

CAPSTONE PROJECT

ON

Customer Churn Prediction



By,
Shobha Kumari Choudhary

Date: - 24th March'24

Contents

Content	Page Number
1) Introduction of the business problem	1
a) Defining problem statement	1
b) Need of the study/project	1
c) Understanding business/social opportunity	1
2) Data Report	2
a) Understanding how data was collected in terms of time, frequency and methodology	2
b) Visual inspection of data (rows, columns, descriptive details)	2
c) Understanding of attributes (variable info, renaming if required)	3
3) Exploratory data analysis	6
a) Univariate analysis (distribution and spread for every continuous attribute, distribution of data in categories for categorical ones)	6
b) Bivariate analysis (relationship between different variables, correlations)	10
a) Removal of unwanted variables (if applicable)	15
b) Missing Value treatment (if applicable)	15
d) Outlier treatment (if required)	16
e) Variable transformation (if applicable)	24
f) Addition of new variables (if required)	24
4) Business insights from EDA	33
a) Is the data unbalanced? If so, what can be done? Please explain in the context of the business	33
b) Any business insights using clustering (if applicable)	33
c) Any other business insights	34
5) Model building and interpretation.	36
a. Build various models (You can choose to build models for either or all of descriptive, predictive or prescriptive purposes)	37 - 116
b. Test your predictive model against the test set using various appropriate performance metrics	
c. Interpretation of the model(s)	
6) Model Tuning and business implication	37 - 116
Ensemble modelling, wherever applicable	
Any other model tuning measures (if applicable)	
Interpretation of the most optimum model and its implication on the business	115

List of Figures

Fig no-1: Outlier Check -----	→ 5
Fig no-2: Count plot -----	→ 6
Fig no-3: Count plot -----	→ 7
Fig no-4: Pie Chart -----	→ 7
Fig no-5: Continuous variable plot -----	→ 8
Fig no-6: Histogram of Tenure -----	→ 9
Fig no-7: Boxplots -----	→ 10
Fig no-8: Count plot w.r.t Target -----	→ 11
Fig no-9: Count plot w.r.t Target -----	→ 11
Fig no-10: Histograms -----	→ 12
Fig no-11: Scatterplot -----	→ 13
Fig no-12: Heatmap -----	→ 13
Fig no-13: Pair plot -----	→ 14
Fig no-14: Missing Value Visualization -----	→ 15
Fig no-15: Missing Value Visualization -----	→ 16
Fig no-16: Boxplot post Outlier Treatment -----	→ 16
Fig no-17: Count plot of City Tier -----	→ 18
Fig no-18: City Tier Vs Revenue -----	→ 18
Fig no-19: City Tier Vs Payment mode -----	→ 19
Fig no-20: City Tier Vs Payment mode Vs Total Revenue -----	→ 20
Fig no-21: Male Vs Female Segmentation -----	→ 21
Fig no-22: Bar plot male customer -----	→ 22
Fig no-23: Facet grid -----	→ 22
Fig no-24: Bar plot Female customer -----	→ 23
Fig no-25: Facet grid -----	→ 23
Fig no-26: Facet grid -----	→ 23
Fig no-27: Facet grid -----	→ 24
Fig no-28: Elbow plot -----	→ 25
Fig no-29: Silhouette plot -----	→ 25
Fig no-30: Cluster-Login Device Vs Revenue -----	→ 26
Fig no-31: Cluster-Login Device Vs Cashback -----	→ 27
Fig no-32: Count plot Churn Vs Clusters -----	→ 27
Fig no-33: Count plot Payment Vs Clusters -----	→ 28
Fig no-34: Count plot account segment Vs Clusters -----	→ 28
Fig no-35: Count plot Gender Vs Clusters -----	→ 28
Fig no-36: Count plot City Tier Vs Clusters -----	→ 29
Fig no-37: Count plot Account user count Vs Clusters -----	→ 29
Fig no-38: Count plot CC Agent score Vs Clusters -----	→ 30
Fig no-39: Count plot Service score Vs Clusters -----	→ 30
Fig no-40: Tenure Vs Clusters -----	→ 31
Fig no-41: Boxplot Account segment Vs Clusters -----	→ 31
Fig no-42: Class imbalanced plot -----	→ 33
Fig no-43: Confusion Matrix LR -----	→ 37

Fig no-44: ROC Curve LR -----	→ 37
Fig no-45: Confusion Matrix LR Smote -----	→ 39
Fig no-46: ROC Curve LR Smote -----	→ 39
Fig no-47: Confusion Matrix LR Smote -----	→ 41
Fig no-48: ROC Curve LR Smote -----	→ 42
Fig no 49: Confusion Matrix LDA -----	→ 43
Fig no 50: ROC Curve LDA -----	→ 44
Fig no 51: Confusion Matrix LDA Smote -----	→ 45
Fig no 52: ROC Curve LDA Smote -----	→ 46
Fig no 53: Confusion Matrix LDA Tuned -----	→ 47
Fig no 54: ROC Curve LDA Tuned -----	→ 48
Fig no 55: Confusion Matrix SVC -----	→ 49
Fig no 56: ROC Curve SVC -----	→ 50
Fig no 57: Confusion Matrix SVC Smote -----	→ 51
Fig no 58: ROC Curve SVC Smote -----	→ 52
Fig no 59: Confusion Matrix SVC Tuned -----	→ 54
Fig no 60: ROC Curve SVC Tuned -----	→ 54
Fig no 61: Confusion Matrix KNN_3 -----	→ 56
Fig no 62: ROC Curve KNN_3 -----	→ 56
Fig no 63: CV Vs K – KNN -----	→ 57
Fig no 64: MCE Vs K – KNN -----	→ 57
Fig no 65: Confusion Matrix KNN_1 -----	→ 59
Fig no 66: ROC Curve KNN_1 -----	→ 59
Fig no 67: Confusion Matrix KNN_3 Smote -----	→ 61
Fig no 68: ROC Curve KNN_3 Smote -----	→ 61
Fig no 69: Confusion Matrix KNN Tuned -----	→ 63
Fig no 70: ROC Curve KNN Tuned -----	→ 64
Fig no 71: Confusion Matrix NB Model -----	→ 65
Fig no 72: ROC Curve NB Model -----	→ 66
Fig no 73: Confusion Matrix NB Smote -----	→ 67
Fig no 74: ROC Curve NB Smote -----	→ 68
Fig no 75: Confusion Matrix DT -----	→ 69
Fig no 76: ROC Curve DT -----	→ 70
Fig no 77: Feature Imp DT -----	→ 70
Fig no 78: Confusion matrix DT smote -----	→ 72
Fig no 79: ROC Curve DT smote -----	→ 72
Fig no 80: Confusion Matrix DT Tuned -----	→ 74
Fig no 81: ROC Curve DT Tuned -----	→ 74
Fig no 82: Feature Imp DT Tuned -----	→ 75
Fig no 83: Confusion Matrix RF -----	→ 76
Fig no 84: ROC Curve RF -----	→ 77
Fig no 85: Feature IMP RF -----	→ 78
Fig no 86: Confusion Matrix RF smote -----	→ 79
Fig no 87: ROC Curve RF smote -----	→ 79
Fig no 88: Confusion Matrix RF Tuned -----	→ 81
Fig no 89: ROC Curve RF Tuned -----	→ 82
Fig no 90: Feature Imp RF Tuned -----	→ 82

Fig no 91: Confusion Matrix Bagging -----	→ 84
Fig no 92: ROC Curve Bagging -----	→ 84
Fig no 93: Confusion Matrix Bagging smote -----	→ 86
Fig no 94: ROC Curve Bagging smote -----	→ 86
Fig no 95: Confusion Matrix Bagging Tuned-----	→ 88
Fig no 96: ROC Curve Bagging Tuned -----	→ 88
Fig no 97: Confusion Matrix Ada Boost -----	→ 90
Fig no 98: ROC Curve Ada Boost -----	→ 91
Fig no 99: Feature Imp Ada Boost -----	→ 91
Fig no 100: Confusion matrix Ada Boost smote -----	→ 93
Fig no 101: ROC Curve Ada Boost smote -----	→ 93
Fig no 102: Confusion Matrix Ada Boost Tuned -----	→ 95
Fig no 103: ROC Curve Ada Boost Tuned -----	→ 95
Fig no 104: Feature Imp Ada Boost Tuned -----	→ 96
Fig no 105: Confusion Matrix GDB -----	→ 97
Fig no 106: ROC Curve GDB -----	→ 98
Fig no 107: Feature Imp GDB -----	→ 98
Fig no 108: Confusion Matrix GDB smote -----	→ 100
Fig no 109: ROC Curve GDB smote -----	→ 100
Fig no 110: Confusion Matrix GDB Tuned -----	→ 102
Fig no 111: ROC Curve GDB Tuned -----	→ 103
Fig no 112: Feature Imp GDB Tuned -----	→ 103
Fig no 113: Confusion Matrix XGB -----	→ 105
Fig no 114: ROC Curve XGB -----	→ 105
Fig no 115: Feature Imp XGB -----	→ 106
Fig no 116: ROC Curve XGB smote-----	→ 108
Fig no 117: Confusion Matrix XGB Tuned -----	→ 109
Fig no 118: ROC Curve XGB Tuned -----	→ 110
Fig no 119: Feature Imp XGB Tuned -----	→ 110
Fig no 120: CM LightGBM -----	→ 112
Fig no 121: ROC Curve LightGBM -----	→ 112
Fig no 122: Feature Imp LightGBM -----	→ 112
Fig no 123: Base Model Comparison -----	→ 113
Fig no 124: Tuned Model Comparison -----	→ 114
Fig no 125: Model Comparison Table -----	→ 116

Introduction of the business problem

1.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 do churn prediction of the accounts and provide segmented offers to the potential churners. In this company, account churn is a major thing because 1 account can have multiple customers. hence by losing one account the company might be losing more than one customer.

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 your recommendation. Hence be very careful while providing campaign recommendation

- Here the objective is to Build a Model to identify the Customers who will potentially Churn and provide useful business recommendations. Retaining a loyal customer is far more important than acquiring a new one. Hence predictive analysis of customer Retention is absolutely necessary in all business.

1.b) Need of the study/project:

- Customer churn is when the customers either switch from their incumbent service provider to its competitor or stop using the service altogether.
- There are so many competitions available in the market for any product /services we use, so it is very necessary to study about customers likes & dislikes, their preferences about the product, their needs etc.
- There is a need to Study about this project in detail because Churning of customers impacts the Revenue of the company.
- A High Churn Rate will result in decrease in Companies Revenue.
- Customer churn prediction using machine learning will help us to identify risky customers and understand why customers are willing to leave.

1.c) Understanding business/social opportunity:

- The Goal of this project is to predict the type of customers that has the potential to churn by identifying various features available that can help in minimizing the customers churn rates and get the right business insights.
- We need to find a way to Retain the customers as long as possible.
- By knowing the customer needs, the kind of services they prefer, we need to keep the customers satisfied.

- By knowing the reason that are affecting customer churn will help us predict the churn and then avoid it.
- There can be many reasons for customers churn - like poor customer Services, another one could be prices.
- attracting new customers is much more expensive than retaining existing ones.

Data Report

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

- Data would be collected from the existing business and their past records of various customers. Data is collected w.r.t different features like gender, Marital status, Service score, Revenue generated etc.

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

❖ Sample of the Dataset:

	AccountID	Churn	Tenure	City_Tier	CC_Contacted_LY	Payment	Gender	Service_Score	Account_user_count	account_segment	CC_Agent_Score
0	20000	1	4	3.0	6.0	Debit Card	Female	3.0	3	Super	2.0
1	20001	1	0	1.0	8.0	UPI	Male	3.0	4	Regular Plus	3.0
2	20002	1	0	1.0	30.0	Debit Card	Male	2.0	4	Regular Plus	3.0
3	20003	1	0	3.0	15.0	Debit Card	Male	2.0	4	Super	5.0
4	20004	1	0	1.0	12.0	Credit Card	Male	2.0	3	Regular Plus	5.0

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

❖ Shape of the Dataset:

- The Number of Rows present in the Dataset are: 11260
- The Number of Columns present in the Dataset are: 19

❖ Statistical Summary:

	count	mean	std	min	25%	50%	75%	max
AccountID	11260.0	25629.500000	3250.626350	20000.0	22814.75	25629.5	28444.25	31259.0
Churn	11260.0	0.168384	0.374223	0.0	0.00	0.0	0.00	1.0
City_Tier	11148.0	1.653929	0.915015	1.0	1.00	1.0	3.00	3.0
CC_Contacted_LY	11158.0	17.867091	8.853269	4.0	11.00	16.0	23.00	132.0
Service_Score	11162.0	2.902526	0.725584	0.0	2.00	3.0	3.00	5.0
CC_Agent_Score	11144.0	3.066493	1.379772	1.0	2.00	3.0	4.00	5.0
Complain_ly	10903.0	0.285334	0.451594	0.0	0.00	0.0	1.00	1.0

❖ Observations:

- Columns like Tenure, Account_user_count, rev_per_month, rev_growth_yoy,
- coupon_used_for_payment, Day_Since_CC_connect and cashback are continuous columns but missing from the Describe output being a numeric column. This indicates the presence of some Anomalies or some Special Character in these columns.
- Columns like City_Tier, Service_Score, CC_Agent_Score and Complain_ly seems Contain Categorical data.
- Average times Customers contacted CC is around 17 times and Max 132 times.
- Average complains raised by the customers is around 28%.
- 75% of customers having service score less than or equal to 3.
- 75% of customers have given a satisfaction score of less than or equal to 4.

2.c) Understanding of attributes (variable info, renaming if required)

❖ Dataset Info:

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

❖ Observations

- Null values are present in the dataset.
- There are 5 float64, 2 int64 and 12 object Datatypes.
- Columns like Tenure, Account_user_count, rev_per_month, rev_growth_yoy, coupon_used_for_payment, Day_Since_CC_connect and cashback contains integer values but their Datatype is object here.
- This may be due to some Anomalies or any special character present in the data.
- We need to inspect such columns for any Anomalies/Special character present in it.

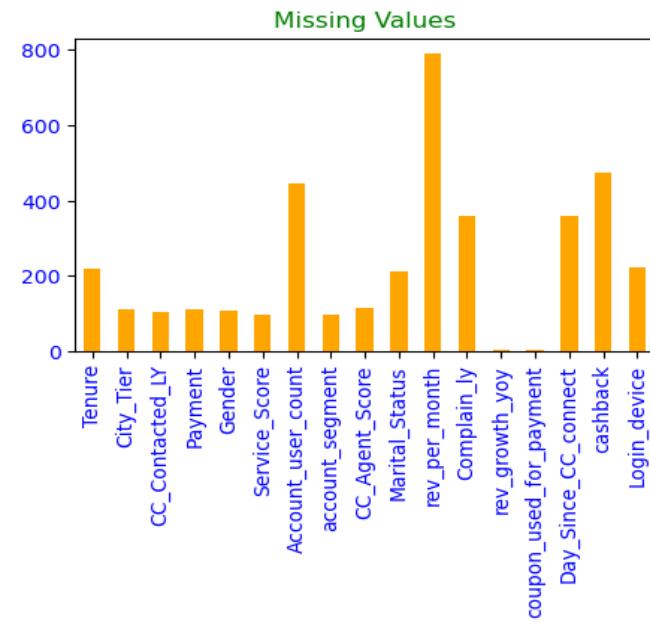
❖ Duplicates:

- There are No Duplicate Rows present in the Dataset.

❖ Missing Values:

Missing Values:

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



❖ Checking for Anomalies/Discrepancies:

- Column Tenure contains a Special character '#' as an entry making it an object datatype.
- Column Gender contains four unique values as 'Male', 'Female', ' M', 'F', which needs to be replaced with 'Male' and 'female'.
- Account_user_count also contains special character '@' as one of the entries.
- Column account_segment contains some Anomalies like 'Regualr plus' and 'Super Plus' is written in two different ways which needs to be corrected.
- Column rev_per_month Contains '+' plus which needs to be replaced.
- Column rev_growth_yoy and Day_Since_CC_connect contains '\$' character.
- Column coupon_used_for_payment contains #, \$ and * characters.
- Column Login_device Contains &&& as one of its categories.

- We need to inspect the Cashback column for any non-digit character present in it.
- We saw that Cashback column contains special character '\$' due to which it is considered as 'object' datatype column.

We will Replace the special characters present in the dataset with np. Nan and later will impute the Null values.

❖ Post Removing Anomalies:

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

❖ Checking for Outliers:

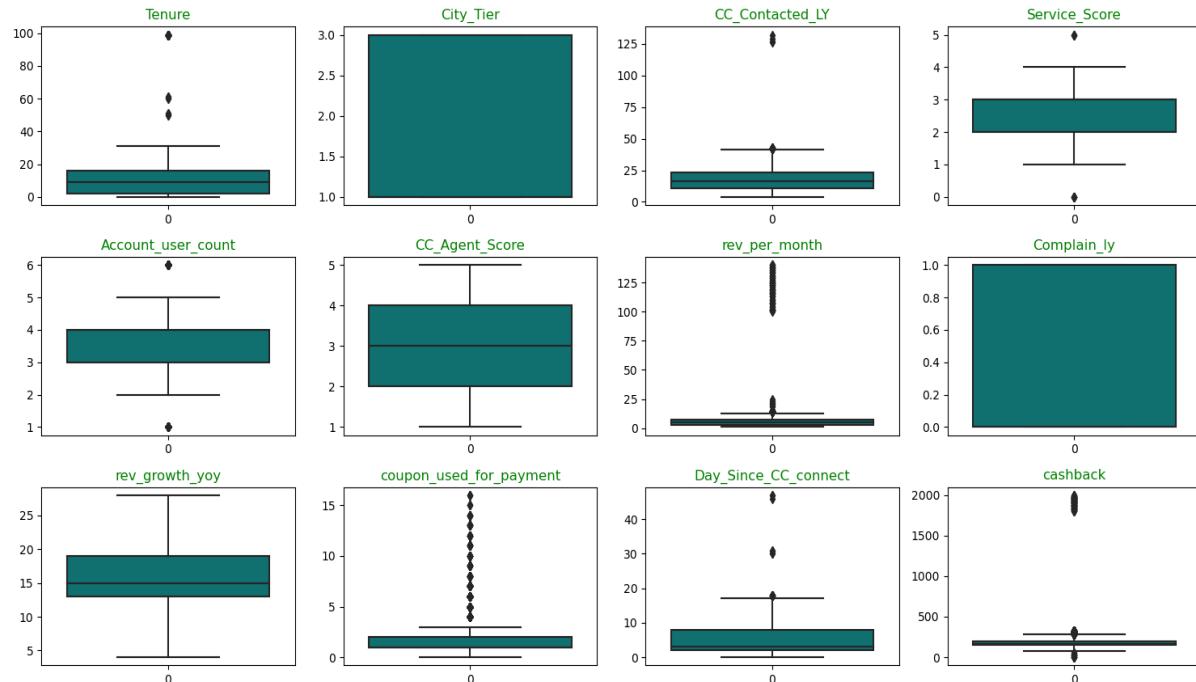


Fig no-1: Outlier Check

❖ Observations:

- There are many features showing the presence of Outliers.

Exploratory data analysis

Univariate Analysis

- ❖ Count plot of all Categorical Columns:

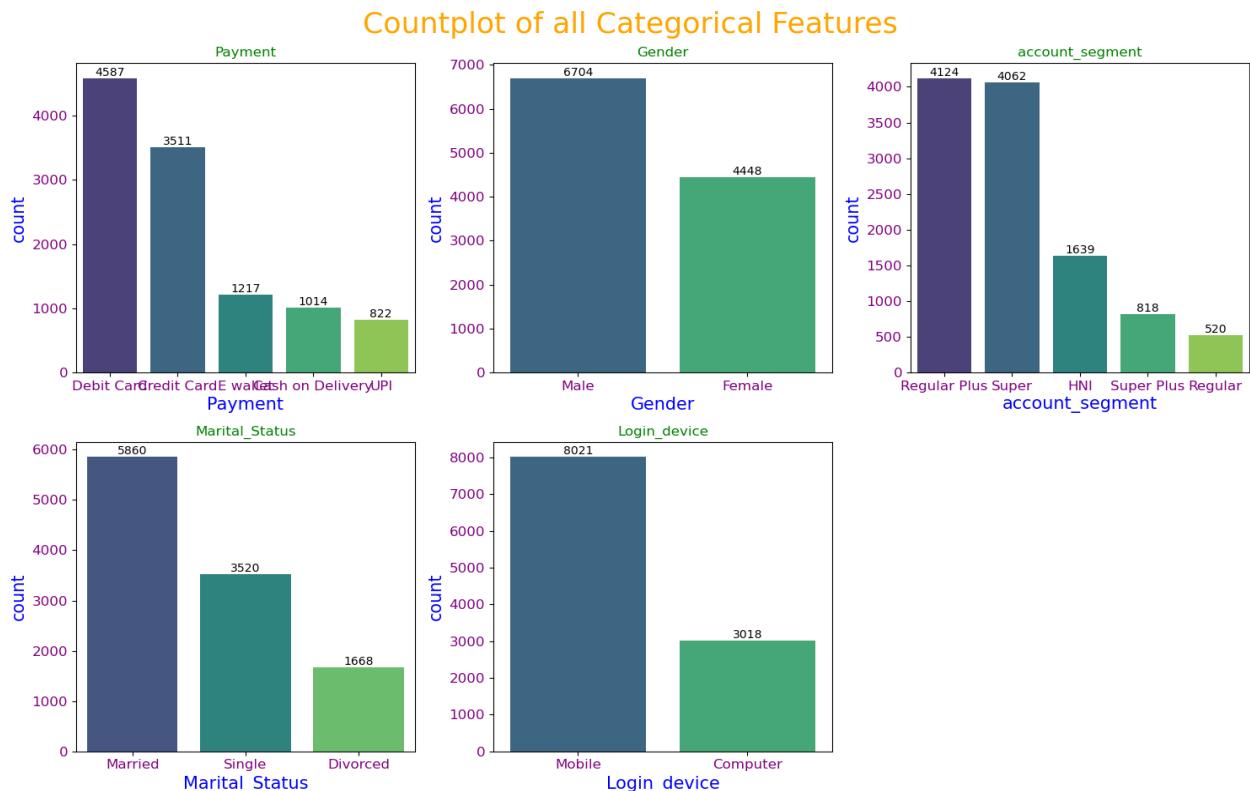


Fig no-2: Count plot

Insights:

- Most Customers Preferred Payment mode are Debit card and Credit Card.
- Very few customers make payment through UPI.
- Male customers are more as compared to female.
- There are more customers that belong to account_segment Regular plus and Super.
- Very less customers belongs to Regular segment on the basis of their spend.
- Married customers are more in number as compared to Singles and Divorced.
- Maximum of the customers prefers to Login Via Mobile as it is convenient to carry and they can Login from any Location.
- Very less customers Login via Computer.

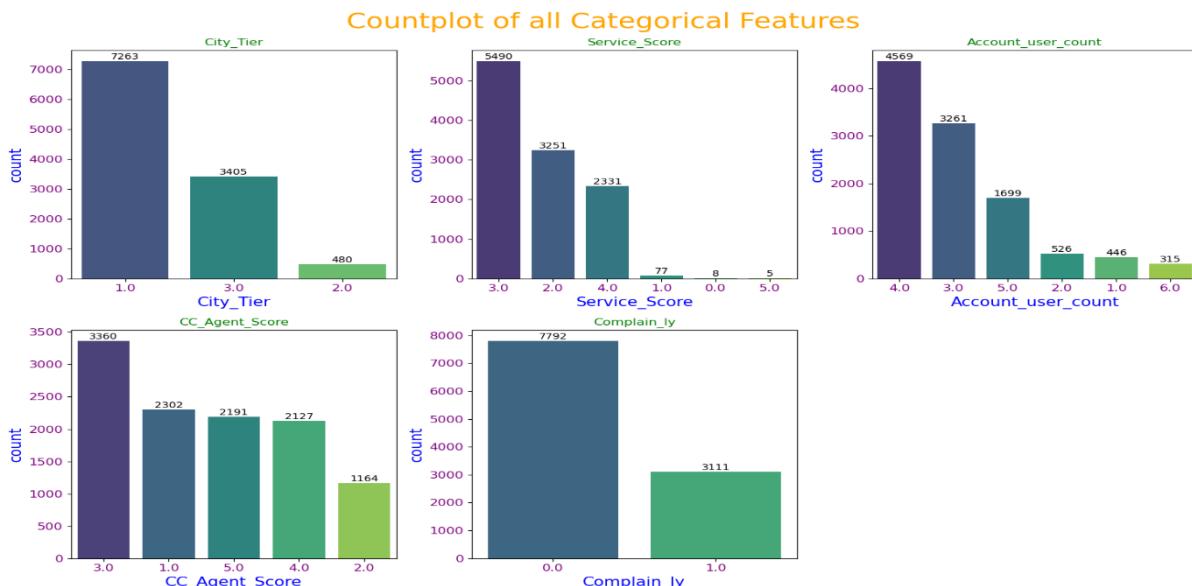


Fig no-3: Count plot

Insights:

- Maximum customers belong to Tier-1 cities followed by Tier-3 cities and very few belongs to Tier-2 cities.
- Service score given by Maximum customers is 3, which states that customers are not fully satisfied by the service provided by the company.
- More numbers of Accounts are linked with 4 members, followed by 3 and 5 members per Account.
- Very few accounts are linked with 1,2 and 6 Members.
- Satisfaction score given by customers on an Average is 3 for Customer Care Services.
- Also, Excellent Score of 4 and 5 for Customer care services are almost equal in number. So, they are Satisfied customers too.
- It seems very less Complaint has been Raised by the customers in last 12 months.

❖ Distribution of Target Variable:

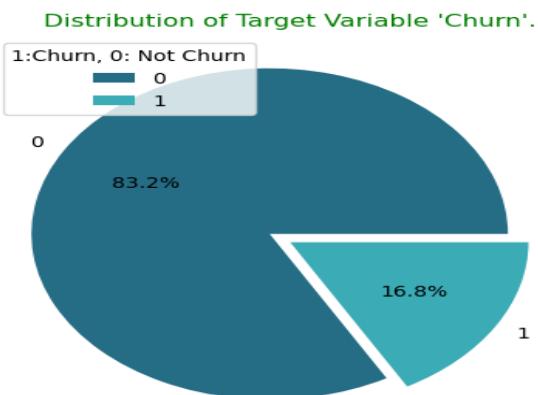


Fig no-4: Pie Chart

Insights:

- 16.8% of the customers have churned whereas 83.2% of the customers have not churned.
- seems Imbalance in Target class.

❖ Histogram and Boxplot of all continuous variables

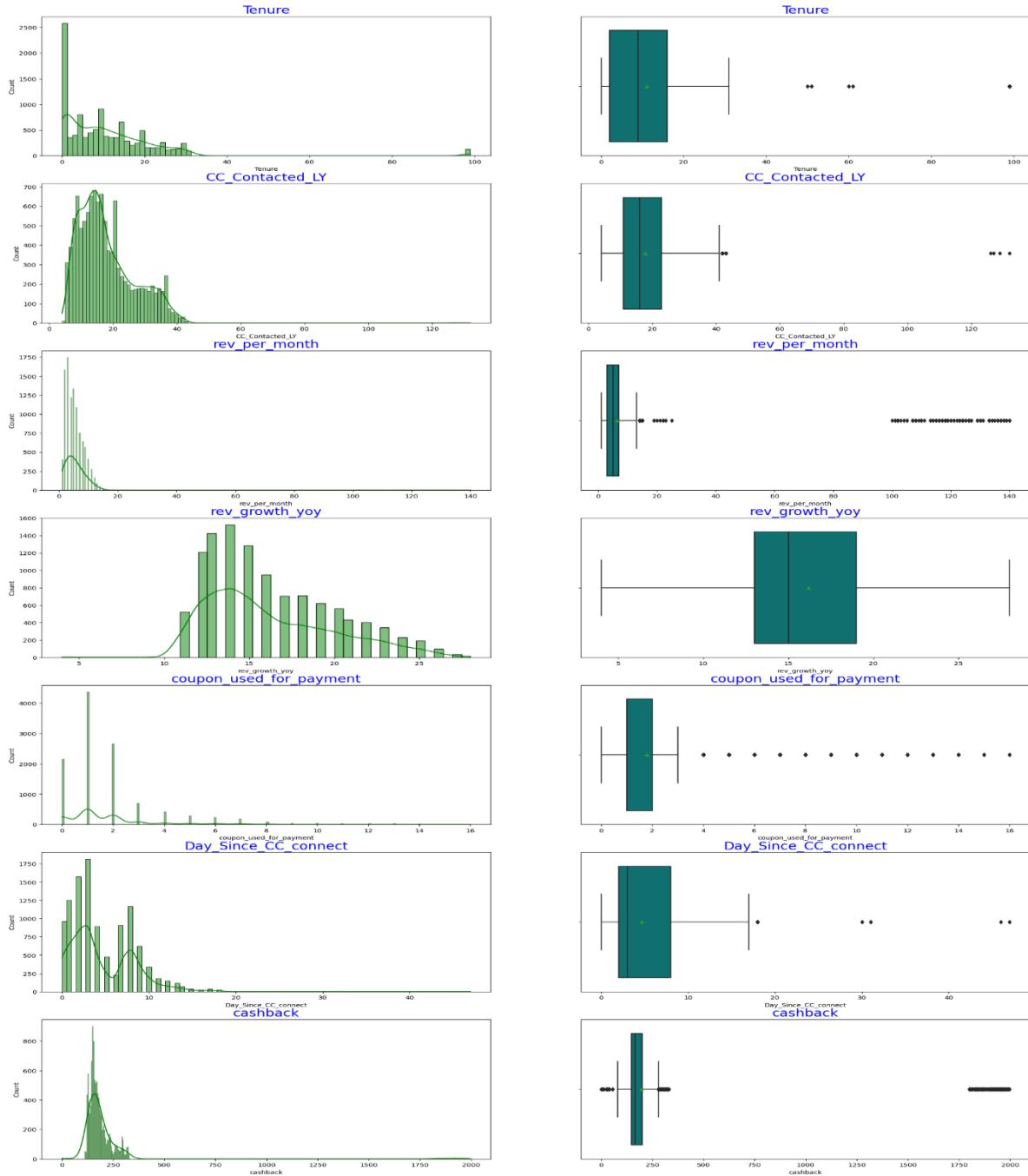


Fig no-5: Continuous variable plot

Insights:

- All the Variables are Right Skewed showing the presence of Outliers.
- Maximum customers have a Tenure of less than a month.
- There are also some customers having a Tenure of more than 50 months, Max up to 100 months.
- Maximum number of customers have contacted Customer care 11 to 23 times in last 12 months.
- The Median of the Monthly average revenue generated by the company is around 10k (Assuming the Currency is in Thousands).
- Also, this feature is highly Right Skewed showing monthly avg revenue generated by the company more than 100k.
- There is an Approx 16% growth in revenue on an average, generated by the account in the last year compared to the previous years.
- On an Average, 2 times coupons were used to do the payment.
- Also, it seems some customers used the coupons more than 4 times to max 16 times.
- Avg no of days since customers have not contacted CC is around 5 days.
- On an Average, Cashback generated by the customers is around Rs 200/- in the last 12 months.
- Also, some customers generated cashback of more than Rs 1750/- monthly.

❖ Visualizing Tenure:

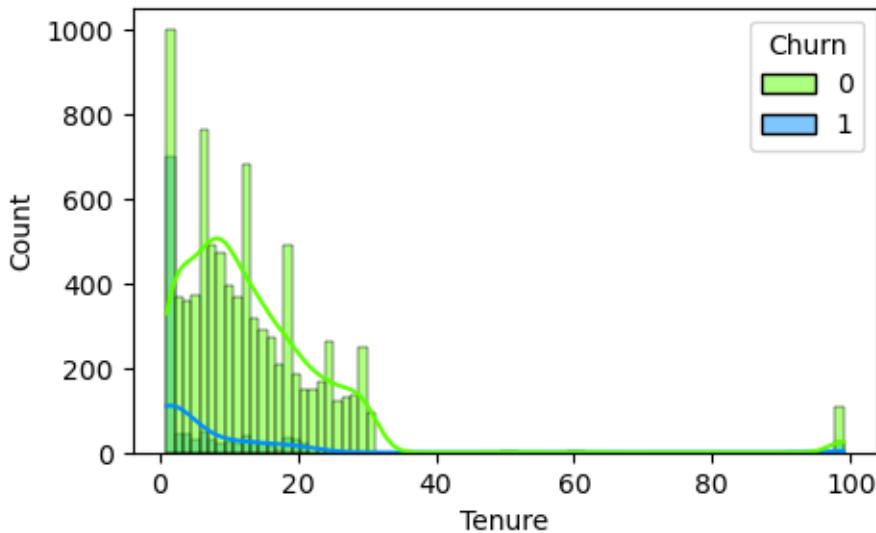


Fig no-6: Histogram of Tenure

Insights:

- Customers having a Tenure of less than 15 months, Churns more.
- There are some Loyal Customers having Tenure of more than 80 months.
- Distribution seems Overlapping.

Bivariate Analysis

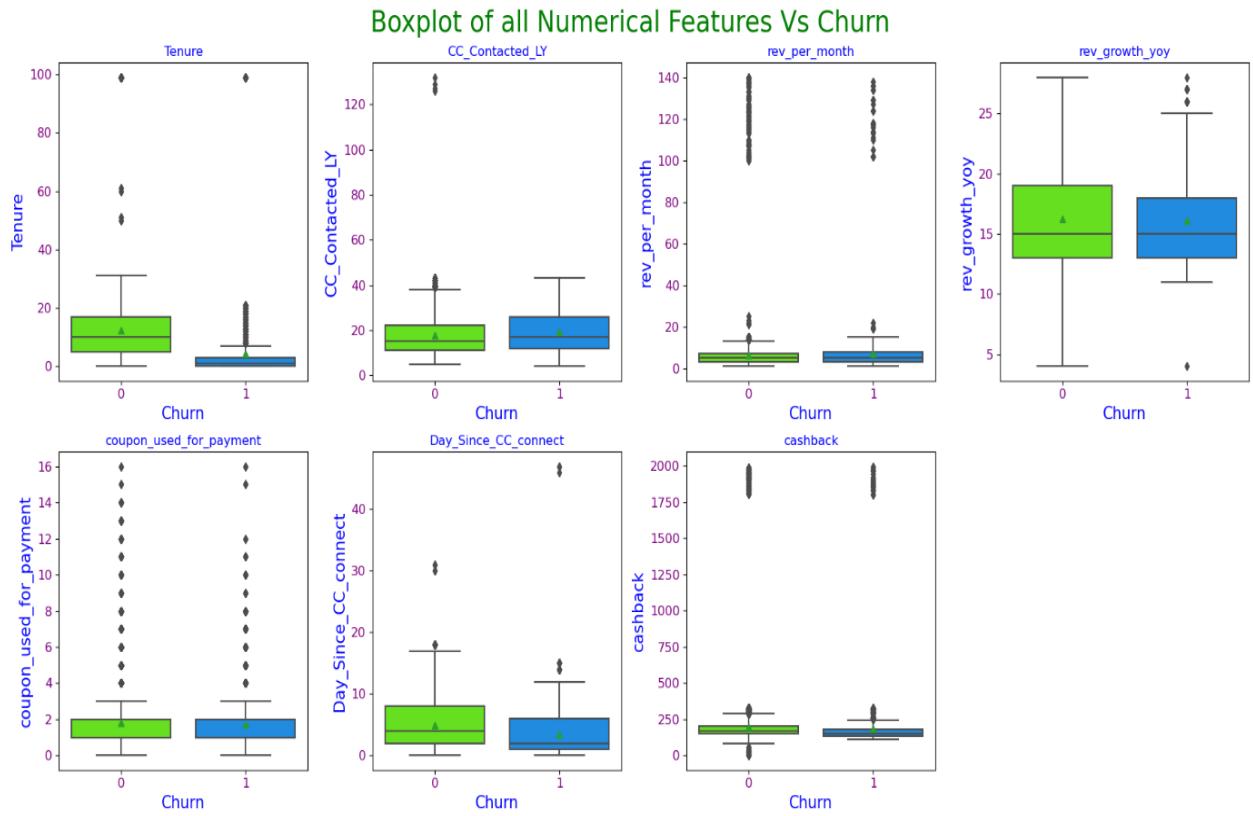


Fig no-7: Boxplots

Insights:

- It seems that when the customers contact the Customers care more their queries gets resolved due to which they don't churn and keep using the services more hence reducing the Churn rate.
- So, we can say Customer Care is also playing an Important role in Retaining Customers.
- We see that median value of the Tenure and Days_since_CC_connect for Churners is less compared to that of non-Churners.
- The Distribution of Coupon_used_for_payment, Cashback and rev_per_month is same for both Churners and Non-churners.
- There is no difference in median rev_growth_yoy between churers and non-churers.

- ❖ Count plot of all Categorical Variables Vs Target Variables

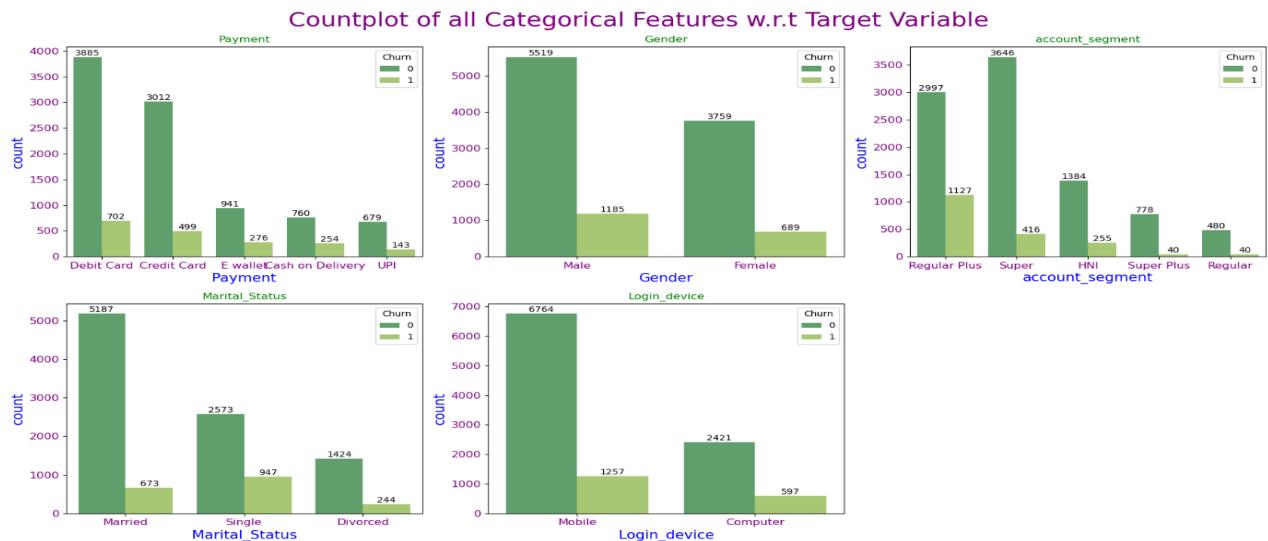


Fig no-8: Count plot w.r.t Target

Insights:

- The proportions of Churners are more of Male customers as compared to Female.
- Most of the churners make payment via Debit Card.
- Non-churners preferable mode of payment are Debit cards and Credit Cards.
- Most of the churners belongs to Regular plus and Super account segment.
- Maximum non-Churners belongs to Super Segment followed by Regular plus.
- Most of the churners are single.
- Non-Churners are Maximum Married Couples.
- Most Preferred Login Device for both Churners and Non-Churners is Mobile since it's handy.

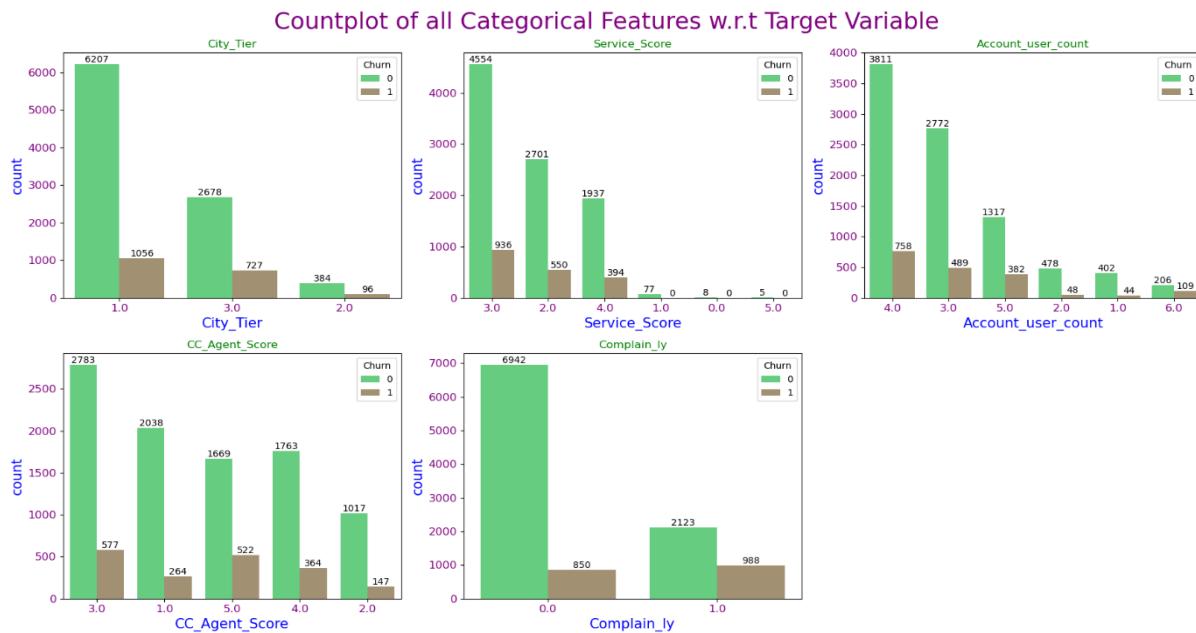


Fig no-9: Count plot w.r.t Target

Insights:

- Maximum non-churners belong to City Tier-1 followed by Tier-3.
- Churning rate of Customers from Tier-2 is very less, means customers tends continue the services seems they are more satisfied.
- Most of the Churners are from Tier-1.
- Maximum Service score given by customers is 3 by both churners and non-churners.
- Account tagged with 3,4 and 5 customers have more churning rate.
- Customers tagged with 2,1 and 6 accounts are mostly non-churners.
- Customers who have given an agent score of 3 and above show the Maximum churn rate.
- Maximum non-churners have given a Agent score of 3 and 1.
- Very few Churners have raised the complaints in the last 12 months. If they would have raised the complaint then may be their queries would have been resolved and they would not churn.
- It is evident from the fact that Customers who have raised the complaint maximum no of times are mostly non-churners. This shows that raising complaint have solved their issue and hence made them retain the use of service and decreasing churn Rate.

❖ Checking separation across the numerical variables:

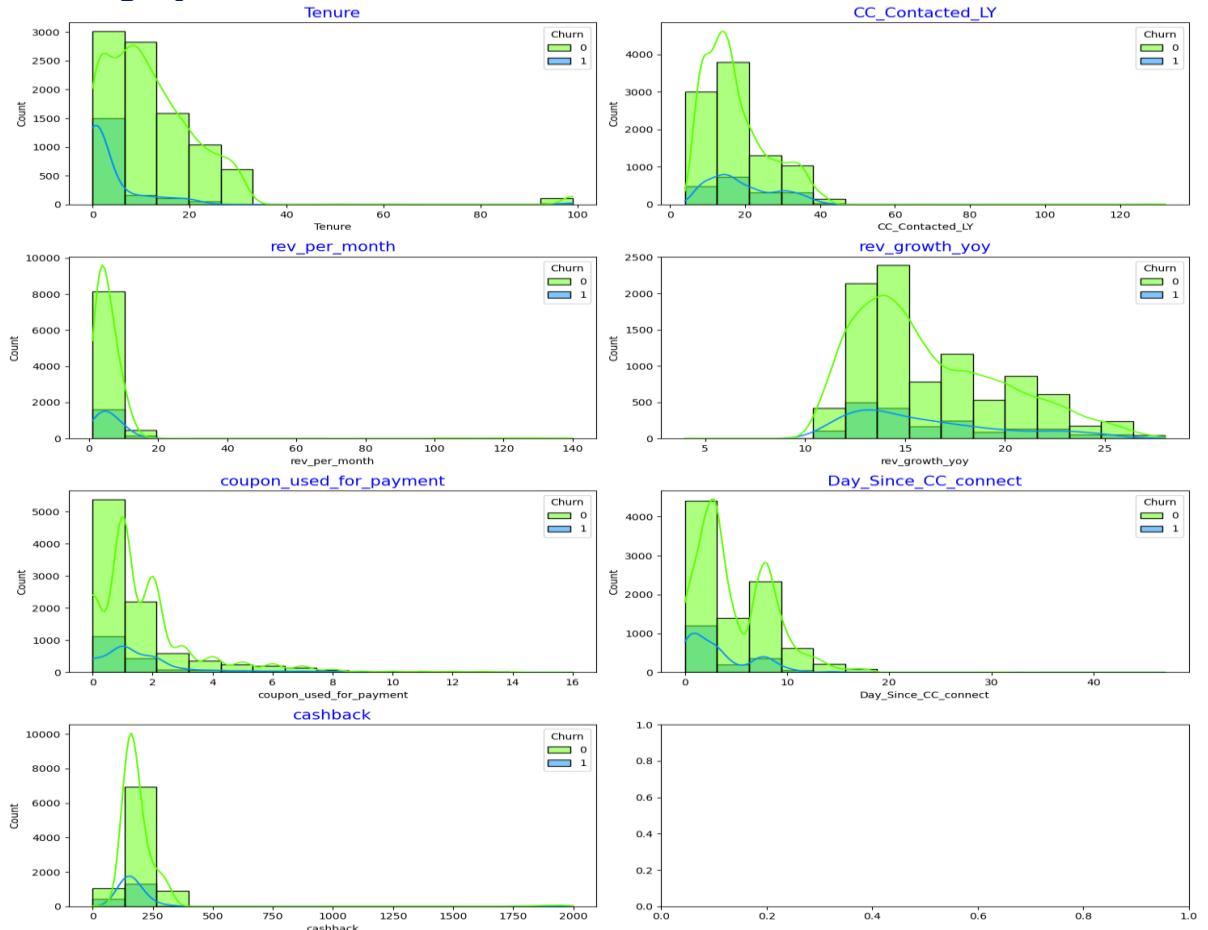


Fig no-10: Histograms

Insights:

- Most of the Distributions are Right Skewed.
- Customers having Short-term Tenure churns more.
- There are some Loyal customers also having Tenure more than 80 months.
- Maximum number of times Customer care was contacted is between 4 to 23 times.
- maximum customers use 1-2 coupons for payment.
- Most of the cashback are generated is around 0 to 300.
- Maximum cashbacks are generated by non-churners.

❖ Scatterplots:

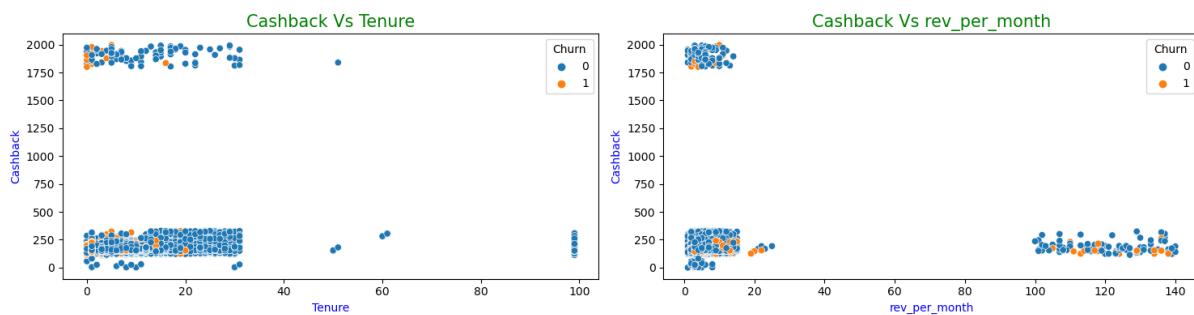


Fig no-11: Scatterplot

Insights:

- Most of the Cashback are for Tenure less than 35 months.
- Cashback Ranges between 0-280 and 1750-2000.
- Cashback of more than 1750 is mostly enjoyed by non-churners.

❖ Correlation Heatmap:

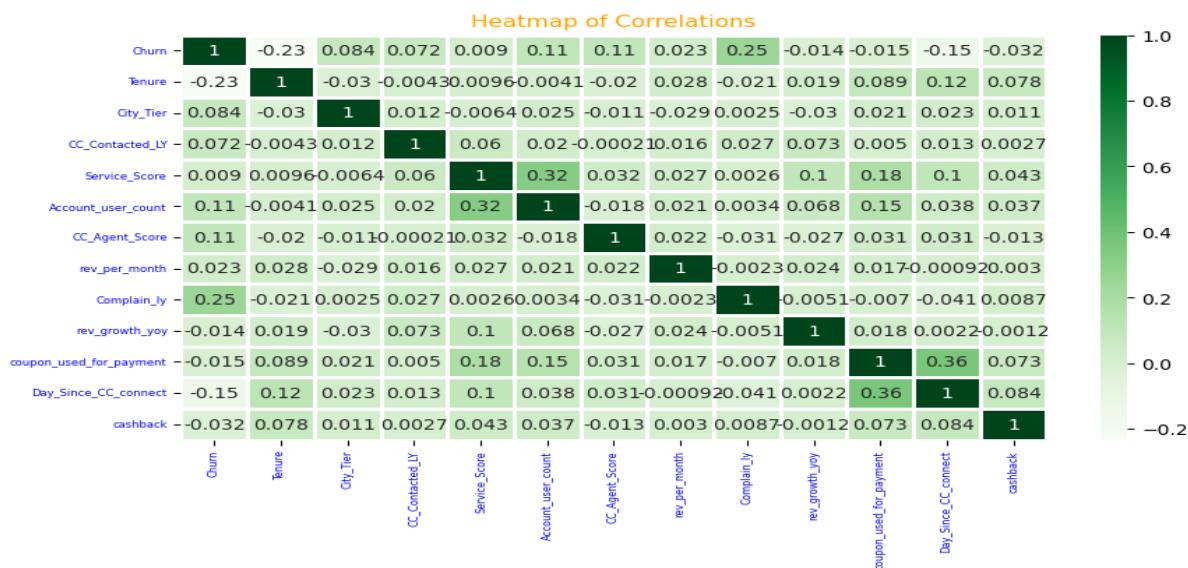


Fig no-12: Heatmap

Insights:

- Churn is negatively correlated with Tenure, having a correlation of -23%.
- The smaller the tenure value, the greater the churn rate.
- Correlation between Churn and Complain_ly is around 25%. The greater the complaint rate, the greater the churn rate.
- Service score has 32% correlation with Account user count and 18% correlation with Coupon used for payment.
- Coupon used for payment has 36% correlation with Days since CC connect, 18% with Service score and 15% with Account user count.
- There are no such features that are Highly Correlated with each other.
- There is no strong positive and Negative Correlations between the Predictors.
- There seems no Multicollinearity present in the data.

❖ Pair plot:

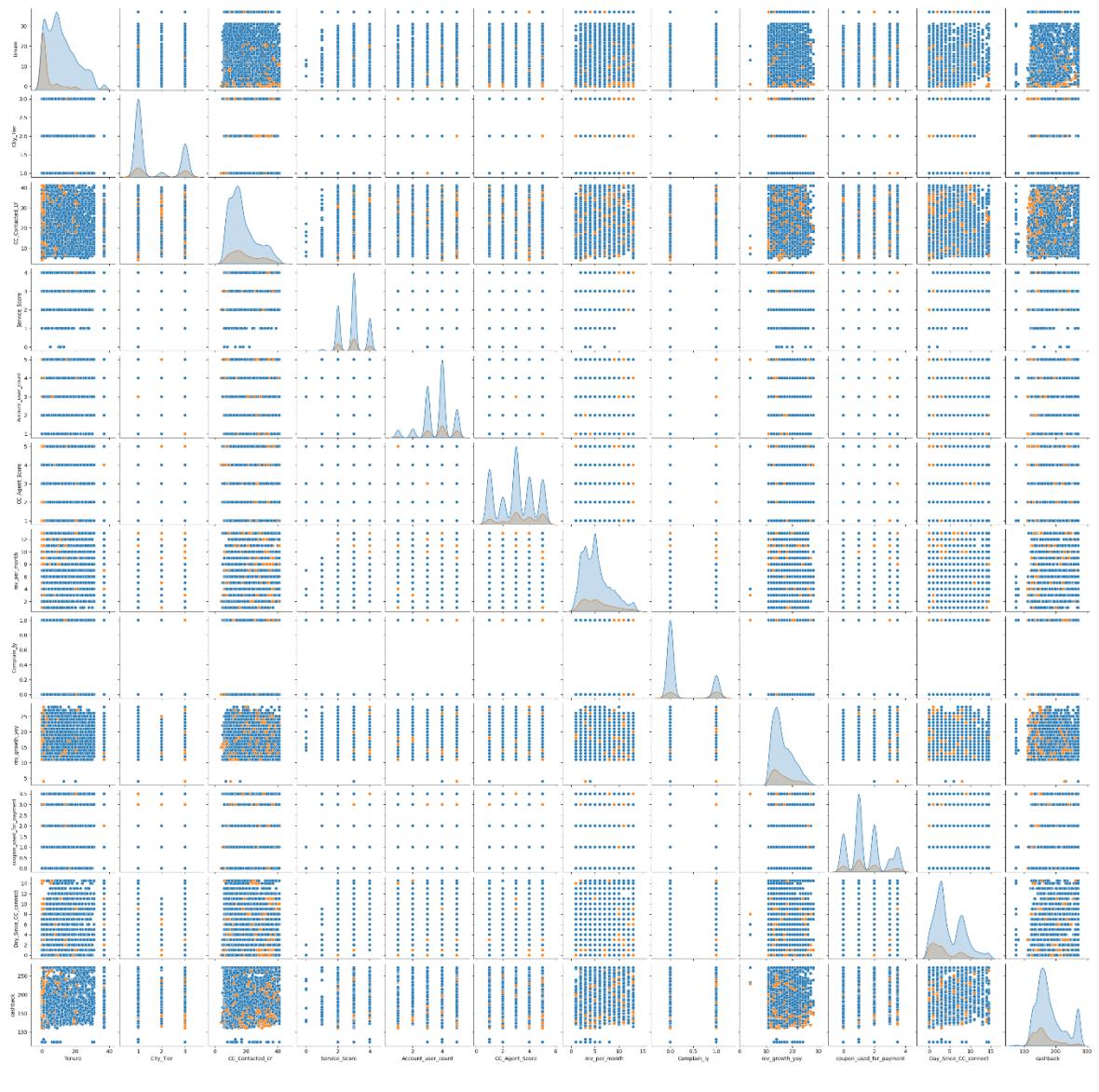


Fig no-13: Pair plot

Insights:

- Churners and non-churners seem overlapping each other in almost all the features.
- There is no Linear Pattern observed.
- Customers Churn more with Lowest Tenure.
- Tier-1 and Tier-3 Customers Churn Rate is more compared to Tier-2 cities customers.

Removal of unwanted variables (if applicable)

- Dropped column AccountID as it is not required for Analysis

Missing Value Treatment

Proportion of Missing Values in the following columns :

```

Churn          0.000000
Tenure         5.703820
City_Tier      2.930403
CC_Contacted_LY 2.668760
Payment        2.851910
Gender          2.825746
Service_Score   2.564103
Account_user_count 11.616954
account_segment  2.537938
CC_Agent_Score   3.035060
Marital_Status    5.546834
rev_per_month    20.695971
Complain_ly     9.340659
rev_growth_yoy   0.078493
coupon_used_for_payment 0.078493
Day_Since_CC_connect 9.366824
cashback         12.375720
Login_device     5.782313
dtype: float64

```

- Since the proportion of Missing Values is less than 30%, so we will impute the missing values rather than dropping them.
- ❖ Let's visually inspect the missing values in our data:

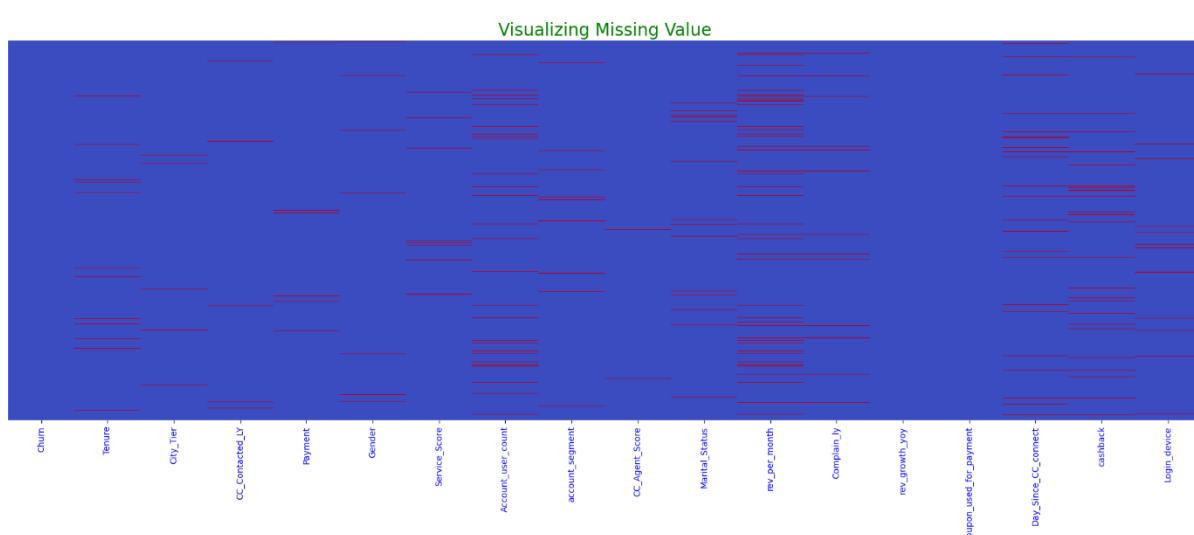


Fig no-14: Missing Value Visualization

❖ Let's Impute Categorical and Numerical Columns using Simple Imputer:

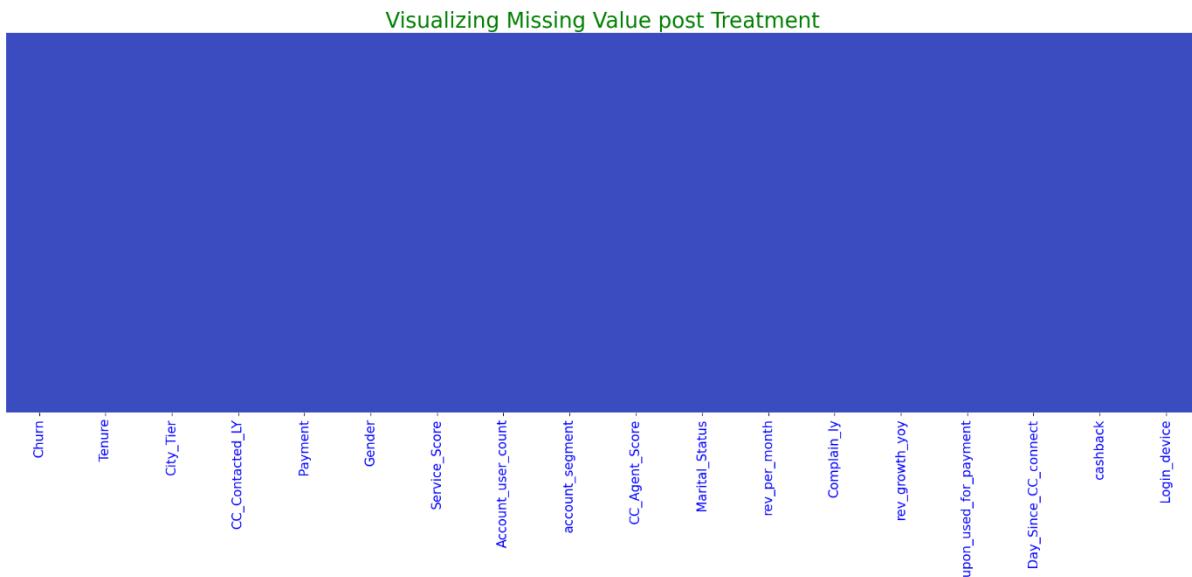


Fig no-15: Missing Value Visualization

➤ There are no Missing values Now.

Outlier Treatment

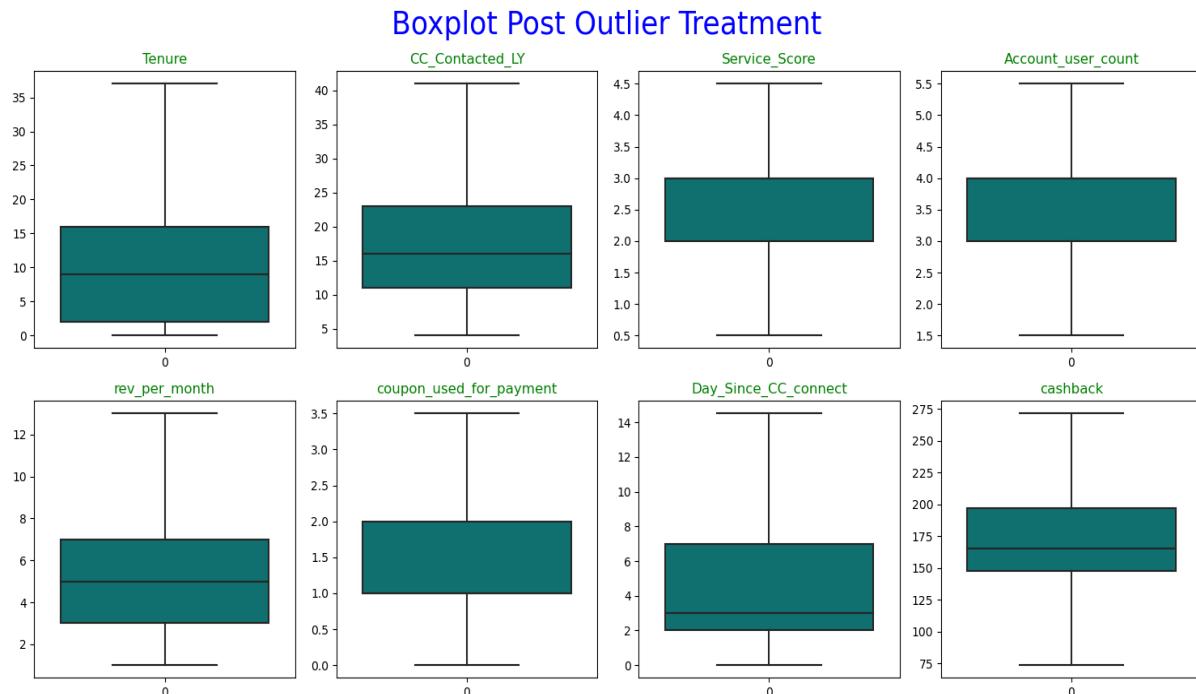


Fig no-16: Boxplot post Outlier Treatment

❖ Statistical Summary of the Customers who have Churned

	count	unique	top	freq	mean	std	min	25%	50%	75%	max
Churn	1896.0	NaN	NaN	NaN	1.0	0.0	1.0	1.0	1.0	1.0	1.0
Tenure	1896.0	NaN	NaN	NaN	3.660865	6.492737	0.0	0.0	1.0	4.0	37.0
City_Tier	1896.0	NaN	NaN	NaN	1.817511	0.957365	1.0	1.0	1.0	3.0	3.0
CC_Contacted_LY	1896.0	NaN	NaN	NaN	19.248945	8.856001	4.0	12.0	17.0	26.0	41.0
Payment	1896	5	Debit Card	724	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Gender	1896	2	Male	1207	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Service_Score	1896.0	NaN	NaN	NaN	2.917722	0.700985	2.0	2.0	3.0	3.0	4.0
Account_user_count	1896.0	NaN	NaN	NaN	3.880802	0.90179	1.0	3.0	4.0	5.0	5.0
account_segment	1896	5	Regular Plus	1145	NaN	NaN	NaN	NaN	NaN	NaN	NaN
CC_Agent_Score	1896.0	NaN	NaN	NaN	3.386603	1.333874	1.0	3.0	3.0	5.0	5.0
Marital_Status	1896	3	Single	947	NaN	NaN	NaN	NaN	NaN	NaN	NaN
rev_per_month	1896.0	NaN	NaN	NaN	5.506857	3.023209	1.0	3.0	5.0	7.0	13.0
Complain_ly	1896.0	NaN	NaN	NaN	0.521097	0.499687	0.0	0.0	1.0	1.0	1.0
rev_growth_yoy	1896.0	NaN	NaN	NaN	16.077532	3.862519	4.0	13.0	15.0	18.0	28.0
coupon_used_for_payment	1896.0	NaN	NaN	NaN	1.429325	1.084681	0.0	1.0	1.0	2.0	3.5
Day_Since_CC_connect	1896.0	NaN	NaN	NaN	3.349156	3.139531	0.0	1.0	3.0	5.0	14.5
cashback	1896.0	NaN	NaN	NaN	162.960654	36.801456	110.09	136.405	153.46	176.645	271.44
Login_device	1896	2	Mobile	1299	NaN	NaN	NaN	NaN	NaN	NaN	NaN

Useful Insights:

- Total Churners = 1896.
- Average Tenure almost 4 months, Maximum Tenure is of 37 months.
- Tier 1 & 3 customers churns more.
- Most of the churners are Male and Singles.
- Preferred payment mode is Debit Cards holding regular Plus account and most likely Login via Mobile.
- Avg Cashback generated is around Rs 163 and Max cashback generated is around Rs 271.

Data Visualization using Segmentation

Let's Segment the data based on City Tiers, Payment Mode and Genders Respectively and will Draw useful insights if any.

Segment on the basis of City Tiers

There are 3 city tiers mentioned in the dataset. Generally, tier 1 cities are considered as the major metro cities where the people tend to use more DTH Services. So, accordingly, can we say that tier 1 city customers tend to generate more avg revenue as compared to tier 2 and tier 3 city customers. Let's visualize this and find it it's True or not?

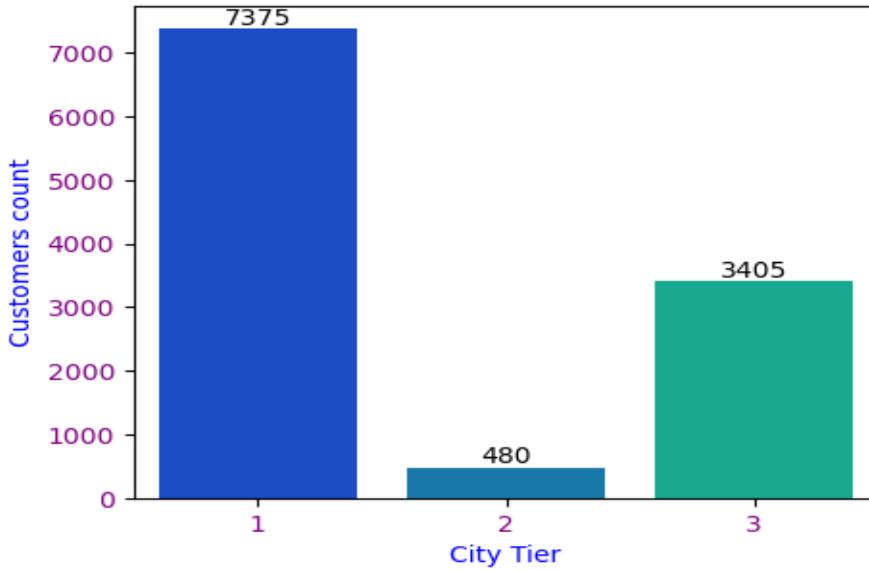


Fig no-17: Count plot of City Tier

Insights:

- Count of customers are more in Tier-1 followed by Tier-3.
- Very less customers belong to Tier-2.

❖ City Tier Vs Monthly Average Revenue:

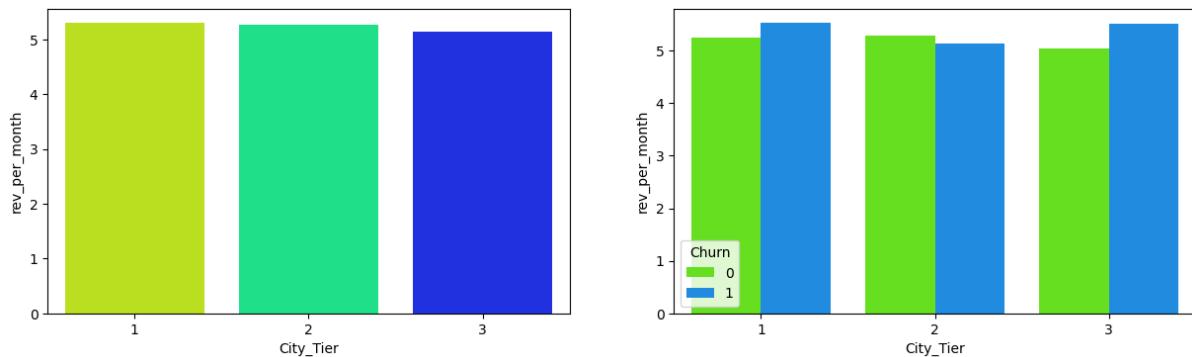


Fig no-18: City Tier Vs Revenue

Insights:

- We can see that Avg Revenue Generation is same across all the City Tiers.
- Though the Customers using the Services in Tier-2 is very less as compared to Tier-1 & 3, But the Avg Revenue generated per month by the account is same across all the Tiers.
- It seems Customers are more satisfied in Tier-2 hence they Retain to use the Services, generating greater revenue.
- As we can see from the plot, City_Tier 1 & 2 has a slightly higher mean of Avg Revenue per month as compared to City_Tier 3 which are more or less the same.

So, our assumption here that tier 1 city customers tend to generate more avg revenue cannot be validated looking at the plot.

Payment Mode

There are different payment modes (CC, DC, COD, E-wallet & UPI). Depending on the city tiers, Let's Visualize and find the preferred payment modes used by the customers

Payment	Cash on Delivery	Credit Card	Debit Card	E wallet	UPI
City_Tier					
1	732	2740	3407	12	484
2	31	100	123	0	226
3	251	671	1166	1205	112

- 46.2 % of customers prefer Debit card as preferred payment mode in Tier 1 cities.
- 47.08 % of customers prefer UPI as preferred payment mode in Tier 2 cities.
- 35.39 % of customers prefer E wallet as preferred payment mode in Tier 3 cities.

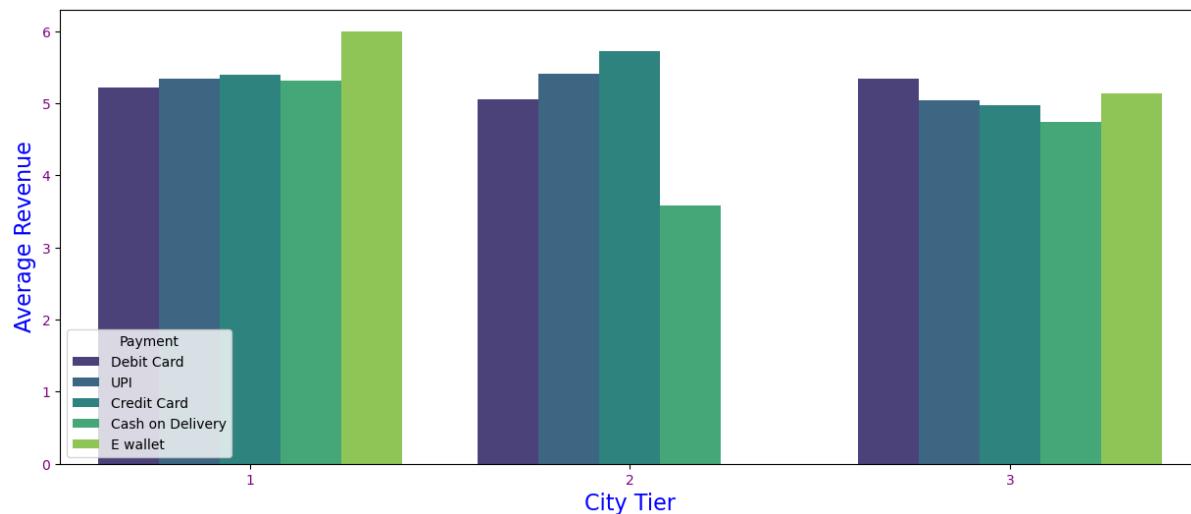


Fig no-19: City Tier Vs Payment mode

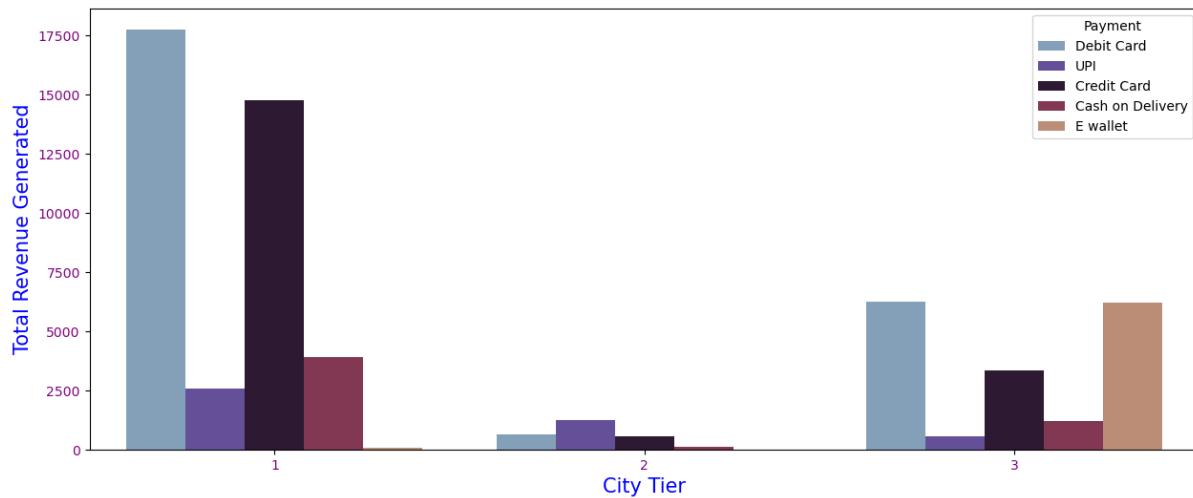


Fig no-20: City Tier Vs Payment mode Vs Total Revenue

Insights:

- As we can see, E wallet is used only by the Tier 1 & Tier 3 cities.
- Tier-2 Customers don't prefer using E-wallet as payment mode.
- Most of the customers prefer using E wallet and Debit Card in tier 3 cities.
- Almost 35% of the tier 3 city customers prefers E wallet as their payment mode, most revenue will be generated from such customers from tier 3 cities.
- Almost 46% of the tier 1 city customers prefers Debit Card as the payment mode, so most revenue will be generated from such customers from tier 1 cities.
- Almost 47% of the tier 2 city customers prefers UPI as the payment mode, so most revenue will be generated from such customers from tier 2 cities.

Segment on the Basis of Gender

❖ Bar plot for Male Vs Female segmentation:

- Male Count: 6812
- Female Count: 4448

❖ Visualization for Male Vs Female segmentation:

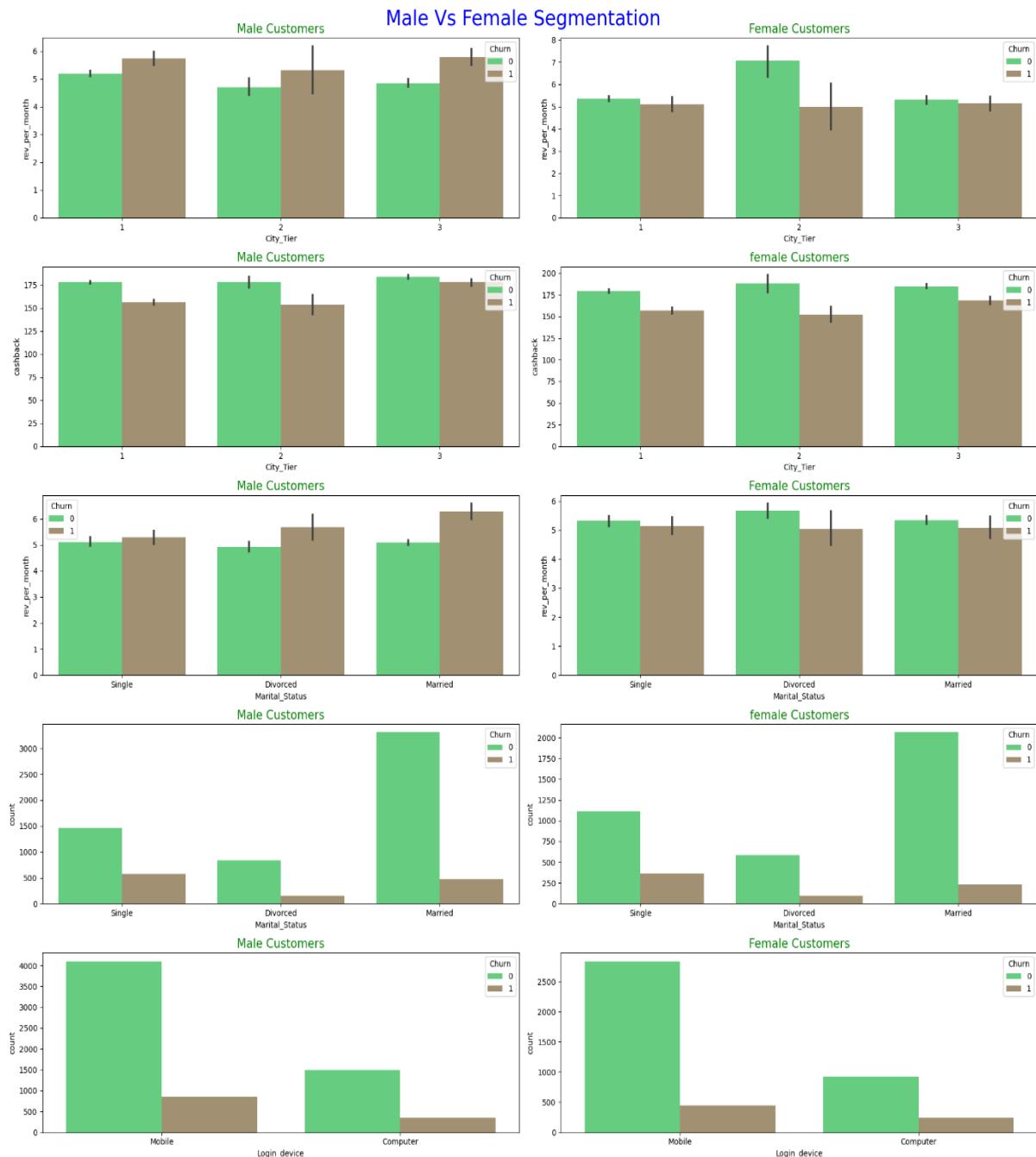


Fig no-21: Male Vs Female Segmentation

Insights:

- In Tier-1 & Tier-2, avg monthly revenue is more generated by Male customer than Females and in Tier-3 by Females.
- Cashback ratio is same for male & female in all Tier and across Marital status.
- Avg revenue generated by Singles Male and Females are almost same in all Tiers.
- Revenue generated by Divorced Females are slightly more than male.
- Count of Marital status is same across Genders.
- Mobile is the most preferred Login device among Male and Females.

❖ Revenue Generated on the basis of Spend (Male Customers):

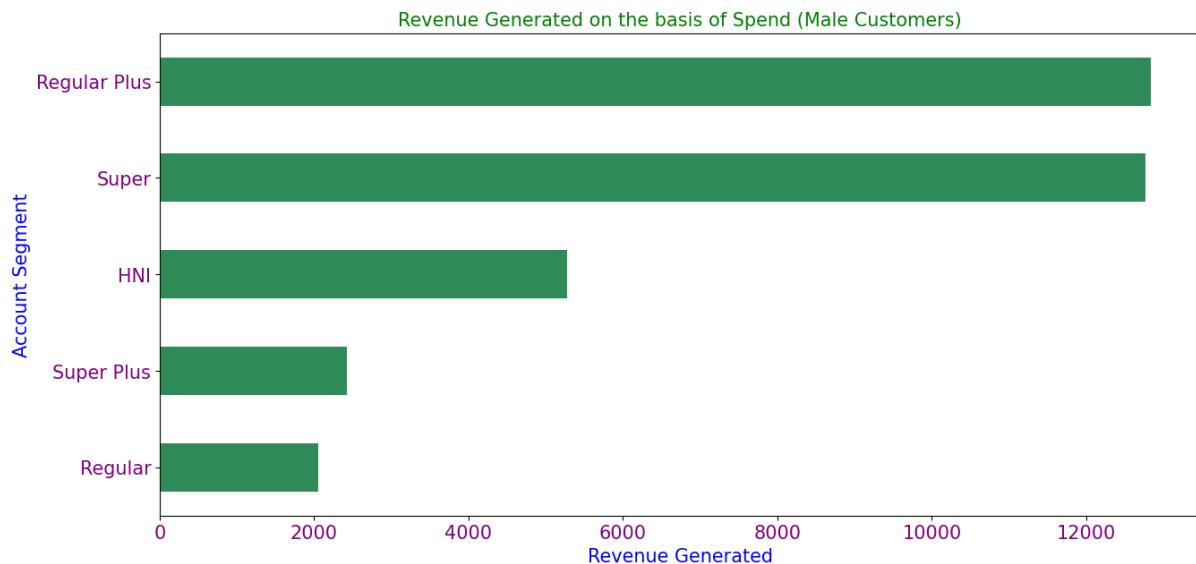


Fig no-22: Bar plot male customer

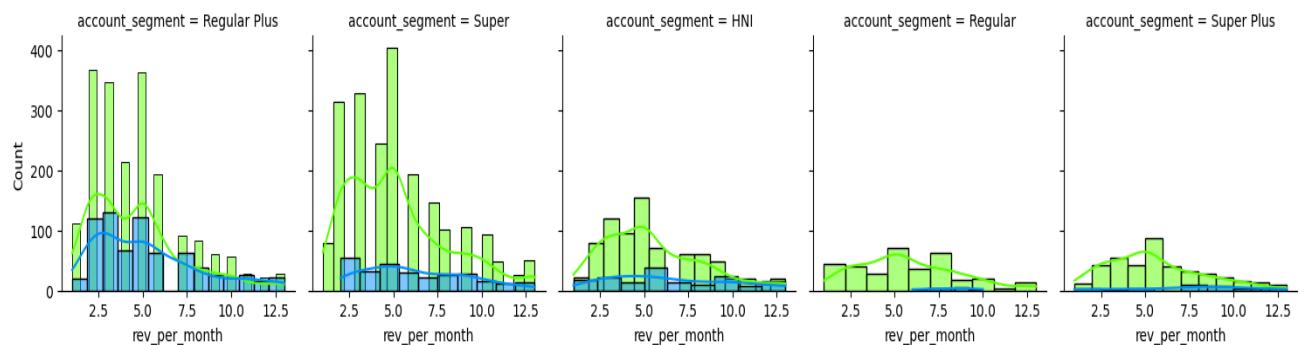


Fig no-23: Facet grid

Insights:

- Maximum Revenue is generated by Regular plus and Super account holder by male customers.
- ❖ Revenue Generated on the basis of Spend (Male Customers):

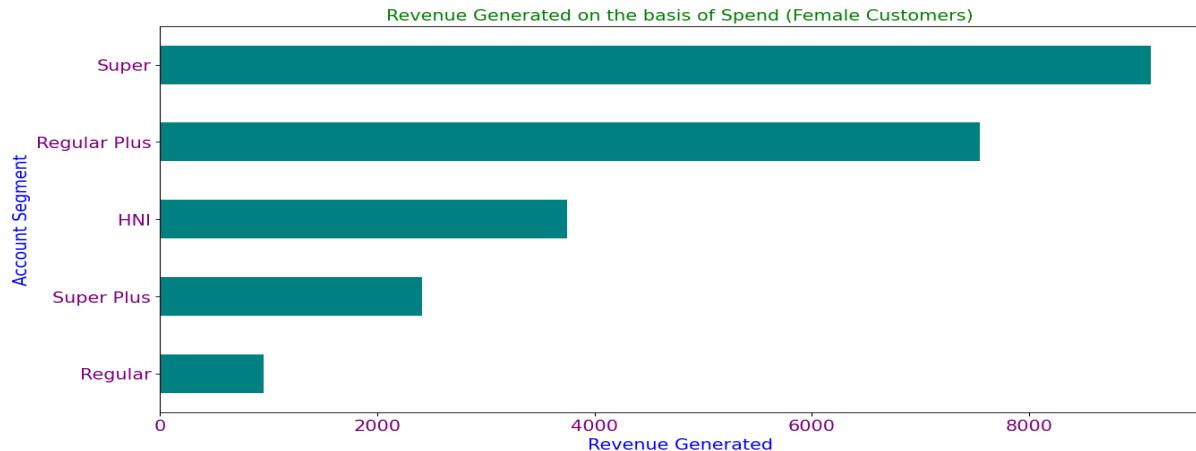


Fig no-24: Bar plot Female customer

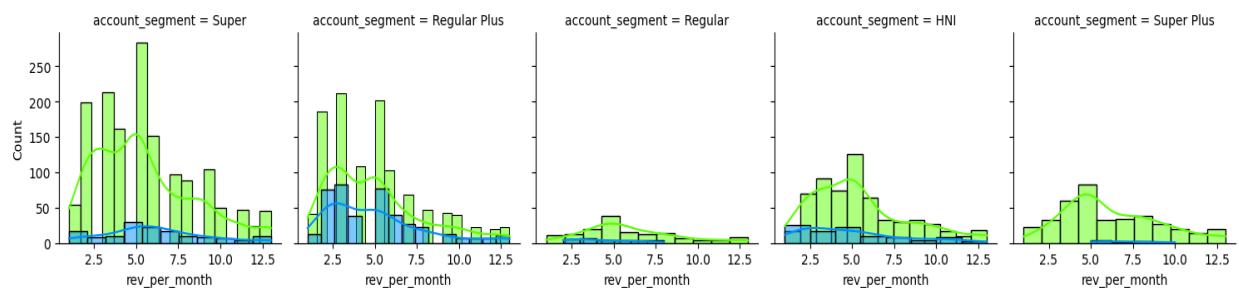


Fig no-25: Facet grid

Insights:

- Maximum revenue is generated by Super account holder followed by Regular Plus Female customers.
- ❖ Revenue Generated on the basis of Account_user_count (Male Customers):

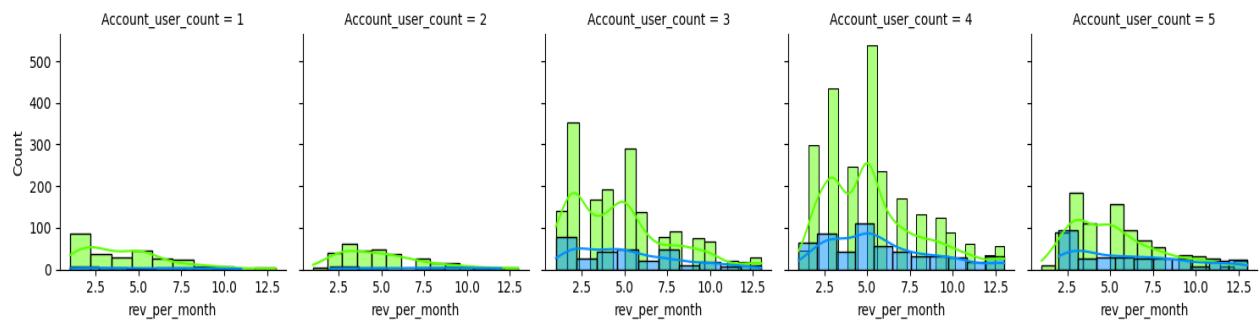


Fig no-26: Facet grid

Insights:

- Account tagged with more users generate more Revenue by Male customers.

❖ Revenue Generated on the basis of Account_user_count (Female Customers):

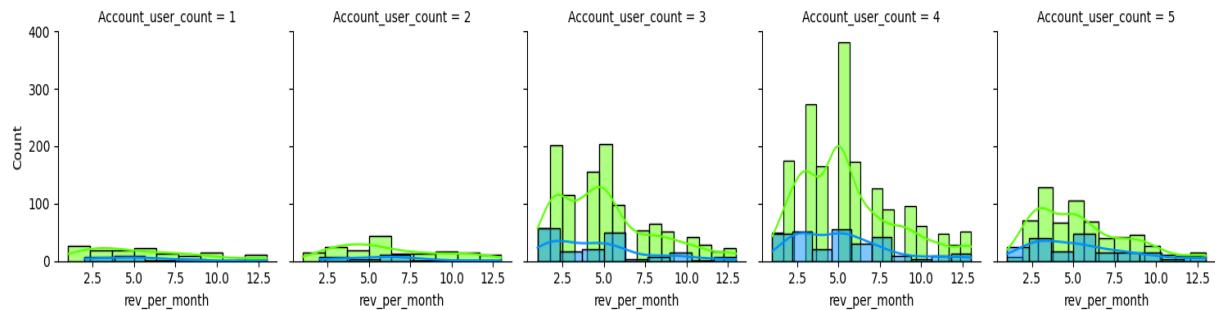


Fig no-27: Facet grid

Insights:

- Account tagged with more users generate more Revenue by Female customers.

Data Encoding

- Data Encoding is done using Label Encoder for Categorical Variable.

Feature Transformation

- Predictors are Scaled using Min Max Scalar

	Tenure	City_Tier	CC_Contacted_LY	Payment	Gender	Service_Score	Account_user_count	account_segment	CC_Agent_Score	Marital_Status
0	0.108108	1.0	0.054054	0.50	0.0	0.75	0.50	0.75	0.25	1.0
1	0.000000	0.0	0.108108	1.00	1.0	0.75	0.75	0.50	0.50	1.0
2	0.000000	0.0	0.702703	0.50	1.0	0.50	0.75	0.50	0.50	1.0
3	0.000000	1.0	0.297297	0.50	1.0	0.50	0.75	0.75	1.00	1.0
4	0.000000	0.0	0.216216	0.25	1.0	0.50	0.50	0.50	1.00	1.0

rev_per_month	Complain_ly	rev_growth_yoy	coupon_used_for_payment	Day_Since_CC_connect	cashback	Login_device
0.666667	1.0	0.291667		0.285714	0.344828	0.435907
0.500000	1.0	0.458333		0.000000	0.000000	0.238466
0.416667	1.0	0.416667		0.000000	0.206897	0.462819
0.583333	0.0	0.791667		0.000000	0.206897	0.305089
0.166667	0.0	0.291667		0.285714	0.206897	0.282477

Addition of new variables (if required).

- Addition of New variables is not required in this problem.

Clustering

- we will be performing clustering via KMeans clustering.
- Plotting Elbow plot (up to n=10) and identifying optimum number of clusters for k-means algorithm.

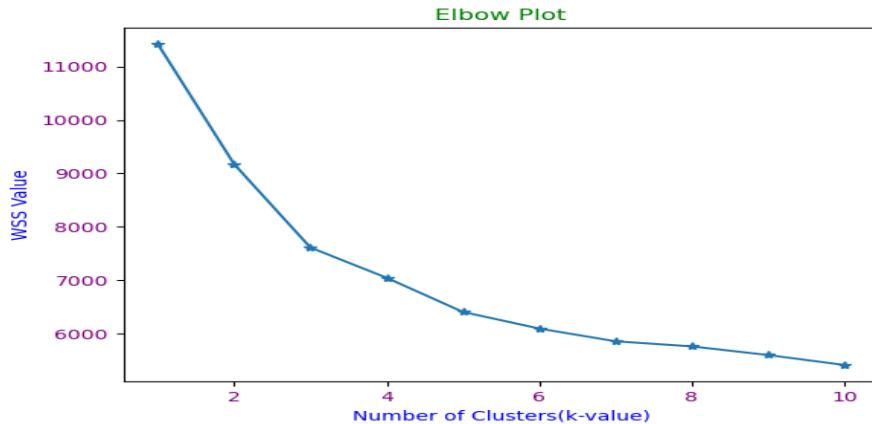


Fig no-28: Elbow plot

- From the above Elbow plot and WSS (within sum of squares) value we can choose optimum number of clusters as 3.
- Up to K=3 the drop in WSS value is significant.
- Beyond K=3, the drop is not significant.
- ❖ Silhouette scores for up to 10 clusters to identify optimum number of clusters.

The Silhouette Score for 2 clusters is 0.21014
 The Silhouette Score for 3 clusters is 0.23507
 The Silhouette Score for 4 clusters is 0.15923
 The Silhouette Score for 5 clusters is 0.18869
 The Silhouette Score for 6 clusters is 0.17312
 The Silhouette Score for 7 clusters is 0.15749
 The Silhouette Score for 8 clusters is 0.14826
 The Silhouette Score for 9 clusters is 0.14614
 The Silhouette Score for 10 clusters is 0.14788



Fig no-29: Silhouette plot

- Optimal number of clusters =3.
- ❖ Adding Clusters to the Churn Data frame for further Visualization:

Churn	Tenure	City_Tier	CC_Contacted_LY	Payment	Gender	Service_Score	Account_user_count	account_segment	CC_Agent_Score	Marital_Status
0	1	4.0	3	6.0	Debit Card	Female	3	3	Super	2
1	1	0.0	1	8.0	UPI	Male	3	4	Regular Plus	3
2	1	0.0	1	30.0	Debit Card	Male	2	4	Regular Plus	3
3	1	0.0	3	15.0	Debit Card	Male	2	4	Super	5
4	1	0.0	1	12.0	Credit Card	Male	2	3	Regular Plus	5

rev_per_month	Complain_ly	rev_growth_yoy	coupon_used_for_payment	Day_Since_CC_connect	cashback	Login_device	Clusters
9.0	1	11.0		1.0	5.0	159.93	Mobile
7.0	1	15.0		0.0	0.0	120.90	Mobile
6.0	1	14.0		0.0	3.0	165.25	Mobile
8.0	0	23.0		0.0	3.0	134.07	Mobile
3.0	0	11.0		1.0	3.0	129.60	Mobile

- ❖ Grouping data on the basis of Clusters and Login Device:

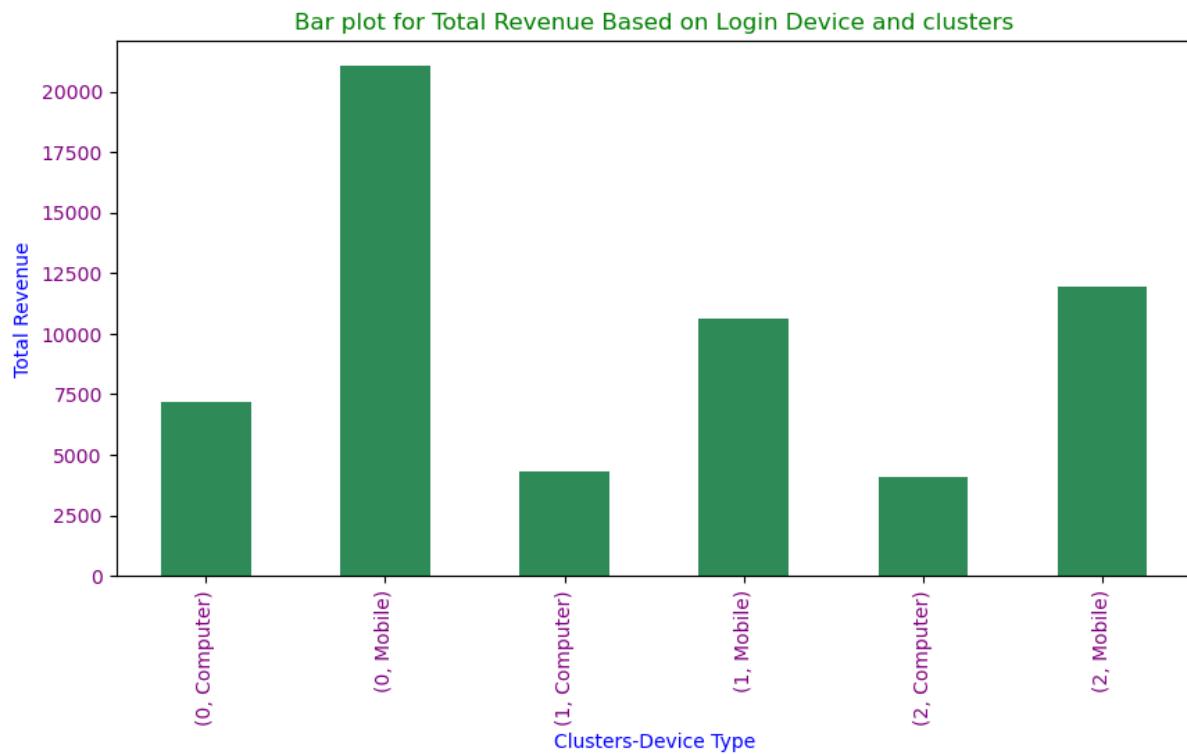


Fig no-30: Cluster-Login Device Vs Revenue

- Total Revenue generated is Maximum for Cluster 0 followed by cluster 2, through Login Device Mobile.
- Also, For Login Device Computer, Maximum Revenue is generated by Cluster 0.

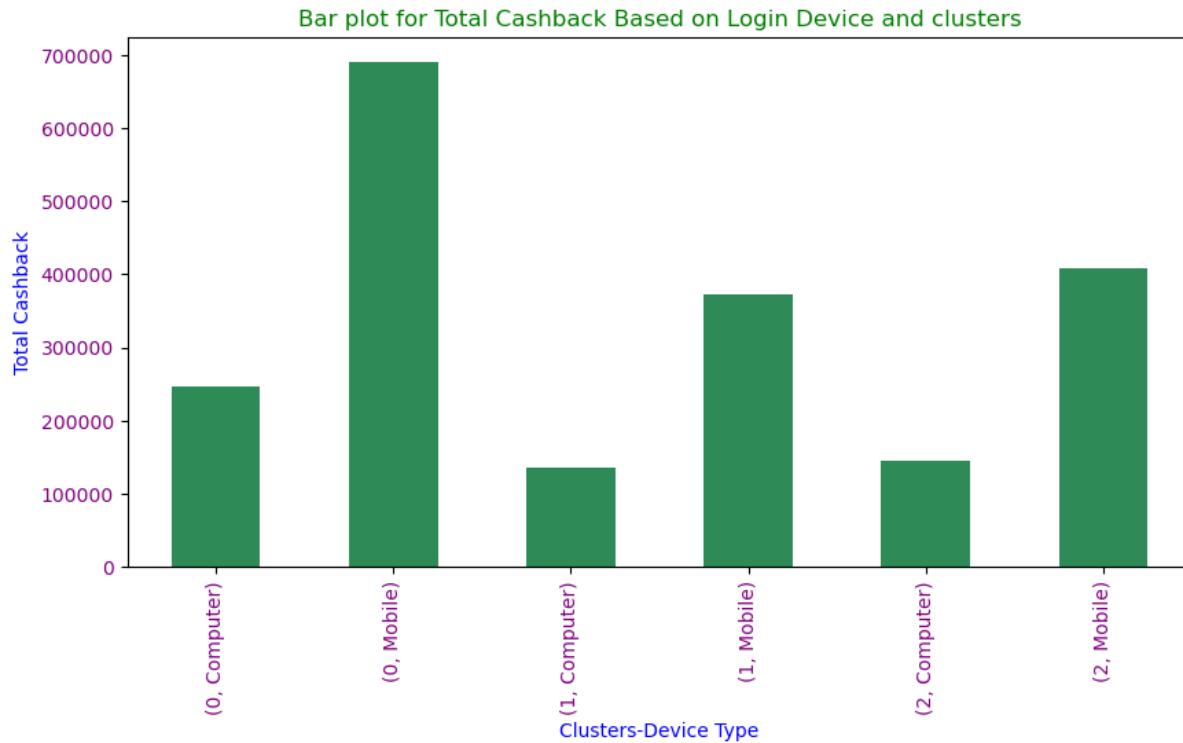


Fig no-31: Cluster-Login Device Vs Cashback

- Total Cashback is Maximum for Cluster 0 followed by cluster 2, through Login Device Mobile.
- Also, For Login Device Computer, Maximum Cashback is generated by Cluster 0.

Segmenting the Dataset based on Clusters

- ❖ Let's Segment the Data based on Clusters Formed

- Creating Three datasets based on Clusters formed for further Visualization.

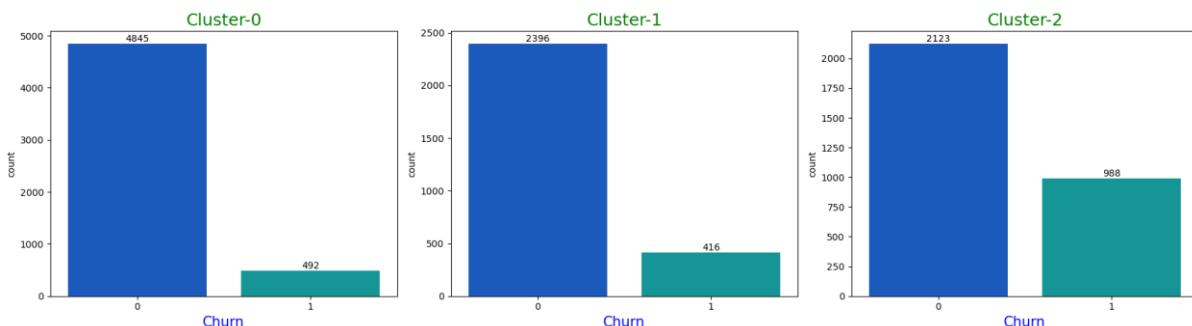


Fig no-32: Count plot Churn Vs Clusters

Insights:

- Churners are more in Cluster-2 as compared to other Clusters.**

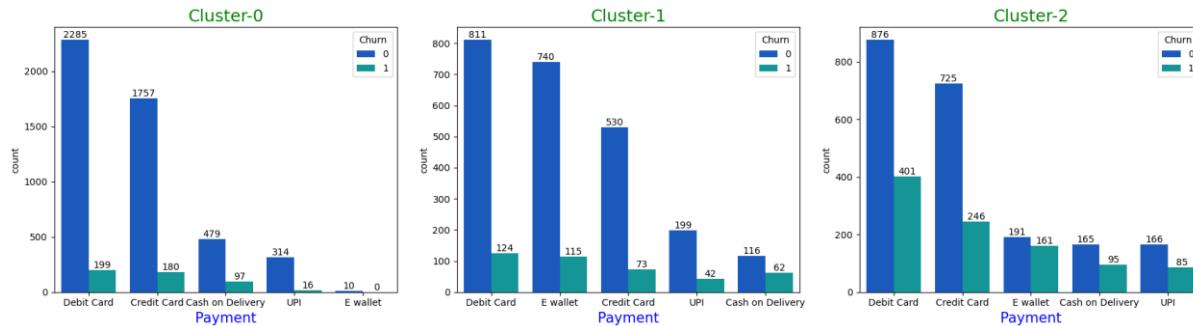


Fig no-33: Count plot Payment Vs Clusters

Insights:

- E-wallet payment mode is mostly used by Cluster-1 customers as compared to other clusters.**
- Debit Card payment mode is used by all three clusters customer.**
- E-wallet is Rarely used in cluster-0.**
- Maximum churners use Debit card as payment mode across clusters.**

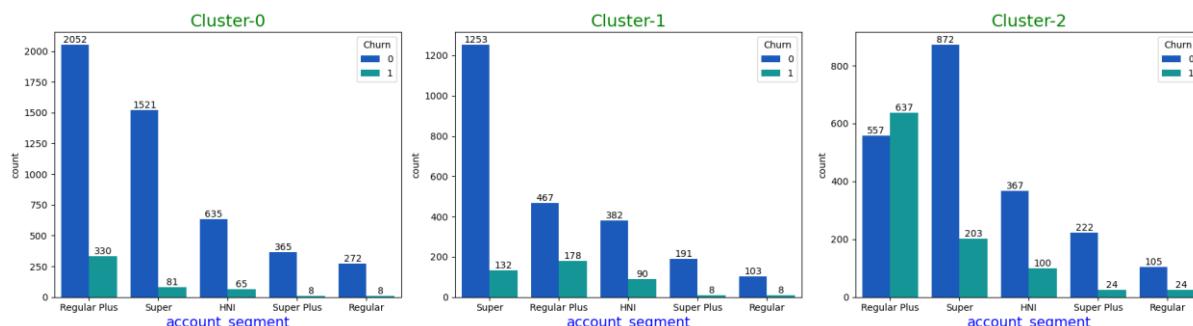


Fig no-34: Count plot account segment Vs Clusters

Insights:

- Super account holder is more in Cluster-1 followed by Cluster-2.**
- Maximum Regular Plus account holders belongs to Cluster-0 and Cluster-2.**
- In cluster-2 more churners belong to Regular plus account.**
- More Churners belong to Regular plus account.**

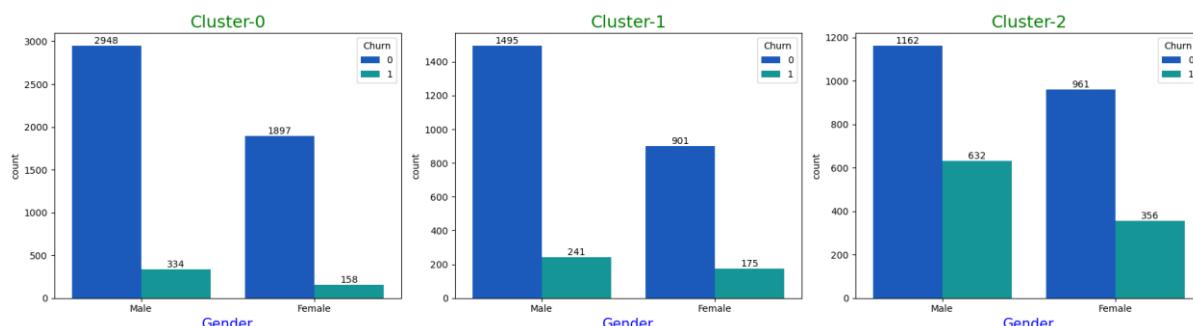


Fig no-35: Count plot Gender Vs Clusters

Insights:

- Females are slightly more in Cluster-2 as compared to other clusters.
- Gender Ratio is almost same in all the Clusters.
- Male Churners are more as compared to Females across clusters.

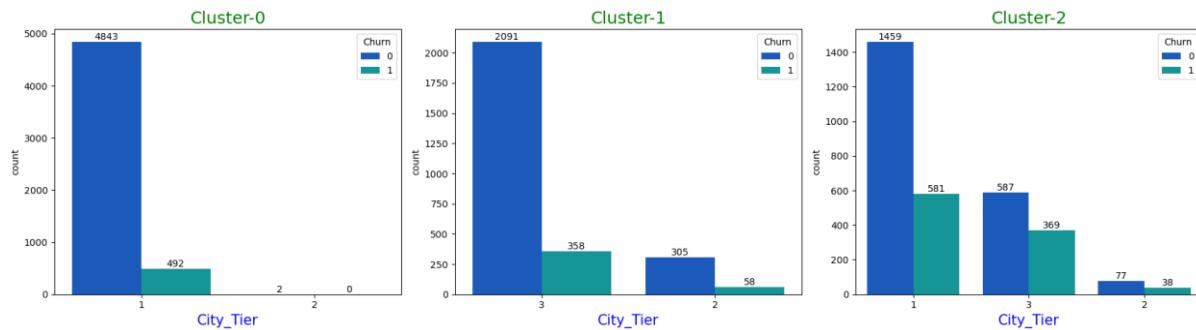


Fig no-36: Count plot City Tier Vs Clusters

Insights:

- Cluster-0 have customers belonging to mostly Tier-1 & very less from Tier-2. There are no Tier-3 customers in cluster-0.
- Cluster-1 have customers belonging to mostly Tier-3 & very less from Tier-2. There are no Tier-1 customers in cluster-1.
- In Cluster-2 maximum customers belong to Tier-1 followed by Tier-1 and then Tier-2.
- More churners belong to cluster-2.

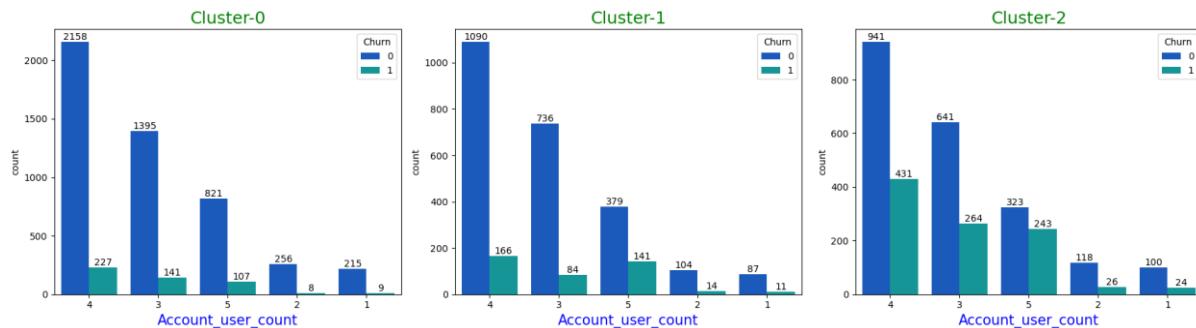


Fig no-37: Count plot Account user count Vs Clusters

Insights:

- Ratio of user tagged with single account is same across all the Clusters.
- More churners belong to account tagged with 3 or 4 users.

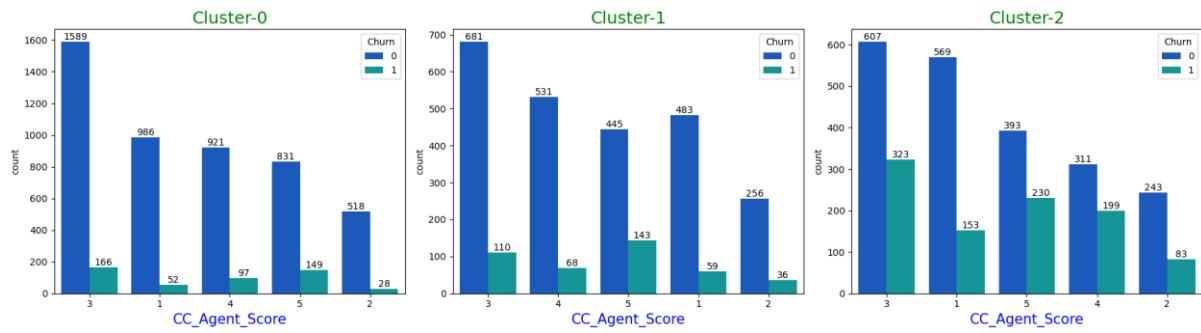


Fig no-38: Count plot CC Agent score Vs Clusters

Insights:

- In cluster-0,1 and 2, satisfaction score given by maximum customers is 3.

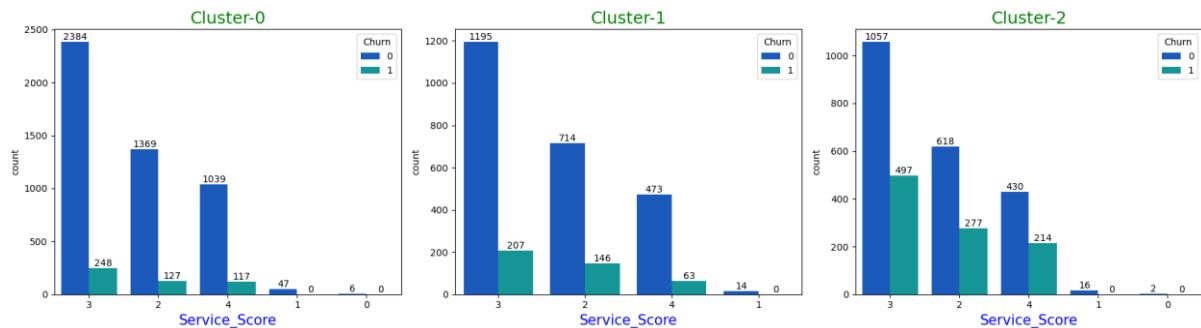


Fig no-39: Count plot Service score Vs Clusters

Insights:

- In cluster-1, no customers have given a score of 0.
- In all the Clusters maximum customers have given a score of 3,2 and 4 respectively.

❖ Tenure Vs Clusters:

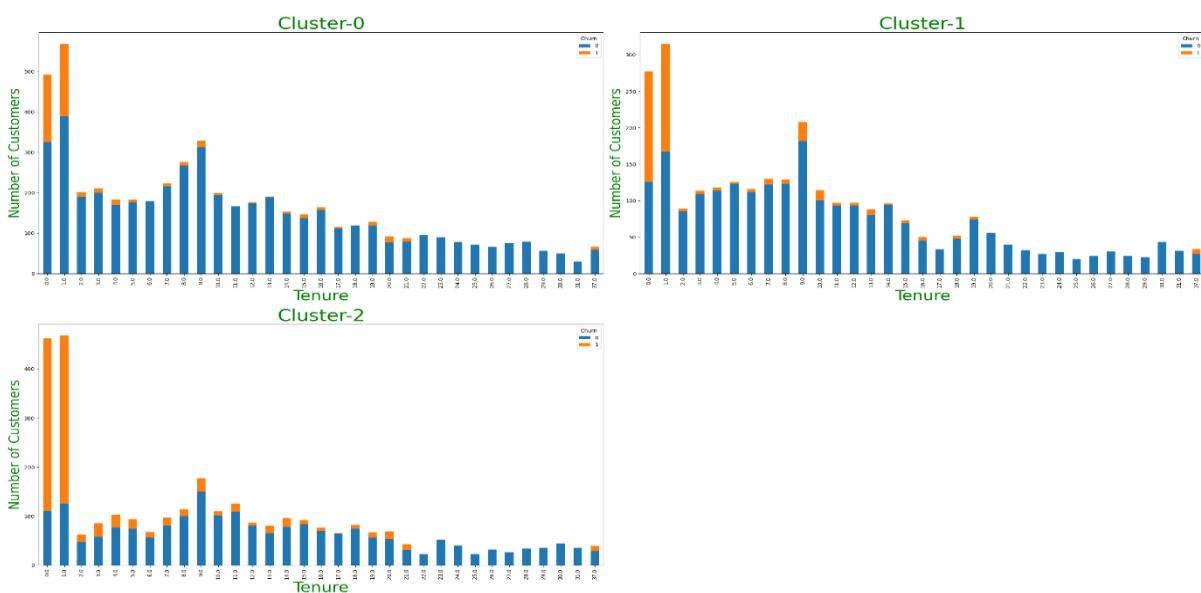


Fig no-40: Tenure Vs Clusters

Insights:

- It is observed that the newly joined customers having low Tenure Churns most, so more concern should be given to the newly joined customers and their cause of churning should be figured out.
- Most of the customers who Churns, customers with less than 2 tenures are classified as new customers.

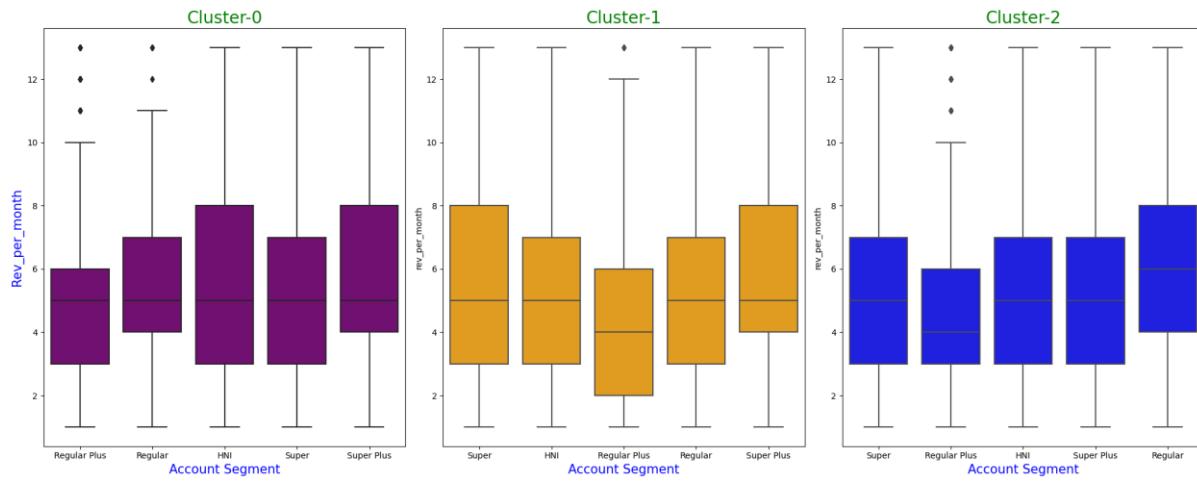


Fig no-41: Boxplot Account segment Vs Clusters

Insights:

- Maximum revenue is generated by HNI accounts in cluster 0.
- In cluster-1, Super, HNI and Super plus generate maximum revenue.
- In cluster-2, Regular accounts generate max Revenue.

Feature Selection

We will use Variance Inflation Factor to Select the Important features using the VIF value.

- We will consider features having VIF < 5
- Features having VIF > 5 are removed from the datasets.
- We removed 15 Features Considering VIF > 5.

	Features	VIF
5	Service_Score	17.592656
12	rev_growth_yoy	10.103968
6	Account_user_count	9.759654
15	cashback	9.384768
7	account_segment	4.447725
3	Payment	4.134365
9	Marital_Status	3.825733
2	CC_Contacted_LY	3.581888
16	Login_device	3.554259
13	coupon_used_for_payment	3.523403
10	rev_per_month	3.465482
14	Day_Since_CC_connect	3.383455
8	CC_Agent_Score	3.135966
0	Tenure	2.991562
4	Gender	2.435963
1	City_Tier	1.630030
11	Complain_ly	1.379092

- We are left with 13 Features now and will Build the Models with these Features.

	Features	VIF
5	account_segment	4.132428
3	Payment	3.925740
7	Marital_Status	3.477064
2	CC_Contacted_LY	3.339777
12	Login_device	3.338953
8	rev_per_month	3.232364
10	coupon_used_for_payment	3.118444
11	Day_Since_CC_connect	3.080066
6	CC_Agent_Score	2.996977
0	Tenure	2.495054
4	Gender	2.335222
1	City_Tier	1.610286
9	Complain_ly	1.362515

Business insights from EDA

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

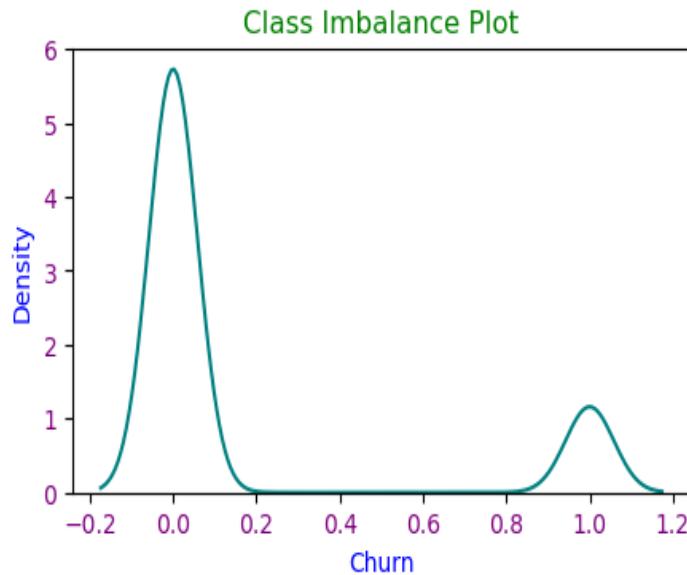


Fig no-42: Class imbalanced plot

- Here the distribution of Target variable is Skewed.
- In the Target feature 'Churn' there is a significant data imbalance, with a Majority class of 83.16% and a Minority class of 16.83%.
- To handle this class imbalance, an Up-sampling process is carried out where synthetically generated data points corresponding to minority class are injected into the datasets.
- We can use SMOTE technique for balancing the Data.

4.b) Any business insights using clustering (if applicable).

- Maximum Customers Prefers Mobile as their Login Device.
- Since Churners are more in Cluster-2, Providing special offers and discounts to cluster-2 customers can decrease the churn rate.
- Some offers should be given for making payment through UPI payment mode to increase its use cluster-0 & Cluster-2.
- E-Wallet payment mode should be provided with offers in cluster-0.
- Maximum churners use Debit card as payment mode across clusters.
- More Churners belong to Regular plus account across clusters.
- Some promotional offers should be given to these customers so that they don't Churn.

- It is observed that the newly joined customers having low Tenure Churns most, so more concern should be given to the newly joined customers and their cause of churning should be figured out.

4.c) Any other business insights.

- Introducing Personalized offers or Promotions can decrease Churn rate.
- Tier-1 customers have high churn rate, suggesting some incentives or discounted offer can help in Retaining customers.
- UPI and E-wallet payment mode should be encouraged to increase customers.
- Tenure period should be Maximized to Retain the customers.
- Customer care services should be improvised so that the customers complaint can be resolved quickly and efficiently.
- Customized services should be offered to different account segment customers to retain customers.
- Launch tailoring offers to multi user account holders to retain customers.

Thank You

Capstone Project Notes – 2

Model building and interpretation

- a. Build various models (You can choose to build models for either or all of descriptive, predictive or prescriptive purposes)
- b. Test your predictive model against the test set using various appropriate performance metrics
- c. Interpretation of the model(s)

Model Tuning and business implication

- a. Ensemble modelling, wherever applicable
- b. Any other model tuning measures (if applicable)
- c. Interpretation of the most optimum model and its implication on the business

Logistic Regression Model

```
LogisticRegression(max_iter=10000, n_jobs=-1, penalty=None, random_state=42,
                   solver='newton-cg', verbose=True)
```

- ❖ Accuracy of Training and Test Data:
 - Accuracy of the Training Data: 0.8822633849276833
 - Accuracy of the Test Data: 0.8875074008288928

- ❖ validness of models (over fitting or under fitting):
 - The Accuracy of both the Train data and Test data are almost same; hence our model is Neither Overfit nor Underfit.

- ❖ Classification Report:

Classification Report of the Training data:				
	precision	recall	f1-score	support
0	0.90	0.97	0.93	6555
1	0.75	0.45	0.56	1327
			accuracy	7882
			macro avg	0.82
			weighted avg	0.87
Classification Report of the Test data:				
	precision	recall	f1-score	support
0	0.90	0.97	0.93	2809
1	0.78	0.47	0.58	569
			accuracy	3378
			macro avg	0.84
			weighted avg	0.88

- ❖ Interpretations:

-----Test set-----

- Accuracy:
 - The Accuracy of model is 89%.
 - 89% of the Data points are identified correctly
- Recall: Represents the correctness of the Model in a particular class.
 - Recall for class 1(Churn) is 47% and recall for class 0(non-churn) is 97%.
 - 47% of those Churned were correctly identified as Churned by the model.
 - Out of all the customers who are Churned, 47% of them are identified correctly as Churned.

- Out of all the customers who are not Churned, 97% of them are identified correctly as Not-Churned.

➤ **Precision:**

- Precision for class 1(Churn) is 78% and for class 0(non-Churn) is 90%.
- 78% of the customers predicted are actually Churned Out of all the Customers that are predicted as Churned.
- 90% of the Customers predicted are actually Not Churned out of all the customers predicted as not Churned.

❖ **Confusion Matrix:**

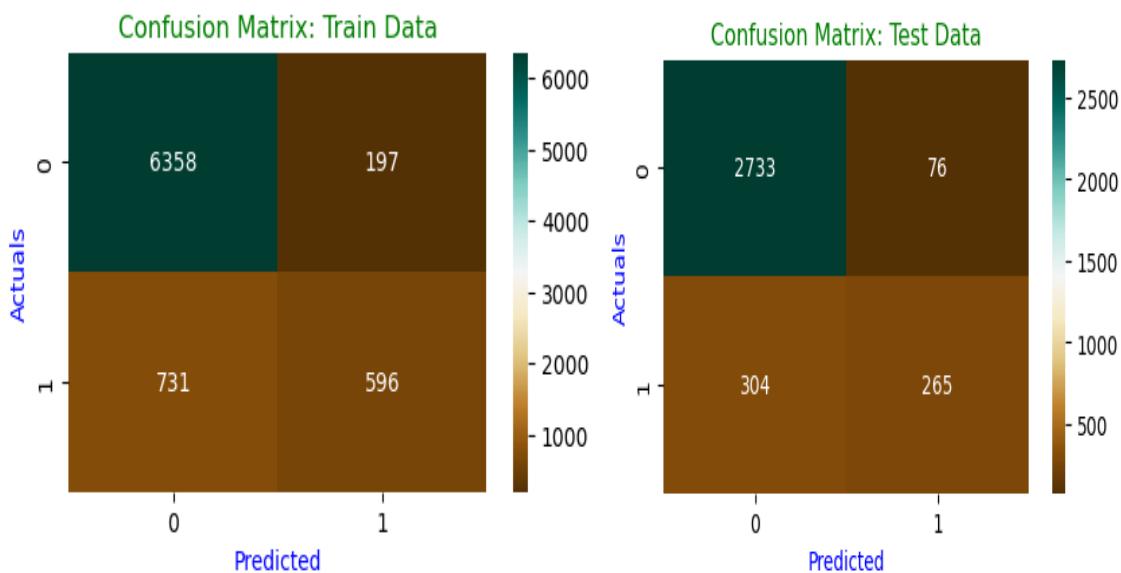


Fig no-43: Confusion Matrix LR

❖ **AUC and ROC Curve:**

- AUC Train: 0.870
- AUC Test: 0.866

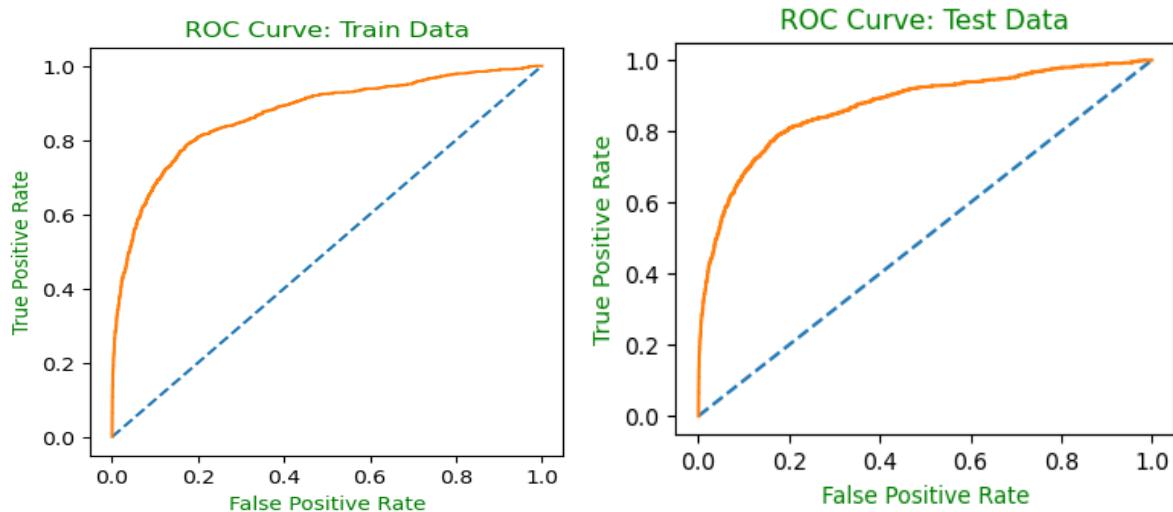
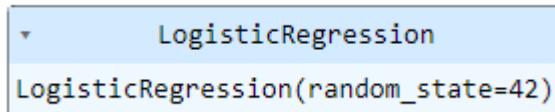


Fig no-44: ROC Curve LR

❖ Observations:

- AUC for Train data is 87% and AUC for Test data is 86%.
- The AUC & ROC curve shows that it is covering almost the same area for Train and Test data. Therefore, this can be considered a good optimized model.

Building Logistic regression model on Balanced dataset



❖ Accuracy of Training and Test Data:

- Accuracy of the Training Data: 0.8136169470839508
- Accuracy of the Test Data: 0.8155713439905269

❖ Classification Report:

Classification Report of the Training data:

	precision	recall	f1-score	support
0	0.84	0.83	0.84	6555
1	0.78	0.79	0.78	4916
accuracy			0.81	11471
macro avg	0.81	0.81	0.81	11471
weighted avg	0.81	0.81	0.81	11471

Classification Report of the Test data:

	precision	recall	f1-score	support
0	0.95	0.82	0.88	2809
1	0.47	0.77	0.58	569
accuracy			0.82	3378
macro avg	0.71	0.80	0.73	3378
weighted avg	0.87	0.82	0.83	3378

❖ Interpretations:

-----Test set Smote Data-----

➤ Accuracy:

- The Accuracy of model is 82%.
- 82% of the Data points are identified correctly

➤ Recall: Represents the correctness of the Model in a particular class.

- Recall for class 1(Churn) is 77% and recall for class 0(non-churn) is 82%.
- 77% of those Churned were correctly identified as Churned by the model.
- Out of all the customers who are Churned, 77% of them are identified correctly as Churned.
- Out of all the customers who are not Churned, 82% of them are identified correctly as Not-Churned.

➤ Precision:

- Precision for class 1(Churn) is 47% and for class 0(non-churn) is 95%.
- 47% of the customers predicted are actually Churned Out of all the Customers that are predicted as Churned.
- 95% of the Customers predicted are actually Not Churned out of all the customers predicted as not Churned.

❖ Confusion Matrix:

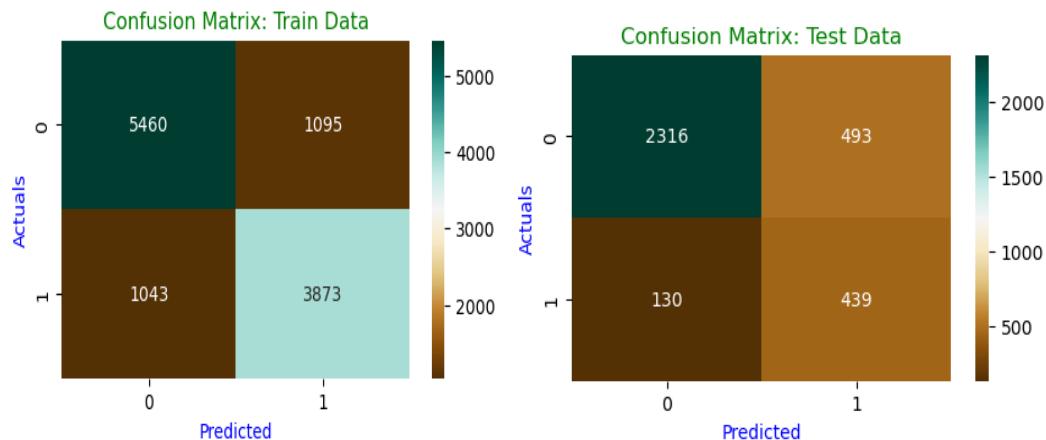


Fig no-45: Confusion Matrix LR Smote

❖ AUC-ROC Curve:

- AUC for the Training Data: 0.877
- AUC for the Test Data: 0.865

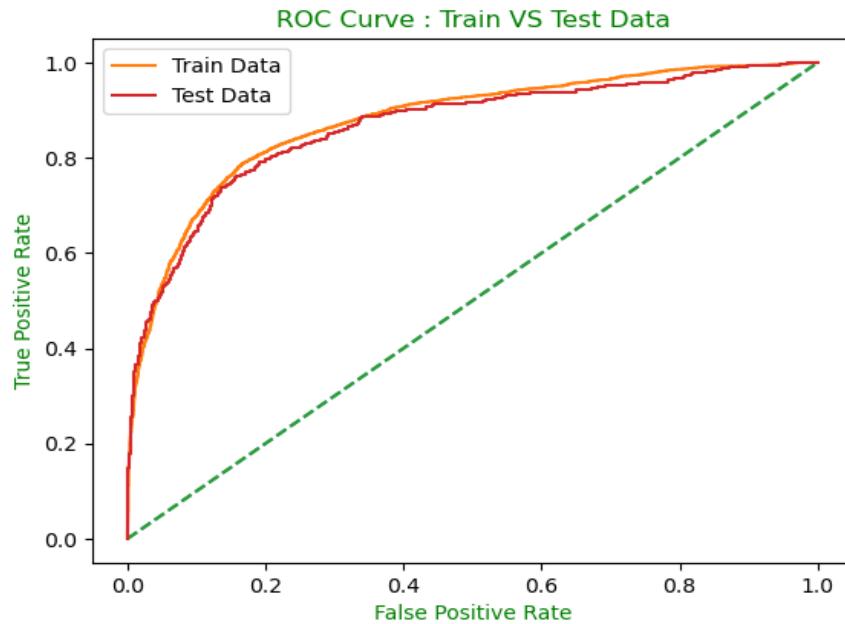


Fig no-46: ROC Curve LR Smote

❖ Observations:

- AUC for Train data is 87% and AUC for Test data is 86%.
- The AUC & ROC curve shows that is covering almost the same area for Train and Test data. Therefore, this can be considered a good Generalized model.

Hyperparameter Tuning

Applying GridSearchCV for Logistic Regression

```

GridSearchCV(cv=5,
             estimator=LogisticRegression(max_iter=10000, n_jobs=-1,
                                           penalty=None, random_state=42,
                                           solver='newton-cg', verbose=True),
             n_jobs=-1,
             param_grid={'penalty': ['l2'],
                         'solver': ['newton-cg', 'liblinear', 'lbfgs', 'sag'],
                         'tol': [0.0001, 1e-05]},
             scoring='f1', verbose=1)
             estimator: LogisticRegression
LogisticRegression(max_iter=10000, n_jobs=-1, penalty=None, random_state=42,
                    solver='newton-cg', verbose=True)
    > LogisticRegression

```

Building model on best parameters

```

LogisticRegression
LogisticRegression(max_iter=10000, n_jobs=-1, random_state=42,
                   solver='newton-cg', verbose=True)

```

- ❖ Accuracy of Train and Test data:
 - Accuracy on Train Data: 0.8818827708703375
 - Accuracy on Test Data: 0.8895796329188869
- ❖ validness of models (over fitting or under fitting):
 - The Accuracy of both the Train data and Test data are almost same; hence our model is Neither Overfit nor Underfit.
- ❖ Classification Report:

Classification Report of the Training data:				
	precision	recall	f1-score	support
0	0.89	0.97	0.93	6555
1	0.76	0.44	0.55	1327
accuracy			0.88	7882
macro avg	0.83	0.70	0.74	7882
weighted avg	0.87	0.88	0.87	7882

Classification Report of the Test data:				
	precision	recall	f1-score	support
0	0.90	0.98	0.94	2809
1	0.80	0.46	0.58	569
accuracy			0.89	3378
macro avg	0.85	0.72	0.76	3378
weighted avg	0.88	0.89	0.88	3378

❖ **Interpretations:**

-----Test set Tuned Model-----

➤ **Accuracy:**

- The Accuracy of model is 89%.
 - 89% of the Data points are identified correctly
- **Recall:** Represents the correctness of the Model in a particular class.
- Recall for class 1(Churn) is 46% and recall for class 0(non-churn) is 98%.
 - 46% of those Churned were correctly identified as Churned by the model.
 - Out of all the customers who are Churned, 46% of them are identified correctly as Churned.
 - Out of all the customers who are not Churned, 98% of them are identified correctly as Not-Churned.

➤ **Precision:**

- Precision for class 1(Churn) is 80% and for class 0(non-churn) is 90%.
- 80% of the customers predicted are actually Churned Out of all the Customers that are predicted as Churned.
- 90% of the Customers predicted are actually Not Churned out of all the customers predicted as not Churned.

❖ **Confusion Matrix:**

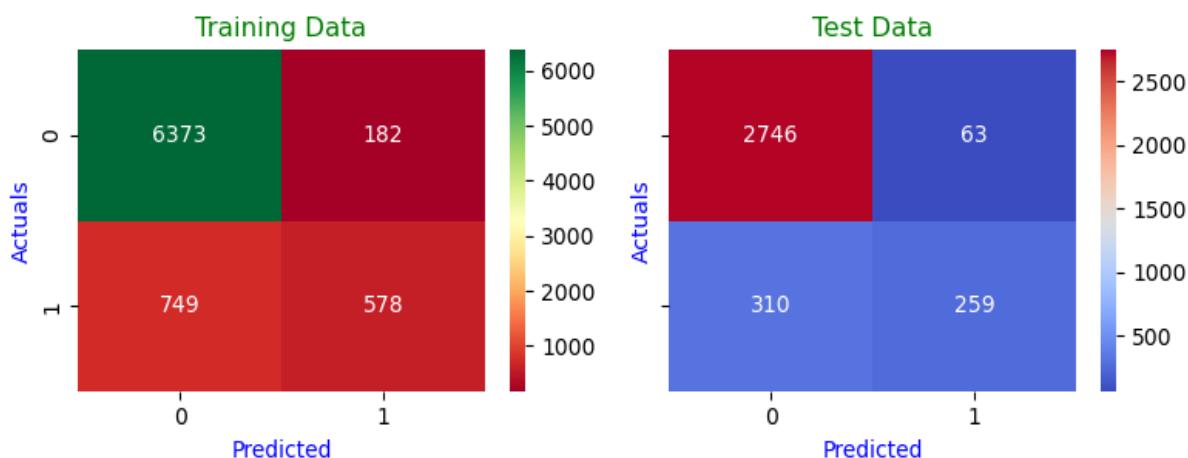


Fig no-47: Confusion Matrix LR Smote

❖ **Measuring and Plotting AUC-ROC Curve:**

- AUC for the Training Data: 0.869
- AUC for the Test Data: 0.866

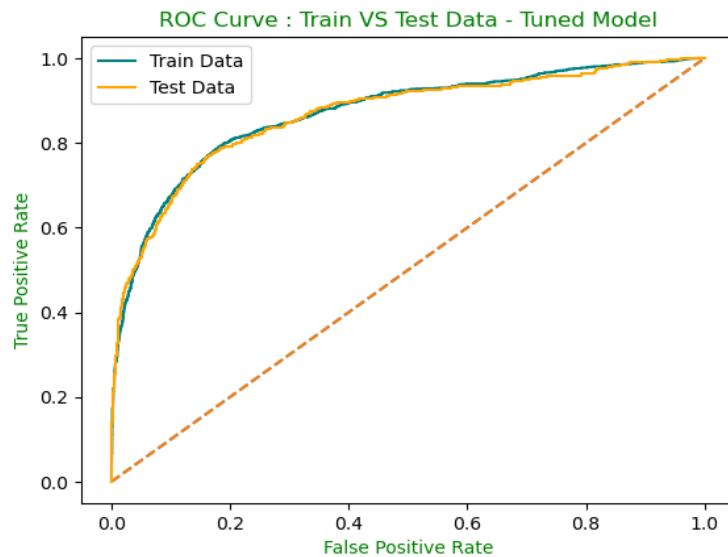


Fig no-48: ROC Curve LR Smote

❖ **Observations:**

- AUC for Train data is 86% and AUC for Test data is 86%.
- The AUC & ROC curve shows that is covering almost the same area for Train and Test data. Therefore, this can be considered a good optimized model.

❖ **Insights:**

- Accuracy is almost same for Base and Tuned Model for Logistic Regression.
- Accuracy on Smote Dataset is less than Base and Tuned Model.

Linear Discriminant Analysis

```
▼ LinearDiscriminantAnalysis
  LinearDiscriminantAnalysis()
```

❖ **Accuracy of Train and Test data:**

- Accuracy of the Training Data: 0.8763004313625983
- Accuracy of the Test Data: 0.8804026050917703

❖ **Validness of models (over fitting or under fitting):**

- The Accuracy of both the Train data and Test data are almost same; hence our model is Neither Overfit nor Underfit.

❖ **Classification Report:**

Classification Report of the Training data:

	precision	recall	f1-score	support
0	0.89	0.97	0.93	6555
1	0.75	0.40	0.52	1327
accuracy			0.88	7882
macro avg	0.82	0.69	0.73	7882
weighted avg	0.87	0.88	0.86	7882

Classification Report of the Test data:

	precision	recall	f1-score	support
0	0.89	0.98	0.93	2809
1	0.78	0.40	0.53	569
accuracy			0.88	3378
macro avg	0.84	0.69	0.73	3378
weighted avg	0.87	0.88	0.86	3378

❖ Interpretations:

Test set

➤ Accuracy:

- The Accuracy of model is 88%.
- 88% of the Data points are identified correctly

➤ Recall: Represents the correctness of the Model in a particular class.

- Recall for class 1(Churn) is 40% and recall for class 0(non-churn) is 98%.
- 40% of those Churned were correctly identified as Churned by the model.
- Out of all the customers who are Churned,40% of them are identified correctly as Churned.
- Out of all the customers who are not Churned, 98% of them are identified correctly as Not-Churned.

➤ Precision:

- Precision for class 1(Churn) is 78% and for class 0(non-churn) is 89%.
- 78% of the customers predicted are actually Churned Out of all the Customers that are predicted as Churned.
- 89% of the Customers predicted are actually Not Churned out of all the customers predicted as not Churned.

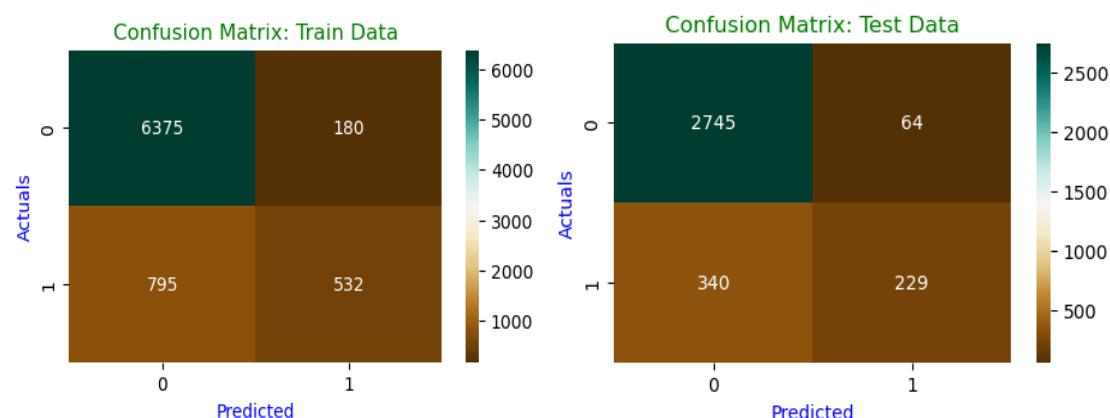
❖ Confusion Matrix:


Fig no 49: Confusion Matrix LDA

❖ **AUC - ROC Curve:**

- **AUC Train: 0.858**
- **AUC Test: 0.857**

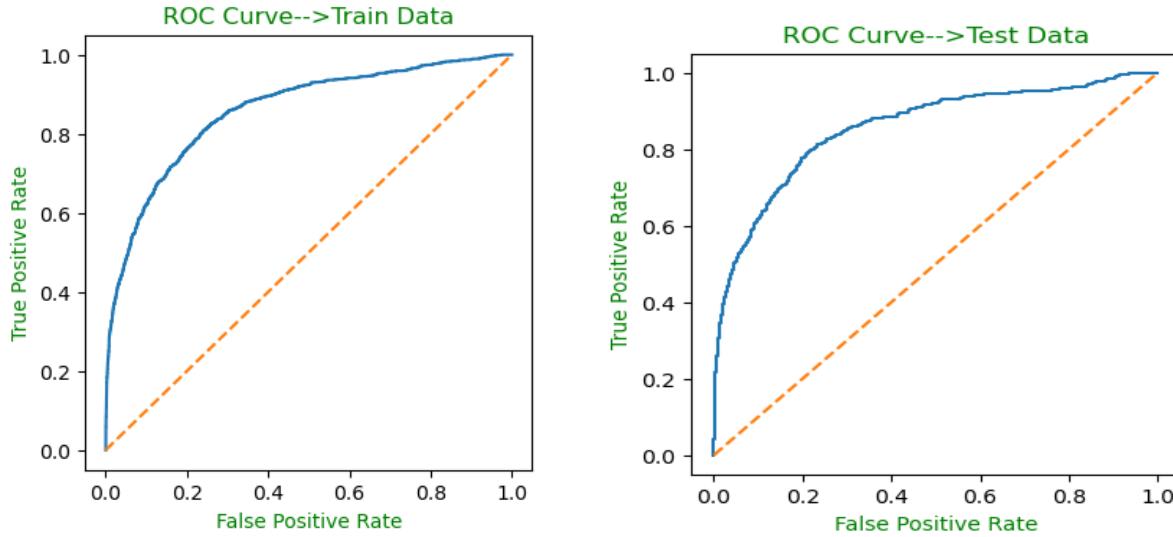
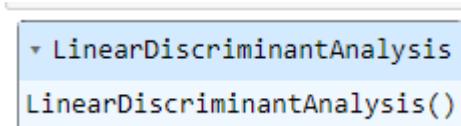


Fig no 50: ROC Curve LDA

❖ **Observations:**

- AUC for Train data is 85% and AUC for Test data is 85%.
- The AUC & ROC curve shows that is covering almost the same area for Train and Test data. Therefore, this can be considered a good optimized model.

Building LDA model on balanced dataset using SMOTE



❖ **Accuracy of Train and Test data:**

- Accuracy of the Training Data: 0.800540493418185
- Accuracy of the Test Data: 0.7960331557134399

❖ **Classification Report:**

Classification Report of the Training data:				
	precision	recall	f1-score	support
0	0.83	0.81	0.82	6555
1	0.76	0.78	0.77	4916
accuracy			0.80	11471
macro avg	0.80	0.80	0.80	11471
weighted avg	0.80	0.80	0.80	11471

Classification Report of the Test data:				
	precision	recall	f1-score	support
0	0.95	0.80	0.87	2809
1	0.44	0.78	0.56	569
accuracy			0.80	3378
macro avg	0.69	0.79	0.71	3378
weighted avg	0.86	0.80	0.82	3378

❖ Interpretations:

-----Test set Smote Data-----

- Accuracy:
 - The Accuracy of model is 80%.
 - 80% of the Data points are identified correctly
- Recall: Represents the correctness of the Model in a particular class.
 - Recall for class 1(Churn) is 78% and recall for class 0(non-churn) is 80%.
 - 78% of those Churned were correctly identified as Churned by the model.
 - Out of all the customers who are Churned,78% of them are identified correctly as Churned.
 - Out of all the customers who are not Churned, 80% of them are identified correctly as Not-Churned.
- Precision:
 - Precision for class 1(Churn) is 44% and for class 0(non-Churn) is 95%.
 - 44% of the customers predicted are actually Churned Out of all the Customers that are predicted as Churned.
 - 95% of the Customers predicted are actually Not Churned out of all the customers predicted as not Churned.

❖ Confusion Matrix:

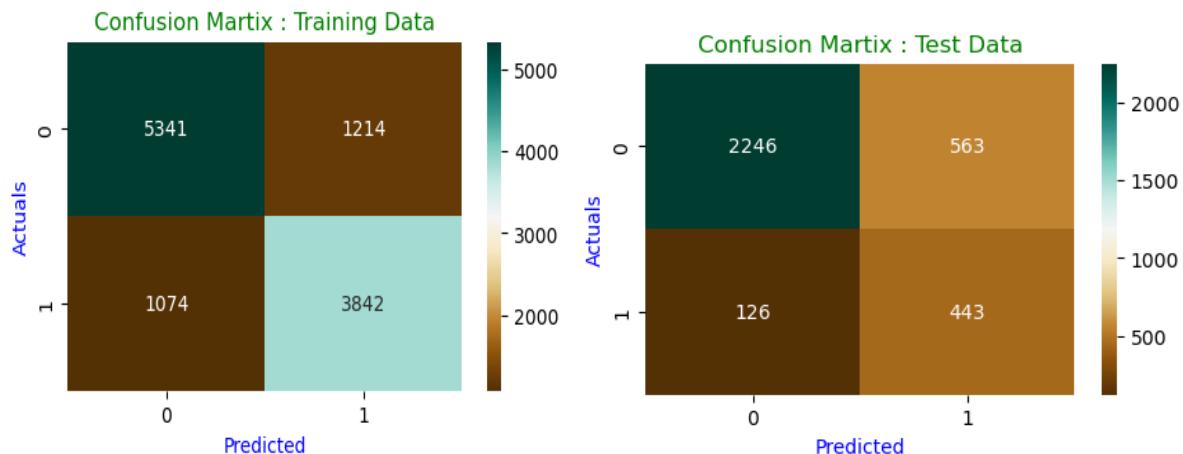


Fig no 51: Confusion Matrix LDA Smote

❖ AUC - ROC Curve:

- AUC Train: 0.873
- AUC Test: 0.863

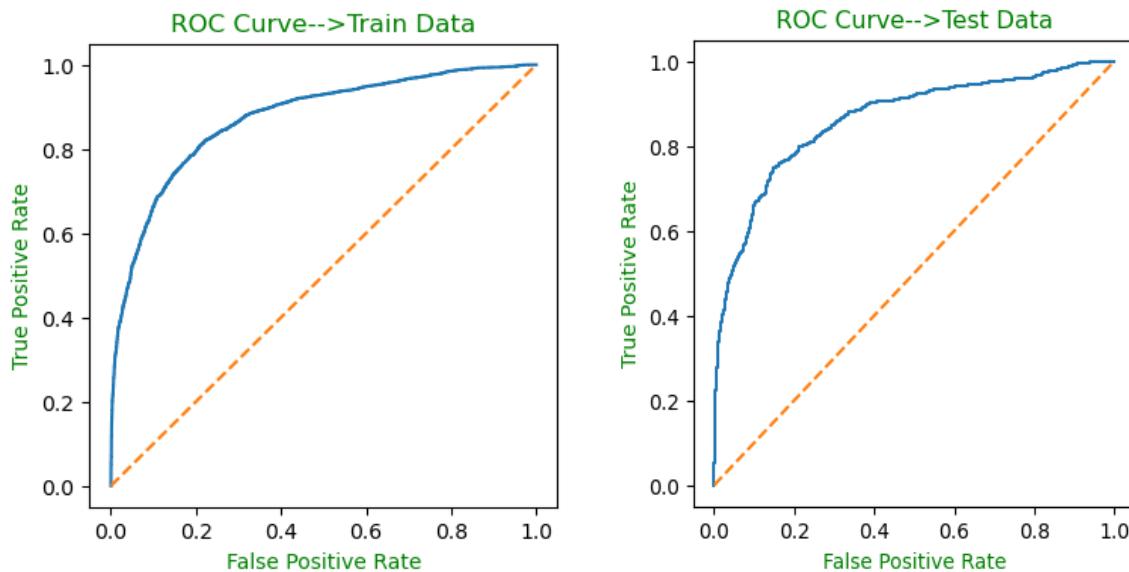


Fig no 52: ROC Curve LDA Smote

❖ Observations:

- AUC for Train data is 87% and AUC for Test data is 86%.
- The AUC & ROC curve shows that is covering almost the same area for Train and Test data. Therefore, this can be considered a good optimized model.

Building LDA model using GridSearchCV

```
GridSearchCV
GridSearchCV(cv=10, estimator=LinearDiscriminantAnalysis(), n_jobs=-1,
            param_grid={'shrinkage': ['auto', 'None'],
                        'solver': ['lsqr', 'eigen'],
                        'tol': [0.0001, 0.0002, 0.0003]},
            scoring='f1')
    -> estimator: LinearDiscriminantAnalysis
        -> LinearDiscriminantAnalysis
```

Building model on best parameters

```
LinearDiscriminantAnalysis
LinearDiscriminantAnalysis(shrinkage='auto', solver='lsqr')
```

❖ Accuracy of Train and Test Data:

- Accuracy on Train Data: 0.8768079167723928
- Accuracy on Test Data: 0.8804026050917703

- ❖ **validness of models (over fitting or under fitting):**
 - The Accuracy of both the Train data and Test data are almost same; hence our model is Neither Overfit nor Underfit.

❖ Classification Report:

Classification Report of the Training data:				
	precision	recall	f1-score	support
0	0.89	0.97	0.93	6555
1	0.75	0.40	0.52	1327
accuracy			0.88	7882
macro avg	0.82	0.69	0.73	7882
weighted avg	0.87	0.88	0.86	7882

Classification Report of the Test data:				
	precision	recall	f1-score	support
0	0.89	0.98	0.93	2809
1	0.78	0.40	0.53	569
accuracy			0.88	3378
macro avg	0.84	0.69	0.73	3378
weighted avg	0.87	0.88	0.86	3378

❖ Interpretations:

-----Test set Tuned Model-----

- **Accuracy:**
 - The Accuracy of model is 88%.
 - 88% of the Data points are identified correctly
- **Recall:** Represents the correctness of the Model in a particular class.
 - Recall for class 1(Churn) is 40% and recall for class 0(non-churn) is 98%.
 - 40% of those Churned were correctly identified as Churned by the model.
 - Out of all the customers who are Churned,40% of them are identified correctly as Churned.
 - Out of all the customers who are not Churned, 98% of them are identified correctly as Not-Churned.
- **Precision:**
 - Precision for class 1(Churn) is 78% and for class 0(Non-Churn) is 89%.
 - 78% of the customers predicted are actually Churned Out of all the Customers that are predicted as Churned.
 - 89% of the Customers predicted are actually Not Churned out of all the customers predicted as not Churned.

❖ Confusion Matrix:

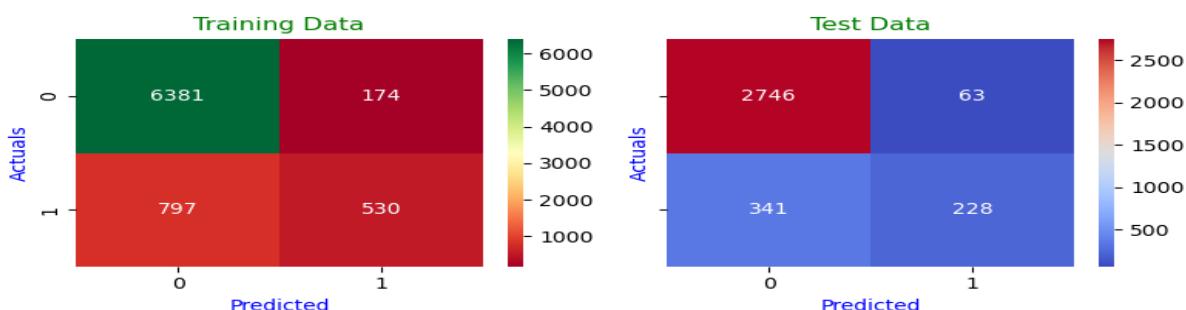


Fig no 53: Confusion Matrix LDA Tuned

❖ **Measuring and Plotting AUC-ROC Curve:**

- AUC for the Training Data: 0.858
- AUC for the Test Data: 0.857

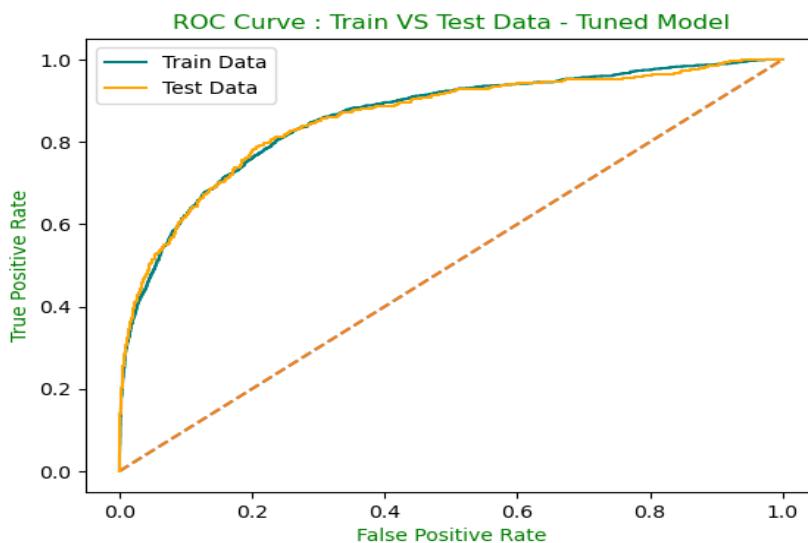


Fig no 54: ROC Curve LDA Tuned

❖ **Insights:**

- AUC for Train data is 85% and AUC for Test data is 85%.
- The AUC & ROC curve shows that is covering almost the same area for Train and Test data. Therefore, this can be considered a good optimized model.

❖ **Observations:**

- For LDA Model, Accuracy and AUC score are almost same for Base and Tuned model.
- Model build on balanced dataset have much less accuracy than Base and Tuned model but AUC score is slightly more than Base and Tuned Model.

Support Vector Classifier (SVC)

```
SVC
SVC(kernel='linear', random_state=42)
```

❖ **Accuracy of Train and Test data:**

- Accuracy of the Training Data: 0.8828977416899264
- Accuracy of the Test Data: 0.8872113676731794

❖ **validness of models (over fitting or under fitting):**

- The Accuracy of both the Train data and Test data are almost same; hence our model is Neither Overfit nor Underfit.

❖ **Classification Report:**

Classification Report of the Training data:				
	precision	recall	f1-score	support
0	0.89	0.98	0.93	6555
1	0.80	0.40	0.54	1327
			accuracy	0.88
			macro avg	0.85
			weighted avg	0.88
Classification Report of the Test data:				
	precision	recall	f1-score	support
0	0.89	0.98	0.94	2809
1	0.83	0.41	0.55	569
			accuracy	0.89
			macro avg	0.86
			weighted avg	0.88

❖ Interpretations:

-----Test set-----

- Accuracy:
 - The Accuracy of model is 89%.
 - 89% of the Data points are identified correctly
- Recall: Represents the correctness of the Model in a particular class.
 - Recall for class 1(Churn) is 41% and recall for class 0(non-churn) is 98%.
 - 41% of those Churned were correctly identified as Churned by the model.
 - Out of all the customers who are Churned,41% of them are identified correctly as Churned.
 - Out of all the customers who are not Churned, 98% of them are identified correctly as Not-Churned.
- Precision:
 - Precision for class 1(Churn) is 83% and for class 0(non-churn) is 89%.
 - 83% of the customers predicted are actually Churned Out of all the Customers that are predicted as Churned.
 - 89% of the Customers predicted are actually Not Churned out of all the customers predicted as not Churned.

❖ Confusion Matrix:

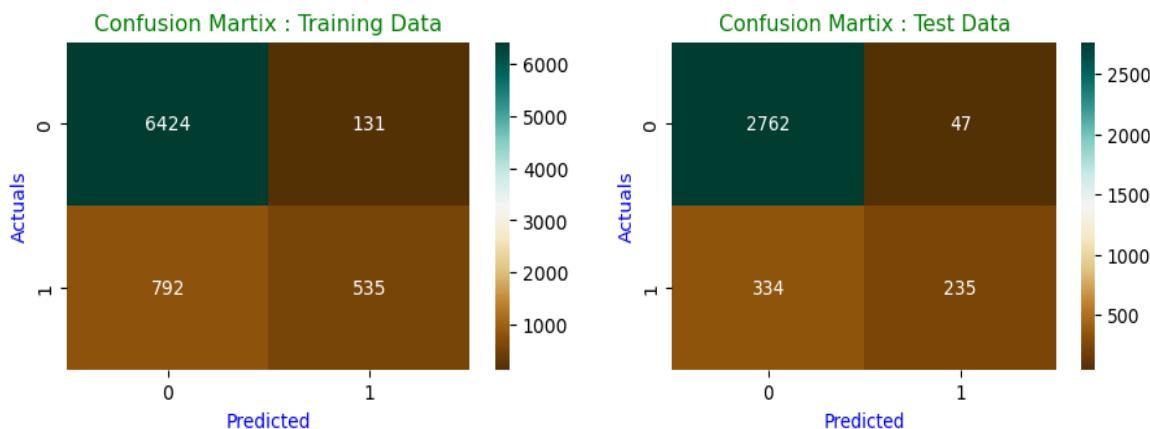


Fig no 55: Confusion Matrix SVC

❖ **AUC - ROC Curve:**

- AUC of Training Data: 0.692
- AUC of Test Data: 0.698

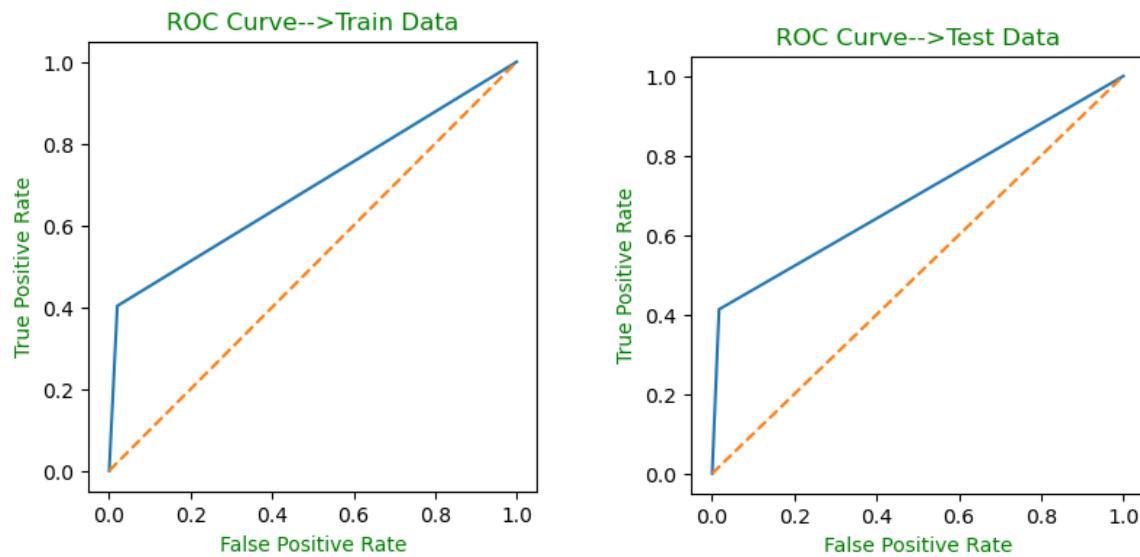


Fig no 56: ROC Curve SVC

❖ **Observations:**

- AUC for Train data is 69% and AUC for Test data is 69%.
- The AUC & ROC curve shows that it is covering almost the same area for Train and Test data.
- Since the AUC score is very less as compared to the above models so it is not considered a Good Model.

Support Vector Machine on Balanced Dataset

```
SVC
SVC(kernel='linear', random_state=42)
```

❖ **Accuracy on Train and Test dataset:**

- Accuracy of the Training Data: 0.8172783541103653
- Accuracy of the Test Data: 0.8164594434576673

❖ **Validness of models (over fitting or under fitting):**

- The Accuracy of both the Train data and Test data are almost same; hence our model is Neither Overfit nor Underfit.

❖ **Classification Report:**

Classification Report of the Training data:				
	precision	recall	f1-score	support
0	0.85	0.83	0.84	6555
1	0.78	0.80	0.79	4916
accuracy			0.82	11471
macro avg	0.81	0.81	0.81	11471
weighted avg	0.82	0.82	0.82	11471

Classification Report of the Test data:				
	precision	recall	f1-score	support
0	0.95	0.82	0.88	2809
1	0.47	0.78	0.59	569
accuracy			0.82	3378
macro avg	0.71	0.80	0.74	3378
weighted avg	0.87	0.82	0.83	3378

❖ Interpretations:

-----Test set Smote Data-----

➤ Accuracy:

- The Accuracy of model is 82%.
 - 82% of the Data points are identified correctly.
- Recall: Represents the correctness of the Model in a particular class.
- Recall for class 1(Churn) is 78% and recall for class 0(non-churn) is 82%.
 - 78% of those Churned were correctly identified as Churned by the model.
 - Out of all the customers who are Churned,78% of them are identified correctly as Churned.
 - Out of all the customers who are not Churned, 82% of them are identified correctly as Not-Churned.

➤ Precision:

- Precision for class 1(Churn) is 47% and for class 0(non-churn) is 95%.
- 47% of the customers predicted are actually Churned Out of all the Customers that are predicted as Churned.
- 95% of the Customers predicted are actually Not Churned out of all the customers predicted as not Churned.

❖ Confusion Matrix:

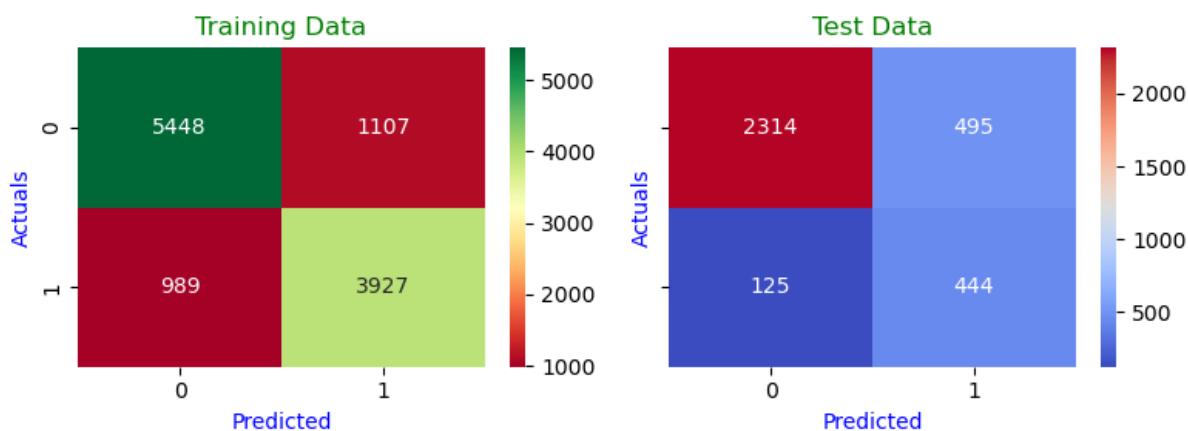


Fig no 57: Confusion Matrix SVC Smote

❖ **AUC - ROC Curve:**

- **AUC for the Training Data: 0.815**
- **AUC for the Test Data: 0.802**

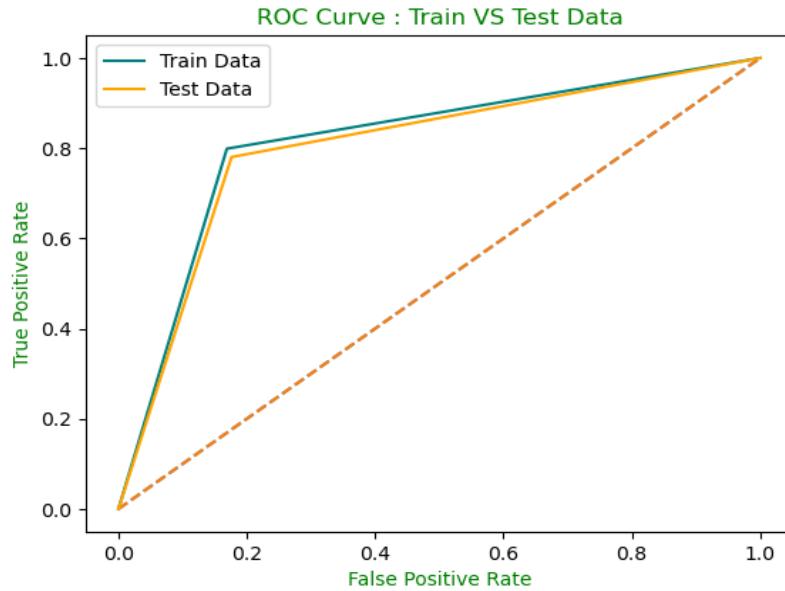


Fig no 58: ROC Curve SVC Smote

❖ **Observations:**

- AUC for Train data is 81% and AUC for Test data is 80%.
- The AUC & ROC curve shows that is covering almost the same area for Train and Test data. Therefore, this can be considered a Good optimized model.

Applying GridSearchCv on SVM

```

GridSearchCV
GridSearchCV(cv=5, estimator=SVC(kernel='linear', random_state=42), n_jobs=-1,
            param_grid={'C': [0.01, 0.1, 1, 10], 'degree': [2, 3],
                        'gamma': [0.01, 0.1, 0.2],
                        'kernel': ['linear', 'rbf', 'poly']},
            scoring='f1')
    estimator: SVC
        SVC(kernel='linear', random_state=42)
            SVC
                SVC(kernel='linear', random_state=42)

```

Building model on best parameters

```

SVC
SVC(C=10, degree=2, gamma=0.2, random_state=42)

```

❖ Accuracy of Train and Test Data:

- Accuracy on Train Data: 0.9340268967267191
- Accuracy on Test Data: 0.9129662522202486

❖ validness of models (over fitting or under fitting):

- The Accuracy of both the Train data and Test data are almost same; hence our model is Neither Overfit nor Underfit.

❖ Classification Report:

```
Classification Report of the Training data:
      precision    recall   f1-score   support
          0       0.94      0.99      0.96     6555
          1       0.92      0.66      0.77     1327
  accuracy                           0.93     7882
  macro avg       0.93      0.83      0.87     7882
  weighted avg    0.93      0.93      0.93     7882
```

```
Classification Report of the Test data:
      precision    recall   f1-score   support
          0       0.92      0.98      0.95     2809
          1       0.85      0.59      0.70      569
  accuracy                           0.91     3378
  macro avg       0.88      0.78      0.82     3378
  weighted avg    0.91      0.91      0.91     3378
```

❖ Interpretations:

-----Test set Tuned Model-----

➤ Accuracy:

- The Accuracy of model is 91%.
- 91% of the Data points are identified correctly .

➤ Recall: Represents the correctness of the Model in a particular class.

- Recall for class 1(Churn) is 59% and recall for class 0(non-churn) is 98%.
- 59% of those Churned were correctly identified as Churned by the model.
- Out of all the customers who are Churned,59% of them are identified correctly as Churned.
- Out of all the customers who are not Churned, 98% of them are identified correctly as Not-Churned.

➤ Precision:

- Precision for class 1(Churn) is 85% and for class 0(non-churn) is 92%.
- 85% of the customers predicted are actually Churned Out of all the Customers that are predicted as Churned.
- 92% of the Customers predicted are actually Not Churned out of all the customers predicted as not Churned.

❖ Confusion Matrix:

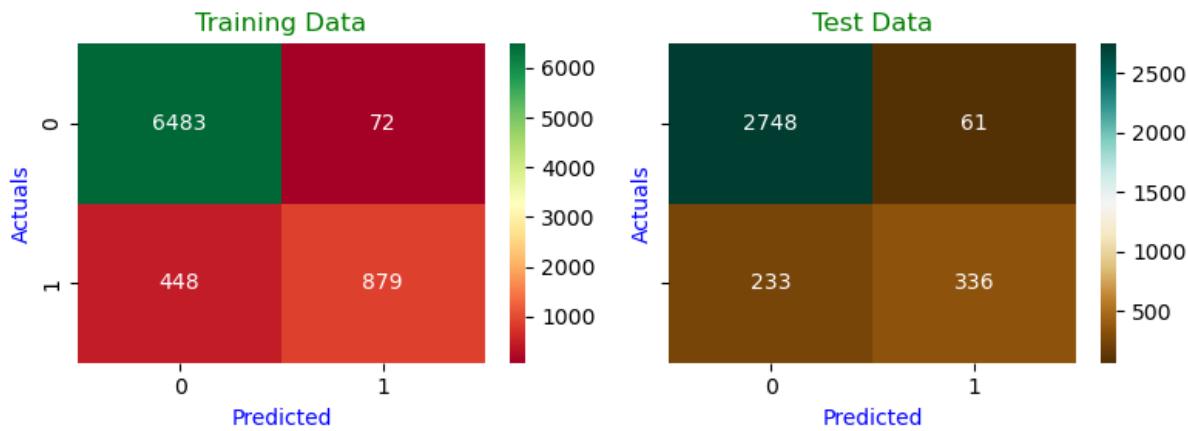


Fig no 59: Confusion Matrix SVC Tuned

❖ **Measuring and Plotting AUC-ROC Curve:**

- AUC for the Training Data: 0.826
- AUC for the Test Data: 0.784

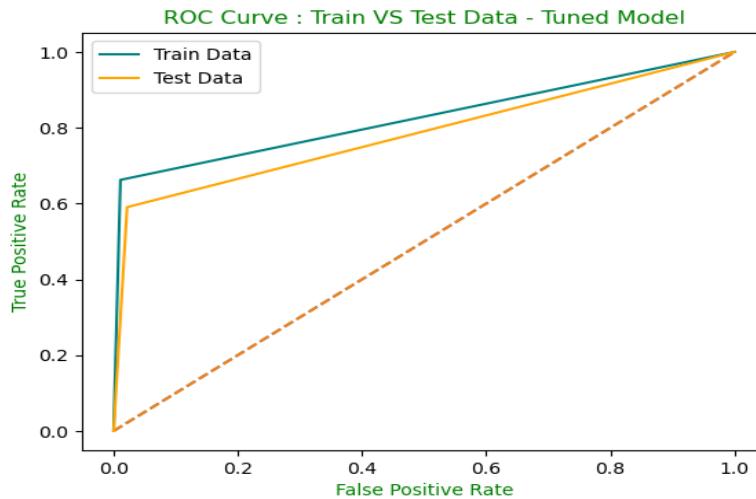


Fig no 60: ROC Curve SVC Tuned

❖ **Observations:**

- AUC for Train data is 82% and AUC for Test data is 78%.
- From the AUC values & ROC curve for the Train and Test data, it shows that Test data is not covering a large area as compared to the train data.
- It is not able to distinguish between Positive and Negative Classes much.

❖ **Insights:**

- SVC Tuned model is giving better accuracy and more AUC Score than the Base and Smote Model.
- Also, the model is neither overfit nor underfit.

K-nearest Neighbour (KNN Model)

Building Model with K = 3, 5, 7 and 9

For K = 3 ----->

- Accuracy Score for Training Data is: 0.9836335955341284
- Accuracy Score for Test Data is: 0.9552989934872705

For K = 5 ----->

- Accuracy Score for Training Data is: 0.9755138289774169
- Accuracy Score for Test Data is: 0.9496743635287153

For K = 7 ----->

- Accuracy Score for Training Data is: 0.9667597056584624
- Accuracy Score for Test Data is: 0.9357608052101836

For K = 9 ----->

- Accuracy Score for Training Data is: 0.9517888860695255
- Accuracy Score for Test Data is: 0.9212551805802249

➤ Here K=3 is giving Better Accuracy.

Building model with K=3 Neighbors

❖ Classification Report:

Classification Report of the Training data:				
	precision	recall	f1-score	support
0	0.99	0.99	0.99	6555
1	0.96	0.94	0.95	1327
accuracy			0.98	7882
macro avg	0.98	0.97	0.97	7882
weighted avg	0.98	0.98	0.98	7882

Classification Report of the Test data:

	precision	recall	f1-score	support
0	0.97	0.98	0.97	2809
1	0.89	0.84	0.86	569
accuracy			0.96	3378
macro avg	0.93	0.91	0.92	3378
weighted avg	0.95	0.96	0.95	3378

❖ Interpretations:

-----Test set KNN_3-----

➤ Accuracy:

- The Accuracy of model is 96%.
- 96% of the Data points are identified correctly.

➤ Recall: Represents the correctness of the Model in a particular class.

- Recall for class 1(Churn) is 84% and recall for class 0(non-churn) is 98%.
- 84% of those Churned were correctly identified as Churned by the model.
- Out of all the customers who are Churned, 84% of them are identified correctly as Churned.
- Out of all the customers who are not Churned, 98% of them are identified correctly as Not-Churned.

➤ **Precision:**

- Precision for class 1(Churn) is 89% and for class 0(non-churn) is 97%.
- 89% of the customers predicted are actually Churned Out of all the Customers that are predicted as Churned.
- 97% of the Customers predicted are actually Not Churned out of all the customers predicted as not Churned.

❖ **Confusion Matrix:**

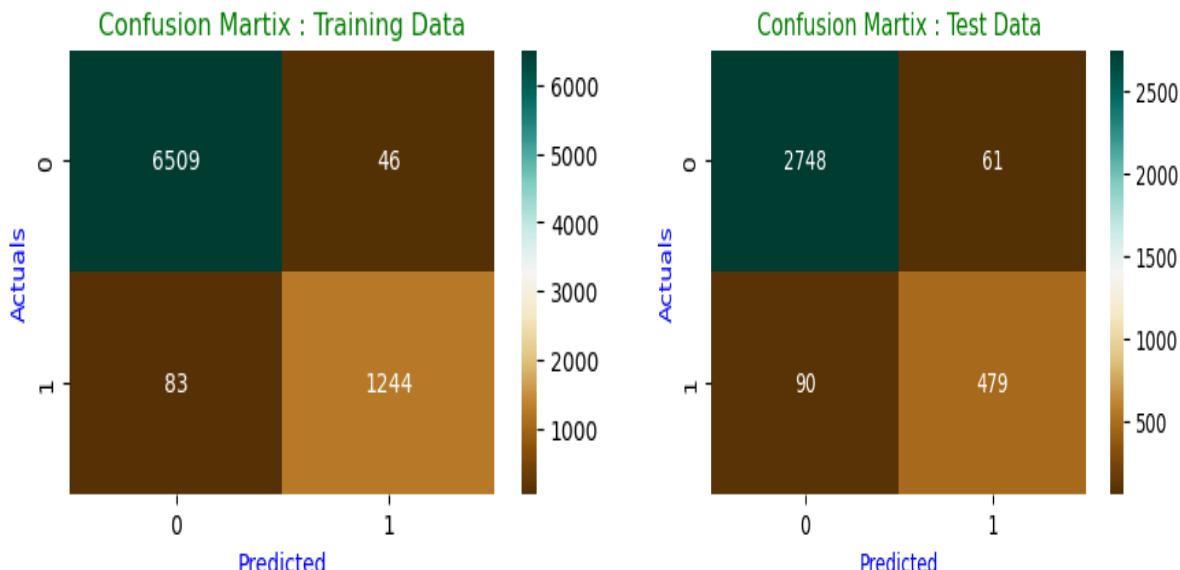


Fig no 61: Confusion Matrix KNN_3

❖ **AUC - ROC Curve:**

- AUC Train: 0.998
- AUC Test: 0.969

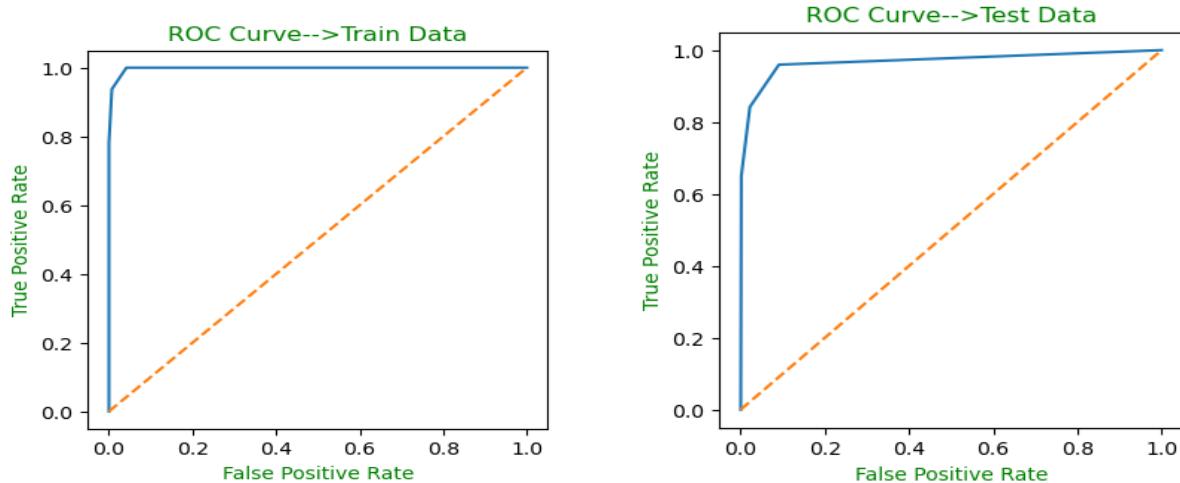


Fig no 62: ROC Curve KNN_3

❖ **Observations:**

- AUC for Train data is 99% and AUC for Test data is 96%.
- The AUC & ROC curve shows that Test data is covering slightly less than Train data.
- The AUC score is very high and is able to distinguish between Positive and Negative classes very well.
- This is a Good Generalized Model.

❖ **Cross Validation on KNN:**

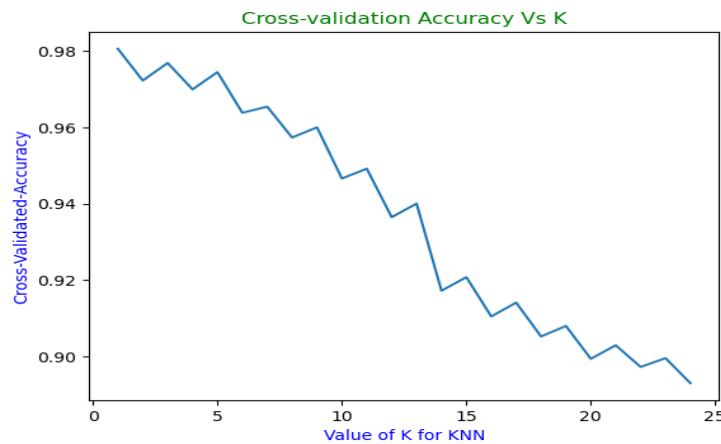


Fig no 63: CV Vs K - KNN

- The above visual representation helps us to understand that accuracy is almost 98% for K=1 for KNN.

❖ **Plotting misclassification error vs k (with k value on X-axis):**

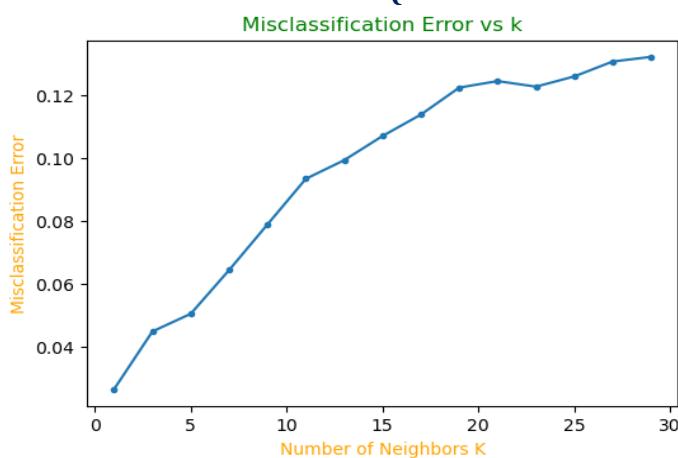


Fig no 64: MCE Vs K - KNN

❖ **Observations:**

- MCE is low for K=1.

Let's Build model on parameter K=1

```
KNeighborsClassifier
KNeighborsClassifier(n_neighbors=1)
```

❖ Accuracy of Train and Test data:

- Accuracy of Train data: 1.0
- Accuracy of Test data: 0.9736530491415039

❖ validness of models (over fitting or under fitting):

- The Accuracy of both the Train data and Test data doesn't differ much; hence our model is Neither Overfit nor Underfit.

❖ Classification Report:

Classification Report of the Training data:

	precision	recall	f1-score	support
0	1.00	1.00	1.00	6555
1	1.00	1.00	1.00	1327
accuracy			1.00	7882
macro avg	1.00	1.00	1.00	7882
weighted avg	1.00	1.00	1.00	7882

Classification Report of the Test data:

	precision	recall	f1-score	support
0	0.98	0.98	0.98	2809
1	0.92	0.92	0.92	569
accuracy			0.97	3378
macro avg	0.95	0.95	0.95	3378
weighted avg	0.97	0.97	0.97	3378

❖ Interpretations:

-----Test set KNN_1-----

➤ Accuracy

- The Accuracy of model is 97%.

- 97% of the Data points are identified correctly.

➤ Recall: Represents the correctness of the Model in a particular class.

- Recall for class 1(Churn) is 92% and recall for class 0(non-churn) is 98%.
- 92% of those Churned were correctly identified as Churned by the model.
- Out of all the customers who are Churned, 92% of them are identified correctly as Churned.
- Out of all the customers who are not Churned, 98% of them are identified correctly as Not-Churned.

➤ Precision:

- Precision for class 1(Churn) is 92% and for class 0(non-churn) is 98%.
- 92% of the customers predicted are actually Churned Out of all the Customers that are predicted as Churned.

- 98% of the Customers predicted are actually Not Churned out of all the customers predicted as not Churned.

❖ Note:

- Here we found that Precision and Recall are same for both the Classes (Churn and Non-Churn).
- This is a Very good Trade-off between Precision and Recall.
- This can be considered as a Very good model in terms of Precision, Recall, f1 score and Accuracy.
- This model is Neither Overfit nor Underfit.

❖ Confusion Matrix:

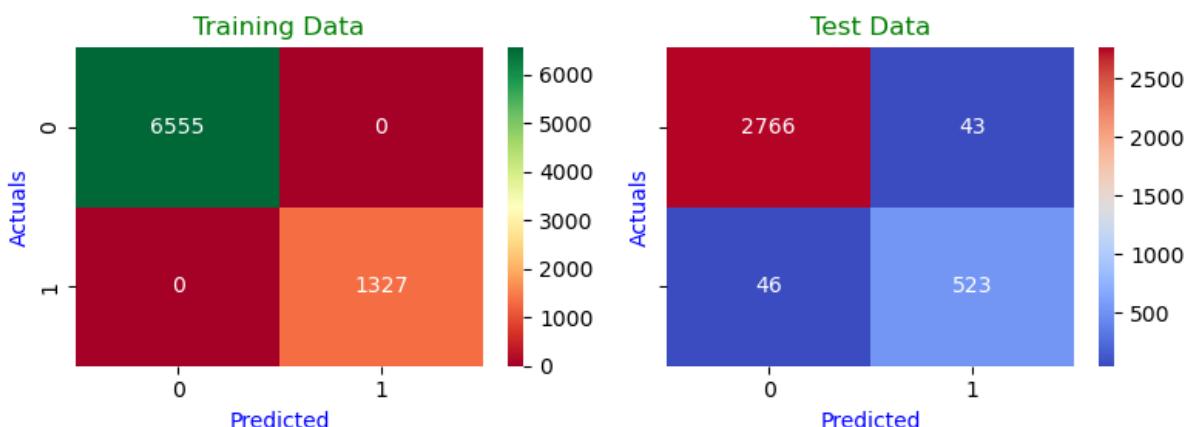


Fig no 65: Confusion Matrix KNN_1

❖ Plotting AUC - ROC Curve:

- AUC for the Training Data: 1.000
- AUC for the Test Data: 0.952

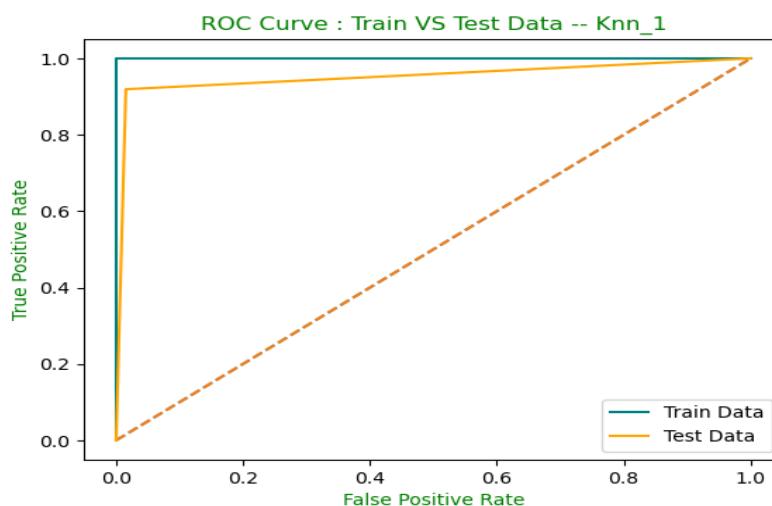


Fig no 66: ROC Curve KNN_1

❖ **Observations:**

- AUC for Train data is 100% and AUC for Test data is 95%.
- It is a Good Generalized Model.
- The AUC & ROC curve shows that Test data is covering slightly lesser area than Train data.
- The AUC score is very high and is able to distinguish between Positive and Negative classes very well.

Building KNN model using SMOTE Data

Building Model with K=3 neighbors

❖ **Accuracy of Train and Test data:**

- Accuracy Score for Training Data is: 0.9899747188562462
- Accuracy Score for Test Data is: 0.9520426287744227

❖ **Classification Report:**

Classification Report of the Training data:				
	precision	recall	f1-score	support
0	1.00	0.98	0.99	6555
1	0.98	1.00	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

Classification Report of the Test data:				
	precision	recall	f1-score	support
0	0.99	0.95	0.97	2809
1	0.81	0.94	0.87	569
accuracy			0.95	3378
macro avg	0.90	0.95	0.92	3378
weighted avg	0.96	0.95	0.95	3378

❖ **Interpretations:**

-----Test set Smote data KNN_3-----

➤ **Accuracy**

- The Accuracy of model is 95%.
- 95% of the Data points are identified correctly.

➤ **Recall:** Represents the correctness of the Model in a particular class.

- Recall for class 1(Churn) is 94% and recall for class 0(non-churn) is 95%.
- 94% of those Churned were correctly identified as Churned by the model.
- Out of all the customers who are Churned, 94% of them are identified correctly as Churned.
- Out of all the customers who are not Churned, 95% of them are identified correctly as Not-Churned.

➤ **Precision:**

- Precision for class 1(Churn) is 81% and for class 0(non-churn) is 99%.
- 81% of the customers predicted are actually Churned Out of all the Customers that are predicted as Churned.

- 99% of the Customers predicted are actually Not Churned out of all the customers predicted as not Churned.

❖ Confusion Matrix:

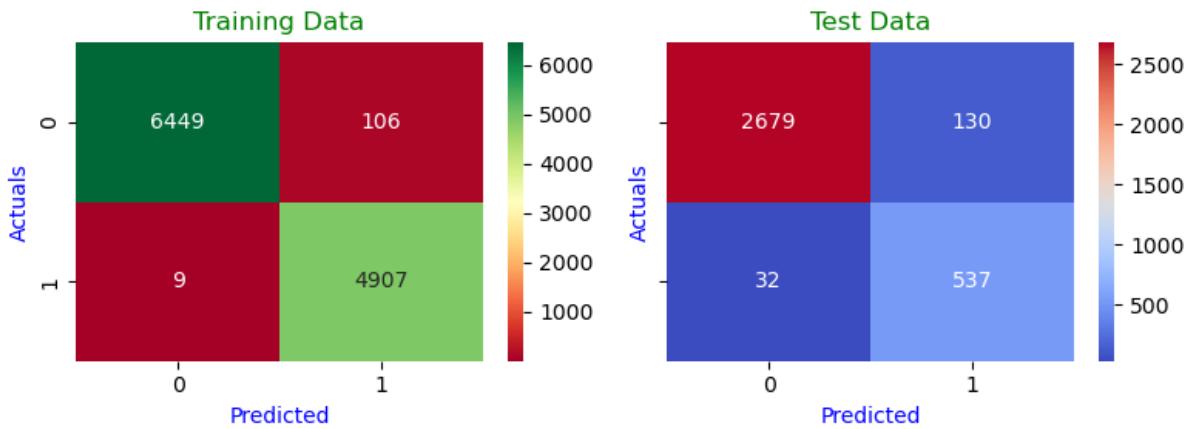


Fig no 67: Confusion Matrix KNN_3 Smote

❖ AUC - ROC Curve:

- AUC for the Training Data: 1.000
- AUC for the Test Data: 0.973

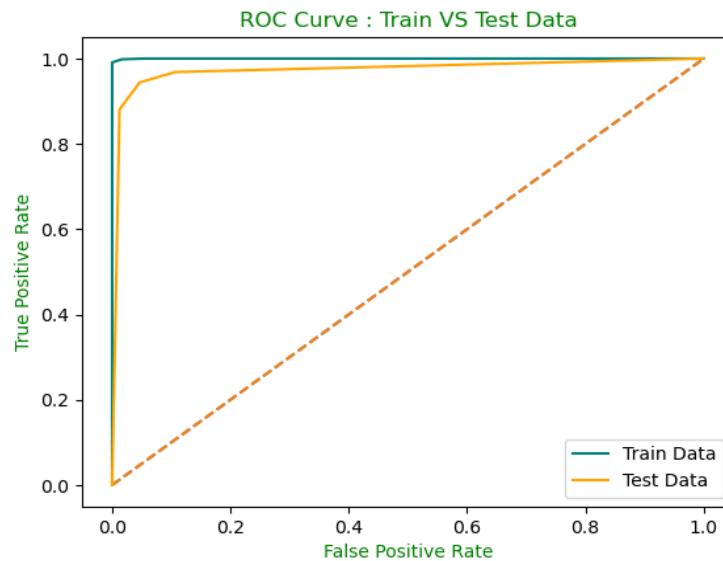


Fig no 68: ROC Curve KNN_3 Smote

❖ Observations:

- AUC for Train data is 100% and AUC for Test data is 97%.
- The AUC & ROC curve shows that is covering almost the same area for Train and Test data. Therefore, this can be considered a Good optimized model.

- Also, the AUC score is very high and is able to distinguish between Positive and Negative classes very well.

GridSearchCV for KNearestNeighbours (KNN)

```

    GridSearchCV
GridSearchCV(cv=10, estimator=KNeighborsClassifier(),
            param_grid={'algorithm': ['auto', 'ball_tree', 'kd_tree', 'brute'],
                        'metric': ['minkowski', 'euclidean', 'manhattan'],
                        'n_neighbors': [1, 2, 3, 4, 5, 6, 7, 8, 9],
                        'weights': ['uniform', 'distance']})
            - estimator: KNeighborsClassifier
              KNeighborsClassifier()
                -> KNeighborsClassifier

```

Building models on best parameters

```

    KNeighborsClassifier
KNeighborsClassifier(metric='manhattan', n_neighbors=1)

```

- Accuracy of Train and Test Data:
 - Accuracy Score for Training Data is: 1.0
 - Accuracy Score for Test Data is: 0.9846062759029012
- validness of models (over fitting or under fitting):
 - The Accuracy of both the Train data and Test data are almost same; hence our model is Neither Overfit nor Underfit.
- Classification Report:

Classification Report of the Training data:				
	precision	recall	f1-score	support
0	1.00	1.00	1.00	6555
1	1.00	1.00	1.00	1327
accuracy			1.00	7882
macro avg	1.00	1.00	1.00	7882
weighted avg	1.00	1.00	1.00	7882

Classification Report of the Test data:				
	precision	recall	f1-score	support
0	0.99	0.99	0.99	2809
1	0.95	0.96	0.95	569
accuracy			0.98	3378
macro avg	0.97	0.97	0.97	3378
weighted avg	0.98	0.98	0.98	3378

❖ **Interpretations:**

-----Test set Tuned Model KNN_1-----

➤ **Accuracy:**

- The Accuracy of model is 98%.
 - 98% of the Data points are identified correctly.
- **Recall:** Represents the correctness of the Model in a particular class.
- Recall for class 1(Churn) is 96% and recall for class 0(non-churn) is 99%.
 - 96% of those Churned were correctly identified as Churned by the model.
 - Out of all the customers who are Churned, 96% of them are identified correctly as Churned.
 - Out of all the customers who are not Churned, 99% of them are identified correctly as Not-Churned.

➤ **Precision:**

- Precision for class 1(Churn) is 95% and for class 0(non-churn) is 99%.
- 95% of the customers predicted are actually Churned Out of all the Customers that are predicted as Churned.
- 99% of the Customers predicted are actually Not Churned out of all the customers predicted as not Churned.

❖ **Confusion Matrix:**

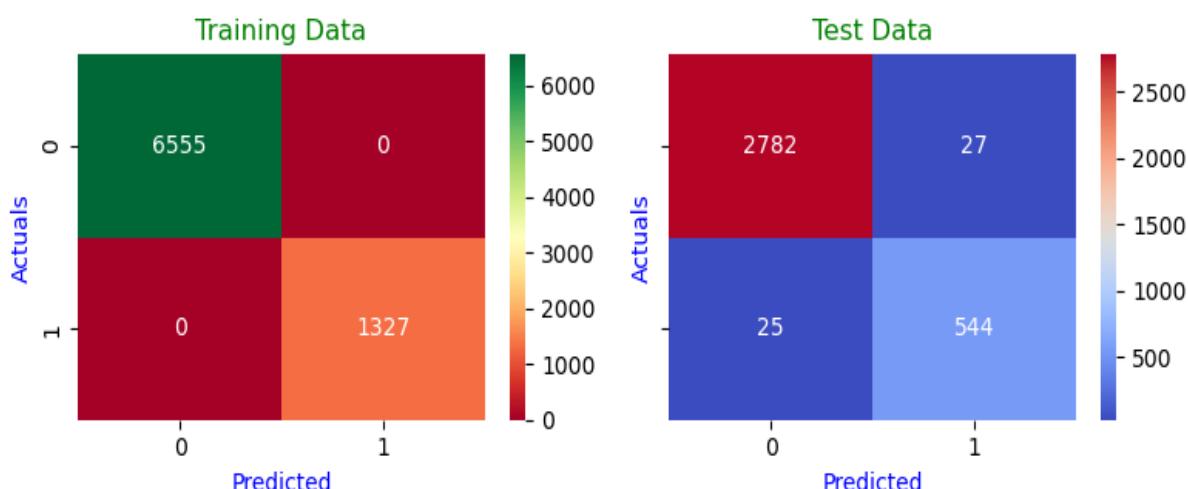


Fig no 69: Confusion Matrix KNN Tuned

❖ **Measuring and Plotting AUC-ROC Curve:**

- AUC for the Training Data: 1.000
- AUC for the Test Data: 0.973

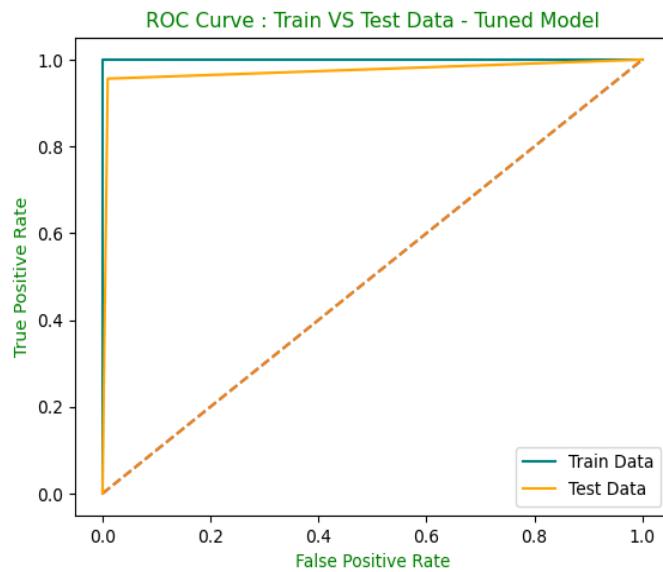


Fig no 70: ROC Curve KNN Tuned

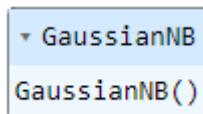
❖ Observations:

- AUC for Train data is 100% and AUC for Test data is 97%.
- The AUC & ROC curve shows that it is covering almost the same area for Train and Test data. Therefore, this can be considered a Good optimized model.
- Also, the AUC score is very high and is able to distinguish between Positive and Negative classes very well.

❖ Observations:

- For KNN Model, we are getting better accuracy with n-neighbours = 1.
- KNN Tuned model is giving better accuracy and AUC Score than KNN base and Smote Model.

Naive Bayes Model



❖ Accuracy of the Train and Test data:

- Accuracy Score for Training Data is: 0.8705912205024106
- Accuracy Score for Test Data is: 0.8587921847246892

❖ validness of models (over fitting or under fitting):

- The Accuracy of both the Train data and Test data are almost same; hence our model is Neither Overfit nor Underfit.

❖ Classification Report:

Classification Report of the Training data:				
	precision	recall	f1-score	support
0	0.91	0.93	0.92	6555
1	0.63	0.57	0.60	1327
accuracy			0.87	7882
macro avg		0.77	0.75	7882
weighted avg		0.87	0.87	7882

Classification Report of the Test data:				
	precision	recall	f1-score	support
0	0.91	0.92	0.92	2809
1	0.58	0.57	0.57	569
accuracy			0.86	3378
macro avg		0.75	0.74	3378
weighted avg		0.86	0.86	3378

❖ Interpretations:

-----Test set-----

➤ Accuracy:

- The Accuracy of model is 86%.
 - 86% of the Data points are identified correctly
- Recall: Represents the correctness of the Model in a particular class.
- Recall for class 1(Churn) is 57% and recall for class 0(non-churn) is 92%.
 - 57% of those Churned were correctly identified as Churned by the model.
 - Out of all the customers who are Churned,57% of them are identified correctly as Churned.
 - Out of all the customers who are not Churned, 92% of them are identified correctly as Not-Churned.

➤ Precision:

- Precision for class 1(Churn) is 58% and for class 0(non-churn) is 91%.
- 58% of the customers predicted are actually Churned Out of all the Customers that are predicted as Churned.
- 91% of the Customers predicted are actually Not Churned out of all the customers predicted as not Churned.

❖ Confusion Matrix:

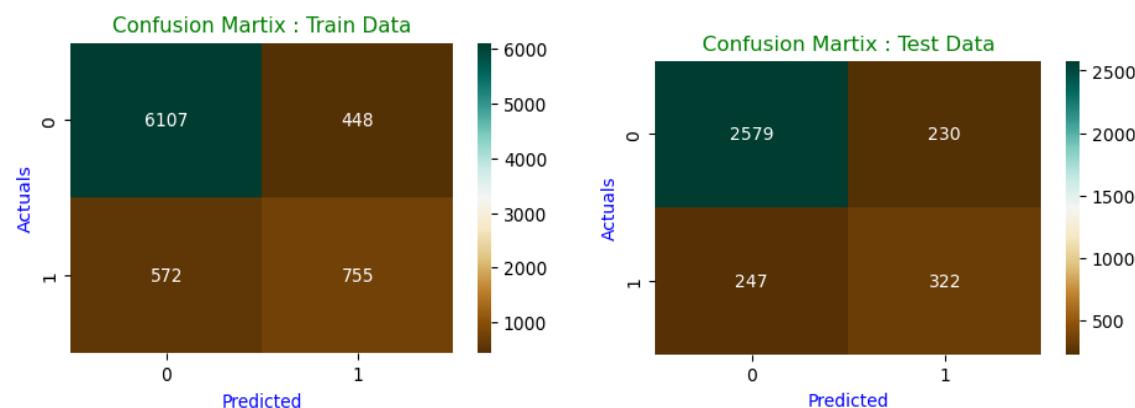


Fig no 71: Confusion Matrix NB Model

❖ **AUC - ROC Curve:**

- AUC Train: 0.831
- AUC Test: 0.819

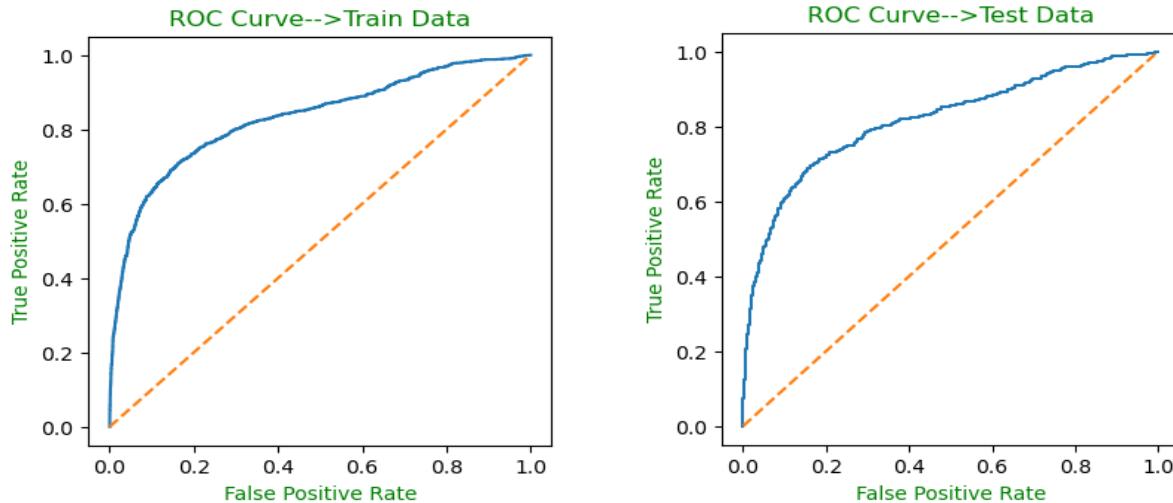
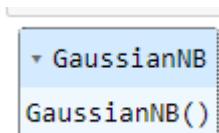


Fig no 72: ROC Curve NB Model

❖ **Observations:**

- AUC for Train data is 83% and AUC for Test data is 81%.
- The AUC & ROC curve shows that it is covering almost the same area for Train and Test data. Therefore, this can be considered a Good optimized model.
- Also, the AUC score is Good and is able to distinguish between Positive and Negative classes.

Building Naive Bayes model using SMOTE data



❖ **Accuracy of the Train and Test data:**

- Accuracy Score for Training Data is: 0.7530293784325691
- Accuracy Score for Test Data is: 0.7181764357608053

❖ **Validness of models (over fitting or under fitting)**

- The Accuracy of Train data is more than the Test data; hence our model is slightly Overfit.

❖ **Classification Report:**

Classification Report of the Training data:

	precision	recall	f1-score	support
0	0.83	0.72	0.77	6555
1	0.68	0.80	0.74	4916
accuracy			0.75	11471
macro avg	0.75	0.76	0.75	11471
weighted avg	0.76	0.75	0.75	11471

Classification Report of the Test data:

	precision	recall	f1-score	support
0	0.94	0.71	0.81	2809
1	0.35	0.77	0.48	569
accuracy			0.72	3378
macro avg	0.64	0.74	0.64	3378
weighted avg	0.84	0.72	0.75	3378

❖ Interpretations:

- Test set Smote Data-----
- Accuracy:
 - The Accuracy of model is 72%.
 - 72% of the Data points are identified correctly.
 - Recall: Represents the correctness of the Model in a particular class.
 - Recall for class 1(Churn) is 77% and recall for class 0(non-churn) is 71%.
 - 77% of those Churned were correctly identified as Churned by the model.
 - Out of all the customers who are Churned,77% of them are identified correctly as Churned.
 - Out of all the customers who are not Churned, 71% of them are identified correctly as Not-Churned.
 - Precision:
 - Precision for class 1(Churn) is 35% and for class 0(non-churn) is 94%.
 - 35% of the customers predicted are actually Churned Out of all the Customers that are predicted as Churned.
 - 94% of the Customers predicted are actually Not Churned out of all the customers predicted as not Churned.

❖ Confusion Matrix:

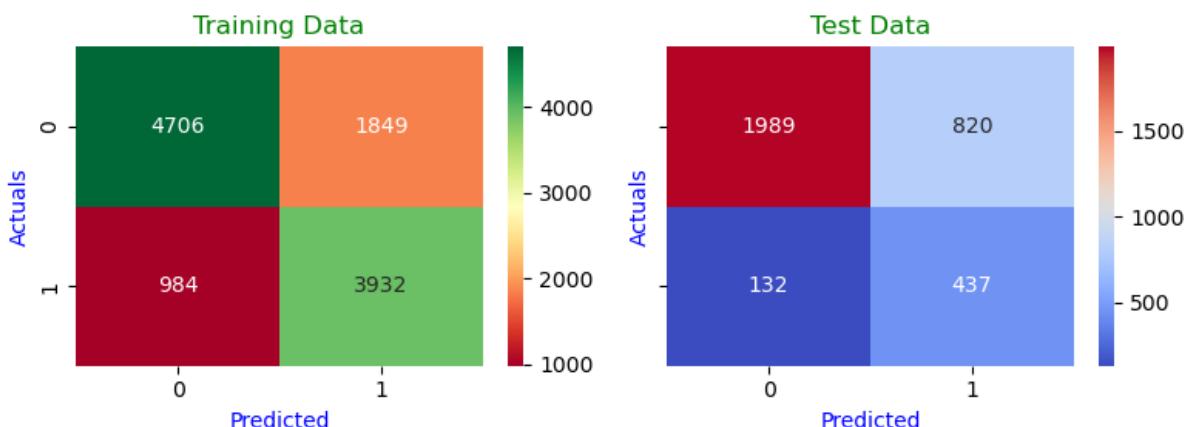


Fig no 73: Confusion Matrix NB Smote

❖ **Measuring and Plotting AUC-ROC Curve:**

- **AUC for the Training Data: 0.837**
- **AUC for the Test Data: 0.810**

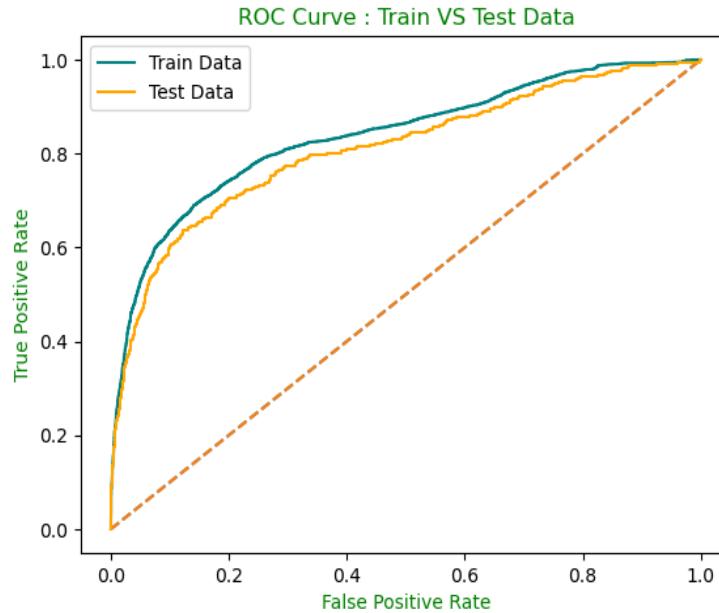


Fig no 74: ROC Curve NB Smote

❖ **Observations:**

- AUC for Train data is 83% and AUC for Test data is 81%.
- The AUC & ROC curve shows that it is covering almost the same area for Train and Test data. Therefore, this can be considered an optimized model.
- Also, the AUC score is Good and is able to distinguish between Positive and Negative classes well.

❖ **Observations:**

- Naive Bayes Base model is giving better Accuracy than Smote Model.

Decision Tree Model

```
DecisionTreeClassifier
DecisionTreeClassifier(random_state=42)
```

❖ **Accuracy of the Train and Test data:**

- Accuracy Score for Training Data is: 1.0
- Accuracy Score for Test Data is: 0.9547069271758437

❖ **Validness of models (over fitting or under fitting):**

- The Accuracy of the Train data is slightly less than Test data, hence our model is Neither Overfit nor Underfit.

❖ Classification Report:

Classification Report of the Training data:

	precision	recall	f1-score	support
0	1.00	1.00	1.00	6555
1	1.00	1.00	1.00	1327
accuracy			1.00	7882
macro avg	1.00	1.00	1.00	7882
weighted avg	1.00	1.00	1.00	7882

Classification Report of the Test data:

	precision	recall	f1-score	support
0	0.97	0.98	0.97	2809
1	0.87	0.85	0.86	569
accuracy			0.95	3378
macro avg	0.92	0.91	0.92	3378
weighted avg	0.95	0.95	0.95	3378

❖ Interpretations:

-----Test set-----

➤ Accuracy:

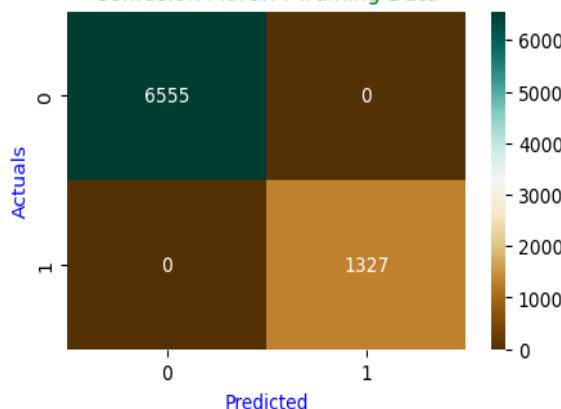
- The Accuracy of model is 95%.
 - 95% of the Data points are identified correctly
- Recall: Represents the correctness of the Model in a particular class.
- Recall for class 1(Churn) is 85% and recall for class 0(non-churn) is 98%.
 - 85% of those Churned were correctly identified as Churned by the model.
 - Out of all the customers who are Churned,85% of them are identified correctly as Churned.
 - Out of all the customers who are not Churned, 98% of them are identified correctly as Not-Churned.

➤ Precision:

- Precision for class 1(Churn) is 87% and for class 0(non-churn) is 97%.
- 87% of the customers predicted are actually Churned Out of all the Customers that are predicted as Churned.
- 97% of the Customers predicted are actually Not Churned out of all the customers predicted as not Churned.

❖ Confusion Matrix:

Confusion Matrix : Training Data



Confusion Matrix : Test Data

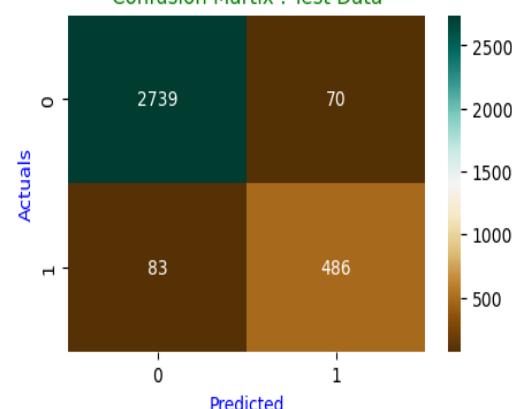


Fig no 75: Confusion Matrix DT

❖ **AUC - ROC Curve:**

- AUC Train: 1.000
- AUC Test: 0.915

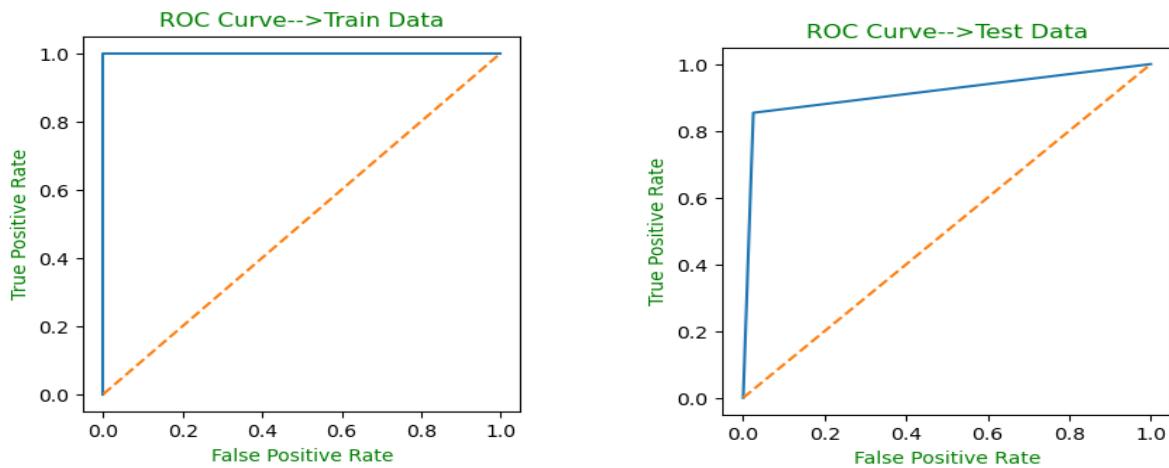


Fig no 76: ROC Curve DT

❖ **Observations:**

- AUC for Train data is 100% and AUC for Test data is 91%.
- From the AUC values & ROC curve for the Train and Test data, it shows that Test data is covering lesser area as compared to the Train data.
- As AUC score is greater than 90%, so it is able distinguish between the Positive and Negative classes well.

❖ **Feature Importance:**

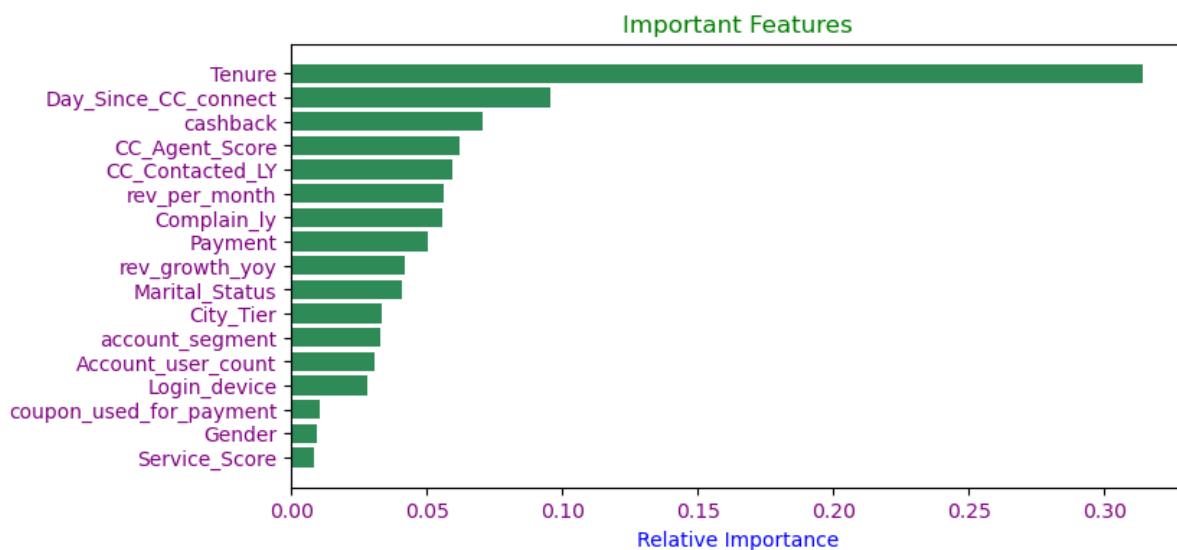


Fig no 77: Feature Imp DT

Insights:

- The 1st Five Important Features of Decision Tree Base Model are: Tenure, Days_since_CC_connect, cashback, CC_Agent_Score and CC_Contacted_LY.

Building Decision Tree model over balanced dataset

```
DecisionTreeClassifier
DecisionTreeClassifier(random_state=42)
```

- ❖ Accuracy of the Train and Test data:
 - Accuracy Score for Training Data is: 1.0
 - Accuracy Score for Test Data is: 0.9283599763173476
- ❖ Validness of models (over fitting or under fitting):
 - The Accuracy of both the Train data and Test data differ by only 8%, Hence our model is Neither Overfit nor Underfit.

❖ Classification Report:

Classification Report of the Training data:				
	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

Classification Report of the Test data:				
	precision	recall	f1-score	support
0	0.96	0.95	0.96	2809
1	0.77	0.81	0.79	569
accuracy			0.93	3378
macro avg	0.87	0.88	0.87	3378
weighted avg	0.93	0.93	0.93	3378

❖ Interpretations:

-----Test set-----

- Accuracy:
 - The Accuracy of model is 93%.
 - 93% of the Data points are identified correctly
- Recall: Represents the correctness of the Model in a particular class.
 - Recall for class 1(Churn) is 81% and recall for class 0(non-churn) is 95%.
 - 81% of those Churned were correctly identified as Churned by the model.
 - Out of all the customers who are Churned, 81% of them are identified correctly as Churned.
 - Out of all the customers who are not Churned, 95% of them are identified correctly as Not-Churned.
- Precision:
 - Precision for class 1(Churn) is 77% and for class 0(non-churn) is 96%.
 - 77% of the customers predicted are actually Churned Out of all the Customers that are predicted as Churned.

- 96% of the Customers predicted are actually Not Churned out of all the customers predicted as not Churned.

❖ Confusion Matrix:

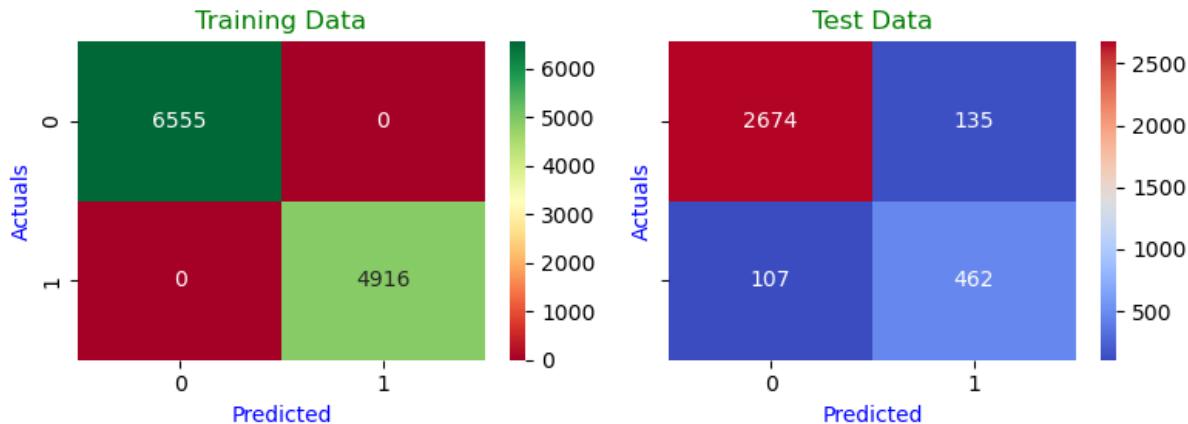


Fig no 78: Confusion matrix DT smote

❖ Measuring and Plotting AUC-ROC Curve:

- AUC for the Training Data: 1.000
- AUC for the Test Data: 0.882

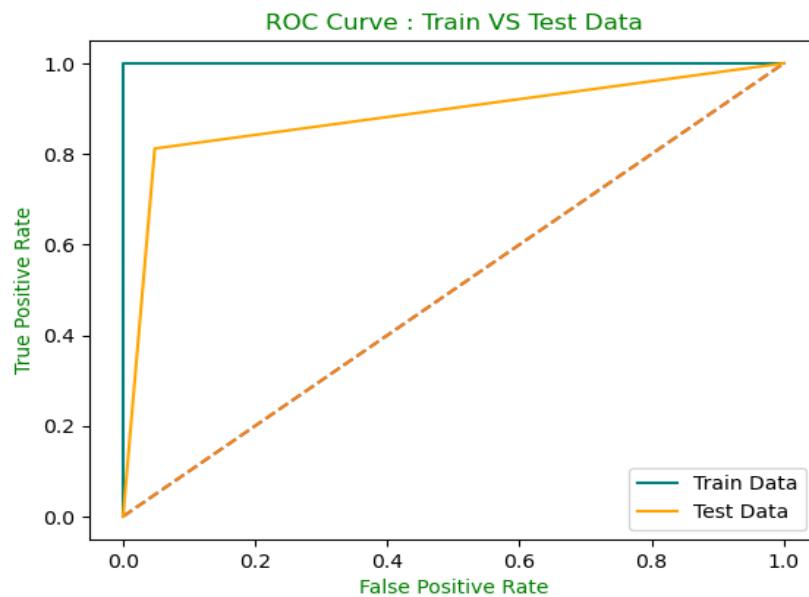


Fig no 79: ROC Curve DT smote

❖ Observations:

- AUC for Train data is 100% and AUC for Test data is 88%.
- From the AUC values & ROC curve for the Train and Test data, it shows that Test data is not covering a large area as compared to the train data.

Applying GridSearchCv on Decision Tree Model

```
Fitting 10 folds for each of 288 candidates, totalling 2880 fits
```

```
GridSearchCV
GridSearchCV(cv=10, estimator=DecisionTreeClassifier(random_state=42),
            n_jobs=-1,
            param_grid={'criterion': ['gini', 'entropy', 'log_loss'],
                        'max_depth': [10, 20, 30],
                        'min_impurity_decrease': [0.001, 0.0001],
                        'min_samples_leaf': [2, 3, 4, 5],
                        'min_samples_split': [2, 3, 5, 7]},
            scoring='accuracy', verbose=1)
    estimator: DecisionTreeClassifier
        DecisionTreeClassifier(random_state=42)
            DecisionTreeClassifier
```

Building model on best parameters

```
DecisionTreeClassifier
DecisionTreeClassifier(criterion='entropy', max_depth=30,
                      min_impurity_decrease=0.0001, min_samples_leaf=2,
                      random_state=42)
```

❖ Accuracy of the Training and Test Data:

- Accuracy of Train data: 0.9911190053285968
- Accuracy of Test data: 0.9440497335701599

❖ validness of models (over fitting or under fitting):

- There is not much difference between Accuracy of the Train data and Test data, hence our model is Neither Overfit nor Underfit.

❖ Classification Report:

Classification Report of the Training data:				
	precision	recall	f1-score	support
0	0.99	1.00	0.99	6555
1	1.00	0.95	0.97	1327
accuracy			0.99	7882
macro avg	0.99	0.97	0.98	7882
weighted avg	0.99	0.99	0.99	7882

Classification Report of the Test data:				
	precision	recall	f1-score	support
0	0.96	0.97	0.97	2809
1	0.85	0.82	0.83	569
accuracy			0.94	3378
macro avg	0.90	0.89	0.90	3378
weighted avg	0.94	0.94	0.94	3378

❖ Interpretations:

-----Test set Tuned Model-----

- **Accuracy:**
 - The Accuracy of model is 94%.
 - 94% of the Data points are identified correctly.
- **Recall:** Represents the correctness of the Model in a particular class.
 - Recall for class 1(Churn) is 82% and recall for class 0(non-churn) is 97%.
 - 82% of those Churned were correctly identified as Churned by the model.
 - Out of all the customers who are Churned, 82% of them are identified correctly as Churned.
 - Out of all the customers who are not Churned, 97% of them are identified correctly as Not-Churned.
- **Precision:**
 - Precision for class 1(Churn) is 85% and for class 0(non-churn) is 96%.
 - 85% of the customers predicted are actually Churned Out of all the Customers that are predicted as Churned.
 - 96% of the Customers predicted are actually Not Churned out of all the customers predicted as not Churned.

❖ **Confusion Matrix:**

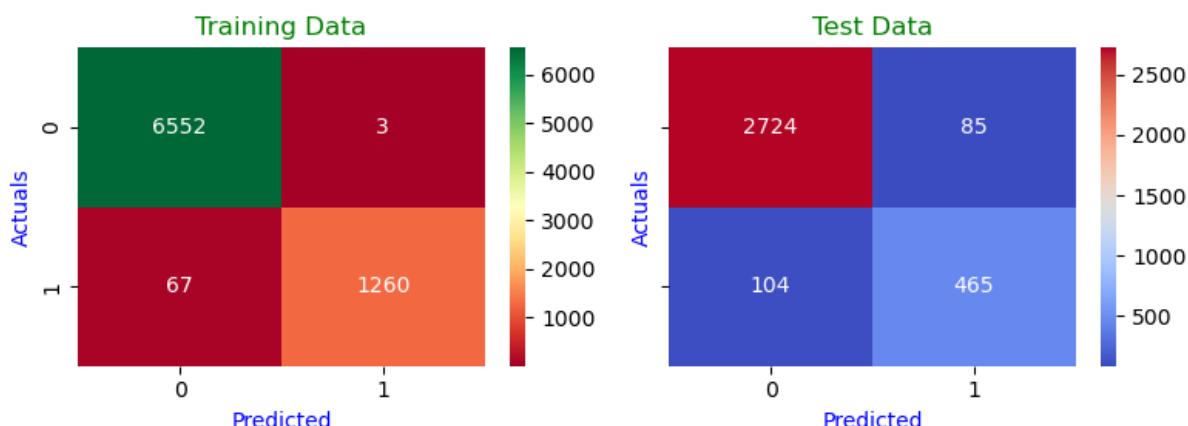


Fig no 80: Confusion Matrix DT Tuned

❖ **Measuring and Plotting AUC-ROC Curve:**

- AUC for the Training Data: 1.000
- AUC for the Test Data: 0.920

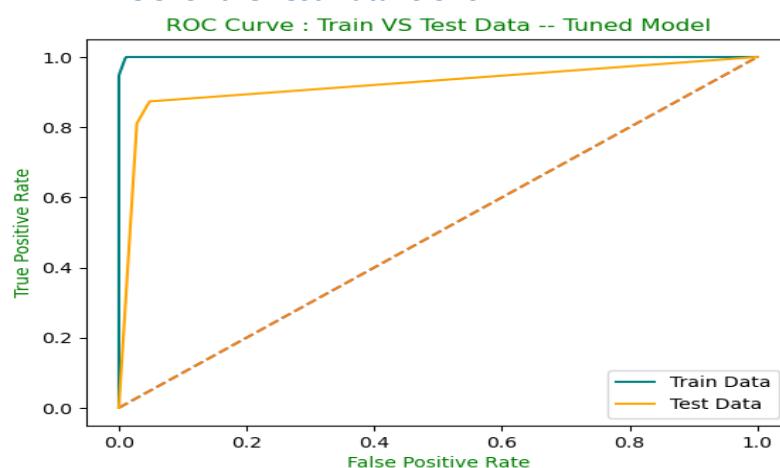


Fig no 81: ROC Curve DT Tuned

❖ **Observations:**

- AUC for Train data is 100% and AUC for Test data is 92%.
- From the AUC values & ROC curve for the Train and Test data, it shows that Test data is covering a less area as compared to the train data.
- Since the AUC score is good it is able to distinguish between Positive and Negative classes well.

❖ **Feature Importance:**

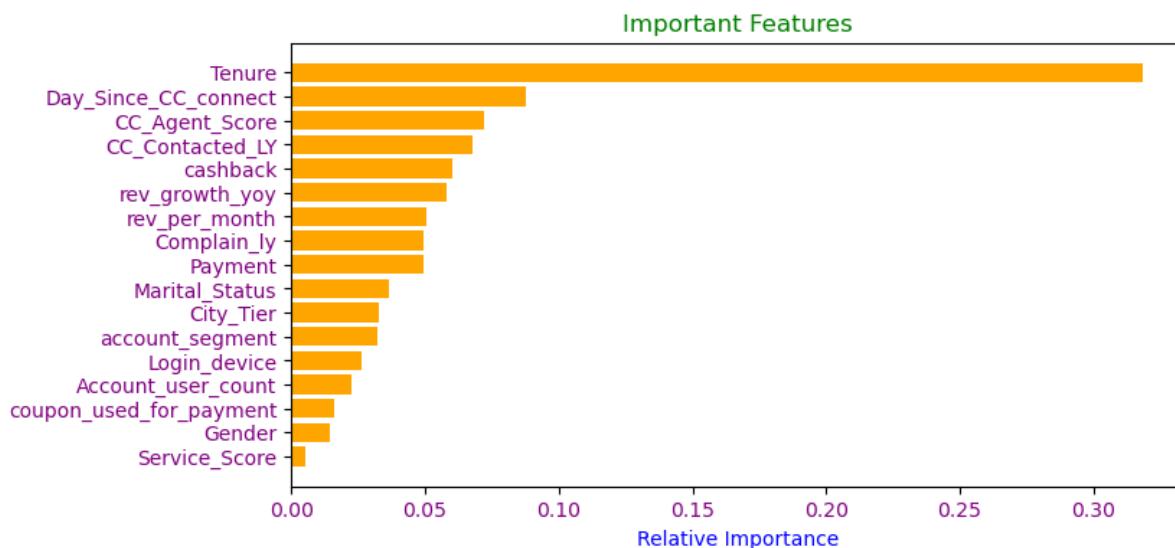


Fig no 82: Feature Imp DT Tuned

- The 1st Five important Features for decision Tree Tuned Model are: Tenure, Day_Since_CC_connect, CC_Agent_Score, CC_Contacted_LY and cashback.

❖ **Observations:**

- Accuracy and AUC Score are almost same for the base and Tuned model.

Random Forest Classifier

```
▼ RandomForestClassifier
RandomForestClassifier(random_state=42)
```

❖ **Accuracy of Train and Test data:**

- Accuracy on Training data: 1.0
- Accuracy on Test data: 0.9683244523386619

❖ **Classification Report:**

Classification Report of the Training data:

	precision	recall	f1-score	support
0	1.00	1.00	1.00	6555
1	1.00	1.00	1.00	1327
accuracy			1.00	7882
macro avg	1.00	1.00	1.00	7882
weighted avg	1.00	1.00	1.00	7882

Classification Report of the Test data:

	precision	recall	f1-score	support
0	0.97	0.99	0.98	2809
1	0.97	0.84	0.90	569
accuracy			0.97	3378
macro avg	0.97	0.92	0.94	3378
weighted avg	0.97	0.97	0.97	3378

❖ Interpretations:

Test set

- Accuracy:
 - The Accuracy of model is 97%.
 - 97% of the Data points are identified correctly.
- Recall: Represents the correctness of the Model in a particular class.
 - Recall for class 1(Churn) is 84% and recall for class 0(non-churn) is 99%.
 - 84% of those Churned were correctly identified as Churned by the model.
 - Out of all the customers who are Churned,84% of them are identified correctly as Churned.
 - Out of all the customers who are not Churned, 99% of them are identified correctly as Not-Churned.
- Precision:
 - Precision for class 1(Churn) is 97% and for class 0(non-churn) is 97%.
 - 97% of the customers predicted are actually Churned Out of all the Customers that are predicted as Churned.
 - 97% of the Customers predicted are actually Not Churned out of all the customers predicted as not Churned.

❖ Confusion Matrix:

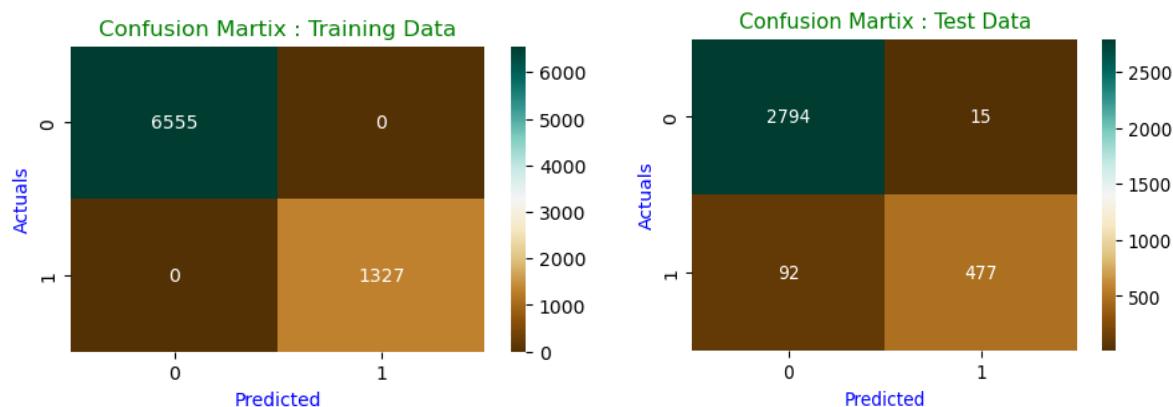


Fig no 83: Confusion Matrix RF

❖ AUC – ROC Curve:

- AUC Train: 1.000
- AUC Test: 0.992

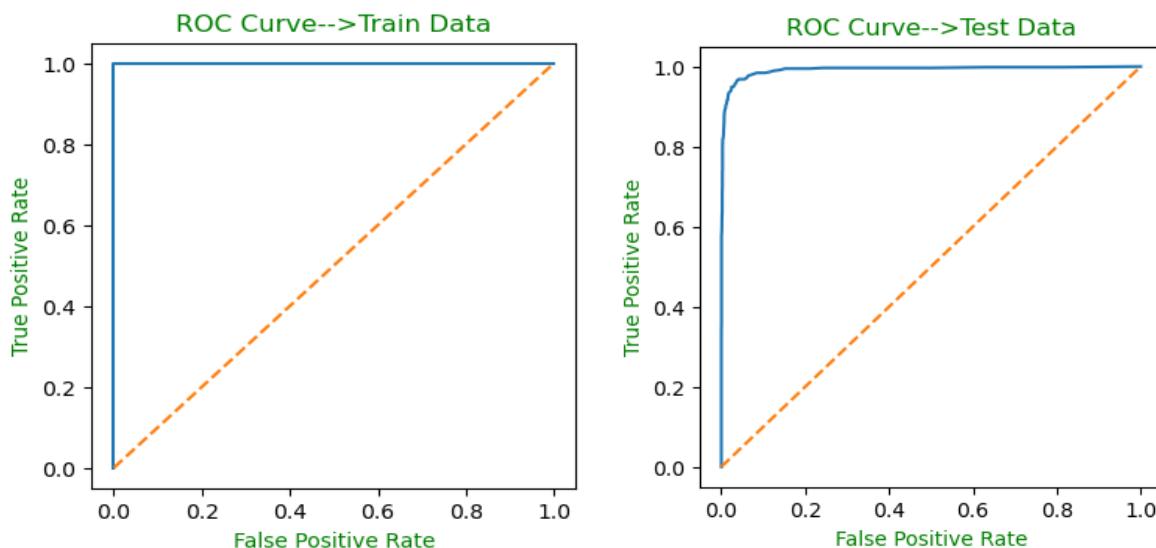


Fig no 84: ROC Curve RF

❖ Observations:

- AUC for Train data is 100% and AUC for Test data is 99%.
- The AUC & ROC curve shows that it is covering the same area for Train and Test data. Therefore, this is considered a Generalized good model.
- Also, the AUC score is very high and is able to distinguish between Positive and Negative classes very well.

❖ Feature importance of Random Forest:

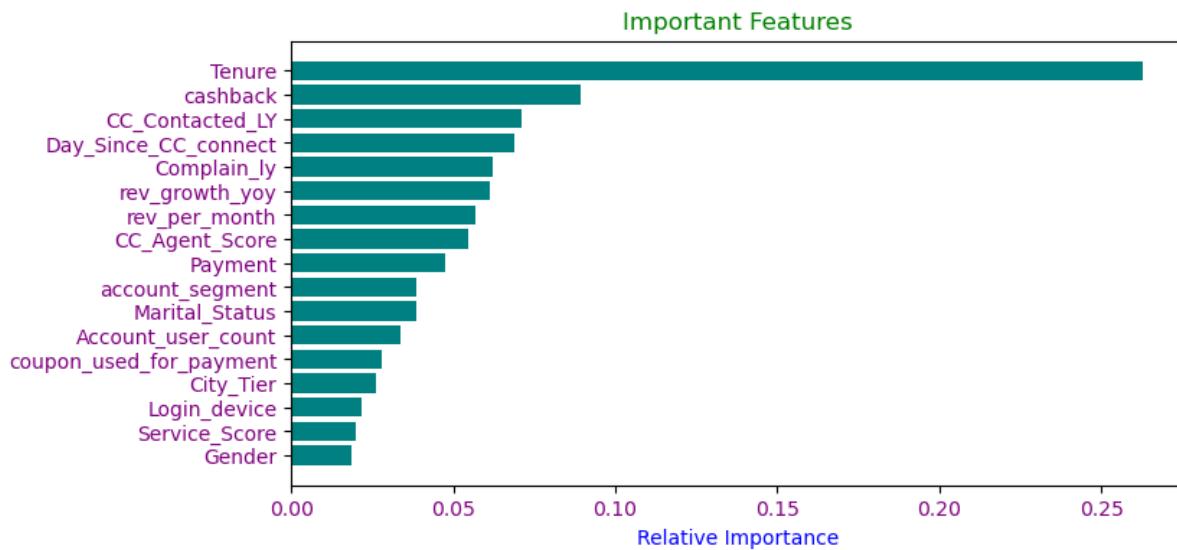


Fig no 85: Feature IMP RF

- The 1st Five Important Features of Random Forest Model are: Tenure, cashback, CC_Contacted_LY, Day_Since_CC_connect and Complain_ly.

Building Random Forest model over balanced dataset

```
RandomForestClassifier
RandomForestClassifier(n_estimators=500, random_state=42)
```

❖ Accuracy of Train and Test data:

- Accuracy on Train data: 1.0
- Accuracy on Test data: 0.9698046181172292

❖ Validness of models (over fitting or under fitting):

- There is not much difference between Train and Test Accuracy, hence our model is Neither Overfit nor Underfit.

❖ Classification Report:

Classification Report of the Training data:				
	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

Classification Report of the Test data:				
	precision	recall	f1-score	support
0	0.98	0.99	0.98	2809
1	0.93	0.88	0.91	569
accuracy			0.97	3378
macro avg	0.95	0.94	0.94	3378
weighted avg	0.97	0.97	0.97	3378

❖ Interpretations:

-----Test set Smote Data-----

➤ Accuracy:

- The Accuracy of model is 97%.
- 97% of the Data points are identified correctly.

➤ Recall: Represents the correctness of the Model in a particular class.

- Recall for class 1(Churn) is 88% and recall for class 0(non-churn) is 99%.
- 88% of those Churned were correctly identified as Churned by the model.
- Out of all the customers who are Churned, 88% of them are identified correctly as Churned.
- Out of all the customers who are not Churned, 99% of them are identified correctly as Not-Churned.

➤ Precision:

- Precision for class 1(Churn) is 93% and for class 0(non-churn) is 98%.
- 93% of the customers predicted are actually Churned Out of all the Customers that are predicted as Churned.
- 98% of the Customers predicted are actually Not Churned out of all the customers predicted as not Churned.

❖ Confusion Matrix:

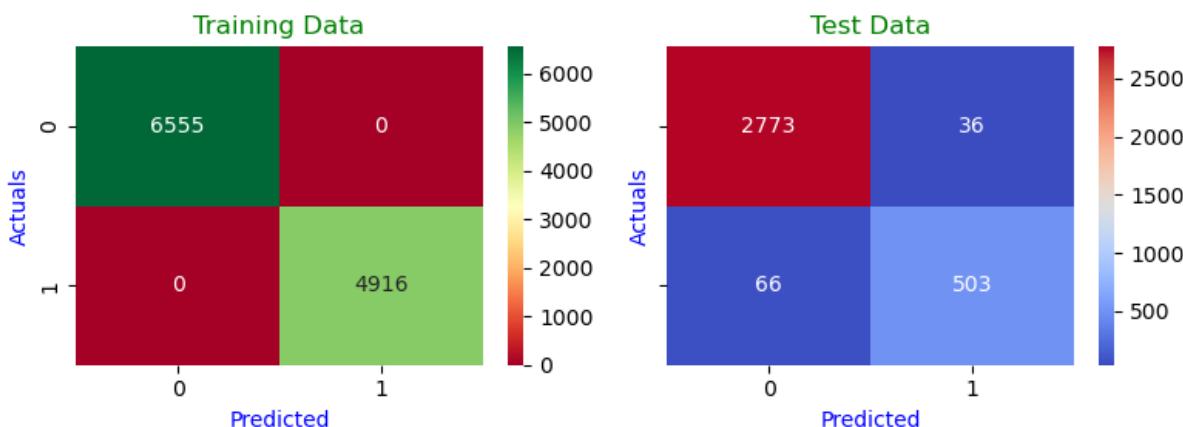


Fig no 86: Confusion Matrix RF smote

❖ Measuring and Plotting AUC-ROC Curve:

- AUC for the Training Data: 1.000
- AUC for the Test Data: 0.991

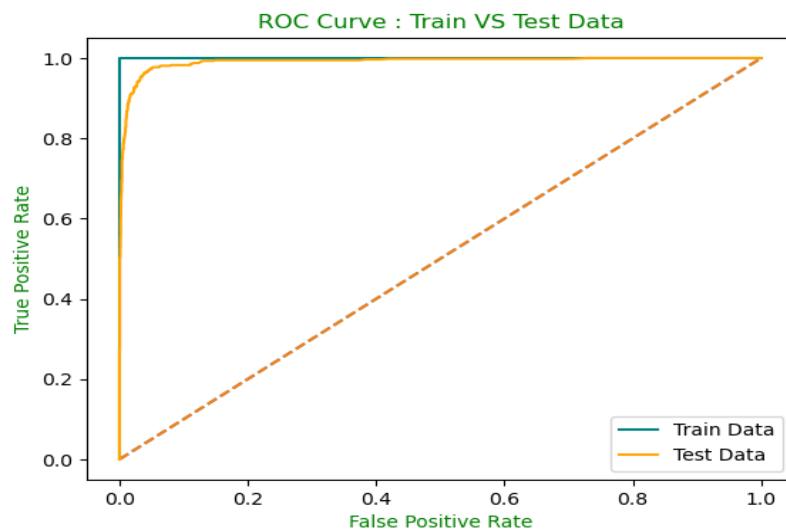


Fig no 87: ROC Curve RF smote

❖ Observations:

- AUC for Train data is 100% and AUC for Test data is 99%.
- The AUC & ROC curve shows that it is covering almost the same area for Train and Test data. Therefore, this is considered a Generalized good model.
- Also, the AUC score is very high and is able to distinguish between Positive and Negative classes very well.

Applying GridSearchCv on Random Forest

```
GridSearchCV
GridSearchCV(cv=5, estimator=RandomForestClassifier(random_state=42),
            param_grid={'max_depth': [5, 10, 15, 20],
                        'max_features': ['sqrt', 'log2'],
                        'min_samples_leaf': [5, 10, 15],
                        'min_samples_split': [10, 20, 30],
                        'n_estimators': [50, 100, 150]})

  estimator: RandomForestClassifier
    RandomForestClassifier(random_state=42)
      RandomForestClassifier
```

Building model on best parameters

```
RandomForestClassifier
RandomForestClassifier(max_depth=15, min_samples_leaf=5, min_samples_split=10,
                      n_estimators=50, random_state=42)
```

❖ Accuracy of Train and Test Data:

- Accuracy on Train data: 0.9691702613549861
- Accuracy on Test data: 0.9342806394316163

❖ Validness of models (over fitting or under fitting):

- The Accuracy of both the Train data and Test data are almost same; hence our model is Neither Overfit nor Underfit.

❖ Classification Report:

Classification Report of the Training data:					
	precision	recall	f1-score	support	
0	0.97	1.00	0.98	6555	
1	0.98	0.83	0.90	1327	
accuracy			0.97	7882	
macro avg	0.97	0.91	0.94	7882	
weighted avg	0.97	0.97	0.97	7882	

Classification Report of the Test data:					
	precision	recall	f1-score	support	
0	0.94	0.99	0.96	2809	
1	0.90	0.68	0.78	569	
accuracy			0.93	3378	
macro avg	0.92	0.83	0.87	3378	
weighted avg	0.93	0.93	0.93	3378	

❖ **Interpretations:**

-----Test set Tuned Model-----

➤ **Accuracy:**

- The Accuracy of model is 93%.
- 93% of the Data points are identified correctly.

➤ **Recall:** Represents the correctness of the Model in a particular class.

- Recall for class 1(Churn) is 68% and recall for class 0(non-churn) is 99%.
- 68% of those Churned were correctly identified as Churned by the model.
- Out of all the customers who are Churned, 68% of them are identified correctly as Churned.
- Out of all the customers who are not Churned, 99% of them are identified correctly as Not-Churned.

➤ **Precision:**

- Precision for class 1(Churn) is 90% and for class 0(non-churn) is 94%.
- 90% of the customers predicted are actually Churned Out of all the Customers that are predicted as Churned.
- 94% of the Customers predicted are actually Not Churned out of all the customers predicted as not Churned.

❖ **Confusion Matrix:**

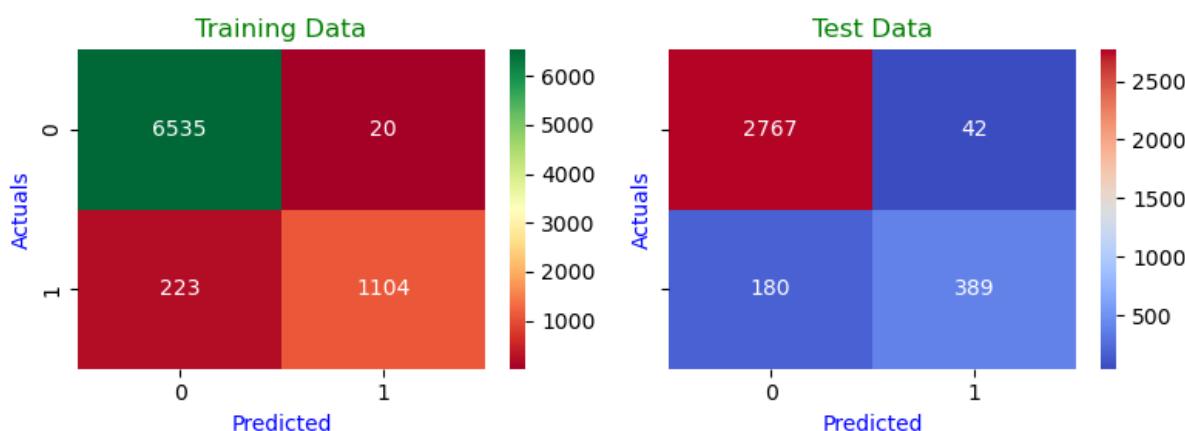


Fig no 88: Confusion Matrix RF Tuned

❖ **Measuring and Plotting AUC-ROC Curve:**

AUC for the Training Data: 0.997
AUC for the Test Data: 0.976

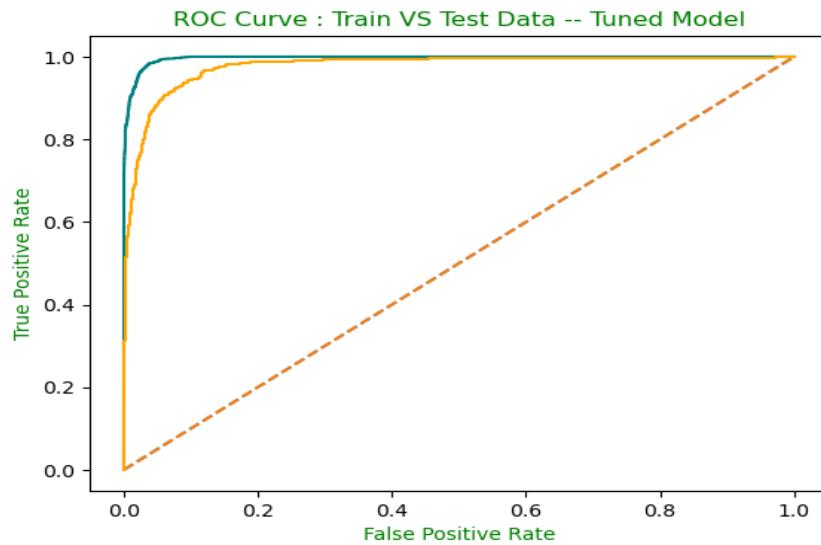


Fig no 89: ROC Curve RF Tuned

❖ **Observations:**

- AUC for Train data is 99% and AUC for Test data is 97%.
- The AUC & ROC curve shows that it is covering almost the same area for Train and Test data. Therefore, this is considered a Generalized good model.
- Also, the AUC score is very high and is able to distinguish between Positive and Negative classes very well.

❖ **Feature Importance:**

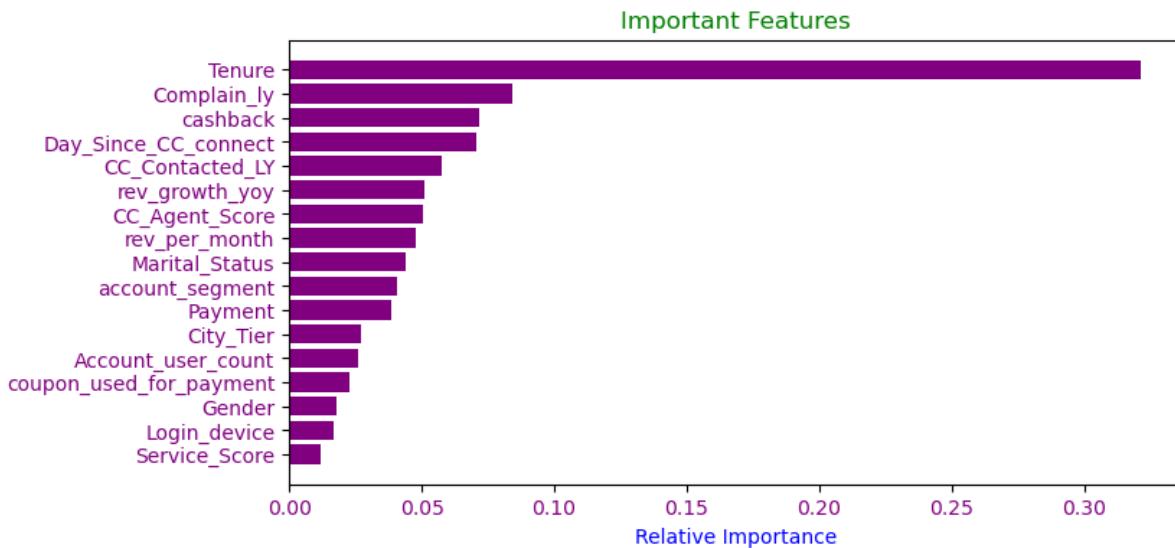


Fig no 90: Feature Imp RF Tuned

- The 1st Five Important Features of Random Forest Tuned model are: Tenure, Complain_ly, cashback, Day_Since_CC_connect and CC_Contacted_LY.

❖ **Observations:**

- Random Forest Base and Smote model are giving better Accuracy and AUC Score than Tuned model.

Bagging Classifier

```
BaggingClassifier
BaggingClassifier(random_state=42)
```

❖ Accuracy of Train and Test Dataset:

Accuracy of Train Data: 0.9965744734838874

Accuracy of Test Data: 0.9600355239786856

❖ Validness of models (over fitting or under fitting):

- The Accuracy of both the Train data and Test data are almost same; hence our model is Neither Overfit nor Underfit.

❖ Classification Report:

Classification Report of the Training data:

	precision	recall	f1-score	support
0	1.00	1.00	1.00	6555
1	1.00	0.98	0.99	1327
accuracy			1.00	7882
macro avg	1.00	0.99	0.99	7882
weighted avg	1.00	1.00	1.00	7882

Classification Report of the Test data:

	precision	recall	f1-score	support
0	0.97	0.99	0.98	2809
1	0.93	0.83	0.87	569
accuracy			0.96	3378
macro avg	0.95	0.91	0.93	3378
weighted avg	0.96	0.96	0.96	3378

❖ Interpretations:

-----Test set-----

- Accuracy:
 - The Accuracy of model is 96%.
 - 96% of the Data points are identified correctly.
- Recall: Represents the correctness of the Model in a particular class.
 - Recall for class 1(Churn) is 83% and recall for class 0(non-churn) is 99%.
 - 83% of those Churned were correctly identified as Churned by the model.

- Out of all the customers who are Churned, 83% of them are identified correctly as Churned.
- Out of all the customers who are not Churned, 99% of them are identified correctly as Not-Churned.

➤ **Precision:**

- Precision for class 1(Churn) is 93% and for class 0(non-churn) is 97%.
- 93% of the customers predicted are actually Churned Out of all the Customers that are predicted as Churned.
- 97% of the Customers predicted are actually Not Churned out of all the customers predicted as not Churned.

❖ **Confusion Matrix:**

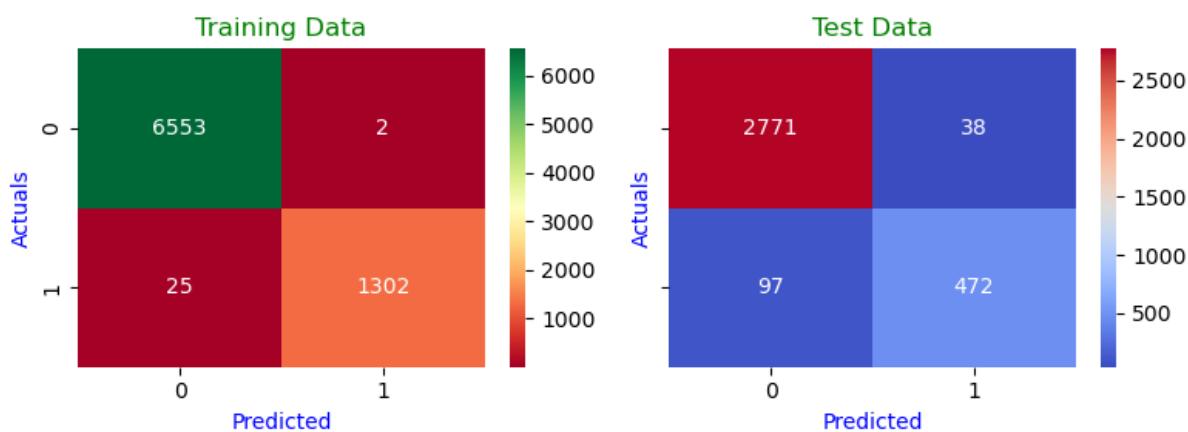


Fig no 91: Confusion Matrix Bagging

❖ **AUC -ROC Curve:**

- AUC Train: 1.000
- AUC Test: 0.983

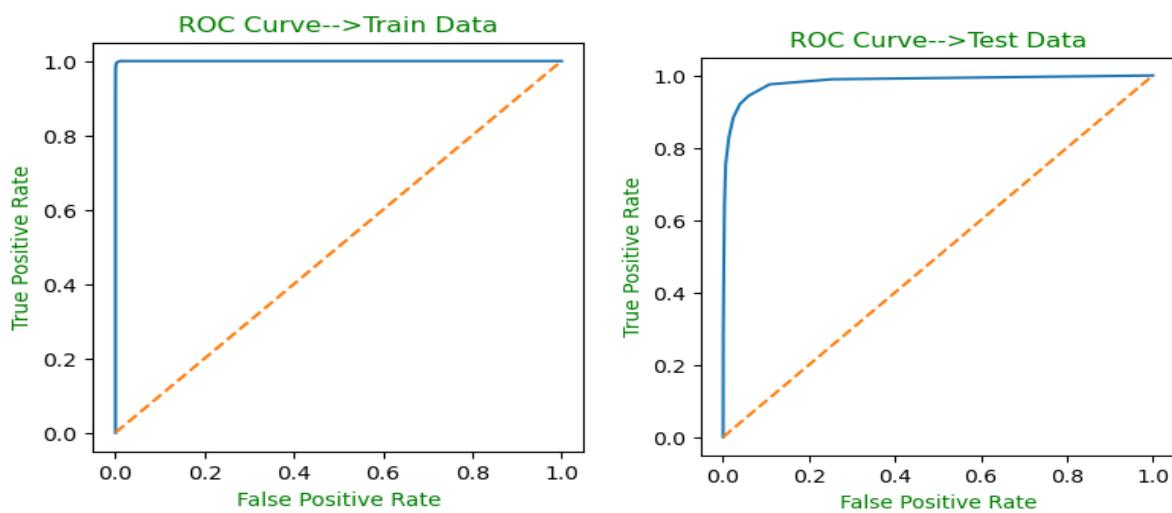


Fig no 92: ROC Curve Bagging

❖ **Observations:**

- AUC for Train data is 100% and AUC for Test data is 98%.
- The AUC & ROC curve shows that it is covering almost the same area for Train and Test data. Therefore, this is considered a Generalized good model.
- Also, the AUC score is very high and is able to distinguish between Positive and Negative classes very well.

Bagging on Smote dataset

```
BaggingClassifier
BaggingClassifier(random_state=42)
```

❖ **Accuracy of Train and Test Dataset:**

- Accuracy of Train Data: 0.9985180019178799
- Accuracy of Test Data: 0.9585553582001184

❖ **Validness of models (over fitting or under fitting):**

- There is not much difference between Train and Test Accuracy, hence our model is Neither Overfit nor Underfit.

❖ **Classification Report:**

Classification Report of the Training data:				
	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

Classification Report of the Test data:				
	precision	recall	f1-score	support
0	0.97	0.98	0.98	2809
1	0.90	0.85	0.87	569
accuracy			0.96	3378
macro avg	0.93	0.92	0.92	3378
weighted avg	0.96	0.96	0.96	3378

❖ **Interpretations:**

-----Test set Smote Data-----

- **Accuracy:**
 - The Accuracy of model is 96%.
 - 96% of the Data points are identified correctly.

- **Recall:** Represents the correctness of the Model in a particular class.
 - Recall for class 1(Churn) is 85% and recall for class 0(non-churn) is 98%.
 - 85% of those Churned were correctly identified as Churned by the model.
 - Out of all the customers who are Churned, 85% of them are identified correctly as Churned.
 - Out of all the customers who are not Churned, 98% of them are identified correctly as Not-Churned.

- **Precision:**
 - Precision for class 1(Churn) is 90% and for class 0(non-Churn) is 97%.
 - 90% of the customers predicted are actually Churned Out of all the Customers that are predicted as Churned.
 - 97% of the Customers predicted are actually Not Churned out of all the customers predicted as not Churned.

❖ **Confusion Matrix:**

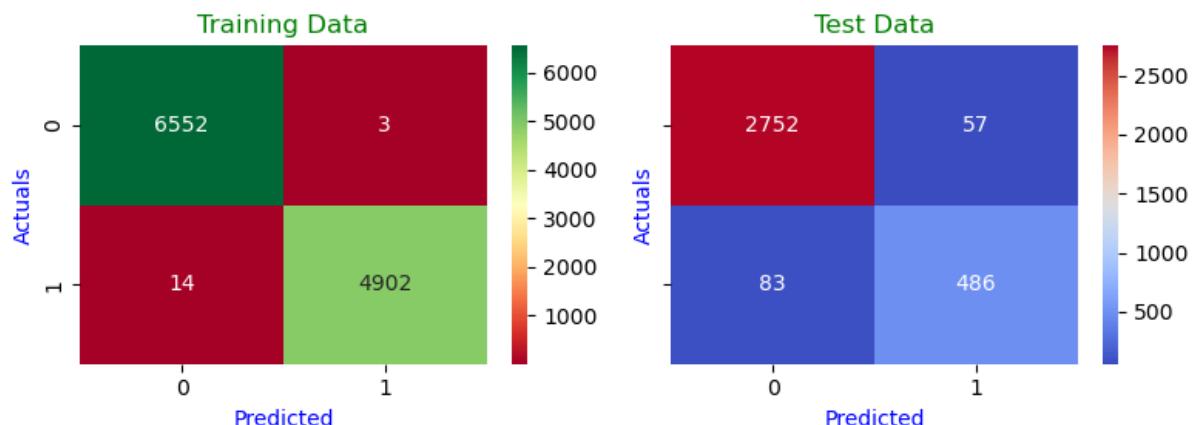


Fig no 93: Confusion Matrix Bagging smote

❖ **Measuring and Plotting AUC-ROC Curve:**

- AUC for the Training Data: 1.000
- AUC for the Test Data: 0.982

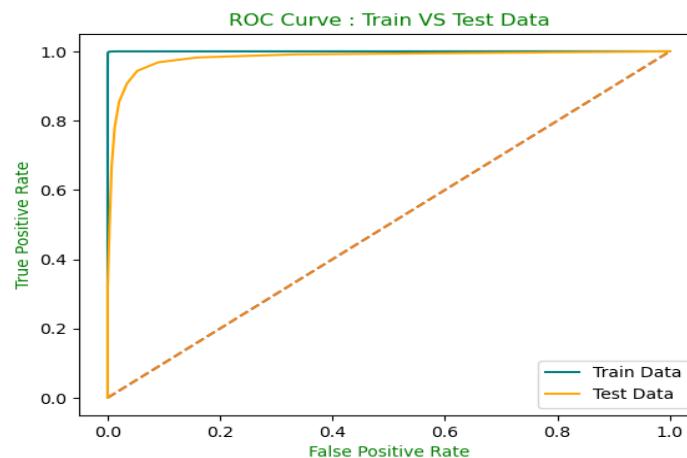


Fig no 94: ROC Curve Bagging smote

❖ Observations:

- AUC for Train data is 100% and AUC for Test data is 98%.
- The AUC & ROC curve shows that it is covering almost the same area for Train and Test data. Therefore, this is considered a Generalized good model.
- Also, the AUC score is very high and is able to distinguish between Positive and Negative classes very well.

Applying GridSearchCV on Bagging

```

GridSearchCV
GridSearchCV(cv=5, estimator=BaggingClassifier(random_state=42),
             param_grid={'max_features': array([0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9]),
                         'max_samples': array([0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9]),
                         'n_estimators': [40, 50, 60]})
    - estimator: BaggingClassifier
        BaggingClassifier(random_state=42)
            - BaggingClassifier

```

Building Model on Best Parameter

```

BaggingClassifier
BaggingClassifier(max_features=0.9000000000000001,
                  max_samples=0.8000000000000003, n_estimators=60,
                  random_state=42)

```

❖ Accuracy of Train and Test Data:

- Accuracy on Train Data: 0.9994925145902055
- Accuracy on Test Data: 0.9659561870929544

❖ Classification Report:

Classification Report of the Training data:				
	precision	recall	f1-score	support
0	1.00	1.00	1.00	6555
1	1.00	1.00	1.00	1327
accuracy			1.00	7882
macro avg	1.00	1.00	1.00	7882
weighted avg	1.00	1.00	1.00	7882

Classification Report of the Test data:				
	precision	recall	f1-score	support
0	0.97	0.99	0.98	2809
1	0.97	0.82	0.89	569
accuracy			0.97	3378
macro avg	0.97	0.91	0.94	3378
weighted avg	0.97	0.97	0.96	3378

❖ Interpretations:

-----Test set Tuned Model-----

- Accuracy:

- The Accuracy of model is 97%.
 - 97% of the Data points are identified correctly.
- **Recall:** Represents the correctness of the Model in a particular class.
- Recall for class 1(Churn) is 82% and recall for class 0(non-churn) is 99%.
 - 82% of those Churned were correctly identified as Churned by the model.
 - Out of all the customers who are Churned, 82% of them are identified correctly as Churned.
 - Out of all the customers who are not Churned, 99% of them are identified correctly as Not-Churned.
- **Precision:**
- Precision for class 1(Churn) is 97% and for class 0(non-churn) is 97%.
 - 97% of the customers predicted are actually Churned Out of all the Customers that are predicted as Churned.
 - 97% of the Customers predicted are actually Not Churned out of all the customers predicted as not Churned.

❖ **Confusion Matrix:**

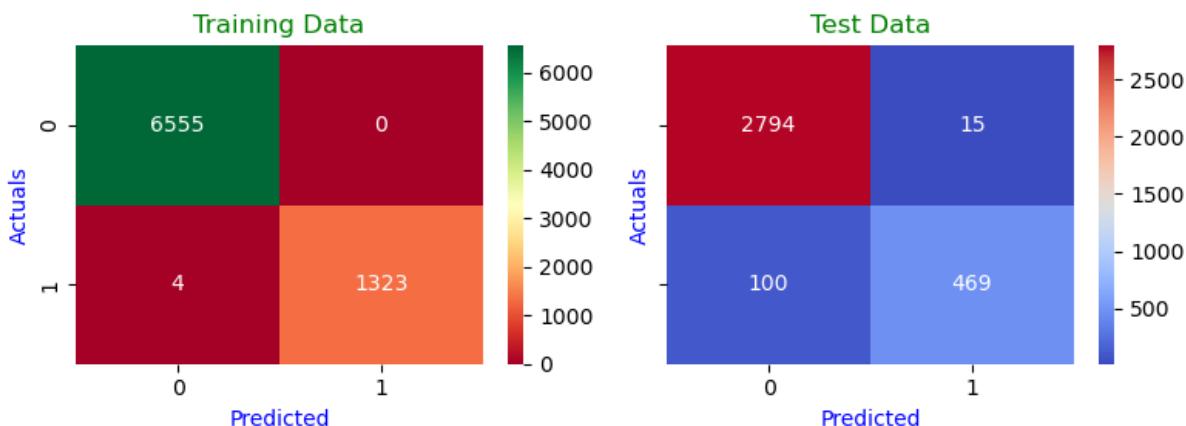


Fig no 95: Confusion Matrix Bagging Tuned

❖ **Measuring and Plotting AUC-ROC Curve:**

- AUC for the Training Data: 1.000
- AUC for the Test Data: 0.991

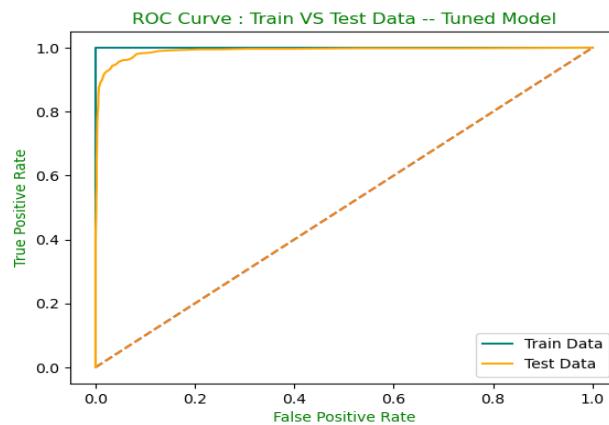


Fig no 96: ROC Curve Bagging Tuned

❖ **Observations:**

- AUC for Train data is 100% and AUC for Test data is 99%.
- The AUC & ROC curve shows that it is covering the same area for Train and Test data. Therefore, this is considered a Generalized good model.
- Also, the AUC score is very high and is able to distinguish between Positive and Negative classes very well.

Observations:

- Accuracy is same for Base, Smote and Tuned model.
- AUC score is better for Tuned model.

AdaBoost Classifier

```
AdaBoostClassifier
AdaBoostClassifier(n_estimators=500, random_state=42)
```

❖ **Accuracy of Train and Test data:**

- Accuracy on Train Data: 0.903577772139051
- Accuracy on Test Data: 0.9011249259917111

❖ **validness of models (over fitting or under fitting):**

- The Accuracy of both the Train data and Test data are almost same; hence our model is Neither Overfit nor Underfit.

❖ **Classification Report:**

Classification Report of the Training data:				
	precision	recall	f1-score	support
0	0.92	0.96	0.94	6555
1	0.77	0.61	0.68	1327
accuracy			0.90	7882
macro avg	0.85	0.79	0.81	7882
weighted avg	0.90	0.90	0.90	7882

Classification Report of the Test data:				
	precision	recall	f1-score	support
0	0.92	0.96	0.94	2809
1	0.76	0.60	0.67	569
accuracy			0.90	3378
macro avg	0.84	0.78	0.81	3378
weighted avg	0.90	0.90	0.90	3378

❖ **Interpretations:**

-----Test set-----

- **Accuracy:**
 - The Accuracy of model is 90%.
 - 90% of the Data points are identified correctly.
- **Recall:** Represents the correctness of the Model in a particular class.
 - Recall for class 1(Churn) is 60% and recall for class 0(non-churn) is 96%.
 - 60% of those Churned were correctly identified as Churned by the model.
 - Out of all the customers who are Churned, 60% of them are identified correctly as Churned.
 - Out of all the customers who are not Churned, 96% of them are identified correctly as Not-Churned.
- **Precision:**
 - Precision for class 1(Churn) is 76% and for class 0(non-churn) is 92%.
 - 76% of the customers predicted are actually Churned Out of all the Customers that are predicted as Churned.
 - 92% of the Customers predicted are actually Not Churned out of all the customers predicted as not Churned.

❖ **Confusion Matrix:**

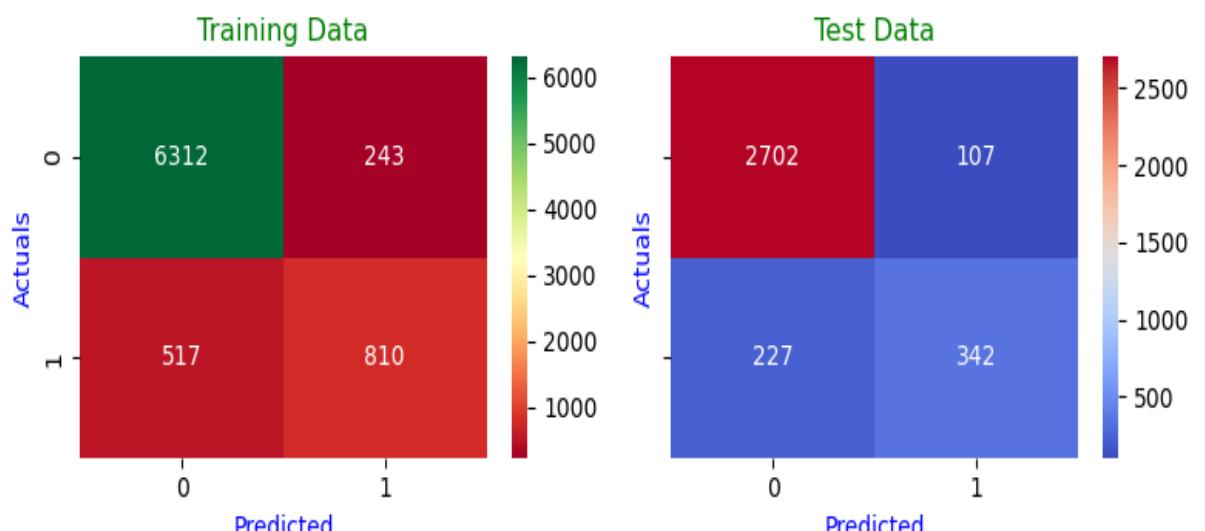


Fig no 97: Confusion Matrix Ada Boost

❖ **AUC - ROC Curve:**

- AUC Train: 0.930
- AUC Test: 0.916

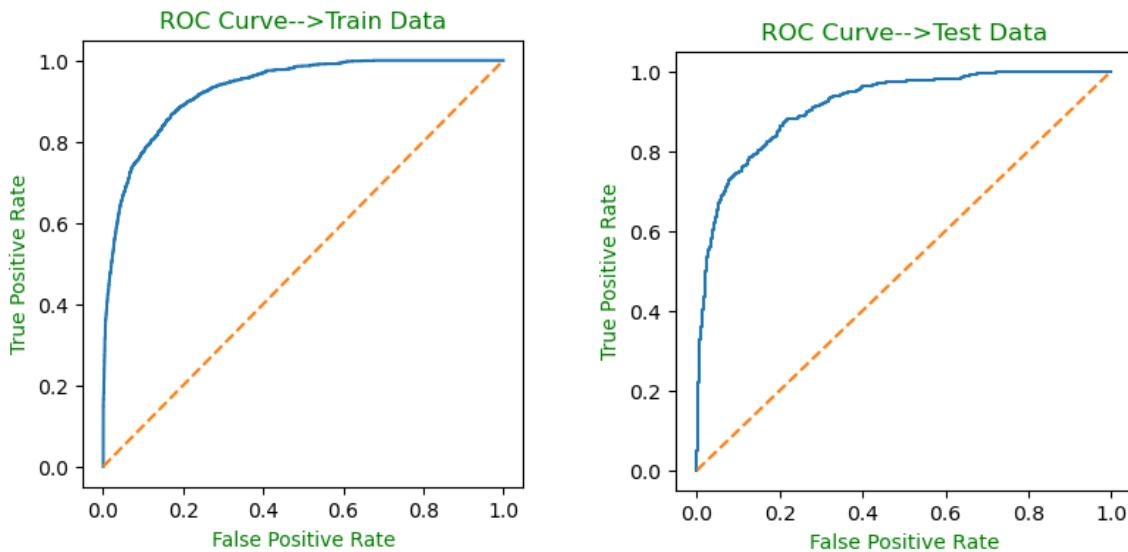


Fig no 98: ROC Curve Ada Boost

❖ **Observations:**

- AUC for Train data is 93% and AUC for Test data is 91%.
- The AUC & ROC curve shows that it is covering almost the same area for Train and Test data. Therefore, this is considered a Generalized good model.
- Also, the AUC score is very high and is able to distinguish between Positive and Negative classes very well.

❖ **Feature Importance:**

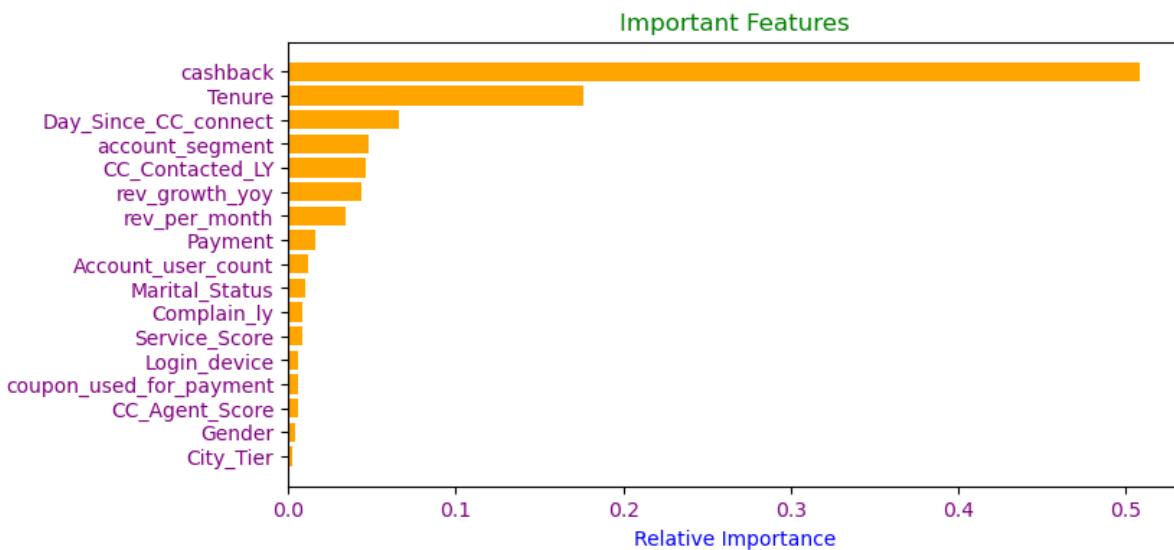


Fig no 99: Feature Imp Ada Boost

- The 1st Five Important features for Ada Boost model are: cashback, Tenure, Day_Since_CC_connect, account_segment and CC_Contacted_LY.

Ada Boost on Smote Dataset

```
AdaBoostClassifier
AdaBoostClassifier(n_estimators=1000, random_state=42)
```

❖ Accuracy of the Training and Test data:

- Accuracy on Train Data: 0.9242437450963299
- Accuracy on Test Data: 0.8946121965660154

❖ validness of models (over fitting or under fitting):

- There is not much difference between Accuracy of the Train data and Test data, hence our model is Neither Overfit nor Underfit.

❖ Classification Report:

Classification Report of the Training data:				
	precision	recall	f1-score	support
0	0.92	0.95	0.93	6555
1	0.93	0.89	0.91	4916
accuracy			0.92	11471
macro avg	0.93	0.92	0.92	11471
weighted avg	0.92	0.92	0.92	11471

Classification Report of the Test data:				
	precision	recall	f1-score	support
0	0.93	0.95	0.94	2809
1	0.71	0.64	0.67	569
accuracy			0.89	3378
macro avg	0.82	0.79	0.80	3378
weighted avg	0.89	0.89	0.89	3378

❖ Interpretations:

-----Test set Smote data-----

- Accuracy:
 - The Accuracy of model is 89%.
 - 89% of the Data points are identified correctly.
- Recall: Represents the correctness of the Model in a particular class.
 - Recall for class 1(Churn) is 64% and recall for class 0(non-churn) is 95%.
 - 64% of those Churned were correctly identified as Churned by the model.
 - Out of all the customers who are Churned, 64% of them are identified correctly as Churned.
 - Out of all the customers who are not Churned, 95% of them are identified correctly as Not-Churned.

- **Precision:**
 - Precision for class 1(Churn) is 71% and for class 0(non-churn) is 93%.
 - 71% of the customers predicted are actually Churned Out of all the Customers that are predicted as Churned.
 - 93% of the Customers predicted are actually Not Churned out of all the customers predicted as not Churned.

❖ **Confusion Matrix:**

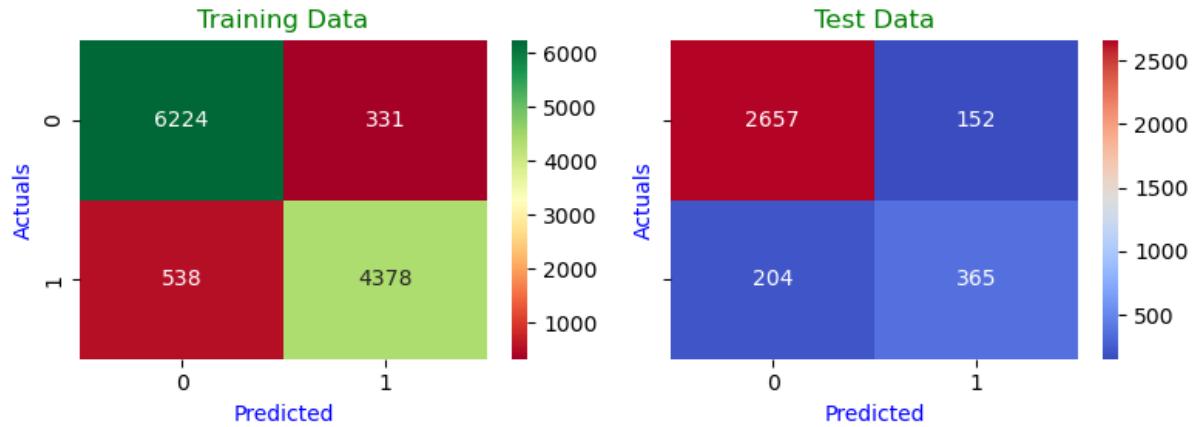


Fig no 100: Confusion matrix Ada Boost smote

❖ **Measuring and Plotting AUC-ROC Curve:**

- AUC for the Training Data: 0.977
- AUC for the Test Data: 0.912

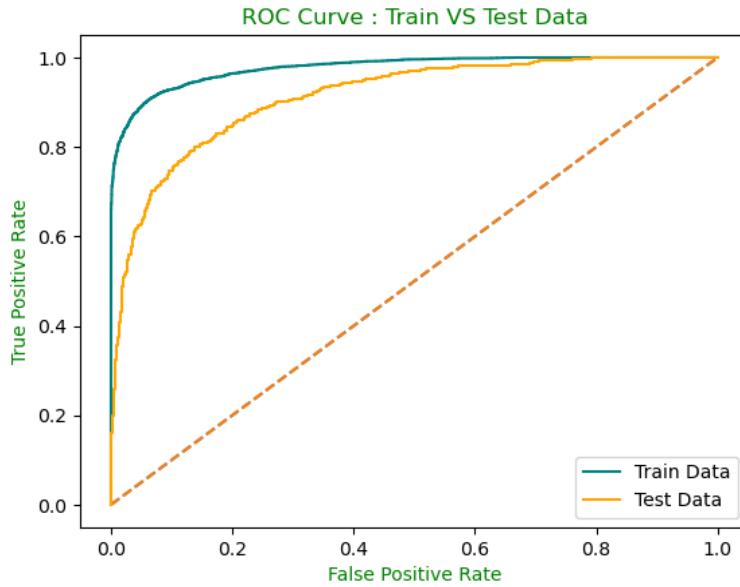


Fig no 101: ROC Curve Ada Boost smote

❖ **Observations:**

- AUC for Train data is 97% and AUC for Test data is 91%.
- From the AUC values & ROC curve for the Train and Test data, it shows that Test data covering a slightly less area as compared to the train data.
- But since AUC is greater than 90% so it is considered a Good Model.

- The AUC score is high and is able to distinguish between Positive and Negative classes very well.

GridSearchCv on ADA Boost

```
GridSearchCV
GridSearchCV(cv=5,
    estimator=AdaBoostClassifier(n_estimators=500, random_state=42),
    param_grid={'algorithm': ['SAMME', 'SAMME.R'],
                'learning_rate': [0.1, 0.01, 0.001, 1.0],
                'n_estimators': [50, 100, 500, 1000]})

    estimator: AdaBoostClassifier
        AdaBoostClassifier(n_estimators=500, random_state=42)
            AdaBoostClassifier
```

❖ Accuracy of Train and Test Data:

- Accuracy on Train Data: 0.8981223039837605
- Accuracy on Test Data: 0.8978685612788633

❖ Validness of models (over fitting or under fitting):

- The Accuracy of both the Train data and Test data are almost same; hence our model is Neither Overfit nor Underfit.

❖ Classification Report:

Classification Report of the Training data:				
	precision	recall	f1-score	support
0	0.92	0.96	0.94	6555
1	0.75	0.59	0.66	1327
accuracy			0.90	7882
macro avg	0.84	0.78	0.80	7882
weighted avg	0.89	0.90	0.89	7882

Classification Report of the Test data:

	precision	recall	f1-score	support
0	0.92	0.96	0.94	2809
1	0.75	0.59	0.66	569
accuracy			0.90	3378
macro avg	0.84	0.78	0.80	3378
weighted avg	0.89	0.90	0.89	3378

❖ Interpretations:

-----Test set Tuned Model-----

➤ Accuracy:

- The Accuracy of model is 90%.
- 90% of the Data points are identified correctly.

- **Recall:** Represents the correctness of the Model in a particular class.
 - Recall for class 1(Churn) is 59% and recall for class 0(non-churn) is 96%.
 - 59% of those Churned were correctly identified as Churned by the model.
 - Out of all the customers who are Churned, 59% of them are identified correctly as Churned.
 - Out of all the customers who are not Churned, 96% of them are identified correctly as Not-Churned.

- **Precision:**
 - Precision for class 1(Churn) is 75% and for class 0(non-churn) is 92%.
 - 75% of the customers predicted are actually Churned Out of all the Customers that are predicted as Churned.
 - 92% of the Customers predicted are actually Not Churned out of all the customers predicted as not Churned.

❖ **Confusion Matrix:**

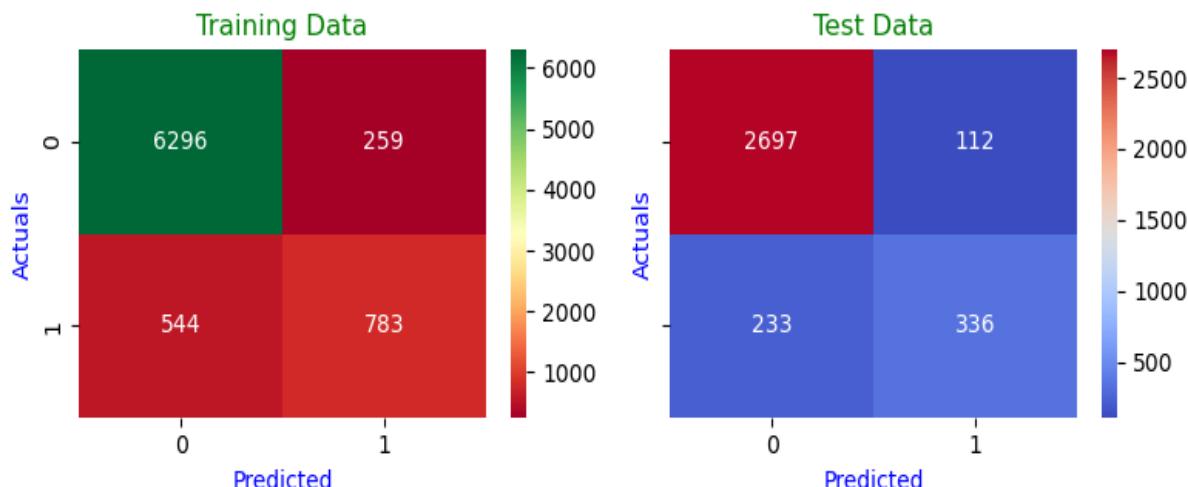


Fig no 102: Confusion Matrix Ada Boost Tuned

❖ **Measuring and Plotting AUC-ROC Curve:**

- AUC for the Training Data: 0.915
- AUC for the Test Data: 0.911

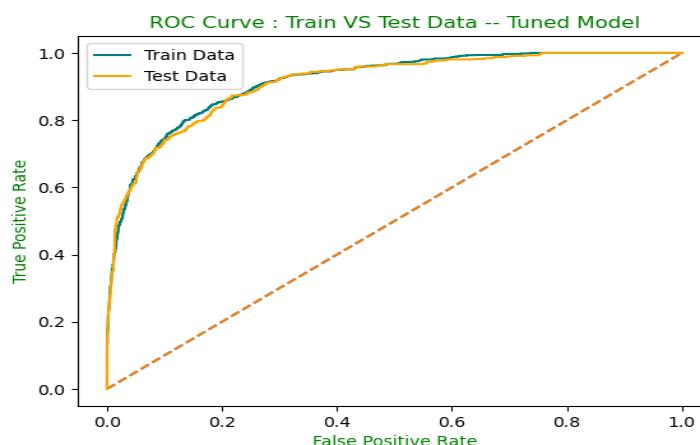


Fig no 103: ROC Curve Ada Boost Tuned

❖ **Observations:**

- ❖ AUC for Train data is 91% and AUC for Test data is 91%.
- ❖ The AUC & ROC curve shows that it is covering the same area for Train and Test data. Therefore, this is considered a Generalized good model.
- ❖ Also, the AUC score is very high and is able to distinguish between Positive and Negative classes very well.

❖ **Feature Importance:**

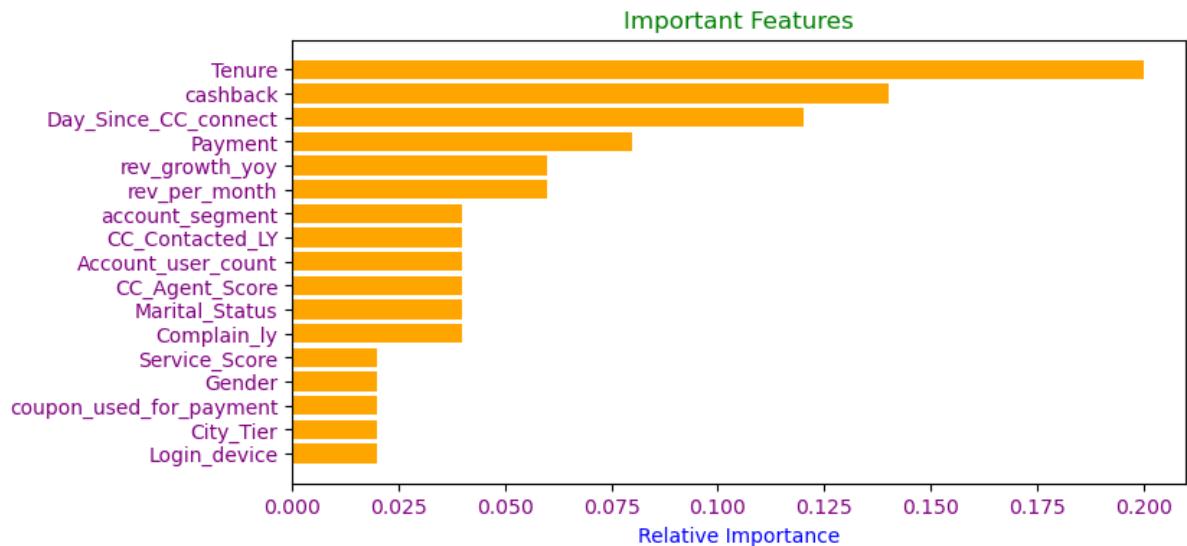


Fig no 104: Feature Imp Ada Boost Tuned

- The 1st Five features of ADB Tuned Model are: Tenure, cashback, Day_Since_CC_connect, payment, rev_growth_yoy and rev_per_month.

Gradient Boost Classifier

```
GradientBoostingClassifier
GradientBoostingClassifier(random_state=42)
```

❖ **Accuracy of Train and Test data:**

- Accuracy on Train Data: 0.9185485917279879
- Accuracy on Test Data: 0.9094138543516874

❖ **validness of models (over fitting or under fitting):**

- The Accuracy of both the Train data and Test data are almost same; hence our model is Neither Overfit nor Underfit.

❖ **Classification Report:**

Classification Report of the Training data:				
	precision	recall	f1-score	support
0	0.93	0.98	0.95	6555
1	0.84	0.64	0.72	1327
			0.92	7882
accuracy		0.89	0.81	7882
macro avg		0.91	0.92	7882
weighted avg		0.91	0.92	7882

Classification Report of the Test data:				
	precision	recall	f1-score	support
0	0.92	0.97	0.95	2809
1	0.82	0.59	0.69	569
			0.91	3378
accuracy		0.87	0.78	3378
macro avg		0.90	0.91	3378
weighted avg		0.90	0.91	3378

❖ Interpretations:

-----Test set-----

❖ Accuracy:

- The Accuracy of model is 91%.
- 91% of the Data points are identified correctly.

❖ Recall: Represents the correctness of the Model in a particular class.

- Recall for class 1(Churn) is 59% and recall for class 0(non-churn) is 97%.
- 59% of those Churned were correctly identified as Churned by the model.
- Out of all the customers who are Churned, 59% of them are identified correctly as Churned.
- Out of all the customers who are not Churned, 97% of them are identified correctly as Not-Churned.

❖ Precision:

- Precision for class 1(Churn) is 82% and for class 0(non-churn) is 92%.
- 82% of the customers predicted are actually Churned Out of all the Customers that are predicted as Churned.
- 92% of the Customers predicted are actually Not Churned out of all the customers predicted as not Churned.

❖ Confusion Matrix:

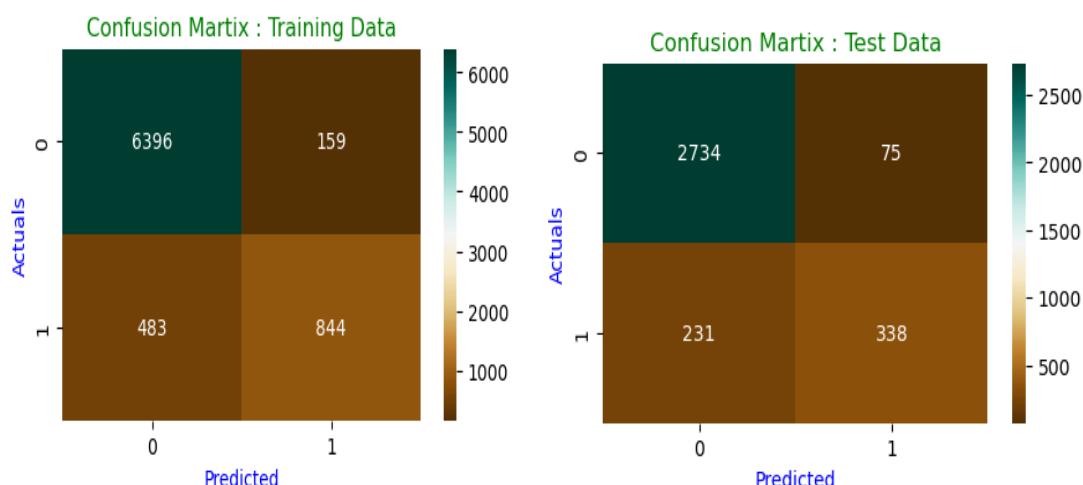


Fig no 105: Confusion Matrix GDB

❖ **AUC - ROC Curve:**

- AUC Train: 0.951
- AUC Test: 0.936

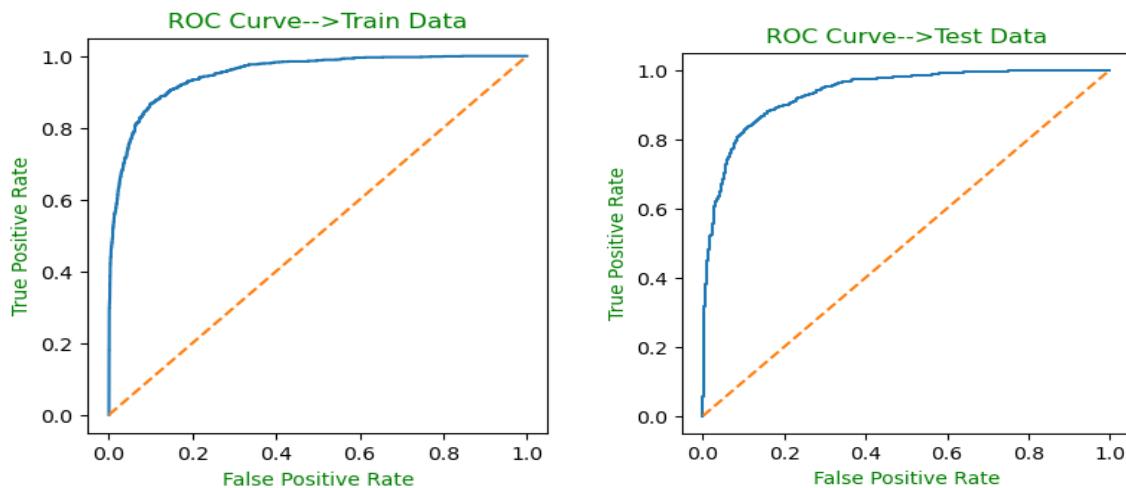


Fig no 106: ROC Curve GDB

❖ **Observations:**

- AUC for Train data is 95% and AUC for Test data is 93%.
- The AUC score is very high and is able to distinguish between Positive and Negative classes very well.
- From the AUC values & ROC curve for the Train and Test data, it shows that Test data is covering a large area as compared to the train data.

❖ **Feature Importance:**

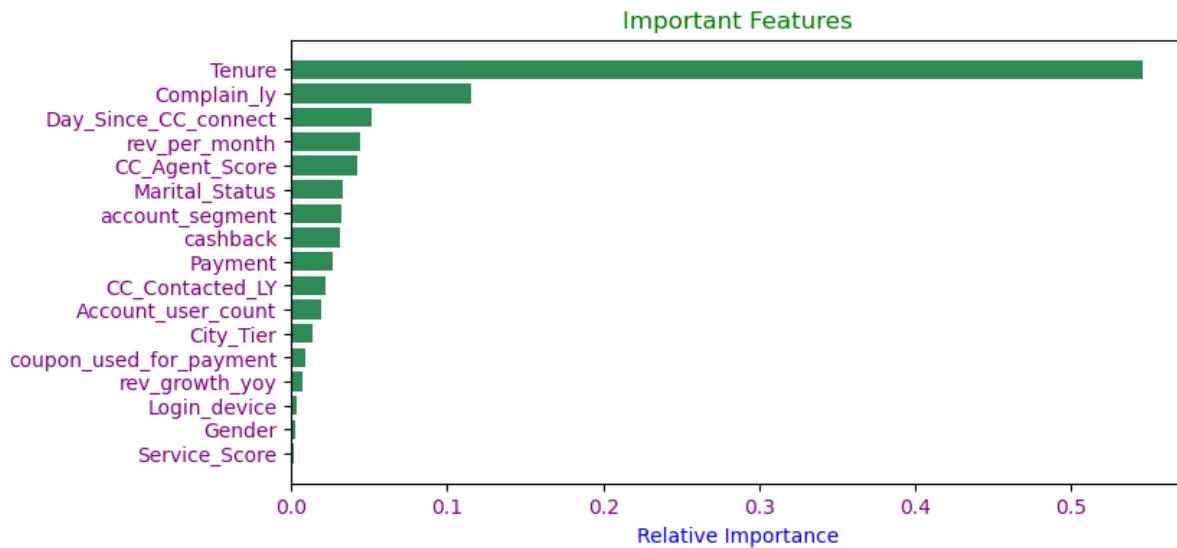


Fig no 107: Feature Imp GDB

- The 1st five important features for Gradient Boost model are: Tenure, Complain_ly, Day_Since_CC_connect, CC_Agent_Score and rev_per_month.

Gradient Boosting on balanced (Smote) dataset

```

    GradientBoostingClassifier
    GradientBoostingClassifier(random_state=42)

```

❖ Accuracy of Train and Test data:

- Accuracy on Train Data: 0.9158748147502397
- Accuracy on Test Data: 0.9049733570159858

❖ validity of models (over fitting or under fitting):

- The Accuracy of both the Train data and Test data are almost same; hence our model is Neither Overfit nor Underfit.

❖ Classification Report:

Classification Report of the Training data:				
	precision	recall	f1-score	support
0	0.91	0.94	0.93	6555
1	0.92	0.88	0.90	4916
accuracy			0.92	11471
macro avg	0.92	0.91	0.91	11471
weighted avg	0.92	0.92	0.92	11471

Classification Report of the Test data:				
	precision	recall	f1-score	support
0	0.94	0.94	0.94	2809
1	0.72	0.72	0.72	569
accuracy			0.90	3378
macro avg	0.83	0.83	0.83	3378
weighted avg	0.91	0.90	0.91	3378

❖ Interpretations:

-----Test set Smote Data-----

❖ Accuracy:

- The Accuracy of model is 90%.
- 90% of the Data points are identified correctly.

❖ Recall: Represents the correctness of the Model in a particular class.

- Recall for class 1(Churn) is 72% and recall for class 0(non-churn) is 94%.
- 72% of those Churned were correctly identified as Churned by the model.
- Out of all the customers who are Churned, 72% of them are identified correctly as Churned.
- Out of all the customers who are not Churned, 94% of them are identified correctly as Not-Churned.

❖ Precision:

- Precision for class 1(Churn) is 72% and for class 0(non-churn) is 94%.
- 72% of the customers predicted are actually Churned Out of all the Customers that are predicted as Churned.

- 94% of the Customers predicted are actually Not Churned out of all the customers predicted as not Churned.

❖ Confusion Matrix:

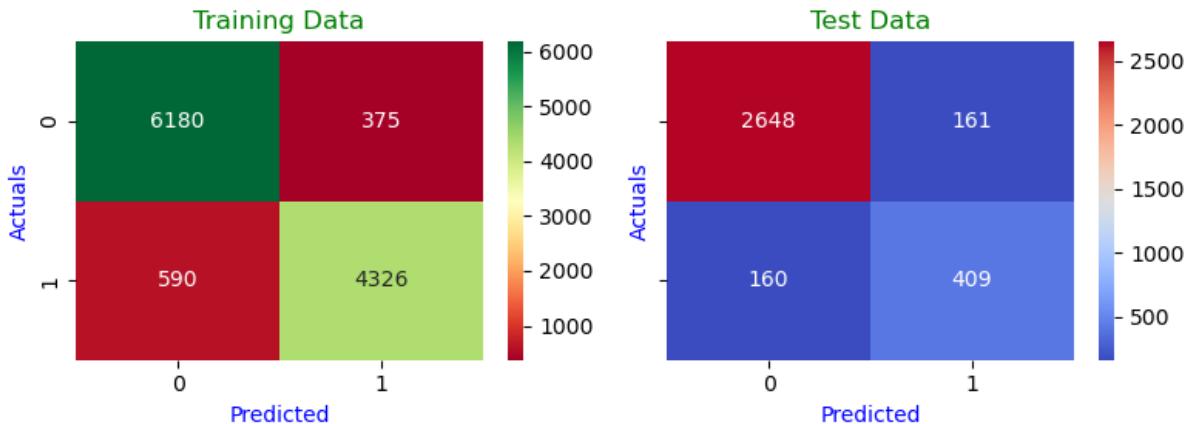


Fig no 108: Confusion Matrix GDB smote

❖ Measuring and Plotting AUC-ROC Curve:

- AUC for the Training Data: 0.975
- AUC for the Test Data: 0.931

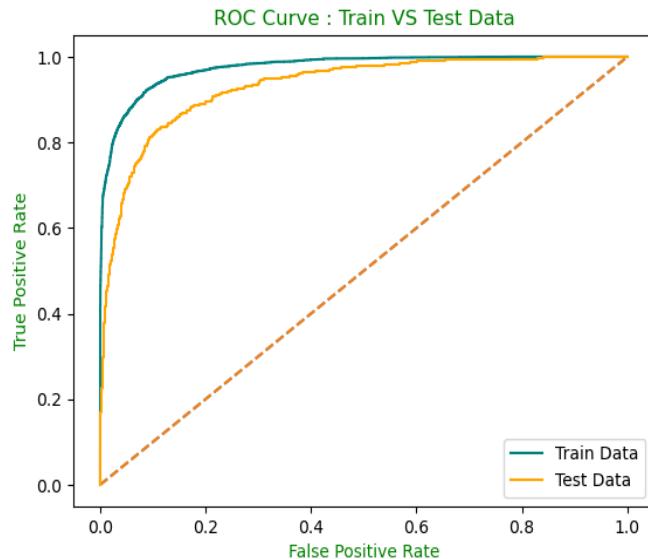


Fig no 109: ROC Curve GDB smote

❖ Observations:

- AUC for Train data is 97% and AUC for Test data is 93%.
- The AUC score is very high and is able to distinguish between Positive and Negative classes very well.
- From the AUC values & ROC curve for the Train and Test data, it shows that Test data is covering a slightly less area as compared to the train data.

GridSearchCV on Gradient Boost

```

        GridSearchCV
GridSearchCV(cv=5, estimator=GradientBoostingClassifier(random_state=42),
            param_grid={'criterion': ['friedman_mse', 'squared_error'],
                        'loss': ['log_loss', 'exponential'],
                        'max_depth': [3, 5], 'max_features': ['sqrt', 'log2'],
                        'max_leaf_nodes': [2, 5],
                        'n_estimators': [50, 100, 150, 200]})

    estimator: GradientBoostingClassifier
    GradientBoostingClassifier(random_state=42)
        GradientBoostingClassifier

```

Building Model on Best Parameter

```

        GradientBoostingClassifier
GradientBoostingClassifier(max_depth=5, max_features='sqrt', max_leaf_nodes=5,
                           n_estimators=200, random_state=42)

```

- ❖ **Accuracy of Train and Test Data:**
 - Accuracy on Train Data: 0.9196904339000254
 - Accuracy on Test Data: 0.9073416222616933
- ❖ **Validness of models (over fitting or under fitting):**
 - The Accuracy of both the Train data and Test data are almost same; hence our model is Neither Overfit nor Underfit.
- ❖ **Classification Report:**

Classification Report of the Training data:				
	precision	recall	f1-score	support
0	0.93	0.98	0.95	6555
1	0.85	0.64	0.73	1327
accuracy			0.92	7882
macro avg			0.89	0.84
weighted avg			0.92	0.91
Classification Report of the Test data:				
	precision	recall	f1-score	support
0	0.92	0.97	0.95	2809
1	0.81	0.59	0.68	569
accuracy			0.91	3378
macro avg			0.86	0.81
weighted avg			0.90	0.90

❖ **Interpretations:**

-----Test set Tuned Model-----

❖ **Accuracy:**

- The Accuracy of model is 91%.
- 91% of the Data points are identified correctly.

❖ **Recall:** Represents the correctness of the Model in a particular class.

- Recall for class 1(Churn) is 59% and recall for class 0(non-churn) is 97%.
- 59% of those Churned were correctly identified as Churned by the model.
- Out of all the customers who are Churned,59% of them are identified correctly as Churned.
- Out of all the customers who are not Churned, 97% of them are identified correctly as Not-Churned.

❖ **Precision:**

- Precision for class 1(Churn) is 81% and for class 0(non-churn) is 92%.
- 81% of the customers predicted are actually Churned Out of all the Customers that are predicted as Churned.
- 92% of the Customers predicted are actually Not Churned out of all the customers predicted as not Churned.

❖ **Confusion Matrix:**

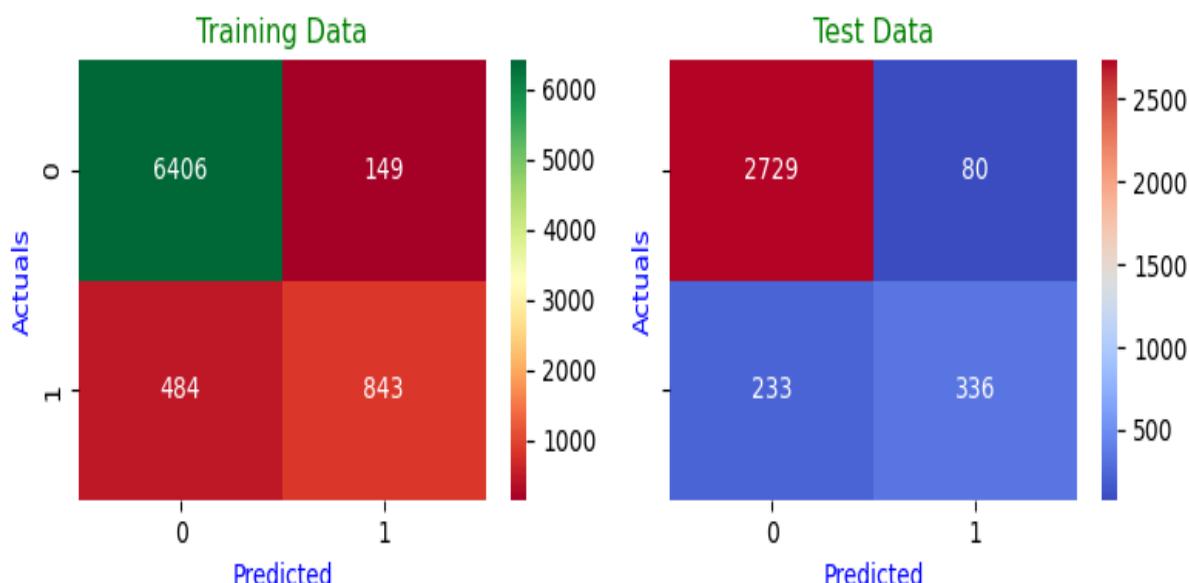


Fig no 110: Confusion Matrix GDB Tuned

❖ **Measuring and Plotting AUC-ROC Curve:**

- AUC for the Training Data: 0.953
- AUC for the Test Data: 0.937

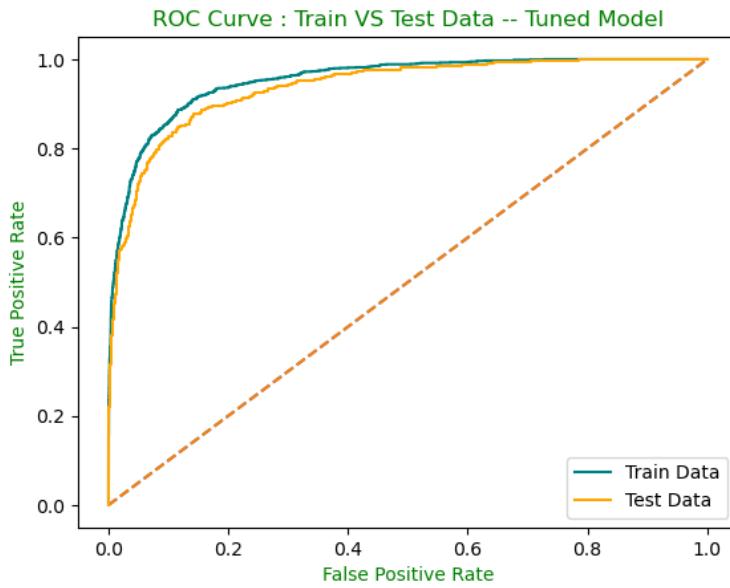


Fig no 111: ROC Curve GDB Tuned

❖ **Observations:**

- AUC for Train data is 95% and AUC for Test data is 93%.
- The AUC & ROC curve shows that it is covering almost the same area for Train and Test data. Therefore, this is considered a Generalized good model.
- Also, the AUC score is very high and is able to distinguish between Positive and Negative classes very well.

❖ **Feature Importance:**

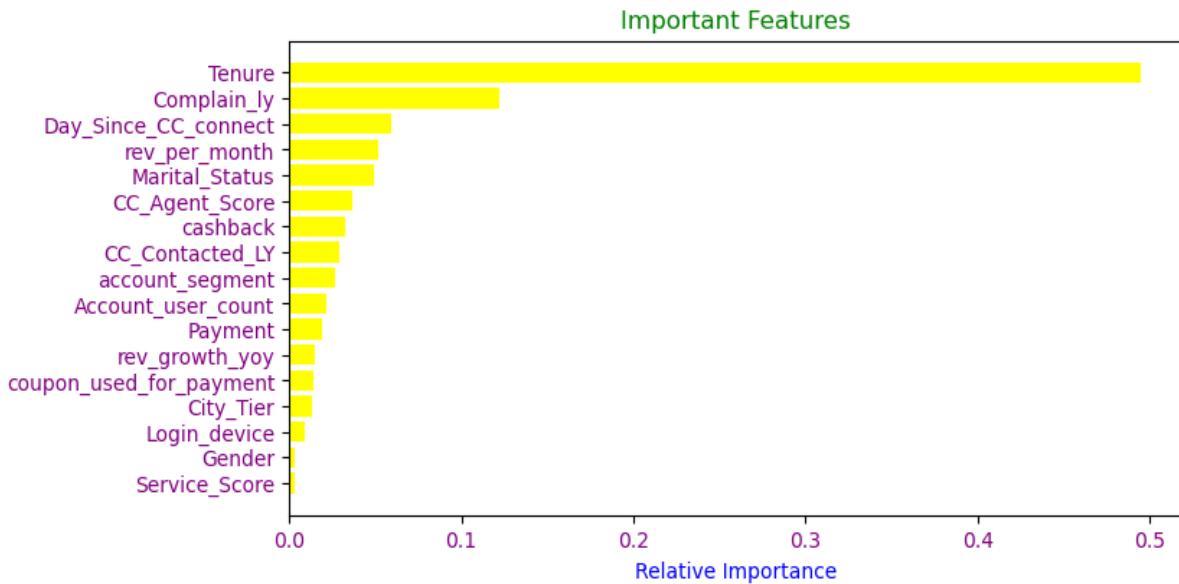


Fig no 112: Feature Imp GDB Tuned

- The 1st five important features for Gradient Boost Tuned model are: Tenure, Complain_ly, Day_Since_CC_connect, CC_Agent_Score and rev_per_month.

XGBoost Classifier

```
XGBClassifier
XGBClassifier(base_score=None, booster=None, callbacks=None,
              colsample_bylevel=None, colsample_bynode=None,
              colsample_bytree=None, device=None, early_stopping_rounds=None,
              enable_categorical=False, eval_metric='logloss',
              feature_types=None, gamma=None, grow_policy=None,
              importance_type=None, interaction_constraints=None,
              learning_rate=None, max_bin=None, max_cat_threshold=None,
              max_cat_to_onehot=None, max_delta_step=None, max_depth=None,
              max_leaves=None, min_child_weight=None, missing=nan,
              monotone_constraints=None, multi_strategy=None, n_estimators=None, v
```

❖ Accuracy of Train and Test data:

- Accuracy on Train Data: 0.9998731286475514
- Accuracy on Test Data: 0.9706927175843695

❖ validness of models (over fitting or under fitting):

- The Accuracy of both the Train data and Test data are almost same; hence our model is Neither Overfit nor Underfit.

❖ Classification Report:

Classification Report of the Training data:				
	precision	recall	f1-score	support
0	1.00	1.00	1.00	6555
1	1.00	1.00	1.00	1327
accuracy			1.00	7882
macro avg			1.00	7882
weighted avg			1.00	7882
Classification Report of the Test data:				
	precision	recall	f1-score	support
0	0.97	0.99	0.98	2809
1	0.95	0.87	0.91	569
accuracy			0.97	3378
macro avg			0.96	3378
weighted avg			0.97	3378

❖ Interpretations:

-----Test set -----

❖ Accuracy:

- The Accuracy of model is 97%.
- 97% of the Data points are identified correctly.

❖ Recall: Represents the correctness of the Model in a particular class.

- Recall for class 1(Churn) is 87% and recall for class 0(non-churn) is 99%.

- 87% of those Churned were correctly identified as Churned by the model.
- Out of all the customers who are Churned, 87% of them are identified correctly as Churned.
- Out of all the customers who are not Churned, 99% of them are identified correctly as Not-Churned.

❖ **Precision:**

- Precision for class 1(Churn) is 95% and for class 0(non-churn) is 97%.
- 95% of the customers predicted are actually Churned Out of all the Customers that are predicted as Churned.
- 97% of the Customers predicted are actually Not Churned out of all the customers predicted as not Churned.

❖ **Confusion Matrix:**

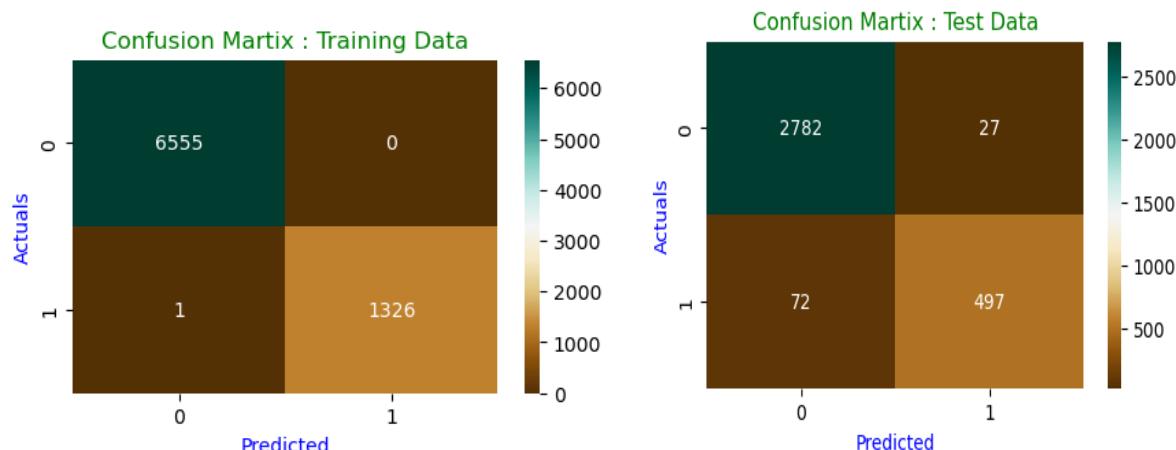


Fig no 113: Confusion Matrix XGB

❖ **AUC - ROC Curve:**

- AUC Train: 1.000
- AUC Test: 0.992

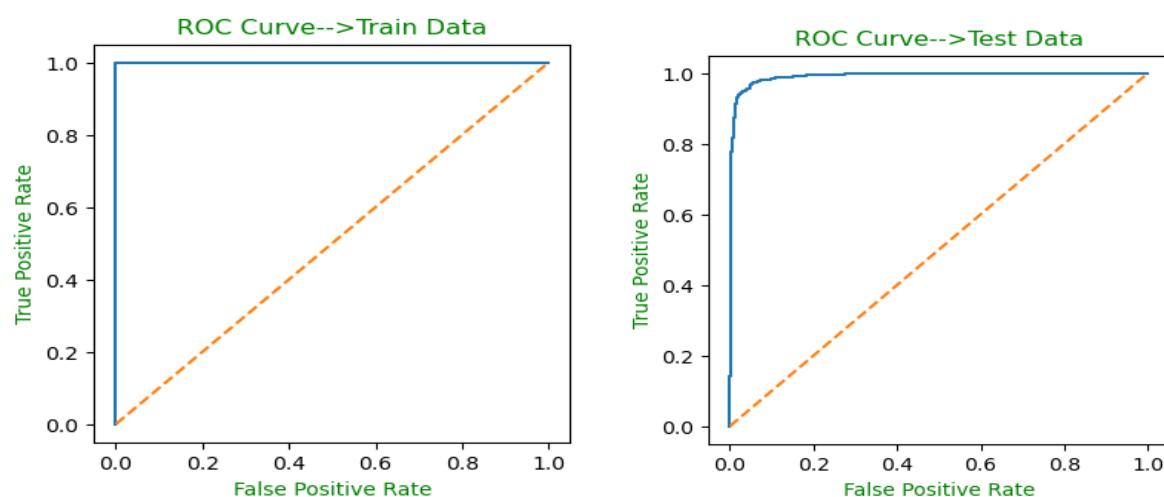


Fig no 114: ROC Curve XGB

❖ **Observations:**

- AUC for Train data is 100% and AUC for Test data is 99%.
- The AUC & ROC curve shows that it is covering the same area for Train and Test data. Therefore, this is considered a Generalized good model.
- Also, the AUC score is very high and is able to distinguish between Positive and Negative classes very well.

❖ **Feature Importance:**

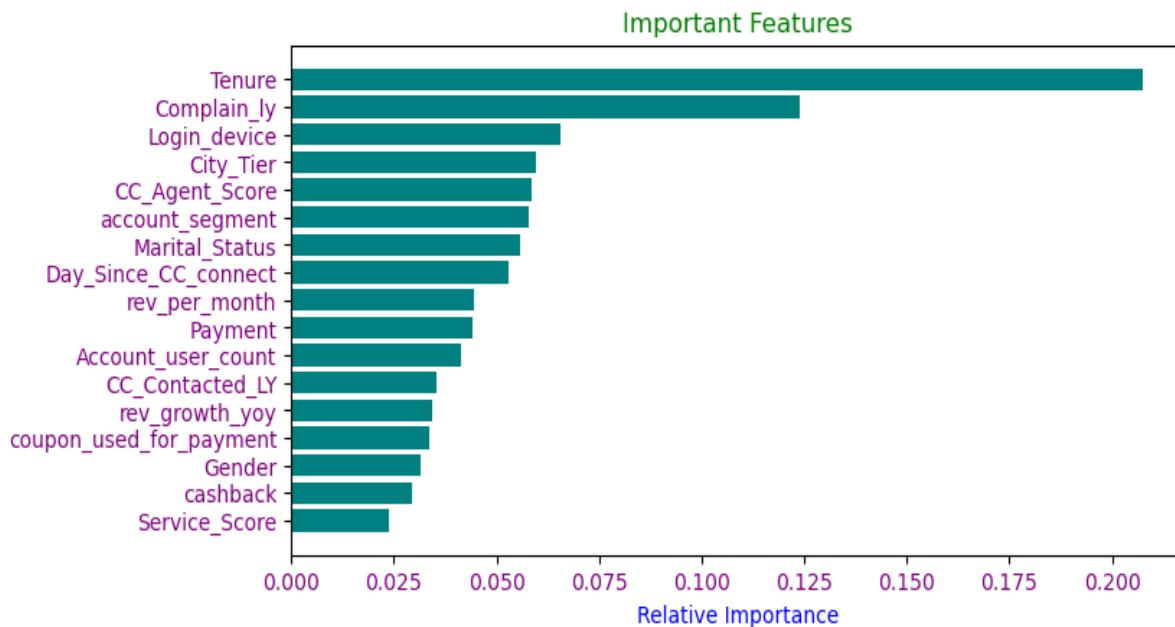


Fig no 115: Feature Imp XGB

- The 1st five important feature of XGBoost model are: Tenure, Complain_ly, Login_device, City_Tier and CC_Agent_Score.

XGBoost on Smote dataset

```
XGBClassifier
XGBClassifier(base_score=None, booster=None, callbacks=None,
              colsample_bylevel=None, colsample_bynode=None,
              colsample_bytree=None, device=None, early_stopping_rounds=None,
              enable_categorical=False, eval_metric='logloss',
              feature_types=None, gamma=None, grow_policy=None,
              importance_type=None, interaction_constraints=None,
              learning_rate=None, max_bin=None, max_cat_threshold=None,
              max_cat_to_onehot=None, max_delta_step=None, max_depth=None,
              max_leaves=None, min_child_weight=None, missing=nan,
              monotone_constraints=None, multi_strategy=None, n_estimators=None, v
```

❖ **Accuracy of Train and Test set:**

- Accuracy on Train Data: 0.9993897654955975
- Accuracy on Test Data: 0.9650680876258141

❖ **validness of models (over fitting or under fitting):**

- **The Accuracy of both the Train data and Test data are almost same; hence our model is Neither Overfit nor Underfit.**

❖ **Classification Report:**

Classification Report of the Training data:

	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

Classification Report of the Test data:

	precision	recall	f1-score	support
0	0.97	0.99	0.98	2809
1	0.92	0.87	0.89	569
accuracy			0.97	3378
macro avg	0.95	0.93	0.94	3378
weighted avg	0.96	0.97	0.96	3378

❖ **Interpretations:**

-----Test set Smote Data-----

❖ **Accuracy:**

- **The Accuracy of model is 97%.**
- **97% of the Data points are identified correctly.**

❖ **Recall: Represents the correctness of the Model in a particular class.**

- **Recall for class 1(Churn) is 87% and recall for class 0(non-churn) is 99%.**
- **87% of those Churned were correctly identified as Churned by the model.**
- **Out of all the customers who are Churned,87% of them are identified correctly as Churned.**
- **Out of all the customers who are not Churned, 99% of them are identified correctly as Not-Churned.**

❖ **Precision:**

- **Precision for class 1(Churn) is 92% and for class 0(non-churn) is 97%.**
- **92% of the customers predicted are actually Churned Out of all the Customers that are predicted as Churned.**
- **97% of the Customers predicted are actually Not Churned out of all the customers predicted as not Churned.**

❖ AUC - ROC Curve:

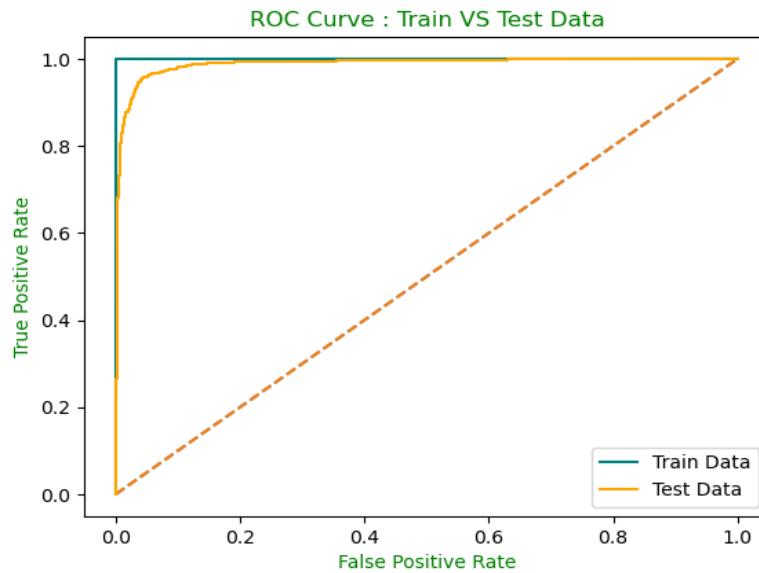


Fig no 116: ROC Curve XGB smote

Applying GridSearchCv on XGBoost

Building Model on Best Parameter

```
XGBClassifier
XGBClassifier(base_score=None, booster=None, callbacks=None,
              colsample_bylevel=1, colsample_bynode=None, colsample_bytree=1,
              device=None, early_stopping_rounds=None, enable_categorical=False,
              eval_metric='logloss', feature_types=None, gamma=1,
              grow_policy=None, importance_type=None,
              interaction_constraints=None, learning_rate=0.2, max_bin=None,
              max_cat_threshold=None, max_cat_to_onehot=None,
              max_delta_step=None, max_depth=None, max_leaves=None,
              min_child_weight=None, missing=nan, monotone_constraints=None,
              multi_strategy=None, n_estimators=200, n_jobs=None,
```

❖ Accuracy of Train and Test Data:

- Accuracy on Train Data: 0.9969550875412332
- Accuracy on Test Data: 0.9600355239786856

❖ validity of models (over fitting or under fitting):

- The Accuracy of both the Train data and Test data are almost same; hence our model is Neither Overfit nor Underfit.

❖ Classification Report:

Classification Report of the Training data:

	precision	recall	f1-score	support
0	1.00	1.00	1.00	6555
1	0.98	1.00	0.99	1327
accuracy			1.00	7882
macro avg	0.99	1.00	0.99	7882
weighted avg	1.00	1.00	1.00	7882

Classification Report of the Test data:

	precision	recall	f1-score	support
0	0.98	0.97	0.98	2809
1	0.87	0.90	0.88	569
accuracy			0.96	3378
macro avg	0.92	0.93	0.93	3378
weighted avg	0.96	0.96	0.96	3378

❖ **Interpretations:**

-----Test set Tuned Model-----

❖ **Accuracy:**

- The Accuracy of model is 96%.
- 96% of the Data points are identified correctly.

❖ **Recall:** Represents the correctness of the Model in a particular class.

- Recall for class 1(Churn) is 90% and recall for class 0(non-churn) is 97%.
- 90% of those Churned were correctly identified as Churned by the model.
- Out of all the customers who are Churned,90% of them are identified correctly as Churned.
- Out of all the customers who are not Churned, 97% of them are identified correctly as Not-Churned.

❖ **Precision**

- Precision for class 1(Churn) is 81% and for class 0(non-churn) is 92%.
- 87% of the customers predicted are actually Churned Out of all the Customers that are predicted as Churned.
- 98% of the Customers predicted are actually Not Churned out of all the customers predicted as not Churned.

❖ **Confusion Matrix:**

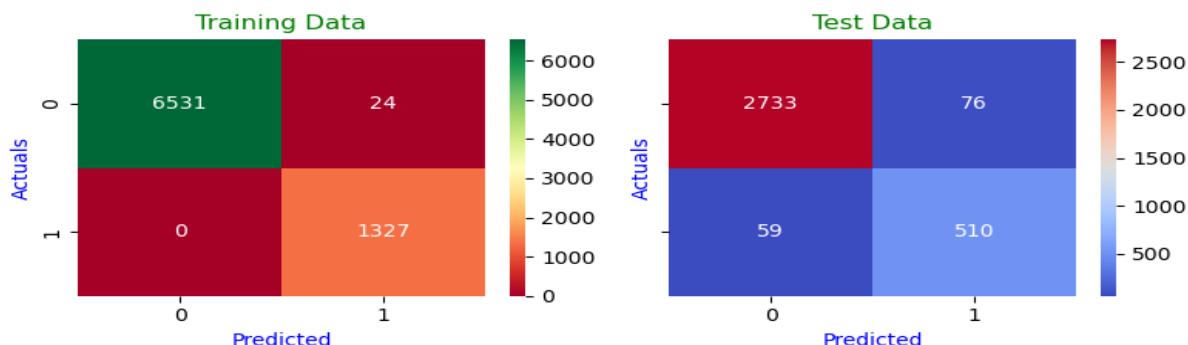


Fig no 117: Confusion Matrix XGB Tuned

❖ **Measuring and Plotting AUC-ROC Curve:**

- AUC for the Training Data: 1.000
- AUC for the Test Data: 0.988

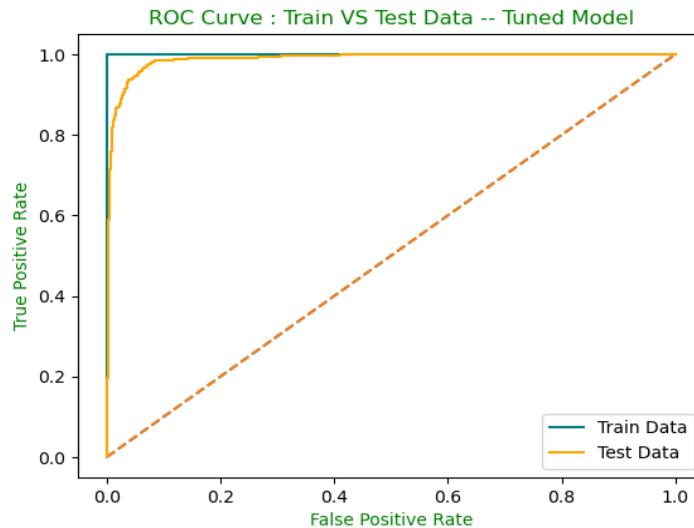


Fig no 118: ROC Curve XGB Tuned

❖ **Observations:**

- AUC for Train data is 100% and AUC for Test data is 98%.
- The AUC & ROC curve shows that it is covering almost the same area for Train and Test data. Therefore, this is considered a Generalized good model.
- Also, the AUC score is very high and is able to distinguish between Positive and Negative classes very well.

❖ **Feature Importance:**

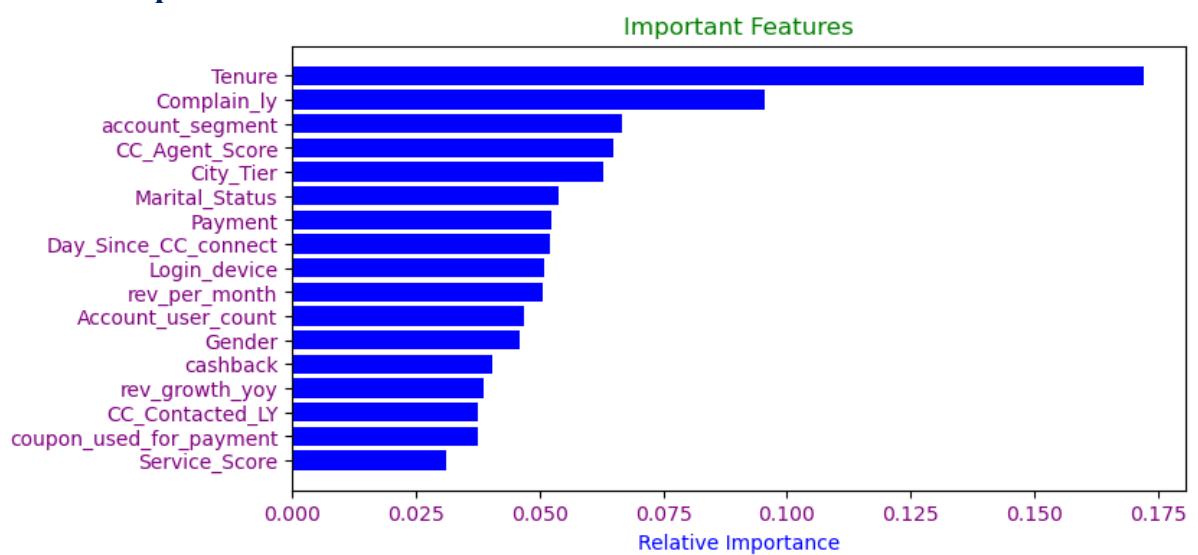


Fig no 119: Feature Imp XGB Tuned

- The 1st five Important features of XGBoost Tuned model are: Tenure, Complain_Ly, CC_Agent_Score, account_segment and City_Tier.

LightGBM Classifier

❖ Accuracy of Train and Test set:

- Accuracy on Train Data: 1.0
- Accuracy on Test Data: 0.9831261101243339

❖ Classification Report:

Classification Report of the Training data:					
	precision	recall	f1-score	support	
0	1.00	1.00	1.00	6555	
1	1.00	1.00	1.00	1327	
accuracy			1.00	7882	
macro avg	1.00	1.00	1.00	7882	
weighted avg	1.00	1.00	1.00	7882	

Classification Report of the Test data:					
	precision	recall	f1-score	support	
0	0.99	0.99	0.99	2809	
1	0.97	0.93	0.95	569	
accuracy			0.98	3378	
macro avg	0.98	0.96	0.97	3378	
weighted avg	0.98	0.98	0.98	3378	

❖ Interpretations:

-----Test set-----

❖ Accuracy:

- The Accuracy of model is 98%.
- 98% of the Data points are identified correctly.

❖ Recall: Represents the correctness of the Model in a particular class.

- Recall for class 1(Churn) is 93% and recall for class 0(non-churn) is 99%.
- 93% of those Churned were correctly identified as Churned by the model.
- Out of all the customers who are Churned, 93% of them are identified correctly as Churned.
- Out of all the customers who are not Churned, 99% of them are identified correctly as Not-Churned.

❖ Precision:

- Precision for class 1(Churn) is 97% and for class 0(non-churn) is 99%.
- 97% of the customers predicted are actually Churned Out of all the Customers that are predicted as Churned.
- 99% of the Customers predicted are actually Not Churned out of all the customers predicted as not Churned.

❖ Confusion Matrix:

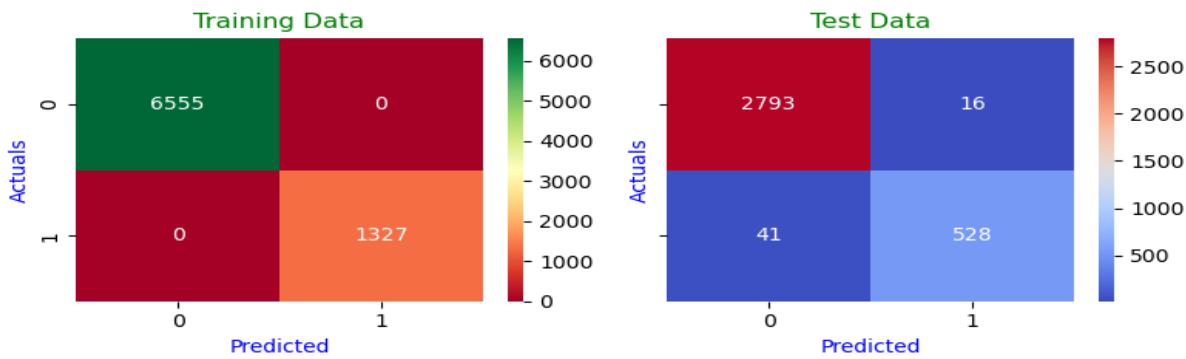


Fig no 120: CM LightGBM

❖ AUC - ROC Curve:

- AUC for the Training Data: 1.000
- AUC for the Test Data: 0.996

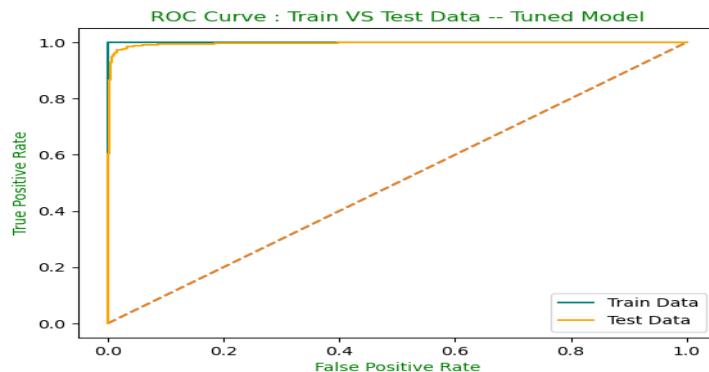


Fig no 121: ROC Curve LightGBM

❖ Observations:

- AUC for Train data is 100% and AUC for Test data is 99%.
- The AUC & ROC curve shows that it is covering the same area for Train and Test data. Therefore, this is considered a Generalized good model.
- Also, the AUC score is very high and is able to distinguish between Positive and Negative classes very well.

❖ Feature Importance:

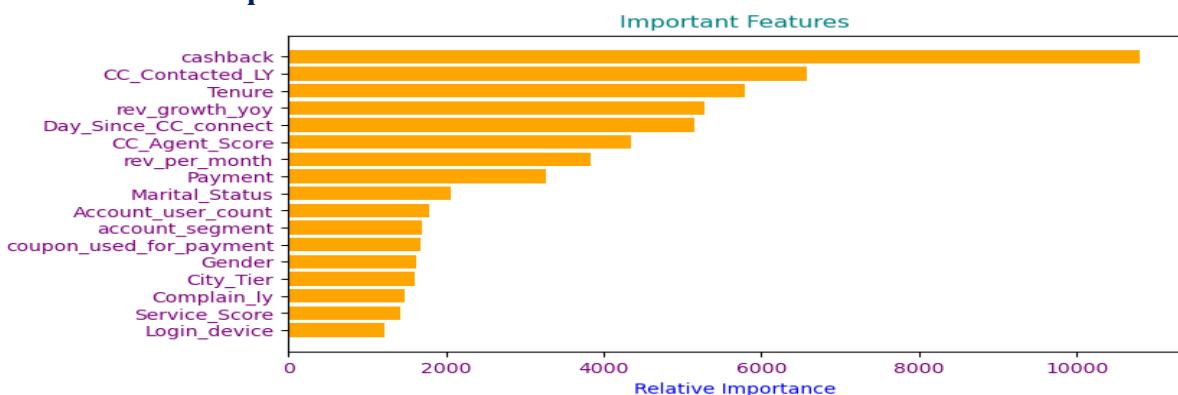


Fig no 122: Feature Imp LightGBM

- The 1st five Important features of LightGBM Model are: cashback, CC_Contacted_LY, Tenure, rev_growth_yoy and Day_Since_CC_connect.

Model Comparison

Comparing the Models built by Plotting the ROC-AUC Curve for Test data on Base model and Tuned Model.

Base Models

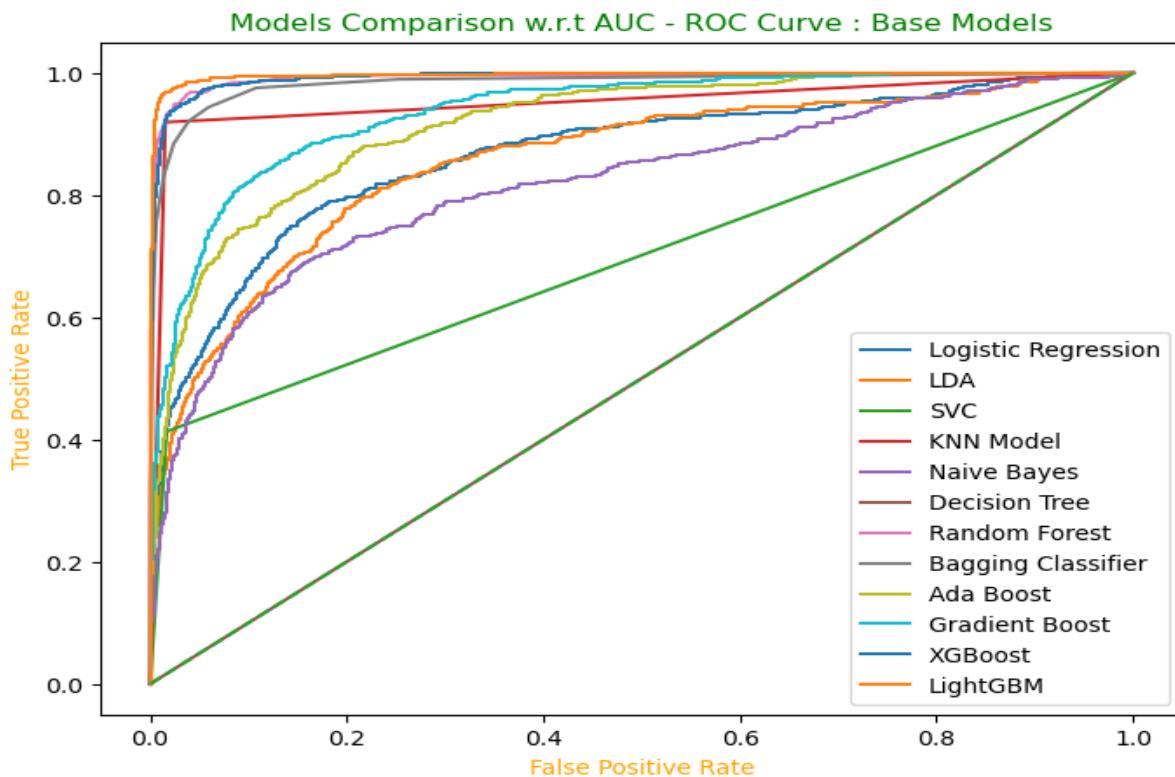


Fig no 123: Base Model Comparison

Models	Logistic Regression	LDA	SVC	KNN	Naïve Bayes	Decision Tree
AUC	0.86	0.85	0.69	0.95	0.81	0.91

Models	Random Forest	Bagging Classifier	Ada Boost	Gradient Boost	XGBoost	LightGBM
AUC	0.99	0.98	0.91	0.93	0.99	0.99

❖ **Interpretations:**

- Comparing the AUC Score we can say that Random Forest, XGBoost and LightGBM Models are performing Better having a High AUC score of 99% on Test data.
- These Models are able to Separate between the Churn and non-churn Classes Very well.
- These models can be considered a Good Generalized model.

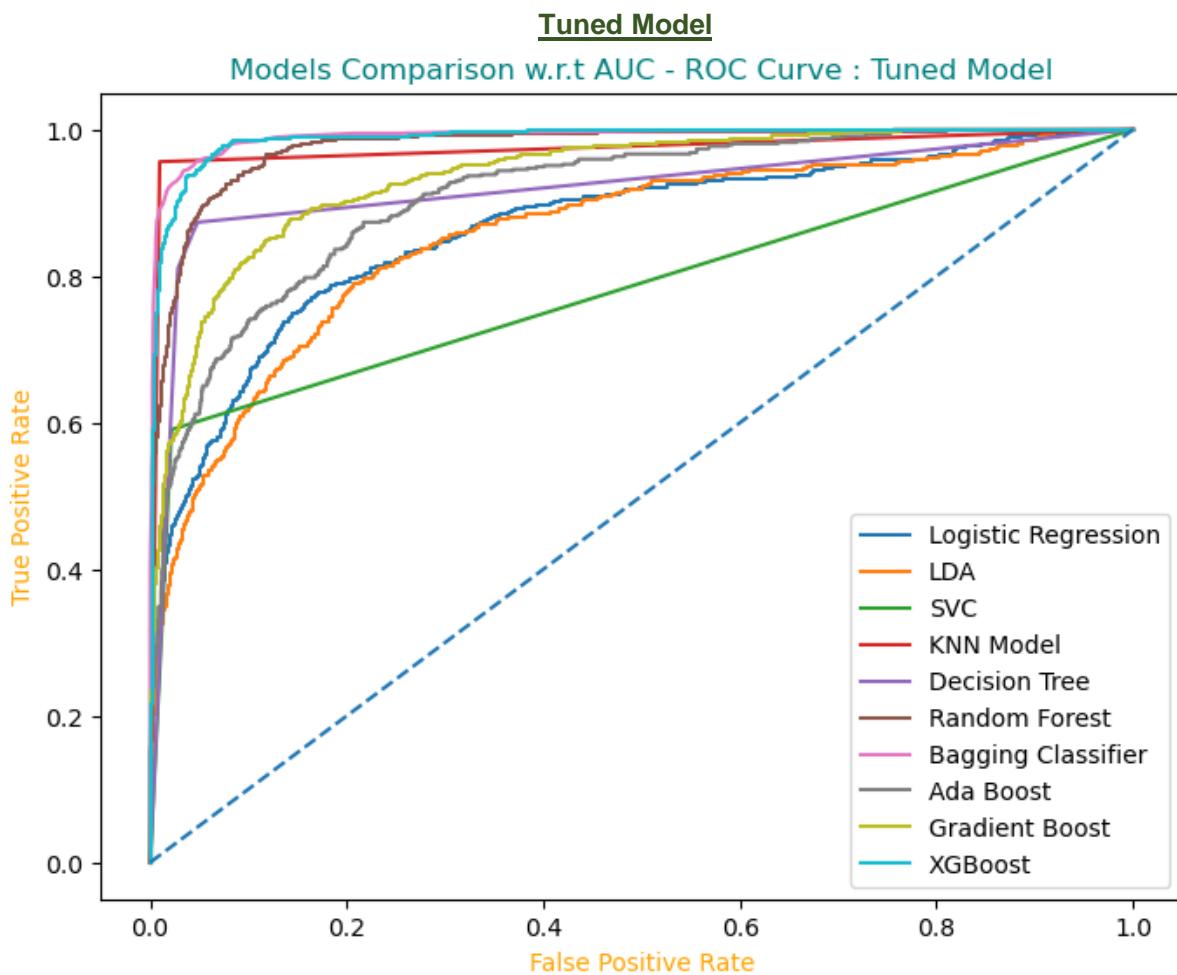


Fig no 124: Tuned Model Comparison

Models	Logistic Regression	LDA	SVC	KNN	Decision Tree
AUC	0.86	0.85	0.78	0.97	0.91

Models	Random Forest	Bagging Classifier	Ada Boost	Gradient Boost	XGBoost
AUC	0.97	0.99	0.91	0.93	0.98

❖ **Interpretations:**

- Optimized Tuned Bagging Model is performing better with highest AUC score of 99%.
- Optimized XGBoost model is also performing better with 2nd highest AUC score of 98%.
- Other models like Random Forest and KNN models are having a Good AUC Score of 97%.
- So, we can say that these Models are able to Separate between the Churn and non-churn Classes Very well.
- These models can be considered a Good Generalized model.

Interpretation of the most optimum model and its implication on the business.

❖ **Interpretations:**

- Among all Built models, LightGBM model is giving Highest Accuracy of 98% on Test data and Highest AUC score of 99%.
- So, LightGBM is the most Optimal model in my case.
- Also, the Recall value is Good for LightGBM model and KNN models.
- Random Forest, Bagging and XGBoost models can be considered 2nd most Optimal Models with Highest AUC Score of 99% and Accuracy of 97%.
- KNN Tuned model LightGBM models are performing better with Highest Accuracy of 98% on Test data.
- Also, XGBoost, Bagging, Random Forest and KNN models can be a good Generalized models with good Accuracy.
- In terms of Precision and Recall, KNN model is showing a good trade-off between Precision and Recall.
- If we look at the Recall Value KNN Tuned and LightGBM model is able to identify maximum numbers of True Positives Correctly in both the classes.

❖ **Business Insights, Implications and Recommendations:**

- The 1st five Important features of LightGBM Model are: cashback, CC_Contacted_LY, Tenure, rev_growth_yoy and Day_Since_CC_connect.
- These 5 features are Contributing most towards the Accuracy of the LightGBM Model.
- Since these 5 features are contributing most towards the Accuracy, the business should ensure that these features should not contain any Anomalies, Null values or any kind of unwanted characters as a part of these columns in the future.
- Also, we should make sure these 5 features are made compulsory for model building and accurate data are captured.
- The business can make use of this model to identify customers who may Churn in future.

Models Comparison Table

Models	Class	Accuracy		Precision		Recall		f1 Score		AUC	
		Train	Test	Train	Test	Train	Test	Train	Test	Train	Test
Logistic Regression	Churn(1)			0.75	0.78	0.45	0.47	0.56	0.58	0.87	0.86
	Not Churn(0)	0.88	0.89	0.9	0.9	0.97	0.97	0.93	0.93		
Logistic Regression Smote	Churn(1)			0.78	0.47	0.79	0.77	0.78	0.58	0.87	0.86
	Not Churn(0)	0.81	0.82	0.84	0.95	0.83	0.82	0.84	0.88		
Logistic Regression Tuned	Churn(1)			0.76	0.8	0.44	0.46	0.55	0.58	0.86	0.86
	Not Churn(0)	0.88	0.89	0.89	0.9	0.97	0.98	0.93	0.94		
LDA	Churn(1)			0.75	0.78	0.4	0.4	0.52	0.53	0.85	0.85
	Not Churn(0)	0.88	0.88	0.89	0.89	0.97	0.98	0.93	0.93		
LDA Smote	Churn(1)			0.76	0.44	0.78	0.78	0.77	0.56	0.87	0.86
	Not Churn(0)	0.8	0.8	0.83	0.95	0.81	0.8	0.82	0.87		
LDA Tuned	Churn(1)			0.75	0.78	0.4	0.4	0.52	0.53	0.85	0.85
	Not Churn(0)	0.88	0.88	0.89	0.89	0.97	0.98	0.93	0.93		
Support Vector Classifier	Churn(1)			0.8	0.83	0.4	0.41	0.54	0.55	0.69	0.69
	Not Churn(0)	0.88	0.89	0.89	0.89	0.98	0.98	0.93	0.94		
SVC Smote	Churn(1)			0.78	0.47	0.8	0.78	0.79	0.59	0.81	0.8
	Not Churn(0)	0.82	0.82	0.85	0.95	0.83	0.82	0.84	0.88		
SVC Tuned	Churn(1)			0.92	0.85	0.66	0.59	0.77	0.7	0.82	0.78
	Not Churn(0)	0.93	0.91	0.94	0.92	0.99	0.98	0.93	0.93		
KNN_3	Churn(1)			0.96	0.89	0.94	0.84	0.95	0.86	0.99	0.96
	Not Churn(0)	0.98	0.96	0.99	0.97	0.99	0.98	0.99	0.97		
KNN_1	Churn(1)			1	0.92	1	0.92	1	0.92	1	0.95
	Not Churn(0)	1	0.97	1	0.98	1	0.98	1	0.98		
KNN_3 Smote	Churn(1)			0.98	0.88	1	0.94	0.99	0.87	1	0.97
	Not Churn(0)	0.99	0.95	1	0.99	0.98	0.95	0.99	0.97		
KNN Tuned	Churn(1)			1	0.95	1	0.96	1	0.95	1	0.97
	Not Churn(0)	1	0.98	1	0.99	1	0.99	1	0.99		
Naïve Bayes	Churn(1)			0.63	0.58	0.57	0.57	0.6	0.57	0.83	0.81
	Not Churn(0)	0.87	0.86	0.91	0.91	0.93	0.92	0.92	0.92		
Naïve Bayes Smote	Churn(1)			0.68	0.35	0.8	0.77	0.74	0.48	0.83	0.81
	Not Churn(0)	0.75	0.72	0.83	0.94	0.72	0.71	0.77	0.81		
Decision Tree	Churn(1)			1	0.87	1	0.85	1	0.86	1	0.91
	Not Churn(0)	1	0.95	1	0.97	1	0.98	1	0.97		
Decision Tree Smote	Churn(1)			1	0.77	1	0.81	1	0.79	1	0.88
	Not Churn(0)	1	0.93	1	0.96	1	0.95	1	0.96		
Decision Tree Tuned	Churn(1)			1	0.85	0.95	0.82	0.97	0.83	1	0.92
	Not Churn(0)	0.99	0.94	0.99	0.96	1	0.97	0.99	0.97		
Random Forest	Churn(1)			1	0.97	1	0.84	1	0.9	1	0.99
	Not Churn(0)	1	0.97	1	0.97	1	0.99	1	0.98		
Random Forest Smote	Churn(1)			1	0.98	1	0.99	1	0.98	1	0.99
	Not Churn(0)	1	0.97	1	0.98	1	0.99	1	0.98		
Random Forest Tuned	Churn(1)			0.98	0.9	0.83	0.68	0.9	0.78	0.99	0.97
	Not Churn(0)	0.97	0.93	0.97	0.94	1	0.99	0.98	0.96		
Bagging Classifier	Churn(1)			1	0.93	0.98	0.83	0.99	0.87	1	0.98
	Not Churn(0)	1	0.96	1	0.97	1	0.99	1	0.98		
Bagging Classifier Smote	Churn(1)			1	0.9	1	0.85	1	0.87	1	0.98
	Not Churn(0)	1	0.96	1	0.97	1	0.98	1	0.98		
Bagging Classifier Tuned	Churn(1)			1	0.97	1	0.92	1	0.89	1	0.99
	Not Churn(0)	1	0.97	1	0.97	1	0.99	1	0.98		
Ada Boost Classifier	Churn(1)			0.77	0.76	0.61	0.6	0.68	0.67	0.93	0.91
	Not Churn(0)	0.9	0.9	0.92	0.92	0.96	0.96	0.94	0.94		
Ada Boost Classifier Smote	Churn(1)			0.93	0.71	0.89	0.64	0.91	0.67	0.97	0.91
	Not Churn(0)	0.92	0.89	0.92	0.93	0.95	0.95	0.93	0.94		
Ada Boost Classifier Tuned	Churn(1)			0.75	0.75	0.59	0.59	0.66	0.66	0.91	0.91
	Not Churn(0)	0.9	0.9	0.92	0.92	0.96	0.96	0.94	0.94		
Gradient Boost	Churn(1)			0.84	0.82	0.64	0.59	0.72	0.69	0.95	0.93
	Not Churn(0)	0.92	0.91	0.93	0.92	0.98	0.97	0.95	0.95		
Gradient Boost Smote	Churn(1)			0.92	0.72	0.88	0.72	0.9	0.72	0.97	0.93
	Not Churn(0)	0.92	0.9	0.91	0.94	0.94	0.94	0.93	0.94		
Gradient Boost Tuned	Churn(1)			0.85	0.81	0.64	0.59	0.73	0.68	0.95	0.93
	Not Churn(0)	0.92	0.91	0.93	0.92	0.98	0.97	0.95	0.95		
XGBoost Classifier	Churn(1)			1	0.95	1	0.87	1	0.91	1	0.99
	Not Churn(0)	1	0.97	1	0.97	1	0.99	1	0.98		
XGBoost Classifier Smote	Churn(1)			1	0.92	1	0.87	1	0.89	1	0.98
	Not Churn(0)	1	0.97	1	0.97	1	0.99	1	0.98		
XGBoost Classifier Tuned	Churn(1)			1	0.87	1	0.9	1	0.88	1	0.98
	Not Churn(0)	1	0.96	1	0.98	1	0.97	1	0.98		
LightLGBM	Churn(1)			1	0.97	1	0.93	1	0.95	1	0.99
	Not Churn(0)	1	0.98	1	0.99	1	0.99	1	0.99		

Fig no 125: Model Comparison Table

Thank You

