10/6/2024

CAPSTONE PROJECT

Customer Churn

Love Kumar Gaur

Table of Contents:

Sr.No.	Name of Figures	Page Number

1. Introduction - What did you wish to achieve while doing the project ?

A) Defining problem statement - An E Commerce company or DTH (you can choose either of these two domains) provider is facing a lot of competition in the current market and it has become a challenge to retain the existing customers in the current situation. Hence, the company wants to develop a model through which they can churn prediction of the accounts and provide segmented offers to the potential churners. In this company, account churn is a major thing because 1 account can have multiple customers. Hence by losing one account the company might be losing more than one customer. You have been assigned to develop a churn prediction model for this company and provide business recommendations on the campaign. Your campaign suggestion should be unique and be very clear on the campaign offer because your recommendation will go through the revenue assurance team. If they find that you are giving a lot of free (or subsidized) stuff thereby making a loss to the company; they are not going to approve of your recommendation. Hence be very careful while providing campaign recommendations.

B) Need for the study/project - Nowadays there is huge competition in the market, as competitor companies attract customers by providing offers and better services. To every organization, it is more difficult and costly to acquire new customers than retain the existing customers. So, we need to retain the existing customers within a company Hence, we need to build model which can analysis their past

data and predict whether customers will churn or not. Once we identify correctly churner customers, we can intervene before they leave. The study is about reducing the overall churn rate and increasing profitability. In this business model, one account may represent multiple customers, such as families sharing an account for purchases or businesses using the platform for employee purchases. Losing an account could mean losing multiple customers at once, resulting in a significant revenue impact. This study will help the company proactively prevent this by identifying at-risk accounts early and taking steps to retain them. The primary objective of this study is to develop a data-driven churn prediction model that accurately identifies at-risk customers and to recommend targeted, cost-effective retention campaigns that maximize customer retention while minimizing revenue loss. This will enable the company to sustain its competitive position in the market by retaining valuable accounts and maintaining profitability.

C) Understanding business/social opportunity - This is a case study of E Commerce company or DTH where they have a unique Account id where multiple customers can be tagged. Business opportunities is to address the critical business churn which impact on a company profitability, loyalty and trust and company reputation. The social opportunities are improving the satisfaction of customers; by preventing churn through timely interventions and personalized offers, the company can improve overall customer satisfaction. When customers feel understood and valued, it enhances their experience with the brand. This contributes to a positive customer relationship, which benefits both the customers and the company. The other social opportunity is creating social value and customer empowerment.

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

- Import the necessary libraries and Read the dataset.

Shape of the data

```
(11260, 19)
```

There are 11,260 observations and 19 columns

Check the first five rows using head function

	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	Compl
0	20000	1	4	3.0	6.0	Debit Card	Female	3.0	3	Super	2.0	Single	9	
1	20001	1	0	1.0	8.0	UPI	Male	3.0	4	Regular Plus	3.0	Single	7	
2	20002	1	0	1.0	30.0	Debit Card	Male	2.0	4	Regular Plus	3.0	Single	6	
3	20003	1	0	3.0	15.0	Debit Card	Male	2.0	4	Super	5.0	Single	8	
4	20004	1	0	1.0	12.0	Credit Card	Male	2.0	3	Regular Plus	5.0	Single	3	

Complain_ly	rev_growth_yoy	coupon_used_for_payment	Day_Since_CC_connect	cashback	Login_device
1.0	11	1	5	159.93	Mobile
1.0	15	0	0	120.9	Mobile
1.0	14	0	3	NaN	Mobile
0.0	23	0	3	134.07	Mobile
0.0	11	1	3	129.6	Mobile

Table 1: Top 5 rows data

Check the last five rows using tail function

	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	C
11255	31255	0	10	1.0	34.0	Credit Card	Male	3.0	2	Super	1.0	Married	9	
11256	31256	0	13	1.0	19.0	Credit Card	Male	3.0	5	HNI	5.0	Married	7	
11257	31257	0	1	1.0	14.0	Debit Card	Male	3.0	2	Super	4.0	Married	7	
11258	31258	0	23	3.0	11.0	Credit Card	Male	4.0	5	Super	4.0	Married	7	
11259	31259	0	8	1.0	22.0	Credit Card	Male	3.0	2	Super	3.0	Married	5	

Complain_ly	rev_growth_yoy	coupon_used_for_payment	Day_Since_CC_connect	cashback	Login_device
0.0	19	1	4	153.71	Computer
0.0	16	1	8	226.91	Mobile
1.0	22	1	4	191.42	Mobile
0.0	16	2	9	179.9	Computer
0.0	13	2	3	175.04	Mobile

Table 2: Last 5 rows data

Let's check the datatypes Information and descriptive statistics

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

Table 3: Datatypes information

Insights:

- There are 11260 observations and 19 variables in the dataset. However, there are missing values as well in many columns.
- There are 5 variables float type, 12 are Object datatype and 2 are integer datatypes.

	count	unique	top	freq	mean	std	min	25%	50%	75%	max
AccountID	11260.0	NaN	NaN	NaN	25629.5	3250.62635	20000.0	22814.75	25629.5	28444.25	31259.0
Churn	11260.0	NaN	NaN	NaN	0.168384	0.374223	0.0	0.0	0.0	0.0	1.0
Tenure	11158.0	38.0	1.0	1351.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN
City_Tier	11148.0	NaN	NaN	NaN	1.653929	0.915015	1.0	1.0	1.0	3.0	3.0
CC_Contacted_LY	11158.0	NaN	NaN	NaN	17.867091	8.853269	4.0	11.0	16.0	23.0	132.0
Payment	11151	5	Debit Card	4587	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Gender	11152	4	Male	6328	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Service_Score	11162.0	NaN	NaN	NaN	2.902526	0.725584	0.0	2.0	3.0	3.0	5.0
Account_user_count	11148.0	7.0	4.0	4569.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN
account_segment	11163	7	Super	4062	NaN	NaN	NaN	NaN	NaN	NaN	NaN
CC_Agent_Score	11144.0	NaN	NaN	NaN	3.066493	1.379772	1.0	2.0	3.0	4.0	5.0
Marital_Status	11048	3	Married	5860	NaN	NaN	NaN	NaN	NaN	NaN	NaN
rev_per_month	11158.0	59.0	3.0	1746.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Complain_ly	10903.0	NaN	NaN	NaN	0.285334	0.451594	0.0	0.0	0.0	1.0	1.0
rev_growth_yoy	11260.0	20.0	14.0	1524.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN
coupon_used_for_payment	11260.0	20.0	1.0	4373.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Day_Since_CC_connect	10903.0	24.0	3.0	1816.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN
cashback	10789.0	5693.0	155.62	10.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Login_device	11039	3	Mobile	7482	NaN	NaN	NaN	NaN	NaN	NaN	NaN

Table 4: Descriptive statistics

Insights:

I have included all the variables in the descriptive summary table. Because some important columns were missed as they are updated as object. There could be a chance that those columns could have special character in it. We will determine them later to know the exact reason.

1. Tenure has 38 unique values where tenure 1 has the highest frequency. This column should be present as numerical, but it is available as object. There are missing values as well.
2. Column City_Tier has 50% value is 1, 75% and Maximum value is 3. This column seems to be categorical type as nature but updated as numerical.
3. Customer contacted last year mean is 17.867091 which suggest that on average, customers has contacted to customer care 18 times in past. Standard deviation is 8.853269, Median value is 16. The 75% data value is 23.00 but maximum value is 132 which is very far to the mean as it suggests that there are some high value presents in the dataset.
4. Column Payment has 5 unique values where debit has highest frequency.
5. Gender has 4 unique values where Male has the highest frequency. There is noticeable thing that gender can not have 4 unique values. Hence, we have to check this variable which are the 4 unique variables. Will check in further steps.
6. Service_Score has minimum value is 0, 25% is 2, 75% value is 3 and maximum is 5. Satisfaction score given by customers of the account on service provided by company. So, this variable should be changed as categorial due to its nature.
7. Account_user_count, Number of customers tagged with this account. It seems that this datatype is updated as a categorial so we will be checking the reason as why it is updated as categorical.
8. Account segment, there are total 7 unique values where super has the highest frequency.
9. CC_Agent_Score, Satisfaction score given by customers of the account on customer care service. So, this variable should be changed as categorial due to its nature.
10. Marital_Status has 3 unique variables where Married has the highest frequency.
11. rev_per_month, this is a very important feature but there is no more information available in descriptive summary as this data type is updated as categorical. We will determine why it is updated as categorical.
12. Complain_ly, Any complaints has been raised by account in last 12 months. It has 0 and 1, 0 means no and 1 means yes.
13. Columns **rev_growth_yoy**, **coupon_used_for_payment, Day_Since_CC_connect and cashback:** There are no more information available in descriptive summary for these columns as their data types are updated as categorical. We will determine why it is updated as categorical.
14. Column Login_device, has 3 unique values where Mobile has the highest frequency.

Let's check irregularities in the data

Upon checking, Column Gender It is observed that there is inconsistency found in data. Male is updated in two way as "Male or M", Similarly, Female is also updated in two way as Female or F. hence, we will keep consistent this column and replace all Male keyword to "M" and all Female keyword to "F".

Gender	count
M	6704
F	4448

dtype: int64

Similarly, Column account segment has inconsistency as there are two values updated for Regular Plus and Regular +, Super Plus and Super +. Hence, we will make this column consistent by replacing Regular + to Regular plus and Super + to Super Plus.

account_segment	count
Regular Plus	4124
Super	4062
HNI	1639
Super Plus	818
Regular	520

dtype: int64

Let's check the following columns to determine why they are updated as categorical since their nature are numerical.

1. TENURE
2. ACCOUNT_USER_COUNT
3. REV_PER_MONTH
4. REV_GROWTH_YOY
5. COUPON_USED_FOR_PAYMENT
6. DAY_SINCE_CC_CONNECT
7. CASHBACK

After reviewing these columns, it is observed that they have special character [$#@+*&]. Therefore, we will replace this special character with null values.

Checking missing values

AccountID	0.00
Churn	0.00
Tenure	1.94
City_Tier	0.99
CC_Contacted_LY	0.91
Payment	0.97
Gender	0.96
Service_Score	0.87
Account_user_count	3.94
account_segment	0.86
CC_Agent_Score	1.03
Marital_Status	1.88
rev_per_month	7.02
Complain_ly	3.17
rev_growth_yoy	0.03
coupon_used_for_payment	0.03
Day_Since_CC_connect	3.18
cashback	4.20
Login_device	6.75

dtype: float64

Table 5: Sum of Missing values

There are missing values present in each column except Account id and Churn. Login device has the highest missing values approx. 7%.
Let's drop the column **AccountID** since it does not provide any help with prediction.

Checking duplicate rows: Upon checking the duplicate, there are no duplicate rows.
Let's change those columns which are categorical by nature but present as numerical in the database.

2. EDA - Univariate / Bi-variate / multi-variate analysis to understand relationship b/w variables. - Both visual and non-visual understanding of the data.

A) Univariate analysis of numerical columns

Tenure

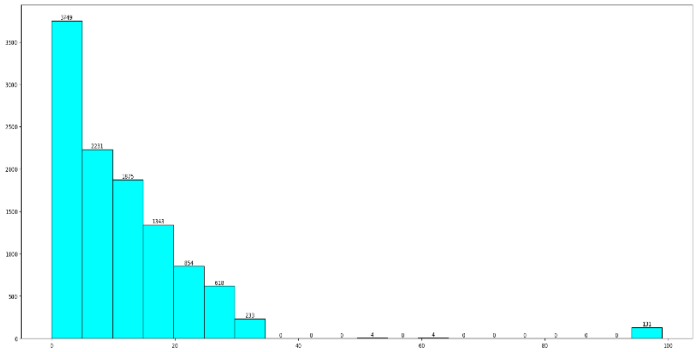


Figure 1: Histogram of Tenure

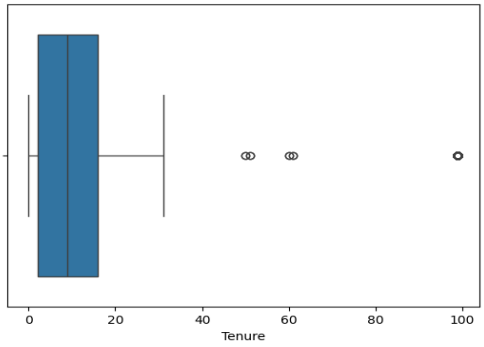


Figure 2: BoxPlot of Tenure

Observation

1. The mean of tenure is 11.025086 which means that on average, the customer has been associated with service for approx. 11 months.
2. The minimum value is 0, which suggest that some customers have recently joined.
3. 25% of customers have 2 months tenure and 50% of customers have a 9-months tenure.
4. 75% of customers are associated with the service for 16 months tenure.
5. Maximum tenure is 99 which is approx. 8 years. The data has right skewed and has outliers in dataset.

CC_Contacted_LY

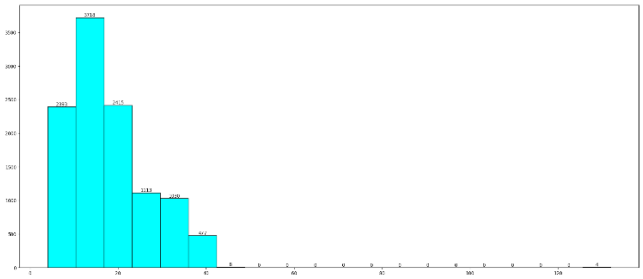


Figure 3: Histogram of Contacted_LY

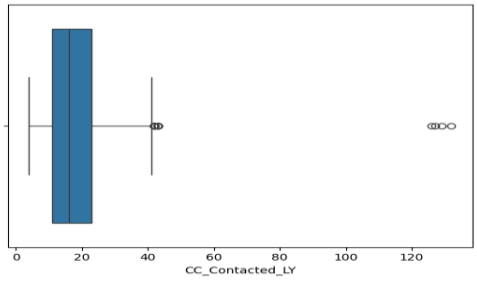


Figure 4: Boxplot of Contacted_LY

Observation

1. Customer contacted last year mean is 17.867091 which suggests that on average, customers has contacted to customer care 18 times in past.
2. The 75% data value is 23.00 but maximum value is 132, which is very far to the mean as it suggests that there are some high value presents in the dataset.

Account_user_count

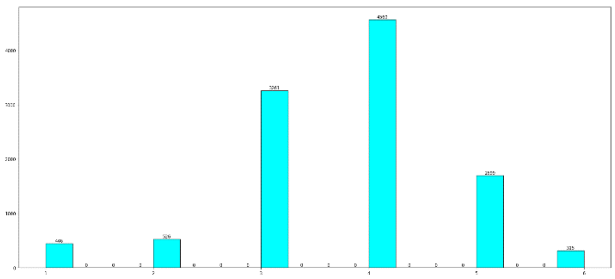


Figure 5: Histogram of Account user count

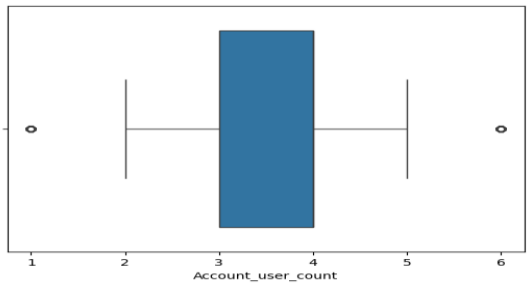


Figure 6: Boxplot of Account user count

Observation

1. The average customers are tagged with an account is approx. 4.
2. The minimum customer tagged with an account is 1. 25% account has 3 customers.
3. 50% and 75% accounts have 4 customers. The maximum account has 6 customers

rev_per_month

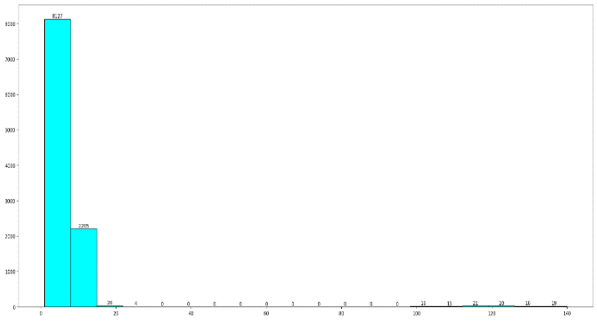


Figure 7: Histogram of Revenue per month

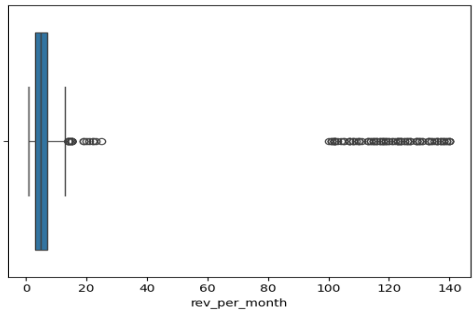


Figure 8: Boxplot of Revenue per month

Observation

1. The mean value of monthly average revenue is 6.37. The standard deviation of 11.91 indicates significant variability in the monthly average revenue across different accounts. Some accounts generate much higher revenue than others, leading to a widespread in the data.
2. The minimum monthly average revenue is 1, 25% monthly average revenue value is 3 and 50% and 75% monthly average revenue values are 5 and 7.
3. The maximum monthly average revenue is 140 which means that there are extremely higher revenue value customers.

rev_growth_yoy

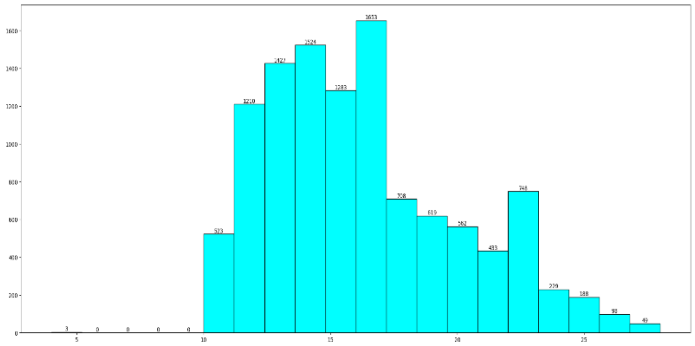


Figure 9: Histogram of rev_growth_yoy

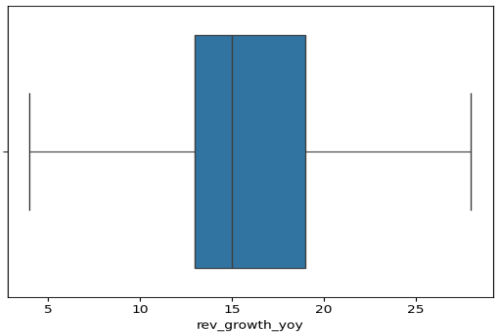


Figure 10: Boxplot of Revenue growth yoy

Observation

1. The mean value of revenue growth percentage of the account is 16. The standard deviation of 3.757721, indicating moderate variability in the data. The values are spread around the mean, but not too widely.
2. The minimum revenue growth percentage of the account is 4.
3. 25% monthly revenue growth percentage of the account is 13 and 50% and 75% revenue growth percentage of the account are 15 and 19.
4. The maximum revenue growth percentage of the account is 28. By seeing boxplot, there is no outlier present in data.

Coupon_used_for_payment

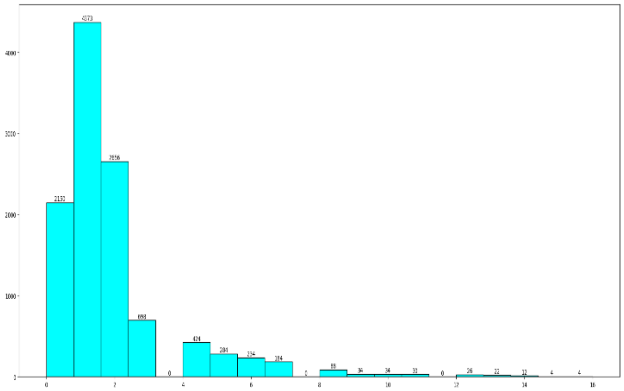


Figure 11: Histogram of coupon_used_for_payment

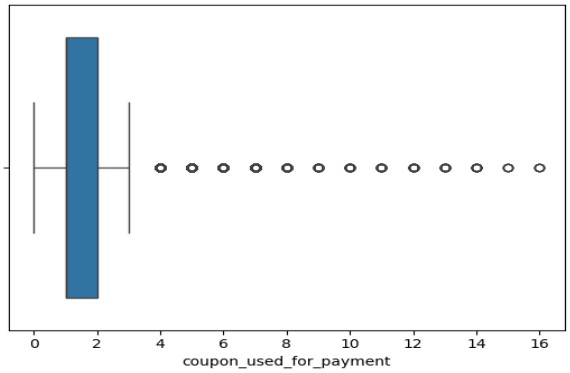


Figure 12: Boxplot of coupon_used_for_payment

Observation

1. The average of coupon_used_for_payment by customer is approx. 2 coupons.
2. The minimum coupon used is 0. This means that there are few customers who never used any coupon.
3. There are 50% of customers who have used 1 coupon, 75% of customers have used 2 coupons.
4. Maximum coupons are used 16.

Day_Since_CC_connect

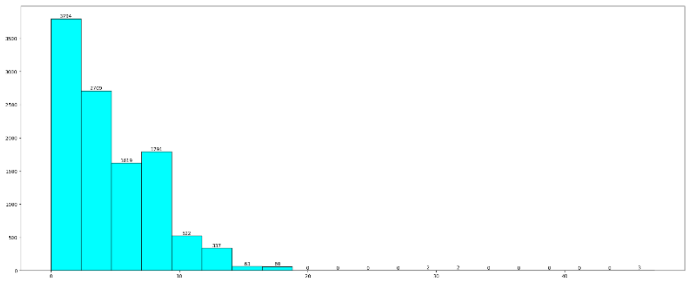


Figure 13: Histogram of Day_Since_CC_connect

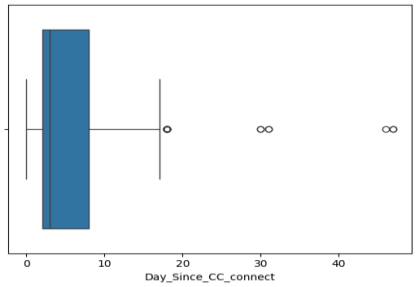


Figure 14: Boxplot of Day_Since_CC_connect

Observation

1. The average number of 4.63 days since no customers in the account has contacted customer care.
2. The minimum coupon used is 0. This means that there are few customers who never used any coupon, There are 50% of customers who have used 1 coupon and 75% of customers have used 2 coupons, Maximum coupons are used 16.

Cashback

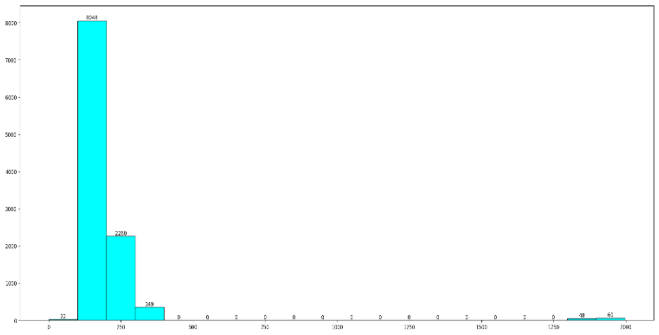


Figure 15: Histogram of Cashback

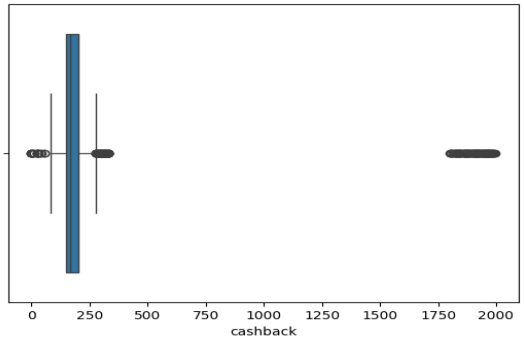


Figure 16: Boxplot of Cashback

Observation

1. The average monthly average cashback is approx. 196.
2. The minimum monthly average cashback is 0. This means that there are few customers who never received cashback, 75% of customers have monthly average cashback received 200.
3. The maximum cashback amount is 1997, indicating a significant outlier.

B) Univariate analysis of categorical columns

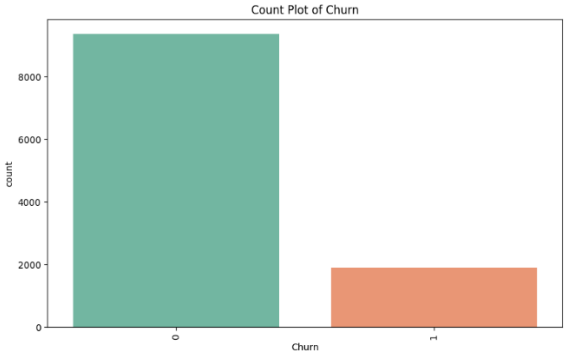


Figure 17: Countplot of Churn

Observation

Churn is the target variable and there are 83% customers non-churners, and 17% customers are about to churn. Instances of one of the two classes is higher than the other, in another way, the number of observations is not the same for all the classes in a classification dataset. This means that data is unbalanced.

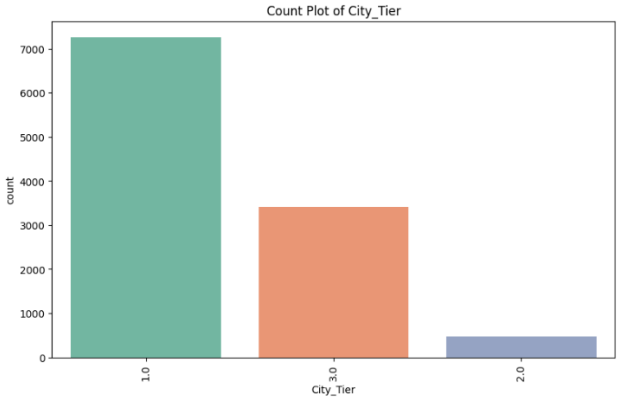


Figure 18: Count plot of City Tier

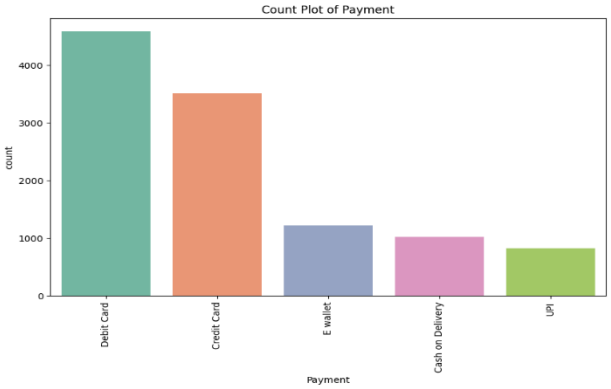


Figure 19: Count plot of Payment

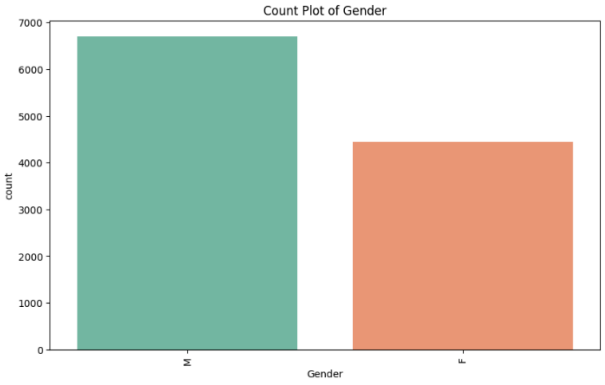


Figure 20: Count plot of Gender

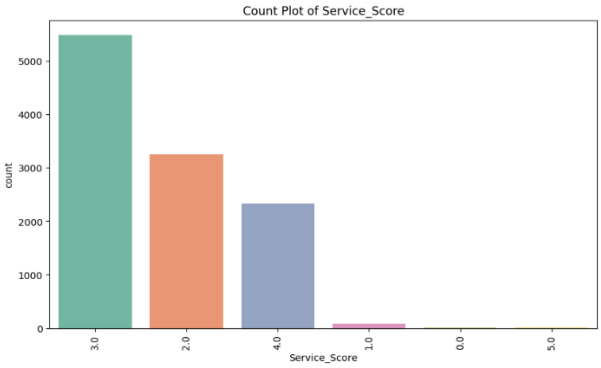


Figure 21: Count plot of Service score

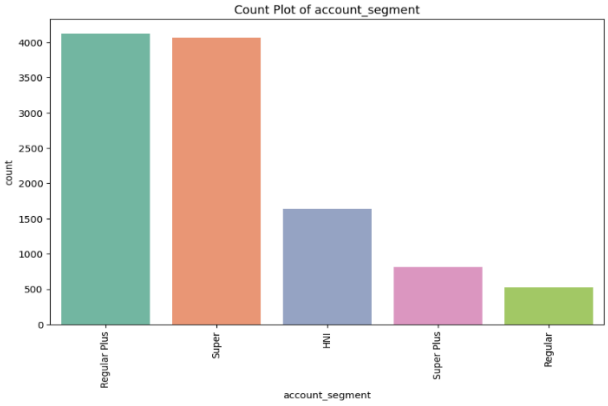


Figure 22: Count plot of Account segment

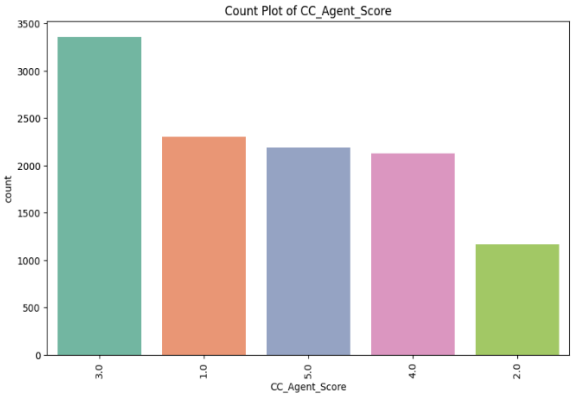


Figure 23: Count plot of CC_Agent_score

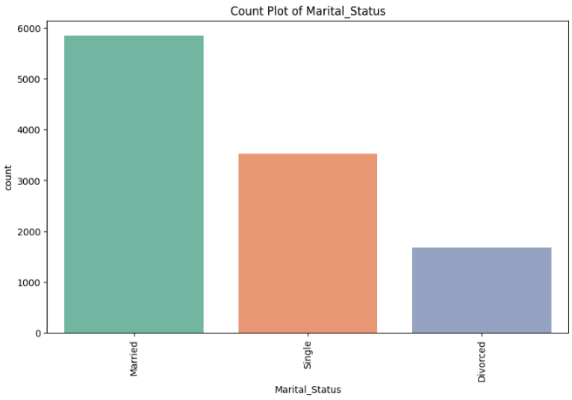


Figure 24: Count plot of Marital status

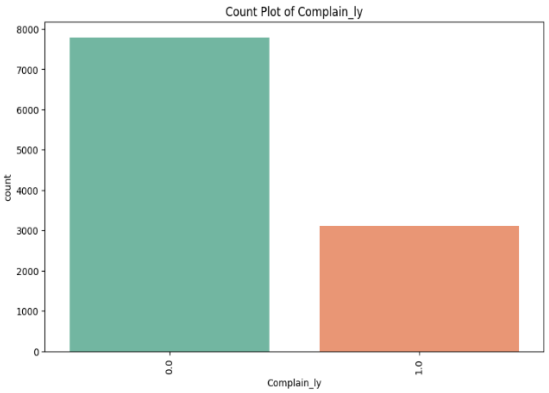


Figure 25: Count plot of complain ly

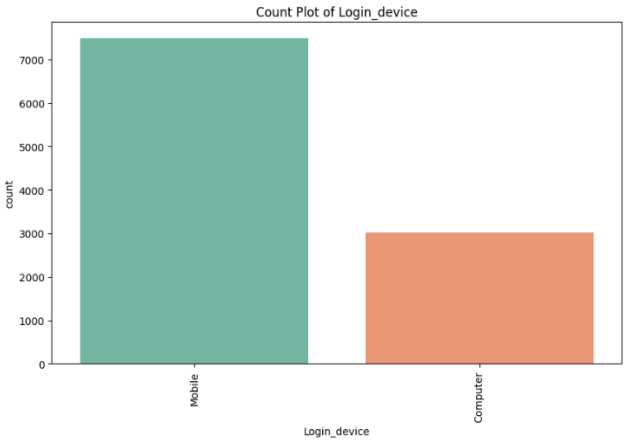


Figure 26: Count plot of Login device

C) Bivariate Analysis

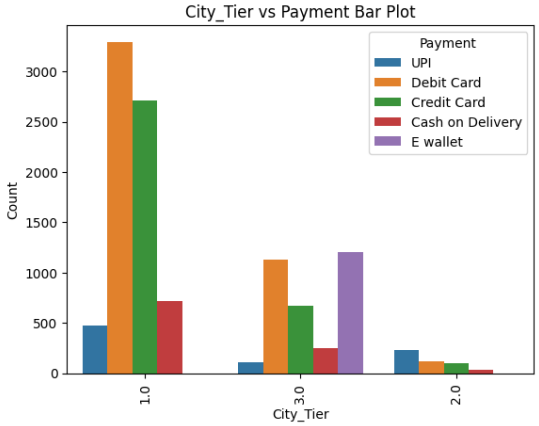


Figure 27: Bar plot of City tier vs Payment

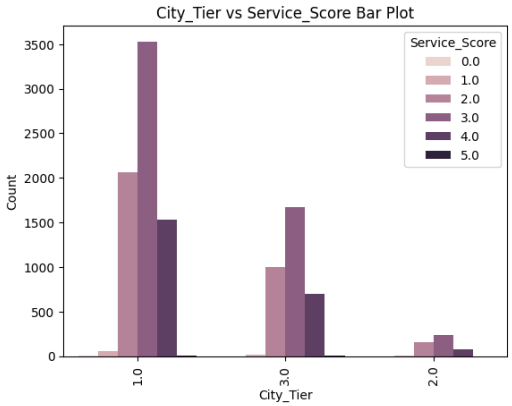


Figure 28: Bar plot of City tier vs Service score

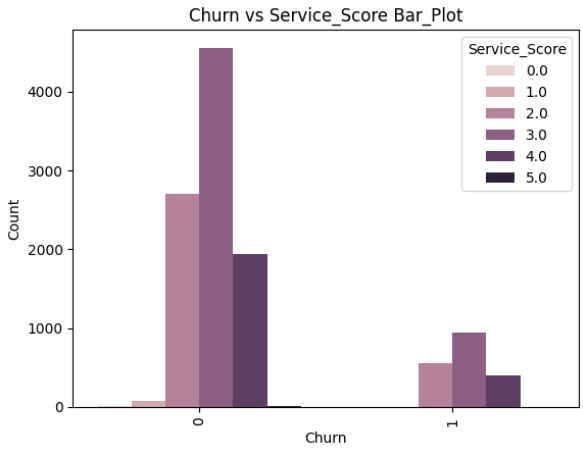


Figure 29: Bar plot of churn vs Service score

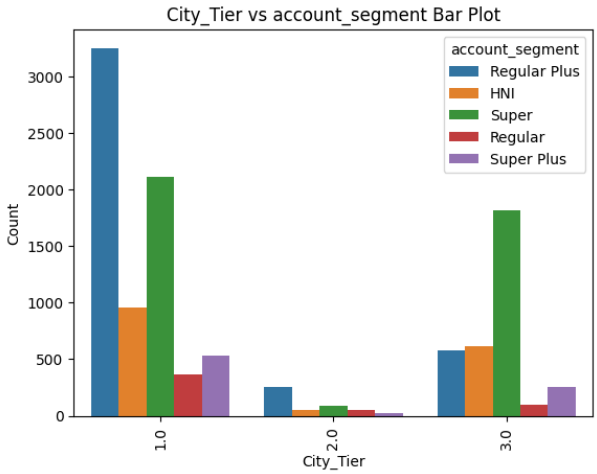


Figure 30: Bar plot of City tier vs account segment

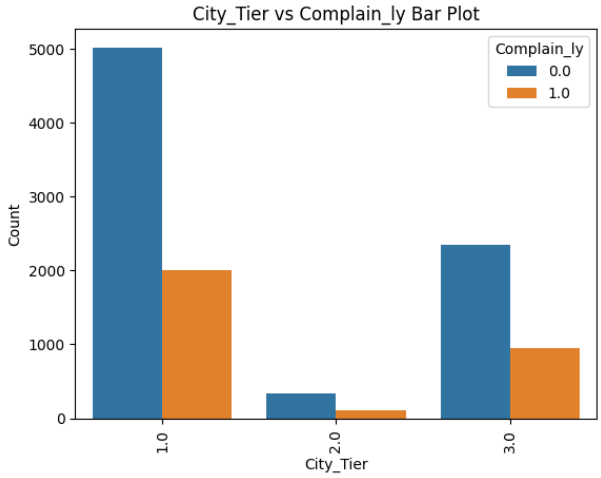


Figure 31: Bar plot of City tier vs complain raised

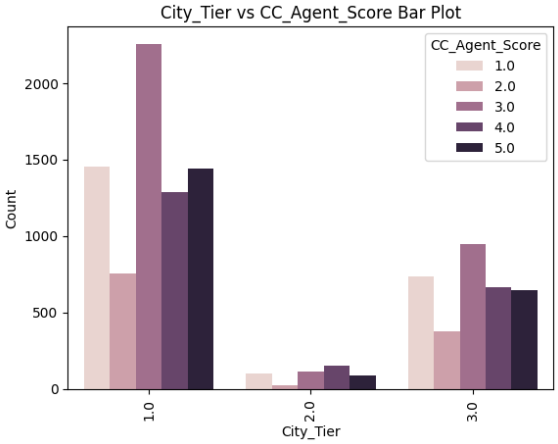


Figure 32: Bar plot of City tier vs CC_agent score

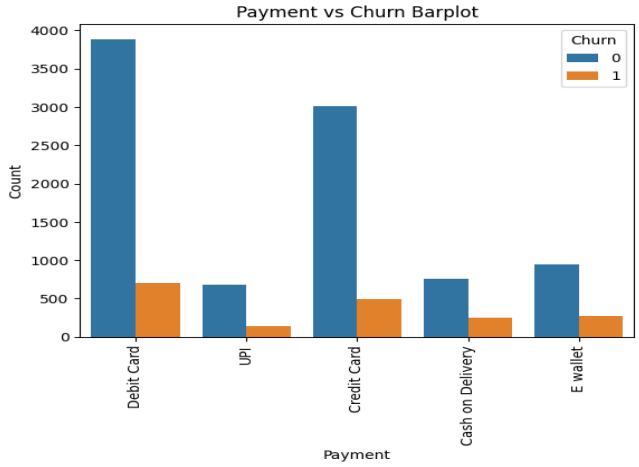


Figure 33: Bar plot of Payment vs Churn

Churn	0	1
Payment		
Cash on Delivery	0.749507	0.250493
Credit Card	0.857875	0.142125
Debit Card	0.846959	0.153041
E wallet	0.773213	0.226787
UPI	0.826034	0.173966

Table 6: Payment vs Churn

Churn	0	1
account_segment		
HNI	0.844417	0.155583
Regular	0.923077	0.076923
Regular Plus	0.726722	0.273278
Super	0.897587	0.102413
Super Plus	0.951100	0.048900

Table 7: Account segment vs Churn

Complain_ly	0.0	1.0
Gender		
F	0.703912	0.296088
M	0.736641	0.263359

Table 8: Gender vs Complain_ly

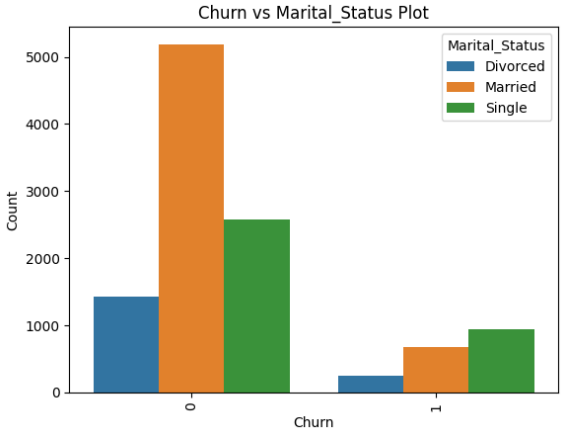


Figure 34: Bar plot of Marital status vs Churn

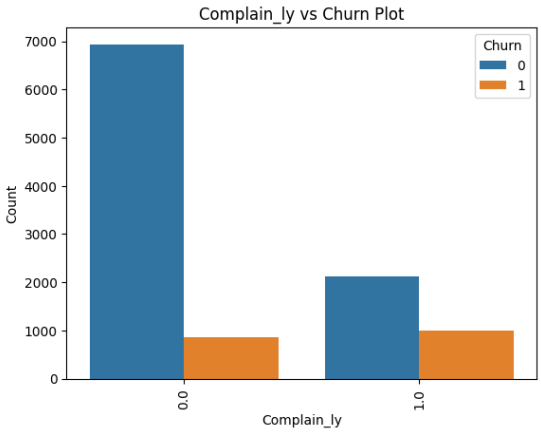


Figure 35: Bar plot of Complain vs Churn

Observation

1. There are more population from tier 1 cities then tier 3 cities and the lowest number of tier 2 cities.
2. Debit card is the most preferred mode of payment then credit card is in second place. E-wallet is at third place, cash on delivery is at fourth place and UPI is the least mode of payment has used by the customers.
3. There are more number of male customers than female.
4. The most satisfaction score is given 3 by customers on service provided by company. Then score 2 has given after score 3, score 4 is standing at 3 places. The least satisfaction score is given 5.
5. Regular plus account has more number of spend and Super has the second highest in the list. HNI is standing at third place, Super plus account is at 4th place. Regular is standing at last place.
6. Satisfaction score 3 given by customers of the account on customer care service provided by company is the highest then score 1 is placed at second place.
7. There are maximum population of customers are married then second highest customers are single and least population of customers are divorced.
8. There are more numbers of non-complaints than complaints.
9. Customers have used mobile devices rather than computer devices.
10. Tier 1 city customers have used Debit cards and credits cards more. They even did not use E wallet as payment mode.
11. Tier 3 city customers have used E wallet more, then debit card and credit card. They have used UPI is the least mode of the payment.
12. Tier 2 city customers have not used E wallet but have used more UPI than other mode of payment.
13. 28% of customers of Tier 1 and Tier 3 city have raised the complaints.24% of customers of Tier 2 have raised the complaints.
14. Tier 1 customers have given the highest score of 3 to customer care agent.
15. Tier 2 customers have given the highest score 4 to customer care agent.
16. Tier 3 customers have given the highest score 3 to customer care agent.
17. Cash on delivery and E Wallet customers are more likely to churn.
18. Regular plus customers are more likely to churn.
19. Super plus and Regular customers have less numbers of churn people.
20. 29% of females out of females are like to churn. Similarly, 26% of males are like to churn.

21. Single people are more to churn.
22. 52% of customers who raised a complain are likely to churn.

D) Bivariate Analysis for Categorical and Numerical

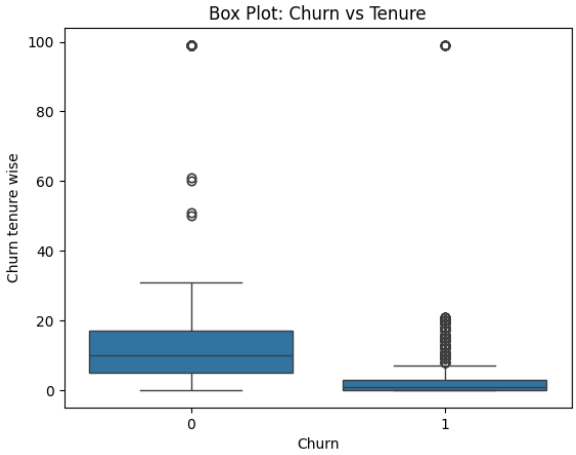


Figure 36: Box plot of Tenure vs Churn

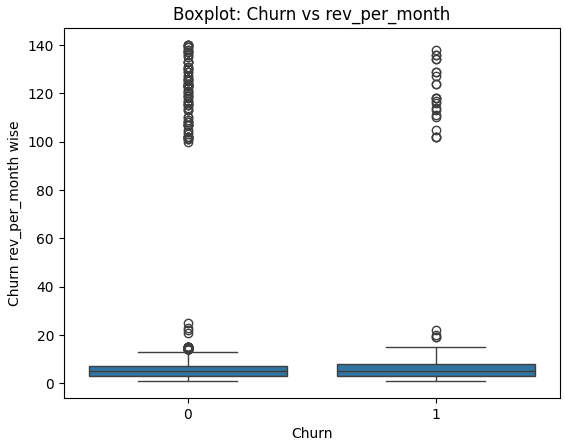


Figure 37: Box plot of Revenue per month vs Churn

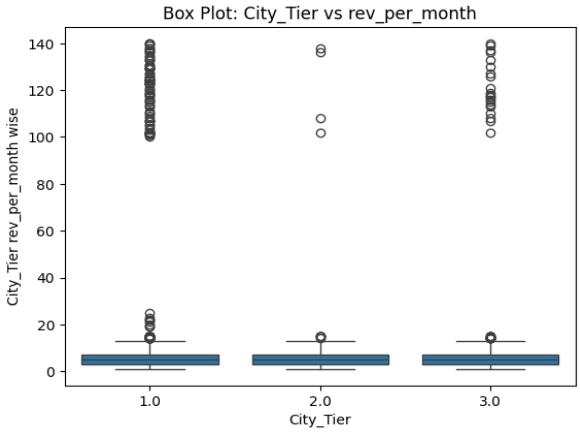


Figure 38: Box plot of City Tier vs Revenue per month

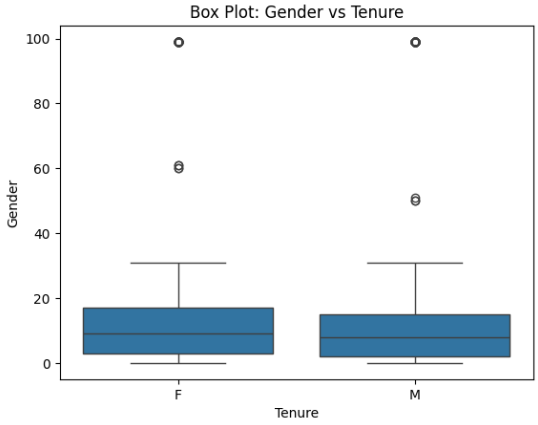


Figure 39: Box plot of Gender vs Tenure

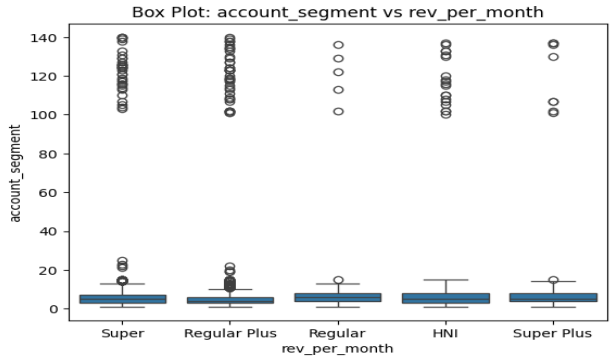


Figure 40: Box plot of Account segment vs Revenue per month

Observation

1. The median of non-churners is higher than churners. Non-churners distribution is wider, which suggests that non-churners customers' tenure is longer than churners. Hence, customers who have small tenure are more likely to churn.
2. The distribution of both churners and non-churners is very tight, which indicates that both groups' revenue is low. However, both groups have some customers whose revenue is high which are considered as outliers.
3. The distribution of each Tier looks similar, and all Tier have the outliers. However, Tier 1 and Tier 3 have a greater number of outliers than 2.
4. The median of female tenure distribution is slightly higher than male which indicates that Female tenure is slightly longer than male.
5. All groups distribution looks same, the median of regular category seems to be high than others and Regular median looks the lower among all.

E) Multivariate Analysis

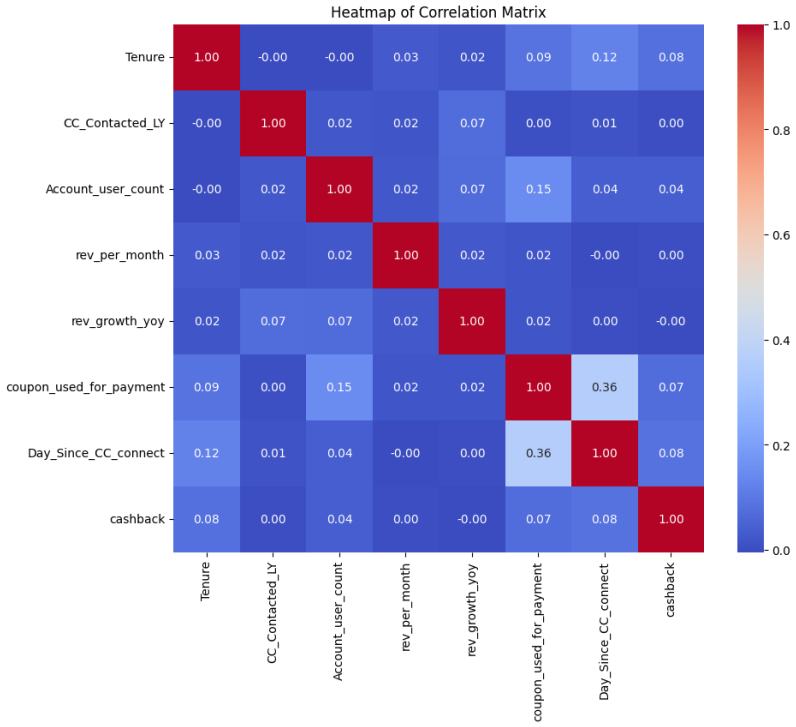


Figure 41: Heatmap

Observation

Most of the correlations in the matrix are close to zero, indicating weak or no linear relationship between the variables. This suggests that, in terms of linear relationships, multicollinearity may not be a significant concern in the data.

Data Cleaning and Pre-processing - Approach used for identifying and treating missing values and outlier treatment (and why) - Need for variable transformation (if any) - Variables removed or added and why (if any

Missing Value Treatment

1 - The following columns are treated with median for missing values.

1. Tenure
2. CC_Contacted_ly
3. Account_user_count
4. rev_growth_yoy
5. coupon_used_for_payment
6. Day_Since_CC_connect
7. Cashback
8. Account_user_count

2 - The following columns are treated with mode for missing values.

1. City_Tier
2. Payment
3. Gender
4. Service score
5. Account Segment
6. Marital_Status
7. Complain_ly
8. Login_device

3 – Below column is treated with KNN Imputer for missing values.

9. rev_per_month

Outliers check and Treatment

Before checking and treating the outliers, let's make two different data frame and copy all the independent variables to one dataframe and copy the target variable to another dataframe.

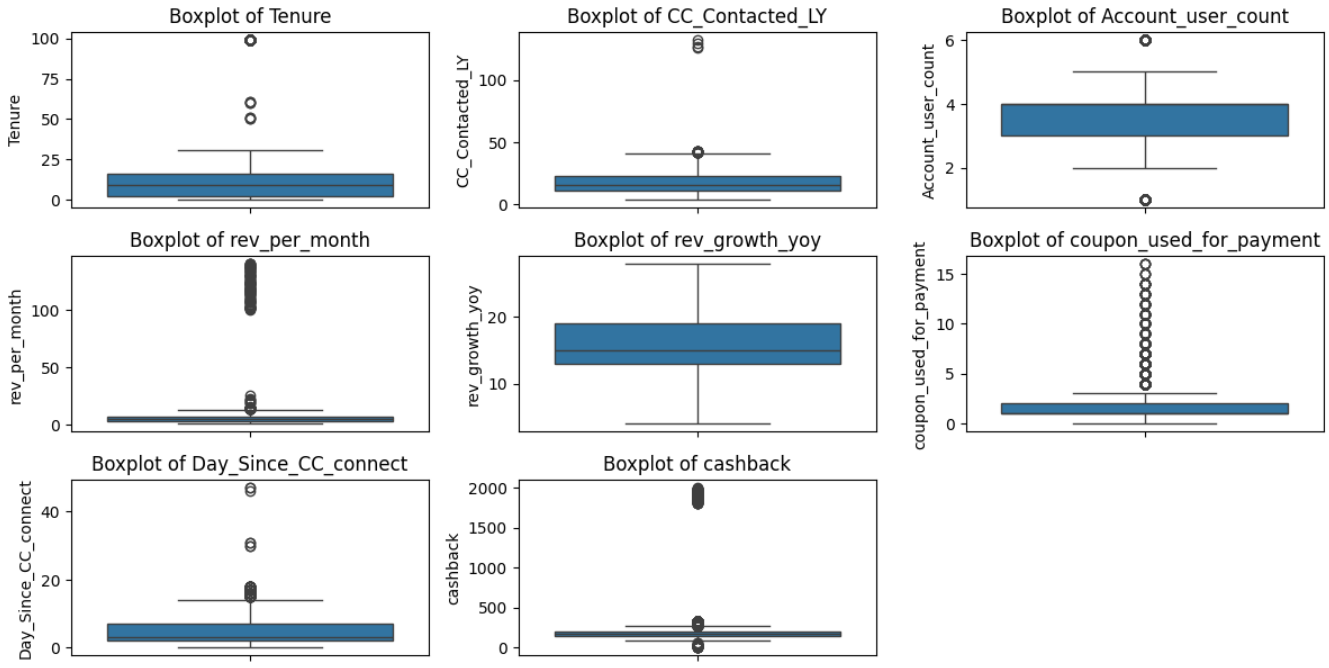


Figure 42: Checking outliers of numerical variables

Observation

We will check the outliers in numerical variables only as it does not make any sense to check for categorical variables. There are outliers present in column Tenure, CC_Contact_Ly, Account_user_count, Revenue per month, coupon used for paymen, Day since customer care connect and Cashback.
Based on 25 and 75 percent of Quantile, let's count of outliers in each column.

	0
Tenure	139
CC_Contacted_LY	42
Account_user_count	761
rev_per_month	185
rev_growth_yoy	0
coupon_used_for_payment	1380
Day_Since_CC_connect	130
cashback	986

Table 9: Checking outliers in each row

Let's remove the outliers with upper range and lower range with following formulas.
IQR=Q3-Q1
lower_range= Q1-(1.5 * IQR)
upper_range= Q3+(1.5 * IQR)

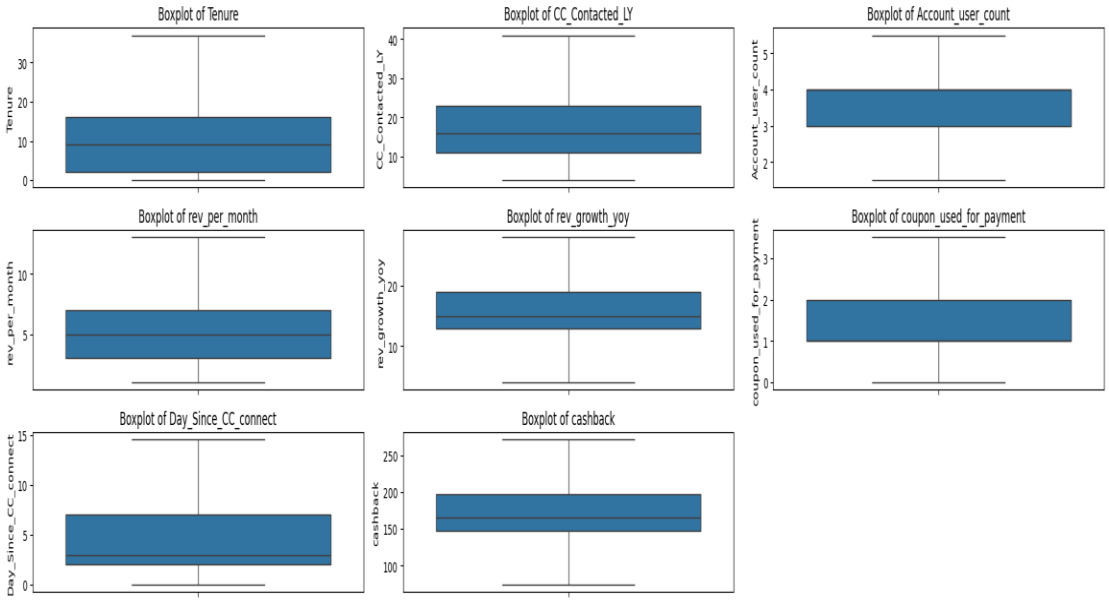


Figure 43: Checking outliers of numerical variables after treatment

Variable transformation - Let's do the encoding for the following variables. Since they are categorical variables and we cannot perform most statistical operations on a string variable, hence we want to turn the string variable into a numeric variable.

1) Column Gender and Login_device has encoded with Label Encoding
2) Column account_segment' has encoded with ordinal encoding as Regular will be marked as 1, Regular plus 2, Super will be 3, Super plus will be marked as 4 and HNI will be marked as 5.
3) Payment', 'Marital_Status' has encoded with One-Hot Encoding.

Now we see that there are different different dimensions in many variables, so we have to bring variables to the same scale, to have a mean of 0 and standard deviation is 1.

	Tenure	City_Tier	CC_Contacted_LY	Gender	Service_Score	Account_user_count	account_segment	CC_Agent_Score	rev_per_month	Complain_ly	...	cashback
0	4	3	6	0	3	3	3	2	9	1	...	159
1	0	1	8	1	3	4	2	3	7	1	...	120
2	0	1	30	1	2	4	2	3	6	1	...	165
3	0	3	15	1	2	4	3	5	8	0	...	134
4	0	1	12	1	2	3	2	5	3	0	...	129

5 rows × 20 columns

Table 10: Top 5 rows of data before Scale

Let's scale the data with the help of StandardScaler from sklearn.

	0	1	2	3	4	5	6	7	8	9	...	13	14	15	16	17
0	-0.703315	1.481914	-1.379652	-1.237528	0.133748	-0.710671	0.094301	-0.776488	1.275330	1.618461	...	-0.408571	0.605123	-0.314588	-0.673120	1.182280
1	-1.153334	-0.709334	-1.146110	0.808063	0.133748	0.339067	-0.817139	-0.047944	0.582018	1.618461	...	-1.303484	0.605123	-0.314588	-0.673120	-0.845824
2	-1.153334	-0.709334	1.422855	0.808063	-1.250443	0.339067	-0.817139	-0.047944	0.235363	1.618461	...	-0.270893	0.605123	-0.314588	-0.673120	1.182280
3	-1.153334	1.481914	-0.328712	0.808063	-1.250443	0.339067	0.094301	1.409143	0.928674	-0.617871	...	-0.982233	0.605123	-0.314588	-0.673120	1.182280
4	-1.153334	-0.709334	-0.679025	0.808063	-1.250443	-0.710671	-0.817139	1.409143	-0.804605	-0.617871	...	-1.096966	0.605123	-0.314588	1.485619	-0.845824

5 rows × 20 columns

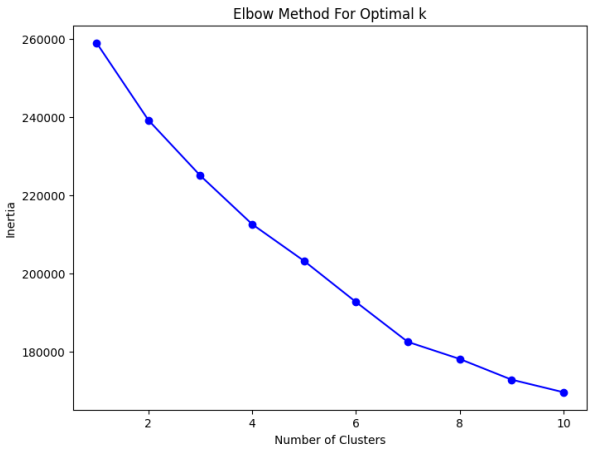
Table 11: Top 5 rows of data after Scale

Any business insights using clustering (if applicable)

Let's use K-means clustering to analysis it. Import Kmeans class from Sklearn.cluster

Define and fit it. Let's take the range of clusters from cluster 1 to 10. Below is their WSS value.

Plot the elbow graph



[258980.0,
239195.73013495476,
225001.912110479,
212562.35119540992,
203172.57203006613,
192669.4632499222,
182453.89372649192,
178084.90560201273,
172790.1932826128,

Figure 44: Elbow method graph

Observation

By looking at the graph, it is observed that after cluster 2 there is a sharp decrease and WSS value also suggests that there is sharp decrease after cluster 2. So, optimal cluster should be 2. But we will test it via **silhouette_score**

For n_clusters=2, the silhouette score is 0.08177977431118202
For n_clusters=3, the silhouette score is 0.09597583323474008
For n_clusters=4, the silhouette score is 0.11480622814632428
For n_clusters=5, the silhouette score is 0.09610019270952365
For n_clusters=6, the silhouette score is 0.10979395181709525
For n_clusters=7, the silhouette score is 0.12083424969768668
For n_clusters=8, the silhouette score is 0.10648497733148284
For n_clusters=9, the silhouette score is 0.1144965176893292
For n_clusters=10, the silhouette score is 0.11064100513733281

After testing the silhouette, it is observed that cluster 7 has the highest score and then cluster 4 has the second highest score.
The Elbow Method suggests that cluster 2 might be optimal based on the WSS values, indicating a balance between simplicity and cluster quality. However, silhouette analysis suggests 7 clusters offer the best separation. Cluster **interpretability** is critical and cluster 7 would be hard to interpret for business insights. Let's go with cluster 4, Based on the silhouette scores, cluster 4 (silhouette score of 0.115) offers a better-defined structure than clusters 2 and 3, and is still manageable from a business perspective. It strikes a balance between cluster quality and interpretability.
Cluster 4 inertia value is 212562.35119540992. Let's add a column by the name of "Clusters" and assign their cluster labels. Combine Column Churn with the dataset and review the dataset.

Cluster	Tenure	City_Tier	CC_Contacted_LY	Gender	Service_Score	Account_user_count	account_segment	CC_Agent_Score	rev_per_month	Complain_ly
0	9.689349	1.525641	17.623274	0.619329	2.892505	3.713018	2.870809	3.185404	5.194280	0.256410
1	10.418684	1.410709	17.996297	0.591854	2.901737	3.647679	2.835090	3.103105	5.393620	0.276559
2	10.175607	1.526459	17.541863	0.618159	2.904132	3.671620	2.863538	3.004712	5.323306	0.276912
3	10.580937	2.980279	18.690222	0.571076	2.913722	3.755957	3.244864	3.135579	5.207067	0.289236

4 rows × 25 columns

Payment_Cash on Delivery	Payment_Credit Card	Payment_Debit Card	Payment_E wallet	Payment_UPI	Marital_Status_Divorced	Marital_Status_Married	Marital_Status_Single	Churn	freq
1.0	0.0	0.000000	0.0	0.000000	0.148915	0.513807	0.337278	0.250493	1014
0.0	1.0	0.000000	0.0	0.000000	0.145827	0.546568	0.307605	0.142125	3511
0.0	0.0	0.851033	0.0	0.148967	0.143893	0.538601	0.317506	0.157122	5518
0.0	0.0	0.000000	1.0	0.000000	0.173377	0.542317	0.284306	0.226787	1217

Table 12: View of dataset after taking mean of all the variables as cluster wise

Business Insight from clusters

Cluster 0 has 1014 accounts frequency where their churn rate is 0.25 and with the following details.

- Tenure: 9.7 months
- City Tier 1 mostly
- Payment method: Cash on delivery
- Revenue per Month: 5.19
- Marital Status: 51% married
- Gender: Male majority

Cluster 1 has 3511 accounts frequency where their churn rate is 0.14.

- Tenure: 10.4 months
- City Tier 1 mostly
- Payment method: Credit card
- Revenue per Month: 5.39
- Marital Status: 54% married
- Gender: Male majority

Cluster 2 has 5518 accounts frequency where their churn rate is 0.15

- Tenure: 10.2 months
- City Tier 1 mostly
- Payment method: Primarily Debit card and some UPI
- Revenue per Month: 5.32

- Marital Status: 53% married
- Gender: Male majority

Cluster 3 has 1217 accounts frequency where their churn rate is 0.22

- Tenure: 10.6 months
- City Tier : 3
- Payment method: E Wallet
- Revenue per Month: 5.20
- Marital Status: 54% married
- Gender: Male majority

1) Cluster 1 and 2 have the lower churn rate and their revenue is also higher than cluster 0 and 3.
2) Cluster 0 is using payment method as Cash on delivery, so they follow traditional payment method. Retention strategies could be focused on offering better online payment options or discount for switching into the digital platform.
3) Cluster 0 has an average tenure of 9.7 months, so it means that they have shorter tenure and higher churn rate. Focus on onboarding and early engagement strategies to solidify their relationship with the company. Personalized communication and first-time discounts or offer can help reduce churn.
4) Cluster 1 has better average tenure as 10.4 with lowest churn rate which indicates that they are most loyal customers to the company. The company should provide them with some reward or loyalty offers for long-term relationships.
5) Cluster 2 has the highest number of customers with second lowest churn rate, their average tenure is also better than cluster 0. The company should focus on this group as well to engage the customers for a long time by providing them loyalty offers or some reward discounts.
6) Cluster 3 has also second highest churn rate after cluster 0, they are using E wallet method for the payment mode, they have the highest average tenure and are from Tier 3 city which indicates that their customers are engaged from a long time but are dissatisfied with either services or geographical restrictions. Company should plan to find their dissatisfaction root cause and mitigate them before they churn as they are long tenure customers. Provide them E wallet top up offers.

Model building - Clear on why was a particular model(s) chosen. - Effort to improve model performance.

Since the dataset is imbalanced as the proportion of churned customers (1's) is significantly lower than non-churned customers (0's), the data is considered unbalanced. To deal with unbalanced data, we are applying SMOTE (Synthetic Minority Over-sampling Technique).

SMOTE works by generating synthetic samples by interpolating between existing minority class samples. It calculates the Euclidean distance between points in the feature space to create new points along the lines connecting nearest neighbors.

```
Shape of Training set :  (11471, 23)
Shape of test set :  (3378, 23)
Percentage of classes in training set:
Churn
0    0.571441
1    0.428559
Name: proportion, dtype: float64
Percentage of classes in test set:
Churn
0    0.831557
1    0.168443
Name: proportion, dtype: float64
```

Table 13: Shape of training and test set after SMOTE

Scale the data with the help of StandardScaler from sklearn

	0	1	2	3	4	5	6	7	8	9	...	13	14	15	16	17
0	-0.001347	-0.727714	-0.721631	0.890294	0.226732	-0.730382	-0.725853	-0.069857	-1.492667	1.546862	...	-0.881518	0.710346	-0.293365	-0.571612	1.362731
1	-0.693333	1.504266	0.093747	0.890294	-1.205448	-0.730382	-0.725853	0.686113	-0.792566	-0.646470	...	-1.118577	0.710346	-0.293365	-0.571612	-0.733820
2	0.459976	1.504266	0.909124	0.890294	0.226732	-0.730382	1.135180	1.442084	-0.092464	-0.646470	...	1.583890	0.710346	-0.293365	-0.571612	-0.733820
3	2.074609	1.504266	-0.954596	0.890294	0.226732	0.351515	1.135180	1.442084	0.257587	-0.646470	...	2.342477	0.710346	-0.293365	-0.571612	-0.733820
4	-0.808664	1.504266	-0.255701	0.890294	0.226732	0.351515	0.204663	-1.581798	-0.792566	-0.646470	...	-0.146637	0.710346	-0.293365	-0.571612	-0.733820

5 rows × 23 columns

Table 14: Top 5 rows of data after Scale

Model Building:

Since this is a binary classification problem we will create the following models.

1 - Logistic Regression Model

Let's import LogisticRegression from Sklearn linear model, define the model with the following Parameters

solver='newton-cg', max_iter=10000,penalty='none',verbose=True,n_jobs=2

Fit the model on train dataset

AUC and ROC for the training data

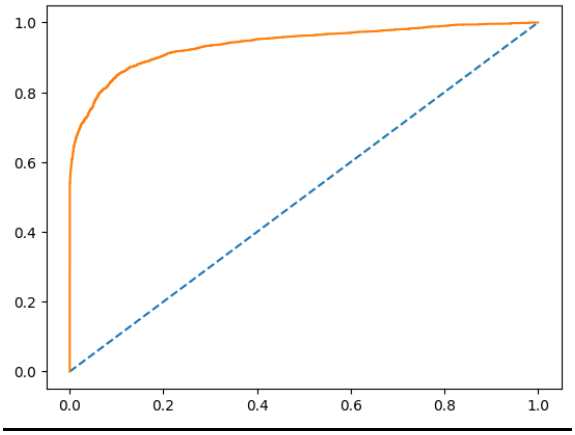


Figure 45: AUC and ROC for the training data

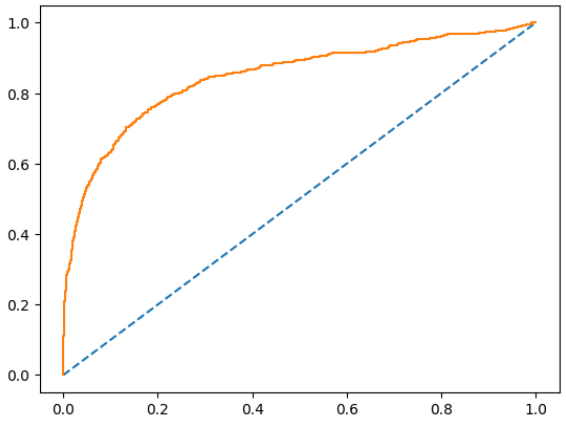


Figure 46: AUC and ROC for the test data

Train: AUC: 0.937 and Test: AUC: AUC: 0.937

Confusion Matrix and Classification report for the training data and Test data

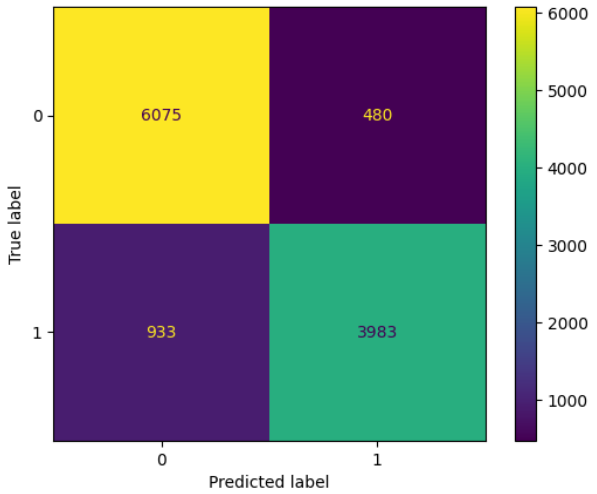


Figure 47: Confusion Matrix for the training data

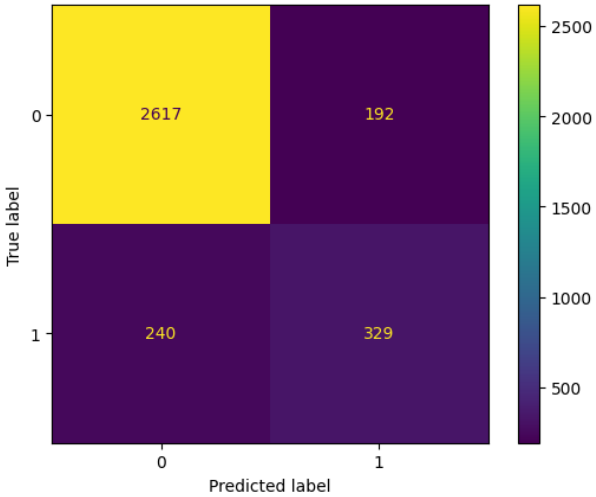


Figure 48: Confusion Matrix for the Test data

	precision	recall	f1-score	support
0	0.87	0.93	0.90	6555
1	0.89	0.81	0.85	4916
accuracy			0.88	11471
macro avg	0.88	0.87	0.87	11471
weighted avg	0.88	0.88	0.88	11471

	precision	recall	f1-score	support
0	0.92	0.93	0.92	2809
1	0.63	0.58	0.60	569
accuracy			0.87	3378
macro avg	0.77	0.75	0.76	3378
weighted avg	0.87	0.87	0.87	3378

Table 15: Classification Report for the training and test data

Insight

Accuracy: Training Accuracy: 0.88 and Test Accuracy: 0.87

The Logistic regression model achieves relatively similar accuracies on both the training and test Datasets.

Recall of class "1":

Training Recall of 1's: 0.81 and Test Recall of 1's: 0.58

The recall of class "1" represents the ability of the model to correctly identify instances belonging to class "1" (e.g., churn). The recall value of train is good, but it drops in test dataset, indicating that the model is overfitting.

Precision of class "1":

Training Recall of 1's: 0.89 and Test Recall of 1's: 0.63

All the instances predicted as class 1, 89% were correct but it drops in test dataset to 63%. It suggests that the model is making more incorrect predictions for class 1 in the test set, resulting in a higher false positive rate for this class.

F1-score of class "1":

Training F1-score of 1's: 0.85 and Test F1-score of 1's: 0.60

The harmonic mean of precision and recall is good in train dataset but drops in test dataset.

Area Under the ROC Curve (AUC):

Training AUC: 0.937 and Test AUC: 0.937

The AUC metric represents the model's ability to discriminate between positive and negative instances. An AUC value closer to 1 indicates better performance. The Linear regression model achieves reasonable AUC scores on both training and test datasets, suggesting that it performs reasonably well in distinguishing between positive and negative instances.

Hyper Tune the model

We will tune the model with GridSearchCV. Below are the parameters.

param_grid = { 'C': [0.01, 0.1, 1, 10], 'solver': ['lbfgs', 'liblinear', 'saga'],'penalty': ['l2'], 'class_weight': [None, 'balanced'], 'max_iter': [1000] }

Get the best parameters and the corresponding model on train and evaluate model on Test dataset.

Best Parameters: LogisticRegression(C=10, class_weight='balanced', max_iter=1000)

	precision	recall	f1-score	support
0	0.89	0.90	0.89	6555
1	0.86	0.84	0.85	4916
accuracy			0.88	11471
macro avg	0.88	0.87	0.87	11471
weighted avg	0.88	0.88	0.88	11471

	precision	recall	f1-score	support
0	0.92	0.90	0.91	2809
1	0.56	0.63	0.59	569
accuracy			0.85	3378
macro avg	0.74	0.77	0.75	3378
weighted avg	0.86	0.85	0.86	3378

Table 16: Classification Report for the train data after HyperTune

Insight

2 - Gaussian Naive Bayes

Let's import GaussianNB, define the model and fit the train dataset.

Confusion Matrix and Classification report for the Train data and Test data

```
0.758434312614419
[[5001 1554]
 [1217 3699]]
```

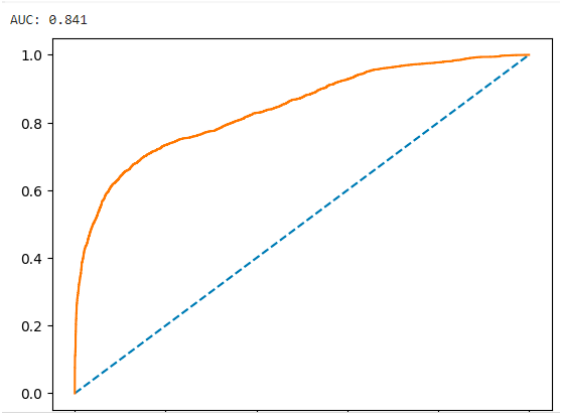
	precision	recall	f1-score	support
0	0.80	0.76	0.78	6555
1	0.70	0.75	0.73	4916
accuracy			0.76	11471
macro avg	0.75	0.76	0.76	11471
weighted avg	0.76	0.76	0.76	11471

```
0.724097098875074
[[2117  692]
 [ 240  329]]
```

	precision	recall	f1-score	support
0	0.90	0.75	0.82	2809
1	0.32	0.58	0.41	569
accuracy			0.72	3378
macro avg	0.61	0.67	0.62	3378
weighted avg	0.80	0.72	0.75	3378

Table 17: Confusion Matrix and Classification Report for the train and test data

AUC and ROC for the train and test data



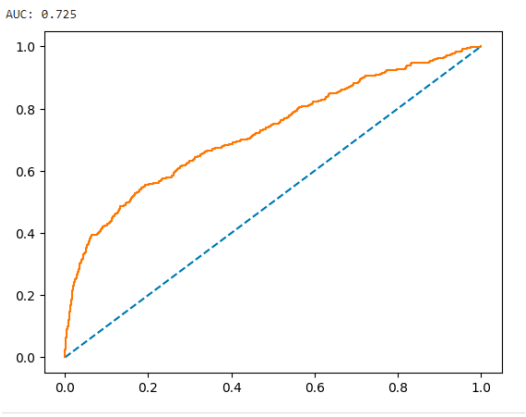


Figure 49: ROC for the Train data

Figure 50: ROC for the Test data

Insight - The model is performed moderately on train dataset; the recall value looks similar for class 0 and 1. But the precision value of class 0 is higher than class 1. However, model performance decreased on unseen data as precision and recall values of class 1 are 0.32 and 0.58.

HyperTunning the Model

We will tune the model with GridSearchCV. Below are the parameters.

param_grid = {'var_smoothing': [1e-9, 1e-8, 1e-7, 1e-6]}

Best parameters: {'var_smoothing': 1e-09}

Let's fit the model and predict the model

Confusion Matrix and Classification report for the train and test data

Train Accuracy after tuning: 0.758434312614419
Train Classification Report after tuning:

	precision	recall	f1-score	support
0	0.80	0.76	0.78	6555
1	0.70	0.75	0.73	4916
accuracy			0.76	11471
macro avg	0.75	0.76	0.76	11471
weighted avg	0.76	0.76	0.76	11471

Test Accuracy after tuning: 0.724097098875074
Test Classification Report after tuning:

	precision	recall	f1-score	support
0	0.90	0.75	0.82	2809
1	0.32	0.58	0.41	569
accuracy			0.72	3378
macro avg	0.61	0.67	0.62	3378
weighted avg	0.80	0.72	0.75	3378

Table 18: Classification Report for the train data

Table 19: Classification Report for the test data

Insight

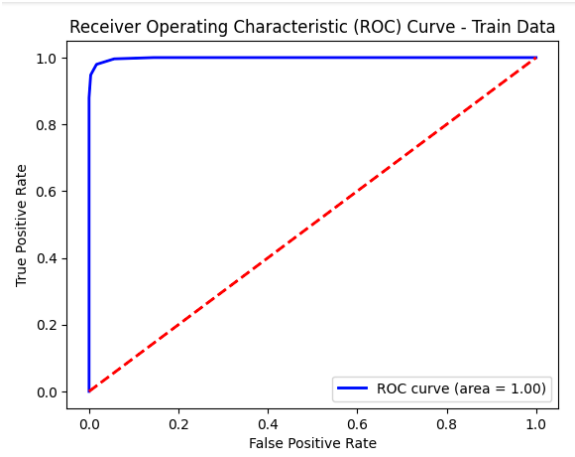
After hyper tunning, there is no sign of Improvement. The result remains the same. Let's build another model.

3 - KNN Model

Let's import KNeighborsClassifier, define the model and fit the train and test dataset.

AUC and ROC for the train data

ROC AUC Train: 0.9985654805460958 and Test: ROC AUC: 0.9707740184856484



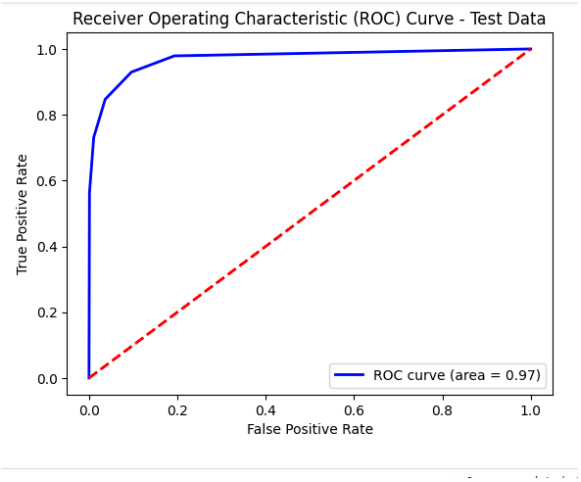


Figure 51: ROC for the Train data

Figure 52: ROC for the Train data

Confusion Matrix and Classification report for the train and test data

```
0.9817801412256996
[[6446  109]
 [ 100 4816]]
              precision    recall  f1-score   support

           0       0.98      0.98      0.98      6555
           1       0.98      0.98      0.98      4916

    accuracy                           0.98     11471
   macro avg       0.98      0.98      0.98     11471
weighted avg       0.98      0.98      0.98     11471
```

Table 20: Classification Report for the train data

```
0.9437537004144464
[[2706  103]
 [  87  482]]
              precision    recall  f1-score   support

           0       0.97      0.96      0.97      2809
           1       0.82      0.85      0.84       569

    accuracy                           0.94      3378
   macro avg       0.90      0.91      0.90      3378
weighted avg       0.94      0.94      0.94      3378
```

Table 21: Classification Report for the test data

Insight

There's a slight drop in performance between the training and test sets, particularly in recall, precision, and F1 score. This indicates minor overfitting. Next step is Cross-Validation, can provide better insight into model stability across different data splits.

Cross-Validation

Let's compute cross-validation scores for a KNN model.

```
KNN Model:
Confusion Matrix:
[[6516   39]
 [ 137 4779]]

Classification Report:
              precision    recall  f1-score   support

           0       0.98      0.99      0.99      6555
           1       0.99      0.97      0.98      4916

    accuracy                           0.98     11471
   macro avg       0.99      0.98      0.98     11471
weighted avg       0.98      0.98      0.98     11471
```

Table 22: Classification Report for the train data

```
KNN Model:
Confusion Matrix:
[[2748   61]
 [ 109  460]]

Classification Report:
              precision    recall  f1-score   support

           0       0.96      0.98      0.97      2809
           1       0.88      0.81      0.84       569

    accuracy                           0.95      3378
   macro avg       0.92      0.89      0.91      3378
weighted avg       0.95      0.95      0.95      3378
```

Table 23: Classification Report for the test data

Insight

After hyperparameter tuning, the model exhibits improved **precision** and **F1-score** for the test set, even though there is a small trade-off in **recall**. The accuracy is also slightly better, making the hypertuned model more reliable overall, especially in terms of reducing false positives.

4 - Random Forest Model

Let's import RandomForestClassifier, define the model and fit the train dataset.

Confusion Matrix and Classification report for the train and test data

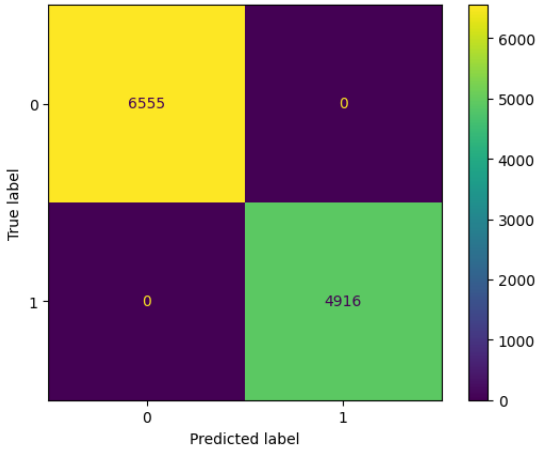


Figure 53: Confusion Matrix for the Train data

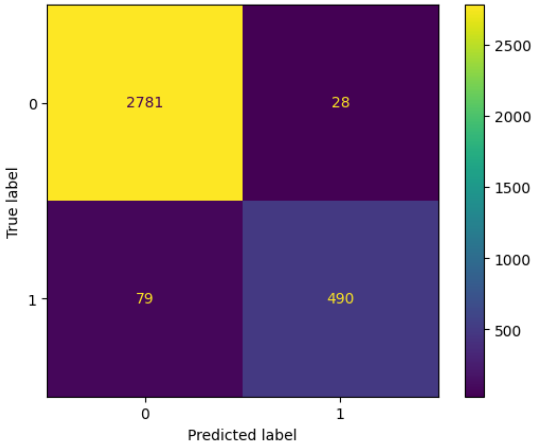


Figure 54: Confusion Matrix for the Test data

Random Forest Classification Report for train:

	precision	recall	f1-score	support
0	1.00	1.00	1.00	6555
1	1.00	1.00	1.00	4916
accuracy			1.00	11471
macro avg	1.00	1.00	1.00	11471
weighted avg	1.00	1.00	1.00	11471

Table 24: Classification Report for the train data

Random Forest Classification Report for test:

	precision	recall	f1-score	support
0	0.97	0.99	0.98	2809
1	0.95	0.86	0.90	569
accuracy			0.97	3378
macro avg	0.96	0.93	0.94	3378
weighted avg	0.97	0.97	0.97	3378

Table 25: Classification Report for the test data

AUC and ROC for the train and test data

Train: ROC AUC: 1.0 and Test: ROC AUC: 0.988

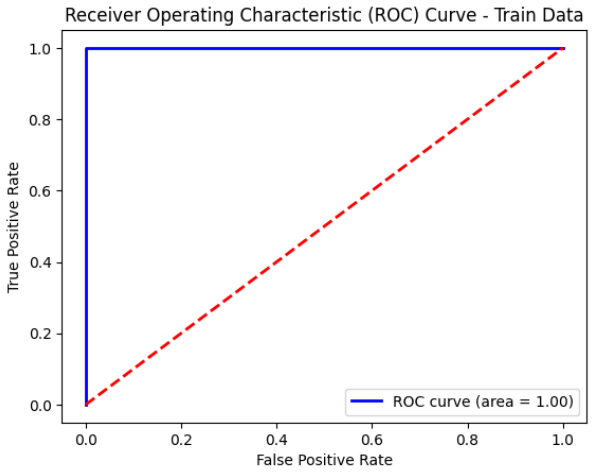


Figure 55: ROC AUC for the Train data

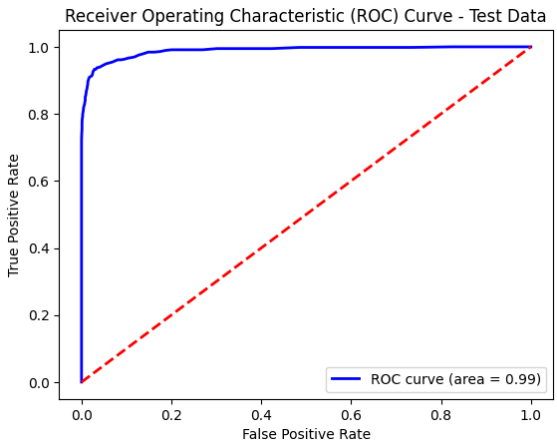


Figure 56: ROC AUC for the Train data

Insight

Accuracy: Training Accuracy: 1.00 and Test Accuracy: 0.97

Random forest model achieves extremely good accuracy on the train dataset, But it drops slightly on the test dataset.

Recall of class "1":

Training Recall of 1's: 1.00 and Test Recall of 1's: 0.86

The recall of class "1" represents the ability of the model to correctly identify instances belonging to class "1" (e.g., churn). The recall value of train is good, but it drops in test dataset, indicating that the model is overfitting.

Precision of class "1":

Training Recall of 1's: 1.00 and Test Recall of 1's: 0.95

All the instances predicted as class 1, 100% were correct but it drops in test dataset to 95%. It suggests that the model is making more incorrect predictions for class 1 in the test set, resulting in a high false positive rate for this class.

F1-score of class "1":

Training F1-score of 1's: 1.00 and Test F1-score of 1's: 0.86

The harmonic mean of precision and recall is good in train dataset but drops in test dataset.

Area Under the ROC Curve (AUC):

Training AUC: 1.00 and Test AUC: 0.98

The AUC metric represents the model's ability to discriminate between positive and negative

instances. AUC 1 in the train dataset indicates that the model is perfectly separating the classes during training. AUC 0.98 in test suggesting strong discrimination capability on the test set.

This performance difference (perfect training but slightly lower test performance) suggests that your model might be overfitting.

Hyper Tune the model

We will tune the model with RandomizedSearchCV. Below are the parameters.

param_dist = {'n_estimators': randint(100, 500), 'max_depth': randint(5, 15), 'min_samples_split': randint(2, 10), 'min_samples_leaf': randint(1, 10), 'max_features': ['auto', 'sqrt'], 'class_weight': [None, 'balanced']}

Here are the best parameters –

```
{'class_weight': 'balanced',
 'max_depth': 12,
 'max_features': 'sqrt',
 'min_samples_leaf': 2,
 'min_samples_split': 3,
 'n_estimators': 248}
```

Fit the model and predict the train and test.

```
[[6468   87]
 [  55 4861]]
```

	precision	recall	f1-score	support
0	0.99	0.99	0.99	6555
1	0.98	0.99	0.99	4916
accuracy			0.99	11471
macro avg	0.99	0.99	0.99	11471
weighted avg	0.99	0.99	0.99	11471

Table 26: Confusion and Classification Report of train data

```
[[2722   87]
 [  95  474]]
```

	precision	recall	f1-score	support
0	0.97	0.97	0.97	2809
1	0.84	0.83	0.84	569
accuracy			0.95	3378
macro avg	0.91	0.90	0.90	3378
weighted avg	0.95	0.95	0.95	3378

Table 27: Confusion and Classification Report of test data

Insight

After tuning, the model's performance on the training set was slightly reduced to more realistic values (precision and recall ~0.99), suggesting reduced overfitting. However, in the test set, performance for class 1 dropped further, with precision and recall around **0.84** and **0.83**, respectively. This indicates the model may still be struggling to capture all instances of class 1 in the test set.

Let's try to change the threshold value

```
Confusion Matrix:
[[6479   76]
 [  60 4856]]
Classification Report:
```

	precision	recall	f1-score	support
0	0.99	0.99	0.99	6555
1	0.98	0.99	0.99	4916
accuracy			0.99	11471
macro avg	0.99	0.99	0.99	11471
weighted avg	0.99	0.99	0.99	11471

Table 28: Classification Report of train data

```
Confusion Matrix:
[[2729   80]
 [  97  472]]
Classification Report:
```

	precision	recall	f1-score	support
0	0.97	0.97	0.97	2809
1	0.86	0.83	0.84	569
accuracy			0.95	3378
macro avg	0.91	0.90	0.91	3378
weighted avg	0.95	0.95	0.95	3378

Table 29: Classification Report of test data

Insight

The model's performance on the training set was slightly reduced to more realistic values (precision 0.98 and recall 0.99), suggesting reduced overfitting. However, in the test set, performance for class 1 dropped further, with precision and recall around **0.86** and **0.83**, respectively. This indicates the model may still be struggling to capture all instances of class 1 in the test set.

Model 5 - Boosting

Let's build another model ***Ada Boost.***

Import the model, define the model and fit the model.

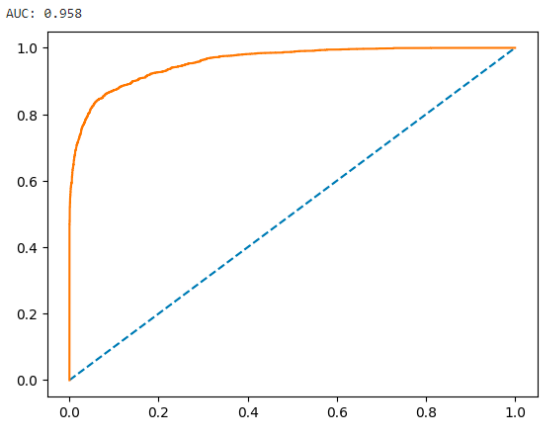
```
AdaBoostClassifier
AdaBoostClassifier(n_estimators=100, random_state=1)
```

Confusion Matrix and Classification report for the train and test data

```
0.893557667160666
[[6036  519]
 [ 702 4214]]
              precision    recall  f1-score   support

           0       0.90      0.92      0.91      6555
           1       0.89      0.86      0.87      4916

    accuracy                           0.89     11471
   macro avg       0.89      0.89      0.89     11471
weighted avg       0.89      0.89      0.89     11471
```

```
0.8863232682060391
[[2602  207]
 [ 177  392]]
              precision    recall  f1-score   support

           0       0.94      0.93      0.93      2809
           1       0.65      0.69      0.67       569

    accuracy                           0.89      3378
   macro avg       0.80      0.81      0.80      3378
weighted avg       0.89      0.89      0.89      3378
```

Table 30: Confusion & Classification Report of train data **Table 31: Confusion & Classification Report of test data**



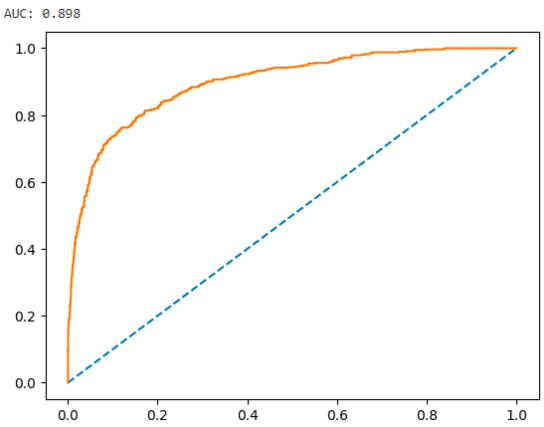


Figure 57: ROC Map for the Train data

Figure 58: ROC Map for the Test data

Insight

This model is also performing good in train data as accuracy is 89, Recall and Precision value is also good such as 86 and 89. However, their values dropped in test dataset as 69 and 65. This model still needs to improve with hyper tunning.

Hyper Tune the model

We will tune the model with DecisionTreeClassifier. Below are the parameters.

base_estimator = DecisionTreeClassifier(max_depth=2, class_weight={0: 1, 1: 5})

Confusion Matrix and Classification report for the train and test data

```
0.7980123790428036
[[4373 2182]
 [ 135 4781]]
              precision    recall  f1-score   support

           0       0.97      0.67      0.79      6555
           1       0.69      0.97      0.80      4916

    accuracy                           0.80     11471
   macro avg       0.83      0.82      0.80     11471
weighted avg       0.85      0.80      0.80     11471
```

```
0.7980123790428036
[[1918  891]
 [  54  515]]
              precision    recall  f1-score   support

           0       0.97      0.68      0.80      2809
           1       0.37      0.91      0.52       569

    accuracy                           0.72      3378
   macro avg       0.67      0.79      0.66      3378
weighted avg       0.87      0.72      0.76      3378
```

Table 32: Confusion matrix and Classification Report of train and Test data

Insight

1. Recall (Test) improved significantly after hyperparameter tuning (from 0.69 to 0.91), which suggests the model became much better at identifying positive cases (class 1).
2. Precision (Test) however, dropped substantially after tuning (from 0.65 to 0.37). This could indicate the model is now classifying more false positives.
3. F1 Score (Test) improved slightly (from 0.67 to 0.72), which is a positive sign as it balances both precision and recall.
4. Accuracy (Test) decreased after tuning (from 0.89 to 0.72), indicating that while recall has improved, the overall classification accuracy has dropped.

After hyperparameter tuning, the model seems to focus more on **recall** at the cost of **precision**, indicating that it's better at identifying positive cases but at the cost of misclassifying negative ones.

Let's find the optimal threshold value. **Changing the threshold value**

Confusion Matrix and Classification report for the train and test data

```
[[6485   70]
 [  71 4845]]
              precision    recall  f1-score   support

           0       0.99      0.99      0.99      6555
           1       0.99      0.99      0.99      4916

    accuracy                           0.99     11471
   macro avg       0.99      0.99      0.99     11471
weighted avg       0.99      0.99      0.99     11471
```

```
[[2739   70]
 [ 102  467]]
              precision    recall  f1-score   support

           0       0.96      0.98      0.97      2809
           1       0.87      0.82      0.84       569

    accuracy                           0.95      3378
   macro avg       0.92      0.90      0.91      3378
weighted avg       0.95      0.95      0.95      3378
```

Table 33: Confusion matrix and Classification Report of train and Test data

Insight

After threshold tuning, the model has shown significant improvement in both train and test performance. After threshold tuning, the model has shown significant improvement in both train and test performance. Recall on the test set decreased slightly from 0.91 to 0.82, but this is a reasonable trade-off for improved precision. Let's build another model

Model 6 – Gradient Boosting

Import the model, define the model and fit the model.

Confusion Matrix and Classification report for the train and test data

```
0.9114288205038793
[[6192  363]
 [ 653 4263]]
              precision    recall  f1-score   support

           0       0.90      0.94      0.92      6555
           1       0.92      0.87      0.89      4916

    accuracy                           0.91     11471
   macro avg       0.91      0.91      0.91     11471
weighted avg       0.91      0.91      0.91     11471
```

```
0.9029011249259917
[[2645  164]
 [ 164  405]]
              precision    recall  f1-score   support

           0       0.94      0.94      0.94      2809
           1       0.71      0.71      0.71       569

    accuracy                           0.90      3378
   macro avg       0.83      0.83      0.83      3378
weighted avg       0.90      0.90      0.90      3378
```

Table 34: Confusion matrix and Classification Report of train and Test data

Insight

As with all models, it also performed well in train dataset, but the performance decreased in test dataset. Let's try to do the hyper tune the model.

Hyper Tune the model

We will tune the model with GridSearchCV. Below are the parameters.

param_grid = { 'n_estimators': [100, 200], 'learning_rate': [0.5, 0.1],'max_depth': [3, 4], 'min_samples_split': [5, 10], 'min_samples_leaf': [2, 4], 'subsample': [0.8, 1.0] }

Here we have the best parameters.

Best Parameters: {'learning_rate': 0.5, 'max_depth': 4, 'min_samples_leaf': 4, min_samples_split': 5, 'n_estimators': 200, 'subsample': 1.0}

Let's fit the train and test dataset and predict the model.

Confusion Matrix and Classification report for the train and test data

```
Train Set Results:
[[6554    1]
 [   1 4915]]
              precision    recall  f1-score   support

           0       1.00      1.00      1.00      6555
           1       1.00      1.00      1.00      4916

    accuracy                           1.00     11471
   macro avg       1.00      1.00      1.00     11471
weighted avg       1.00      1.00      1.00     11471
```

```
Test Set Results:
[[2755   54]
 [  82  487]]
              precision    recall  f1-score   support

           0       0.97      0.98      0.98      2809
           1       0.90      0.86      0.88       569

    accuracy                           0.96      3378
   macro avg       0.94      0.92      0.93      3378
weighted avg       0.96      0.96      0.96      3378
```

Table 35: Confusion matrix and Classification Report of train and Test data

Insight

The Precision, recall, F1-score and accuracy of train dataset was 1 which later dropped in dataset which indicates that model is overfitting. **Let's try to change the parameters and fit the model again on the train and test dataset.**

Changing the parameters

param_grid = { 'n_estimators': [400], 'learning_rate': [0.3], 'max_depth': [5], 'min_samples_split': [2], 'min_samples_leaf': [5], 'subsample': [0.9]}

Here are the Best Parameters:

{'learning_rate': 0.3, 'max_depth': 5, 'min_samples_leaf': 5, 'min_samples_split': 2, 'n_estimators': 400, 'subsample': 0.9}

Let's predict the train and test dataset.

```
Train Set Results:
[[6555    0]
 [   0 4916]]
              precision    recall  f1-score   support

           0       1.00      1.00      1.00      6555
           1       1.00      1.00      1.00      4916

    accuracy                           1.00     11471
   macro avg       1.00      1.00      1.00     11471
weighted avg       1.00      1.00      1.00     11471
```

```
Test Set Results:
[[2787   22]
 [  59  510]]
              precision    recall  f1-score   support

           0       0.98      0.99      0.99      2809
           1       0.96      0.90      0.93       569

    accuracy                           0.98      3378
   macro avg       0.97      0.94      0.96      3378
weighted avg       0.98      0.98      0.98      3378
```

Table 36: Confusion matrix and Classification Report of train and Test data

Insight

The model has improved substantially after adjusting hyperparameters, especially in terms of test recall, precision, and F1 score. The model continues to perfectly fit the training data which indicates overfitting.

1 – Accuracy: The accuracy of the train dataset is 1 which dropped on test data set to 98. However, the accuracy of the test data set has increased after the tune.

2 – Recall: The recall value of train is 1 which dropped on test data set to 90. However, the recall value of test data set is increased after hyper tune.

3 – Precision: The precision value of the train is 1 which dropped on test data set to 96 which is very good, and it has improved after hyper tune.

4 - F1 Score: Both recall and precision are balanced well, with the final F1 score at 0.93 for the test set, indicating solid model performance.

Overall, this model worked well with minor overfitting. Let's check another model then will decide on final model.

Model 6 - Bagging

Import BaggingClassifier from Sklearn.ensemble

Import the model, define the model and fit the model.

Confusion Matrix and Classification report for the train and test data

```
1.0
[[6555    0]
 [   0 4916]]
              precision    recall  f1-score   support

           0       1.00      1.00      1.00      6555
           1       1.00      1.00      1.00      4916

    accuracy                           1.00     11471
   macro avg       1.00      1.00      1.00     11471
weighted avg       1.00      1.00      1.00     11471
```

```
0.9632918886915335
[[2769   40]
 [  84  485]]
              precision    recall  f1-score   support

           0       0.97      0.99      0.98      2809
           1       0.92      0.85      0.89       569

    accuracy                           0.96      3378
   macro avg       0.95      0.92      0.93      3378
weighted avg       0.96      0.96      0.96      3378
```

Table 37: Confusion matrix and Classification Report of train and Test data

Insight The model was able to identify all instances perfectly in train dataset as the recall, precision and accuracy is 1. However, it dropped in unseen data which indicates that the model is overfitting.

Let's hyper tune the model.

Hyper Tune the model

We will tune the model with DecisionTreeClassifier(). Below are the parameters.

param_dist = { 'n_estimators': [50, 100, 200], 'max_samples': [0.8, 1.0], 'max_features': [0.8, 1.0], 'bootstrap': [True, False], bootstrap_features': [True, False],'base_estimator__max_depth': [None, 10, 20], 'base_estimator__min_samples_split': [2, 5, 10], 'base_estimator__min_samples_leaf': [1, 2, 4]}

Here we have the best parameters.

Best parameters from RandomizedSearchCV: {'n_estimators': 200, 'max_samples': 1.0, 'max_features': 0.8, 'bootstrap_features': True, 'bootstrap': False, 'base_estimator__min_samples_split': 5, 'base_estimator__min_samples_leaf': 1, 'base_estimator__max_depth': 20}

Confusion Matrix and Classification report for the train and test data

```
1.0
[[6555    0]
 [   0 4916]]
              precision    recall  f1-score   support

           0       1.00      1.00      1.00      6555
           1       1.00      1.00      1.00      4916

    accuracy                           1.00     11471
   macro avg       1.00      1.00      1.00     11471
weighted avg       1.00      1.00      1.00     11471
```

```
0.9748371817643576
[[2800    9]
 [  76  493]]
              precision    recall  f1-score   support

           0       0.97      1.00      0.99      2809
           1       0.98      0.87      0.92       569

    accuracy                           0.97      3378
   macro avg       0.98      0.93      0.95      3378
weighted avg       0.98      0.97      0.97      3378
```

Table 38: Confusion matrix and Classification Report of train and Test data

Insight - The performance of the model has increased after hyper tune. The accuracy of class 1, recall of class 1, Precision and F1 score has increased in test data. However, in train data, it has excellent performance as value 1 for all the matrix shows that the model is overfit.

Select Best Model & mention reason:

	Accuracy Train	Accuracy Test	Recall Train	Recall Test	Precision Train	Precession Test	F1 score Train	F1 score Test	AUC train	AUC test
Linear Regression	0.88	0.87	0.81	0.58	0.89	0.63	0.85	0.6	0.937	0.937
Regularized Linear regression model	0.88	0.85	0.84	0.63	0.87	0.56	0.85	0.59	0.937	0.848
Linear Regression Hypertunning	0.88	0.85	0.84	0.63	0.86	0.56	0.85	0.59	0.937	0.847
Linear Regression changed threshold	0.85	0.85	0.89	0.63	0.79	0.56	0.84	0.59	0.937	0.847
Gaussian Naive Bayes	0.76	0.72	0.75	0.58	0.7	0.32	0.76	0.41	0.84	0.72
Gaussian Naive Bayes Hypetunning	0.76	0.72	0.75	0.58	0.7	0.32	0.73	0.41	0.841	0.725
KNN model	0.98	0.94	0.98	0.85	0.98	0.82	0.98	0.84	0.99	0.97
KNN model Hypertune	0.98	0.95	0.97	0.81	0.99	0.88	0.98	0.95	0.99	0.97
RFM	1	0.97	1	0.86	1	0.95	1	0.9	1	0.98
RFM Hypertunning	0.99	0.95	0.99	0.83	0.98	0.84	0.99	0.84	0.999	0.971
Ada Boosting	0.89	0.89	0.86	0.69	0.89	0.65	0.87	0.67	0.95	0.89
Ada Boosting Hypertunning	0.8	0.72	0.97	0.91	0.69	0.37	0.8	0.72	0.952	0.9
Ada Boosting change threshold value	0.99	0.95	0.99	0.82	0.99	0.87	0.99	0.84	0.987	0.898
Gradient Boost	0.91	0.9	0.87	0.71	0.92	0.71	0.89	0.71	0.97	0.92
Gradient Boost hypertunning	1	0.96	1	0.86	1	0.9	1	0.88	1	0.99
Gradient Boost adjusted parameters	1	0.98	1	0.9	1	0.96	1	0.93	1	0.992
Bagging	1	0.96	1	0.85	1	0.92	1	0.89	1	0.973
Bagging Hypertunning	1	0.97	1	0.87	1	0.98	1	0.92	1	0..96

Table 39: Comparison table of all models

As per the above table, we have built several models such as Logistic Regression, Naive Bayes, KNN, Random Forest, Ada Boosting, Gradient Boosting and Bagging. While building the model and test it on the unseen data. We noticed that the data was imbalanced, so we used SMOTE technique. To improve the model performance, we hyper tune the model, adjusted the parameters. After all the efforts, we came up with a final model is ***Gradient Boost After hyper tune adjusted parameters.***

Accuracy:

Training Accuracy: 1.00 and Test Accuracy: 0.98
Gradient Boost After hyper tune adjusted parameters achieves extremely good accuracies on the training, but it drops slightly on the test datasets.

Recall of class "1":

Training Recall of 1's: 1.00 and Test Recall of 1's: 0.90

The recall of class "1" represents the ability of the model to correctly identify instances belonging to class "1" (e.g., churn). The recall value of train is extremely good, but it drops in test dataset, but still, it is 90 which is 10% deviation so we can consider it very good model on test dataset.

Precision of class "1":

Training precision of 1's: 1.00 and Test precision of 1's: 0.96

All the instances predicted as class 1, 100% were correct but it drops in test data set to 96%. It suggests that the model is making 4% incorrect predictions for class 1 in the test set.

F1-score of class "1":

Training F1-score of 1's: 1.00 and Test F1-score of 1's: 0.93

The harmonic mean of precision and recall is good in train dataset but drops in test dataset.

Area Under the ROC Curve (AUC):

Training AUC: 1.00 and Test AUC: 0.99 - The AUC is on the train and test is very good.

Gradient Boost after hyper tune adjusted parameters outperforms other models in terms of accuracy, recall of positive cases, and AUC score on both training and test sets.

Bagging (Hypertuning) also performs very well with slightly lower recall, but higher precision compared to Gradient Boost. Since our focus is on recall more, ***Gradient Boost after hyper tune adjusted parameters*** is our first choice.

Check the most important features in the final model and draw inferences.

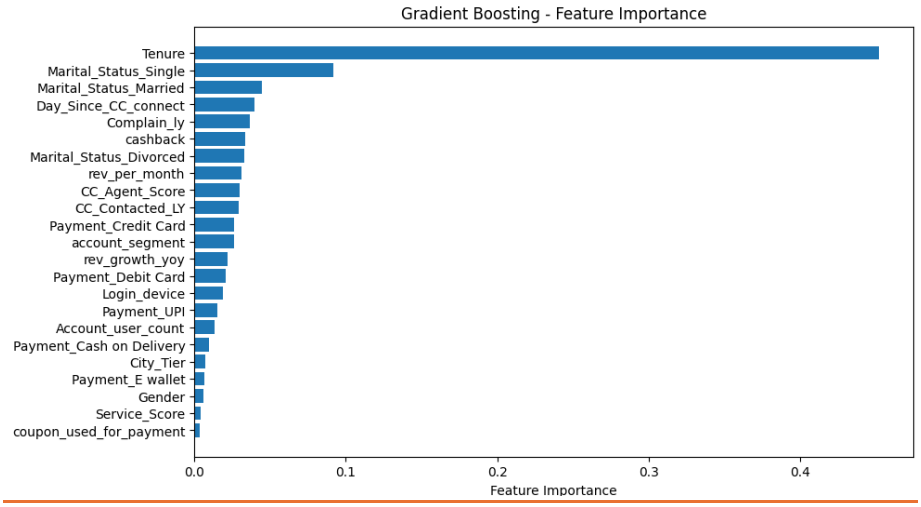


Figure 59: Important features in final model

Observation - Tenure is the most important feature, followed by Marital status as per the tuned gradient boosting

model.

Actionable Insights & Recommendations

1 - Churn Insights by City Tier and Payment method

1. Customers in Tier 1 cities are likely more using traditional payment methods like debit and credit cards and No E-Wallet Usage, the lack of E-Wallet usage suggests that Tier 1 customers either don't see the need for them or prefer to stick with more traditional payment methods. E-wallets may not offer significant benefits to this group.
2. Tier 3 city customers are more inclined towards using **E-wallets**, likely due to the convenience and ease of use. These customers may have limited access to banking services or may prefer alternatives like E-wallets for smaller or day-to-day transactions. **Low UPI Usage**: UPI, although widely adopted in urban centers, might not be as common or preferred in Tier 3 cities due to either limited awareness, internet connectivity issues, or a preference for the simplicity of E-wallets over linking bank accounts for UPI transactions. This suggests a shift in behavior where customers are comfortable with mobile payments, but they may not fully trust or utilize the more direct forms of digital banking, like UPI.
3. In Tier 2 cities, UPI seems to be the dominant mode of payment. This shows that Tier 2 city customers are adopting more recent and tech-forward payment methods like UPI but have not embraced E-wallets. They might perceive UPI as more convenient or secure.
4. COD and E-wallet have a higher churn rate compared to other payment methods. This could be because COD might be associated with a higher risk of order cancellations or returns, leading to dissatisfaction or inconvenience for both customers and businesses. For E-Wallet, this could be due to issues such as limited acceptance, perceived lack of security, or dissatisfaction with e-wallet services.

2 - Churn Insights by Customer Segment

5. **Regular Plus Customers** - These customers have the highest likelihood of churning. This could be due to several factors such as Value Perception (They may feel they are not receiving enough value relative to their spending or benefits), Service Expectations (They might have high expectations that are not being met, leading to dissatisfaction), Competition (They might be more inclined to switch to competitors offering better deals or services).
6. **HNI (High Net-Worth Individuals)** - HNI customers are also more likely to churn due to Personalized Service (HNIs often expect highly personalized and premium services. Any shortfall in meeting these expectations could lead to higher churn), Exclusive Offers (They might be more tempted by exclusive offers from competitors, leading to higher churn rates if those offers are more attractive).
7. **Super Plus & Regular Customers** - These customers have a lower churn rate compared to Regular Plus and HNI but a moderate rate. This suggests they will likely receive a good balance of value and service, which keeps them engaged, they are likely content with their current level of service and benefits, leading to higher retention.
8. **Super Customers -** Super customers have a moderate churn rate. They are more stable than Regular Plus and HNI but not as secure as Regular customers. This might be due to Service Quality and Competitive Pressure.

3 – Other insights

9. Single people are more to churn, are more than **50%** of the churn group, indicating they are the most at risk for leaving. This could be due to their lifestyle flexibility or willingness to explore other services or products.
10. Customers who have small tenure are more likely to churn.
11. Males are slightly more likely to churn compared to females. Males might be more inclined to explore competitive alternatives or might be less satisfied with their current service, leading to higher churn.

Business recommendations:

City Tier and Payment Method

Tier 1 City Customers

1. To encourage continued loyalty, offer incentives such as cashback or discounts for customers using debit or credit cards.
2. Launch awareness campaigns highlight the benefits of using e-wallets (e.g., faster transactions, exclusive offers). Collaborate with e-wallet providers to offer promotions tailored to Tier 1 customers, emphasizing convenience and security.

Tier 3 City Customers:

1. Continuing offering and improving e-wallet-based offers and promotions, as these customers are more inclined to use them.
2. **Educate on UPI Benefits:** Conduct awareness campaigns or workshops about the ease and security of UPI payments to increase its adoption in Tier 3 cities. Offering simple guides or incentives for first-time UPI users may also encourage adoption.

Tier 2 City Customers:

1. Focus on promoting UPI-related discounts, cashback, or other rewards, as these customers have embraced the convenience of UPI.

COD and E-Wallet Usage:

1. Introduce incentives for prepaid transactions or offer faster delivery and easier returns for prepaid orders to shift away from COD and reduce risks of order cancellations.
2. Partner with more merchants to increase the acceptance of e-wallets and resolve potential user issues, such as security concerns or limited usability.

Customer Segment

Regular Plus Customers:

1. Increase Value Perception: Offer personalized rewards, exclusive offers, or better loyalty points to Regular Plus customers. Highlight the additional benefits of staying loyal, such as priority customer service or access to premium features.
2. Monitor Satisfaction: Implement frequent satisfaction surveys or engagement tracking to proactively address any dissatisfaction before it leads to churn.

High Net-Worth Individuals (HNIs):

1. Invest in customer relationship management to offer tailored services for HNIs. This could include dedicated relationship managers, exclusive events, or personalized financial or service advice.
2. Create Exclusive Offers.

Super Plus & Regular Customers:

1. Ensure that customer service and product offerings remain consistent to retain these customers

Super Customers:

1. Reassess pricing, customer service, and product offerings for this segment to prevent competitors from drawing them away. Offering regular updates on new features or improvements can keep them engaged.

Demographic and Behavioral

Single People: More likely to churn, offer flexible pricing plans or services that cater to the lifestyle of single individuals. Ensure engagement through personalized offers, tailored communications, and targeted messaging around convenience or excitement.

Short Tenure Customers: Create a seamless and engaging onboarding process for new customers. Provide extra attention in the first 3-6 months with special welcome offers, product tutorials, and proactive support to help them fully realize the value of your service.

Males: Males are slightly more likely to churn than females. Personalize communication and offers to target male customers based on their preferences. Competitive pricing and innovative product features may help retain their interest.