who are most likely to churn.
- For a **balanced model** which provided a good trade-off between precision & recall, we can consider **Catboost model** with recall as **93.69% & having an excellent f1-score of 95.87% for minority class**. The model also has best **precision of 98.15%** among the models.
- It's important to note that both **SVM** and **CatBoost** might result in some false positives (high precision) which could lead to additional retention costs. The business need to weigh the cost of false positives against the benefits of identifying potential churners.
- Using above model for predictions, the **business can follow below steps to reduce churn** basis the most important Features.
 - **Introduce pre-defined customer segmentation** and according to customers' needs and usage. Business can segment customers based on **Tenure, complaints, City Tier, Payment** etc. and can have different strategy for each set of customers
 - **Increase customers loyalty** as compared to competitors, **customer with high tenure tends to churn less. (Tenure- Most important feature)**
 - Business needs to **increase in visibility in Tier-2 city** for better customer acquisition. The business can **introduce curated contents for Tier 2 cities** as same content might not be liked by all city tiers. **(City Tier)**
 - Delighting customers by **improving customer services** as customer who complaint tends to churn (**Complaint_Ly**).
 - **Rather providing cashbacks** to future churning customers company should focus to retain them by providing them some exclusive offers like **account upgrade to higher segment** which can reduce the company burden on cashbacks.
 - Business can **introduce referral drive for existing customers** to acquire new customers as well as get them cashback/coupons/upgrade to next tier in exchange, which will help to reduce churn as well as acquire new users.

APPENDIX

Model Evaluation Parameters

When it comes to model validation of a classification problem statement we cannot just get relied on accuracy, we need to look at various others parameter like F1 score, Recall, precision, ROC curve and AUC score, along with confusion matrix. The details of these parameters are described below: -

Confusion Matrix:

Confusion Matrix usually causes a lot of confusion even in those who are using them regularly. Terms used in defining a confusion matrix are TP, TN, FP, and FN.

True Positive (TP): - The actual is positive in real and at the same time the prediction was classified correctly.

False Positive (FP): - The actual was actually negative but was falsely classified as positive.

True Negative: - The actuals were actually negative and was also classified as negative which is the right thing to do.

False Negative: - The actuals were actually positive but was falsely classified as negative.

	Predicted Positive	Predicted Negative	
Actual Positive	TP <i>True Positive</i>	FN <i>False Negative</i>	Sensitivity $\frac{TP}{(TP + FN)}$
Actual Negative	FP <i>False Positive</i>	TN <i>True Negative</i>	Specificity $\frac{TN}{(TN + FP)}$
	Precision $\frac{TP}{(TP + FP)}$	Negative Predictive Value $\frac{TN}{(TN + FN)}$	Accuracy $\frac{TP + TN}{(TP + TN + FP + FN)}$

Various Components Of Classification Report: -

Accuracy: This term tells us how many right classifications were made out of all the classifications.

$$\text{Accuracy} = (TP + TN) / (TP + FP + TN + FN)$$

Precision: Out of all that were marked as positive, how many are actually truly positive.

$$\text{Precision} = TP / (TP + FP)$$

Recall or Sensitivity: Out of all the actual real positive cases, how many were identified as positive.

$$\text{Recall} = TP / (TN + FN)$$

Specificity: Out of all the real negative cases, how many were identified as negative.

$$\text{Specificity} = TN / (TN + FP)$$

F1-Score: As we saw above, sometimes we need to give weightage to FP and sometimes to FN. F1 score is a weighted average of Precision and Recall, which means there is equal importance given to FP and FN. This is a very useful metric compared to "Accuracy". The problem with using accuracy is that if we have a highly imbalanced dataset for training (for example, a training dataset with 95% positive class and 5% negative class), the model will end up learning how to predict the positive class properly and will not learn how to identify the negative class. But the model will still have very high accuracy in the test dataset too as it will know how to identify the positives really well.

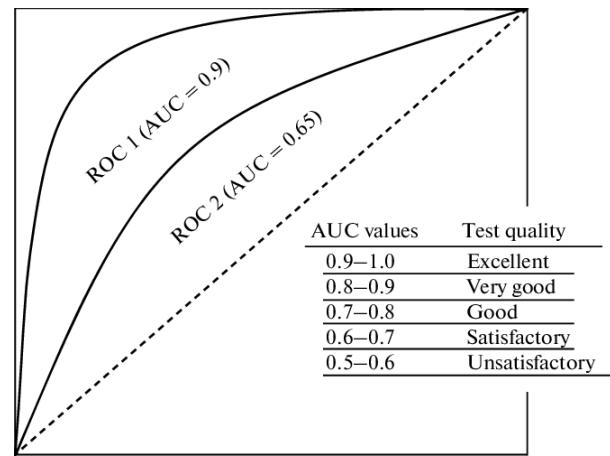
$$F1 \text{ score} = 2 * (\text{Precision} * \text{Recall}) / (\text{Precision} + \text{Recall})$$

Area Under Curve (AUC) and ROC Curve:

AUC or Area Under Curve is used in conjunction with ROC Curve which is Receiver Operating Characteristics Curve. AUC is the area under the ROC Curve. So, let's first understand the ROC Curve. A ROC Curve is drawn by plotting TPR or True Positive Rate or Recall or Sensitivity (which we saw above) in the y-axis against FPR or False Positive Rate in the x-axis. $FPR = 1 - Specificity$ (which we saw above).

$$TPR = TP / (TP + FN)$$

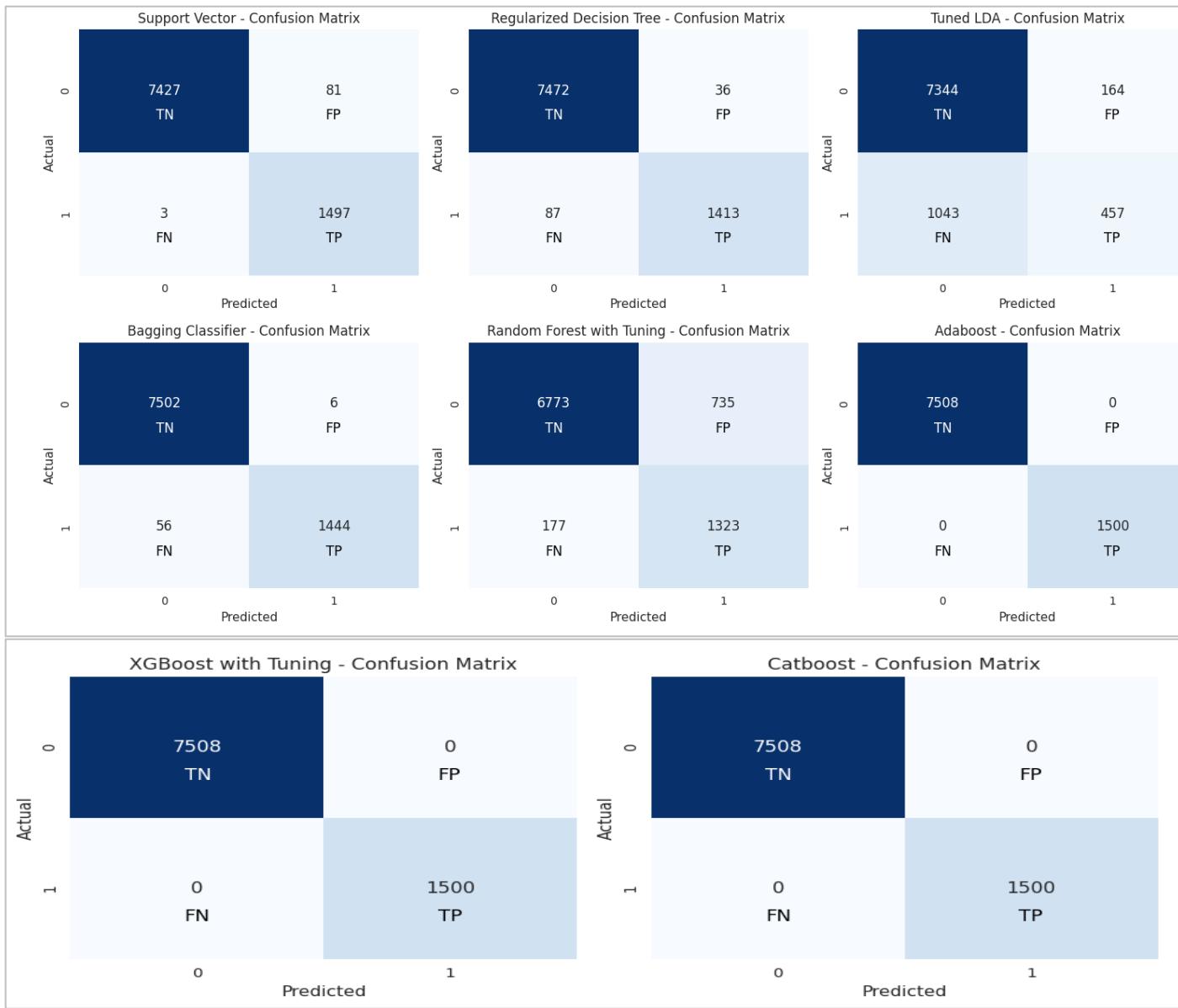
$$FPR = 1 - TN / (TN + FP) = FP / (TN + FP)$$



When we want to select the best model, we want a model that is closest to the perfect model. In other words, a model with AUC close to 1. When we say a model has a high AUC score, it means the model's ability to separate the classes is very high (high separability). This is a very important metric that should be checked while selecting a classification model.

Confusion Matrix of all models on Train data

		Decision Tree - Confusion Matrix		Random Forest - Confusion Matrix		Linear Discriminant Analysis - Confusion Matrix	
Actual	Predicted	0	1	0	1	0	1
		TN	FP	TN	FP	TN	FP
0	0	7508	0	7508	0	7344	164
	1	0	1500	0	1500	1043	457
		FN	TP	FN	TP	FN	TP
		0	1	0	1	0	1
		KNN - Confusion Matrix		Naive Bayes - Confusion Matrix		Neural Network - Confusion Matrix	
Actual	Predicted	0	1	0	1	0	1
		TN	FP	TN	FP	TN	FP
0	0	7488	20	7323	185	7493	15
	1	38	1462	1212	288	253	1247
		FN	TP	FN	TP	FN	TP
		0	1	0	1	0	1



Confusion Matrix of all models on Test data





Evaluation report of all models on Test data

Models	Metric Comparison Train Data				Metric Comparison Test Data				
	Accuracy	Precision	Recall	F1 Score	Accuracy	Precision	Recall	F1 Score	
Decision Tree	1.00	1.00	1.00	1.00	0.94	0.84	0.84	0.84	
Random Forest	1.00	1.00	1.00	1.00	0.97	0.96	0.89	0.92	
Linear Discriminant Analysis	0.87	0.74	0.30	0.43	0.86	0.76	0.33	0.46	
KNN	0.99	0.99	0.97	0.98	0.97	0.96	0.89	0.92	
Naive Bayes	0.84	0.61	0.19	0.29	0.84	0.67	0.21	0.32	
Neural Network	0.97	0.99	0.83	0.90	0.94	0.93	0.74	0.82	
Support Vector	0.99	0.95	1.00	0.97	0.97	0.91	0.95	0.93	
Regularized Decision Tree	0.99	0.98	0.94	0.96	0.94	0.84	0.78	0.81	
Tuned LDA	0.87	0.74	0.30	0.43	0.86	0.76	0.33	0.46	
Bagging Classifier	0.99	1.00	0.96	0.98	0.96	0.93	0.83	0.88	
Random Forest with Tuning	0.90	0.64	0.88	0.74	0.89	0.63	0.84	0.72	
Adaboost	1.00	1.00	1.00	1.00	0.97	0.98	0.87	0.92	
XGBoost with Tuning	1.00	1.00	1.00	1.00	0.97	0.96	0.87	0.91	
Catboost	1.00	1.00	1.00	1.00	0.99	0.98	0.94	0.96	
	Metrics					Metrics			