

A collection of business-related icons. In the center is a magnifying glass over a line graph on a presentation board. To the right is a money bag with a dollar sign. Below that is a safe. To the left is a wallet. Below the wallet is a plus sign. To the right of the plus sign is a pyramid. Above the pyramid is a target. To the left of the target is a checkmark. To the right of the target is a diamond. Below the diamond is a rainbow. To the left of the rainbow is a bar chart. Below the bar chart is a grid of dots.

A collection of business-related icons including a pie chart, envelope, bank building, gear, dollar bill, calendar, bar chart, calculator, and various geometric shapes.

greatlearning
Power Ahead

TABLE OF CONTENTS

1

Customer Churn Analysis.....6

CONTENTS		
1	Introduction of the Business Problem	6
1.1	Problem Statement	6
1.2	Need of the Project	7
1.3	Objective	7
1.4	Constraints	8
1.5	Understanding business/social opportunity	8
2	EDA & Business Implication	9
2.1	Data Dictionary	9
2.2	Understanding how data was collected in terms of time, frequency and methodology	10
2.3	Visual inspection of data (rows, columns, descriptive details)	11
2.4	Information of the Features	14
2.5	Renaming the columns of the data frame	16
2.6	Exploratory Data Analysis	17
3	Data Cleaning & Pre-Processing	34
3.1	Removal of unwanted variables	34
3.2	Addition of new features	34
3.3	Missing value & Null value treatment	35
4	Model Building	43
4.1	Algorithms applicable for the given problem	43
4.2	Evaluation metrics for model comparison	43
4.3	Improving model performance	45
4.4	Comparison of metrics for all the models	46
5	Model Validation	49
5.1	Criteria for best performing model	49
5.2	Why Tuned XG Boost Model is the Optimal Model?	50

5.3	Business Implications	51
5.4	How Can Business Use these metrics?	52
5.5	Model Interpretations from Feature Importance	53
6	Business Recommendations & Insights	57

LIST OF FIGURES

Fig.1	Customer Churn analysis	4
Fig.2	Details of the dataset columns	11
Fig.3	Descriptive summary of the numeric columns	12
Fig.4	Descriptive summary of the string columns	12
Fig.5	Distribution of the target variable - Churn	17
Fig.6	Histograms of continuous Features	19
Fig.7	Skewness of continuous features	19
Fig.8	Boxplot of continuous features	19
Fig.9	Count plot of categorical features	21
Fig.10	Count plot of features vs Target column	27
Fig.11	Pair plot of all the features	30
Fig.12	Correlation Heatmap	31
Fig.13	WSS Plot	32
Fig.14	Clustered Profile – 3 segments	33
Fig.15	Null values in the dataset and its visualization	35
Fig.16	Duplicate values in the dataset	36
Fig.17	Distribution of target variable in duplicate records	36
Fig.18	Duplicate Rows	37
Fig.19	Null Values before pre-processing	38
Fig.20	Data Cleanup Table	39
Fig.21	Outlier Proportion in each column before & after treatment	40
Fig.22	Descriptive summary before & after scaling	41
Fig.23	Sample rows of scaled data frame	42
Fig.24	Splitting of train and test data	44
Fig.25	Distribution of Target Variable in Train Dataset before & after SMOTE	45
Fig.26	Performance metrics summary for all models	46
Fig.27	Performance metrics – Sorted by Test F1-Score	47

Fig.28	Performance metrics – Sorted by Test Recall	47
Fig.29	Performance metrics – Sorted by Test Precision	48
Fig.30	Performance metrics of Optimal Model	50
Fig.31	Ada Boost Feature Importance	53
Fig.32	Gradient Boost Feature Importance	54
Fig.33	XG Boost Feature Importance	54
Fig.34	Count plot of account_tenure vs Target variable	57
Fig.35	Count plot of payment method vs Target variable	58
Fig.36	Count plot of coupon used vs Target variable	59
Fig.37	Tuned XG Boost Model	64
Fig.38	Best parameters of Tuned XG Boost Model	64
Fig.39	Tuned XG Boost – AUC & ROC curve	65
Fig.40	Tuned XG Boost – Confusion Matrix	65
Fig.41	Logistic Regression Model with Smote	66
Fig.42	LR Smote – AUC & ROC curve	67
Fig.43	LR Smote – Confusion Matrix	67
Fig.44	LR Smote – Classification Report	68
Fig.45	LR Smote – Metrics Summary	68
Fig.46	Logistic Regression Model	69
Fig.47	LR – AUC & ROC curve	69
Fig.48	LR – Confusion Matrix	69
Fig.49	LR – Classification Report	70
Fig.50	LR – Metrics Summary	70
Fig.51	Tuned Logistic Regression Model	72
Fig.52	Best parameters of Tuned Logistic Regression Model	72
Fig.53	Tuned LR – AUC & ROC curve	73
Fig.54	Tuned LR – Confusion Matrix	73
Fig.55	Tuned LR – Classification Report	74
Fig.56	Tuned LR – Metrics Summary	74
Fig.57	LDA – AUC & ROC curve	76
Fig.58	LDA – Confusion Matrix	76
Fig.59	LDA – Classification Report	77
Fig.60	LDA – Metrics Summary	77
Fig.61	Ada Boost Model	79
Fig.62	Ada Boost – AUC & ROC curve	79
Fig.63	Ada Boost – Confusion Matrix	79
Fig.64	Ada Boost – Classification Report	80
Fig.65	Ada Boost – Metrics Summary	80

Fig.66	Gradient Boost Model	82
Fig.67	Gradient Boost – AUC & ROC curve	82
Fig.68	Gradient Boost – Confusion Matrix	82
Fig.69	Gradient Boost – Classification Report	83
Fig.70	Gradient Boost – Metrics Summary	83
Fig.71	XG Boost Model	84
Fig.72	XG Boost – AUC & ROC curve	85
Fig.73	XG Boost – Confusion Matrix	85
Fig.74	XG Boost – Classification Report	86
Fig.75	XG Boost – Metrics Summary	86
Fig.76	Tuned Ada Boost Model	88
Fig.77	Best parameters of Tuned Ada Boost Model	88
Fig.78	Tuned Ada Boost – AUC & ROC curve	89
Fig.79	Tuned Ada Boost – Confusion Matrix	89
Fig.80	Tuned Ada Boost – Classification Report	90
Fig.81	Tuned Ada Boost – Metrics Summary	90
Fig.82	Tuned Gradient Boost Model	92
Fig.83	Best parameters of Tuned Gradient Boost Model	92
Fig.84	Tuned Gradient Boost – AUC & ROC curve	93
Fig.85	Tuned Gradient Boost – Confusion Matrix	93
Fig.86	Tuned Gradient Boost – Classification Report	94
Fig.87	Tuned Gradient Boost – Metrics Summary	94
Fig.88	RF Model	96
Fig.89	RF – AUC & ROC curve	96
Fig.90	RF – Confusion Matrix	96
Fig.91	RF – Classification Report	97
Fig.92	RF – Metrics Summary	97
Fig.94	Tuned RF Model	98
Fig.95	Best parameters of Tuned RF Model	98
Fig.96	Tuned RF – AUC & ROC curve	99
Fig.97	Tuned RF – Confusion Matrix	99
Fig.98	Tuned RF – Classification Report	100
Fig.99	Tuned RF – Metrics Summary	100
Fig.100	ANN Model	102
Fig.101	ANN – AUC & ROC curve	102
Fig. 102	ANN – Confusion Matrix	102
Fig. 103	ANN – Classification Report	103
Fig. 104	ANN – Metrics Summary	103
Fig. 105	Tuned ANN Model	104

Fig. 106	Best parameters of Tuned ANN Model	104
Fig. 107	Tuned ANN – AUC & ROC curve	105
Fig. 108	Tuned ANN – Confusion Matrix	105
Fig. 109	Tuned ANN – Metrics Summary	112
Fig. 110	KNN Model	113
Fig. 111	KNN – AUC & ROC curve	108
Fig. 112	KNN – Confusion Matrix	108
Fig. 113	KNN – Classification Report	108
Fig. 114	KNN – Metrics Summary	108
Fig. 115	Tuned KNN Model	109
Fig. 116	Tuned KNN – AUC & ROC curve	109
Fig. 117	Tuned KNN – Confusion Matrix	111
Fig. 118	Tuned KNN – Classification Report	111
Fig. 119	Tuned KNN – Metrics Summary	111
Fig. 120	Bagging Model	112
Fig. 121	Bagging – AUC & ROC curve	112
Fig. 122	Bagging – Confusion Matrix	114
Fig. 123	Bagging – Classification Report	114
Fig. 124	Bagging – Metrics Summary	114
Fig. 125	Performance metrics summary for all models	115

LIST OF TABLES

Table 1	Sample of first 5 rows of the dataset	11
---------	---------------------------------------	----

Customer Churn Analysis

I. Introduction of the Business Problem

I.1 Problem Statement

The problem at hand is to reduce customer churn for a DTH provider in a highly competitive market. The company is facing a significant challenge in retaining its existing customers and wants to develop a model to predict customer churn and provide targeted offers to potential churners. This is crucial because one account can represent multiple customers, and the loss of one account can result in the loss of multiple customers.

The goal is to develop a churn prediction model that accurately identifies potential churners and provides segmented offers to retain them without incurring significant costs for the company. The model should take into account various factors such as customer demographics, purchase history, and usage patterns.

The recommendation for the campaign must be unique and clear on the campaign offer, taking into consideration the concerns of the revenue assurance team. The campaign should not result in significant losses for the company, and any free or subsidized offers must be carefully considered to ensure their impact on the bottom line.

The problem statement, therefore, is to develop a churn prediction model and provide a clear and unique campaign recommendation that will retain existing customers while ensuring the financial viability of the company.



Fig.1 Customer Churn analysis

I.2 Need of the Project

The need for this study/project arises from the significant challenge faced by the DTH provider in retaining its existing customers in a highly competitive market. The loss of one account can result in the loss of multiple customers, making account churn a major concern for the company.

As a result, the company is experiencing significant losses in both revenue and customer acquisition. The cost of acquiring new customers is five times higher than the cost of retaining existing ones. However, increasing customer retention by just 5% can expand the company's revenue by more than 25%. It is crucial to retain loyal customers as they are five times more likely to repurchase, forgive, and refer others to the company, and seven times more likely to try new offerings. Additionally, poor customer service can cause 33% of customers to consider switching companies after just one incident.

Currently, the DTH industry's rate of churn stands at 14-16%, while our company's churn rate is an alarming 16.84%. Addressing this issue of customer churn is critical to the company's success and growth in the competitive DTH industry.

I.3 Objective

The goal is to develop a churn prediction model to identify customers who are more likely to churn and offer them tailored campaign offers, resulting in increased revenue and profitability for the business.

In addition to identifying these customers, the findings from the exploratory data analysis and the best-performing models will be utilized to develop strategic recommendations for the business.

I.4 Constraints

The focus of previous retention campaigns is unknown – cashbacks, coupons:

The company has not disclosed the focus of their previous retention campaigns. It is unclear whether they have offered cashbacks, coupons, or any other type of incentives to retain their customers. Without this information, it may be difficult to determine what types of offers have been successful in the past and what types of offers may be effective in the future.

Campaign Budget is needed to effectively provide business recommendations:

In order to provide effective business recommendations, it is necessary to know the budget allocated for the campaign. This will help in determining the types of offers that can be made and the scale at which they can be implemented. If the budget is limited, it may be necessary to prioritize certain customer segments or offer smaller incentives to retain customers. On the other hand, if the budget is more flexible, the company may be able to offer larger incentives and more targeted retention campaigns. Understanding the budget is essential for developing effective retention strategies

I.5 Understanding business/social opportunity

This project has the potential to reduce customer churn, increase customer loyalty, and improve the bottom line for the DTH provider by accurately predicting potential churners and providing targeted offers.

Retaining customers and improving customer satisfaction can also lead to a positive reputation, new customers, and further growth for the company.

Additionally, reducing customer churn can contribute to the stability of the company and the economy.

2. EDA & Business Implication

2.1 Data Dictionary

The following table shows the attribute names, their description. Although some of the variable names are slightly long, they do not have blanks or special characters in them. Some of the variable names will be renamed such that they are self-explanatory and would be easy to understand and interpret when seen in the plots as part of univariate and bivariate analysis.

Variable Name	Description
AccountID	account unique identifier
Churn	account churn flag (Target)
Tenure	Tenure of account
City_Tier	Tier of primary customer's city
CC_Contacted_L12m	How many times all the customers of the account has contacted customer care in last 12months
Payment	Preferred Payment mode of the customers in the account
Gender	Gender of the primary customer of the account
Service_Score	Satisfaction score given by customers of the account on service provided by company
Account_user_count	Number of customers tagged with this account
account_segment	Account segmentation on the basis of spend
CC_Agent_Score	Satisfaction score given by customers of the account on customer care service provided by company
Marital_Status	Marital status of the primary customer of the account
rev_per_month	Monthly average revenue generated by account in last 12 months

Complain_l12m	Any complaints has been raised by account in last 12 months
rev_growth_yoy	revenue growth percentage of the account (last 12 months vs last 24 to 13 month)
coupon_used_l12m	How many times customers have used coupons to do the payment in last 12 months
Day_Since_CC_connect	Number of days since no customers in the account has contacted the customer care
cashback_l12m	Monthly average cashback generated by account in last 12 months
Login_device	Preferred login device of the customers in the account

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

- The collected data consists of information from 11260 unique customer accounts and includes various attributes related to these accounts. The data collected belongs to a period of one year.
- The variables such as " CC_Contacted_L12m ", "rev_per_month", "Complain_l12m", "rev_growth_yoy", "coupon_used_l12m", "Day_Since_CC_connect", and "cashback_l12m" indicate that the data was collected over a period of the last 12 months.
- The data has a total of 19 variables, with 18 being independent and 1 being the dependent or target variable, which indicates whether the customer has churned or not.

2.3 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	Marital_S
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	

Table 1. Sample of first 5 rows of the dataset

Dataset has 19 columns which captures customer account information and other attributes of the customer.

Let us check the types of variables in the data frame and check for missing values in the dataset

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 11260 entries, 0 to 11259
Data columns (total 19 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   account_id                            11260 non-null  int64
1   Churn                                11260 non-null  int64
2   account_tenure                        11158 non-null  object
3   city_tier                             11148 non-null  float64
4   cust_care_contacts_12m               11158 non-null  float64
5   payment_method                       11151 non-null  object
6   gender                               11152 non-null  object
7   service_score                        11162 non-null  float64
8   customers_per_account                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  revenue_per_month                    11158 non-null  object
13  account_complaints_12m              10903 non-null  float64
14  rev_growth_yoy                      11260 non-null  object
15  coupons_used                         11260 non-null  object
16  days_since_cc_contact               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
```

Fig.2 Details of the dataset columns

- **The dataset has 19 variables and 11260 rows in total.** The "account_id" column can be **deleted** because it is not necessary for our modelling.
- The columns 'account_tenure', 'payment_method', 'gender', 'customers_per_account', 'account_segment', 'marital_Status', 'revenue_per_month', 'rev_growth_yoy', 'coupons_used', 'days_since_cc_contact', 'cashback', 'login_device', are of the **string type**
- The **other columns** 'account_id', 'Churn', 'city_tier', 'cust_care_contacts_12m', 'service_score', 'cc_agent_score', 'account_complaints_12m' are of the **integer type**.
- **Additionally, we can observe from the data above that most of the columns have missing values in them.**
- **“Churn” column will be the target variable and the remaining columns are predictor variables**

	count	mean	std	min	25%	50%	75%	max
account_id	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
cust_care_contacts_12m	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	3.086493	1.379772	1.0	2.00	3.0	4.00	5.0
account_complaints_12m	10903.0	0.285334	0.451594	0.0	0.00	0.0	1.00	1.0

Fig.3 Descriptive summary of the numeric columns

	count	unique	top	freq
account_tenure	11158	38	1	1351
payment_method	11151	5	Debit Card	4587
gender	11152	4	Male	6328
customers_per_account	11148	7	4	4569
account_segment	11163	7	Super	4062
marital_Status	11048	3	Married	5860
revenue_per_month	11158	59	3	1746
rev_growth_yoy	11260	20	14	1524
coupons_used	11260	20	1	4373
days_since_cc_contact	10903	24	3	1816
cashback	10789.0	5693.0	155.62	10.0
login_device	11039	3	Mobile	7482

Fig.4 Descriptive summary of the string columns

Observation:

- Approximately three-quarters of accounts are not expected to cancel their service (churn)
- The majority of customers, 50%, originate from cities categorized as tier 1 out of a total of three tiers
- The majority of accounts, 75%, have contacted customer support fewer than 23 times, with an average contact frequency of 18
- The average customer rating for the service provided is 2.9, with only 25% of customers giving a rating of 3 or higher on a scale of 0 to 5
- The average rating given to customer service agents is 3, with only 25% of customers giving a score of 4 or higher
- There are very few complaints reported on average, with only 25% of accounts having submitted at least one complaint
- The most commonly used payment method among accounts is Debit card out of the five options available
- The majority of customers identify as male, with four gender categories to choose from
- The average number of customers per account is 4
- The largest segment of accounts belongs to the "Super" category out of seven segments
- The majority of customers are married, with three marital status categories
- The website is most frequently accessed through mobile devices

2.4 Information of the Features

There are 18 attributes in the project that impact the target variable. Let's examine each of these variables individually:

1. **account_id** – This is a unique identifier for each consumer and is an integer data type with no null values.
2. **Churn** – This is the target variable that shows whether or not a customer has cancelled their service. It has no null values and is a categorical variable, with “0” representing “No” and “1” representing “Yes”
3. **account_tenure** – This indicates the length of time the account has been open and has 102 null values. **It is assumed that the tenure is represented in months** and It is a continuous variable.
4. **city_tier** – This attribute categorizes consumers based on the city where they reside and has 112 null values. It is a categorical variable with three groups.
5. **cust_care_contacts_12m** – This variable shows the number of times all customers associated with an account_id have contacted customer service in the past 12 months. It has 102 null values and is a continuous variable.
6. **payment_method**: This displays the customer's preferred method of payment and has 109 null values. It is a categorical variable.
7. **gender** – This attribute indicates the gender of the primary account holder and has 108 null values. It is a categorical variable.
8. **service_score** – This is a rating based on the quality of the company's service, with 98 null values. It is a categorical variable.
9. **customers_per_account** – This variable shows how many customers are associated with an account_id and has 112 null values. It is a continuous variable.
10. **account segment** – This attribute divides customers into various segments based on their spending and revenue generation, with 97 null values. It is a categorical variable.

11. **cc_agent_score** – This is a rating based on the performance of the company's customer service representatives and has 116 null values. It is a categorical variable.
12. **marital_Status** – This indicates the marital status of the primary account holder, with 212 null values. It is a categorical variable.
13. **revenue_per_month** – This shows the average monthly revenue for each account_id over the past 12 months, with 102 null values. It is a continuous variable.
14. **account_complaints_12m** – This variable indicates whether or not a customer has filed a complaint in the past 12 months, with 357 null values. It is a categorical variable.
15. **rev growth yoy** – This is a continuous variable that compares the revenue growth over the past 24 to 13 months to the growth over the past 12 months. It has no null values. **It is assumed that yoy revenue growth is represented as %.**
16. **coupons_used** – This counter shows the number of times customers have used promotional codes to pay their bills. It is a continuous variable with no null values.
17. **days_since_cc_contact** – This shows the number of days since the customer last contacted customer service. The service is considered better when the number of days is higher, with 357 null values. It is a continuous variable.
18. **cashback l12m** – This displays the amount of cash back received by the customer after paying their bill, with 471 null values. It is a continuous variable
19. **login_device** – This variable indicates if a consumer is using a phone or a computer to access the services. This has 221 null values and is a category variable.

2.5 Renaming the columns of the data frame

The below mentioned columns of the data frame have been renamed as shown.

Original Column Name	Renamed Column Name
AccountID	account_id
churn	churn
Tenure	account_tenure
City_Tier	city_tier
CC_Contacted_LY	cust_care_contacts_12m
Payment	payment_method
Gender	gender
Service_Score	service_score
Account_user_count	customers_per_account
CC_Agent_Score	cc_agent_score
Marital_Status	marital_Status
rev_per_month	revenue_per_month
Complain_ly	account_complaints_12m
coupon_used_for_payment	coupons_used
Day_Since_CC_connect	days_since_cc_contact
Login_device	login_device

2.6 Exploratory Data Analysis

Univariate Analysis

The purpose of univariate analysis is to examine the distribution and spread of every continuous attribute and the distribution of data in categories for categorical ones. This is accomplished through the use of box plots and histograms for continuous variables and count plots for categorical variables

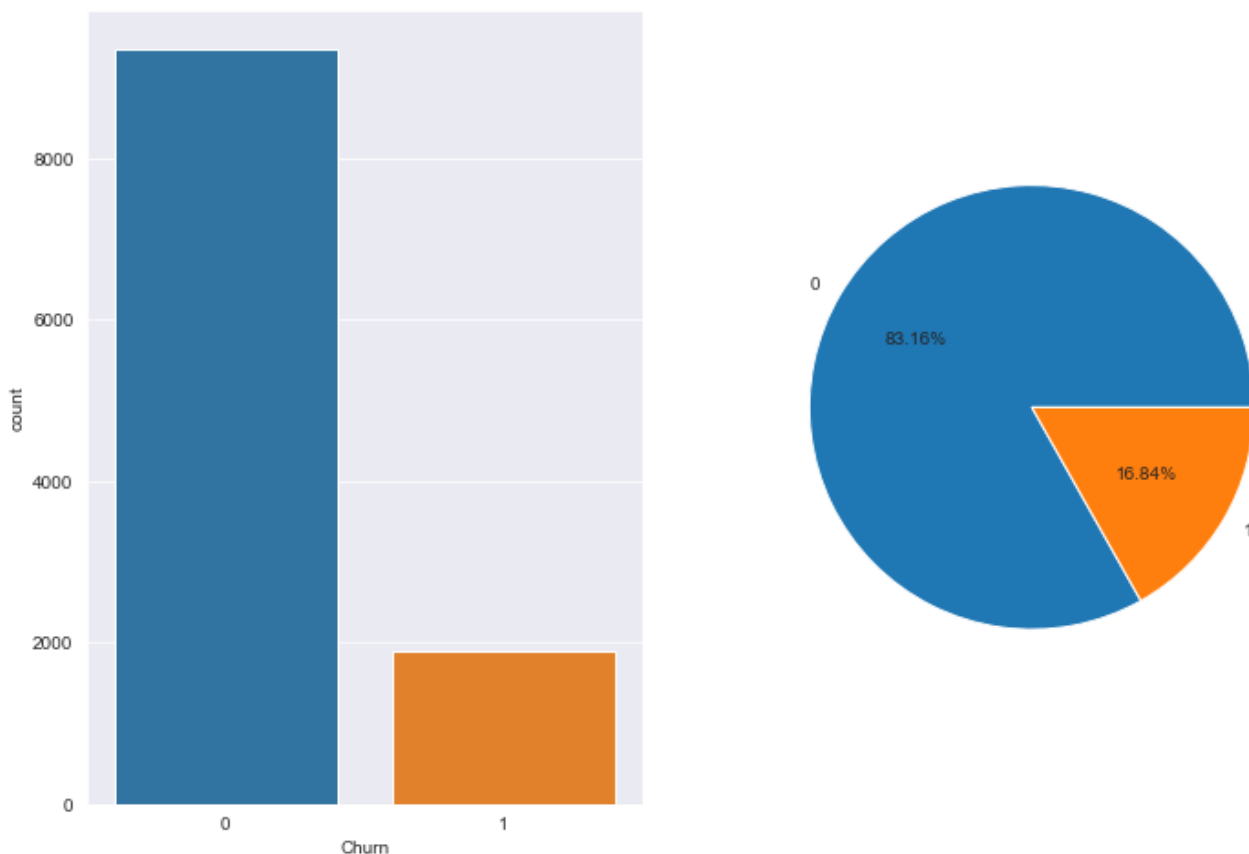
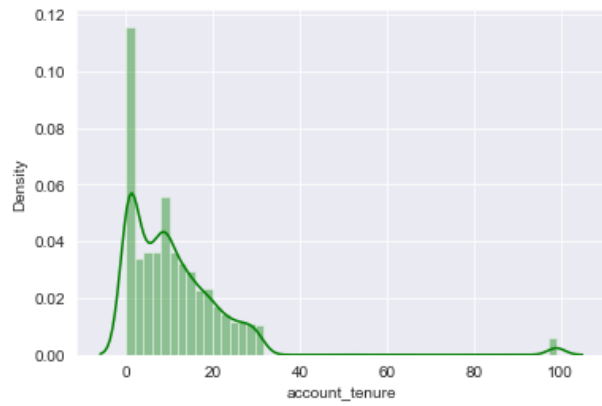


Fig.5 Distribution of the target variable - Churn

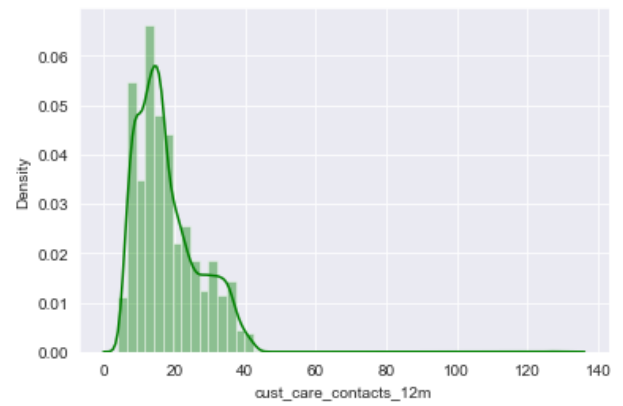
The data set presented has an imbalance. The target variable, "Churn," has a large disparity in its categorical count with 84% for "0" and 17% for "1."

To address this imbalance, we can apply the **SMOTE (Synthetic Minority Over-sampling Technique)** method, which generates additional data points to balance the distribution.

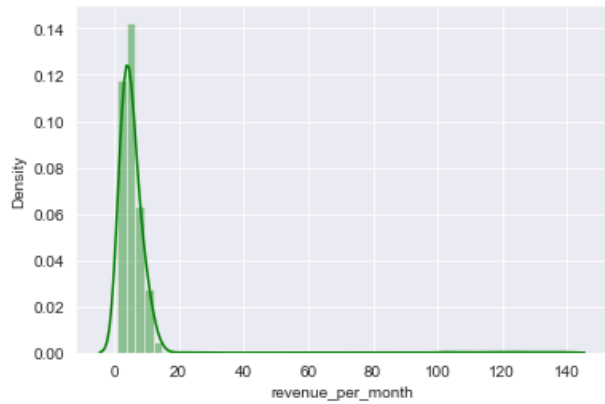
Histogram of account_tenure



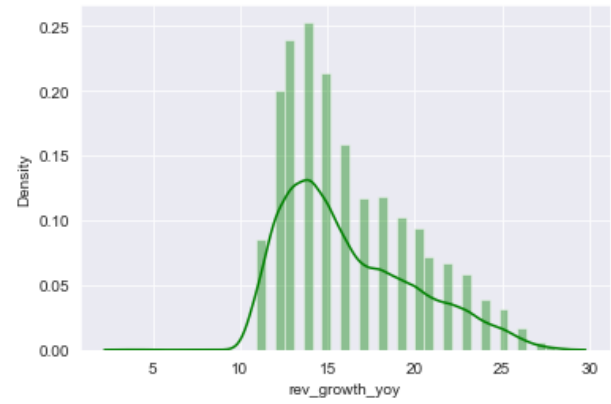
Histogram of cust_care_contacts_12m



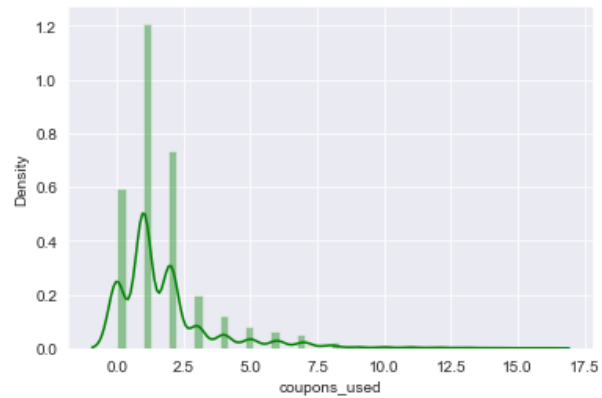
Histogram of revenue_per_month



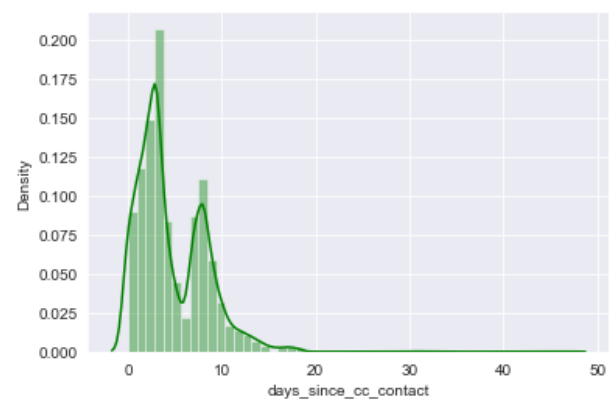
Histogram of rev_growth_yoy



Histogram of coupons_used



Histogram of days_since_cc_contact



Histogram of cashback

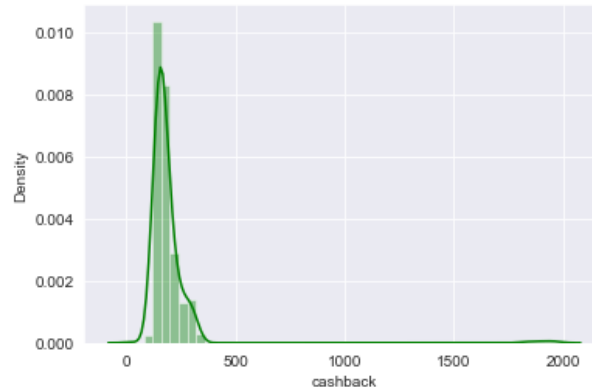


Fig.6 Histograms of continuous Features

Skewness	
account_tenure	3.94
cust_care_contacts_12m	1.44
customers_per_account	-0.44
revenue_per_month	9.35
account_complaints_12m	1.00
rev_growth_yoy	0.75
coupons_used	2.55
days_since_cc_contact	1.33
cashback	8.88

Fig.7 Skewness of continuous features

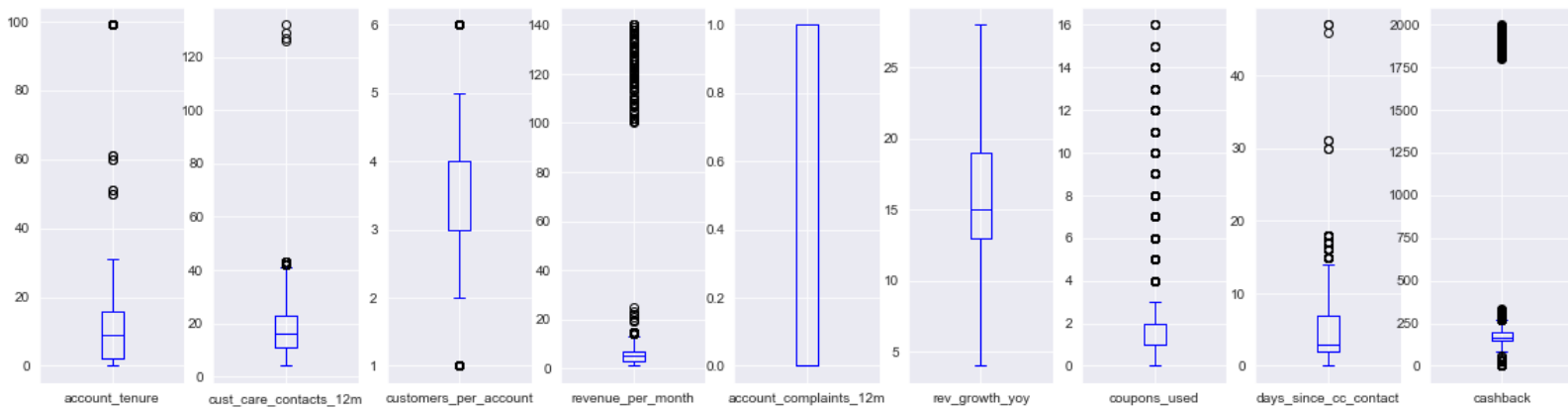
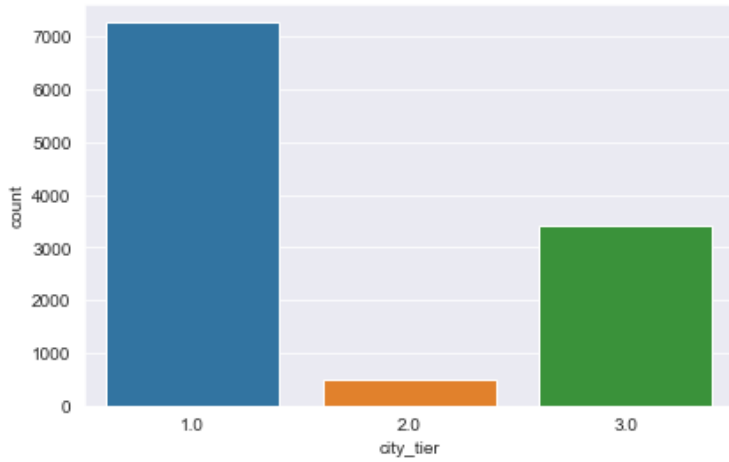
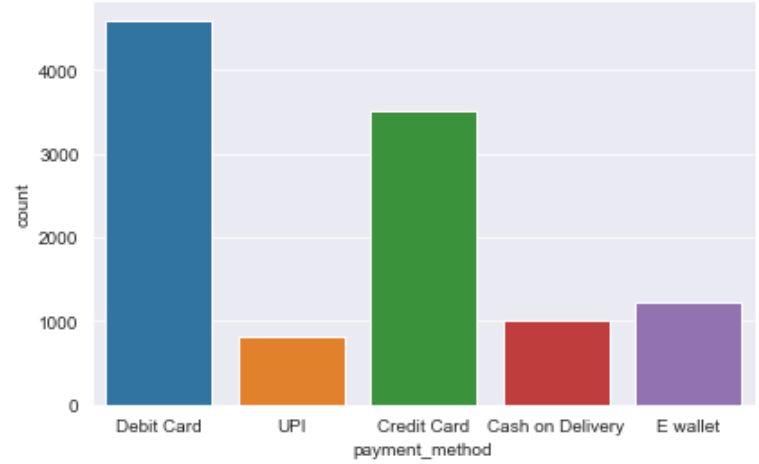


Fig.8 Boxplot of continuous features

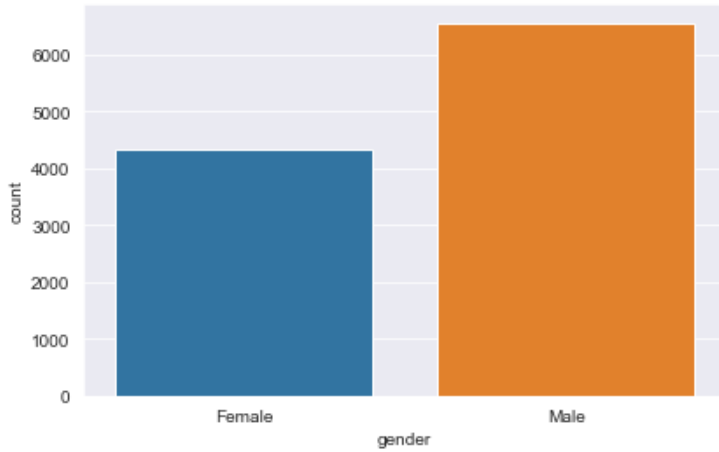
Count Plot of city_tier



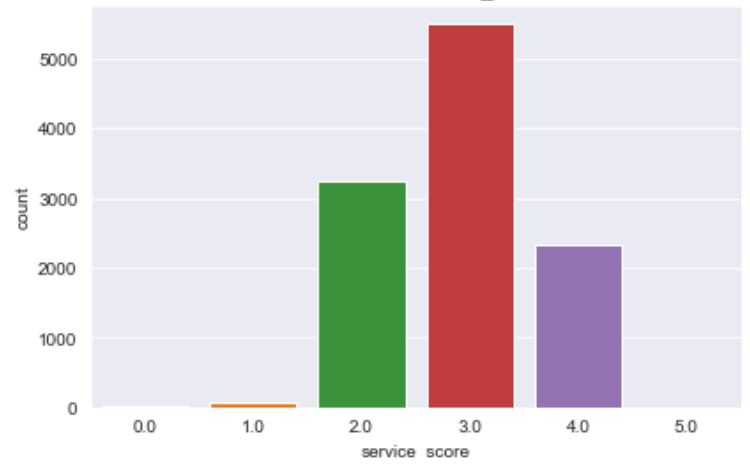
Count Plot of payment_method



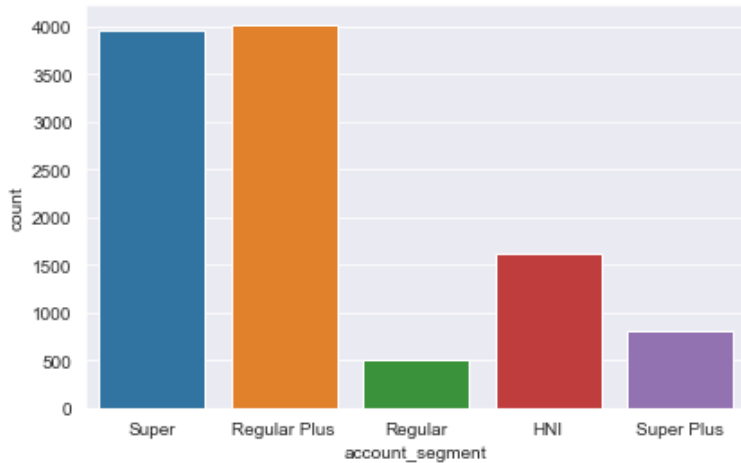
Count Plot of gender



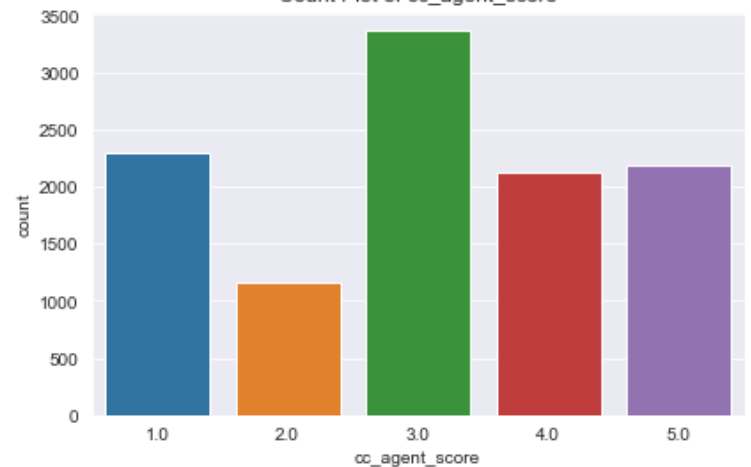
Count Plot of service_score



Count Plot of account_segment



Count Plot of cc_agent_score



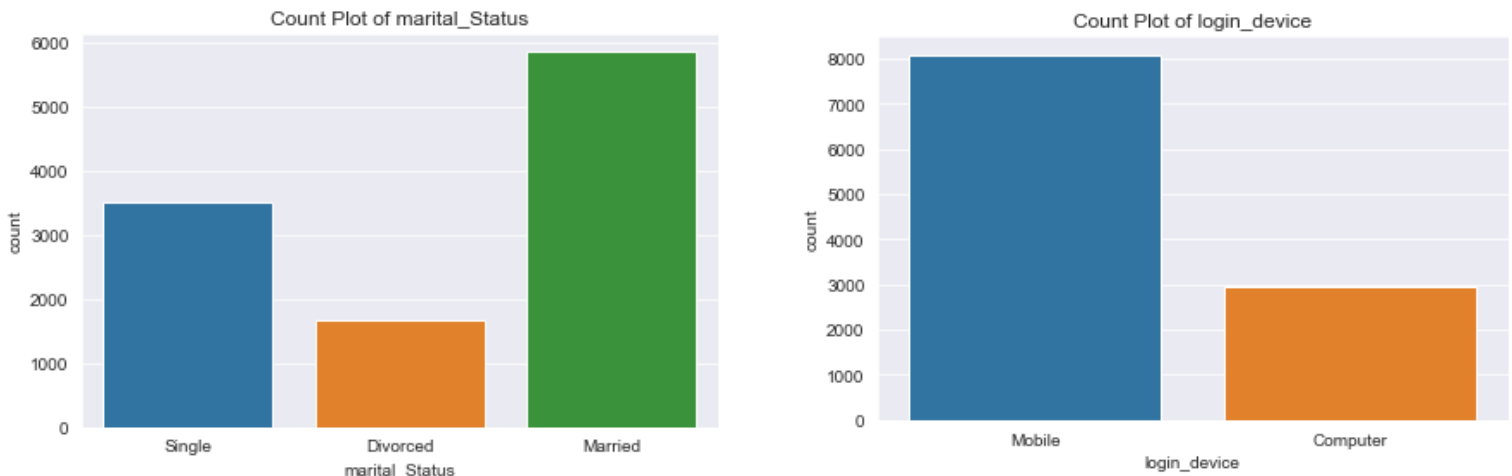


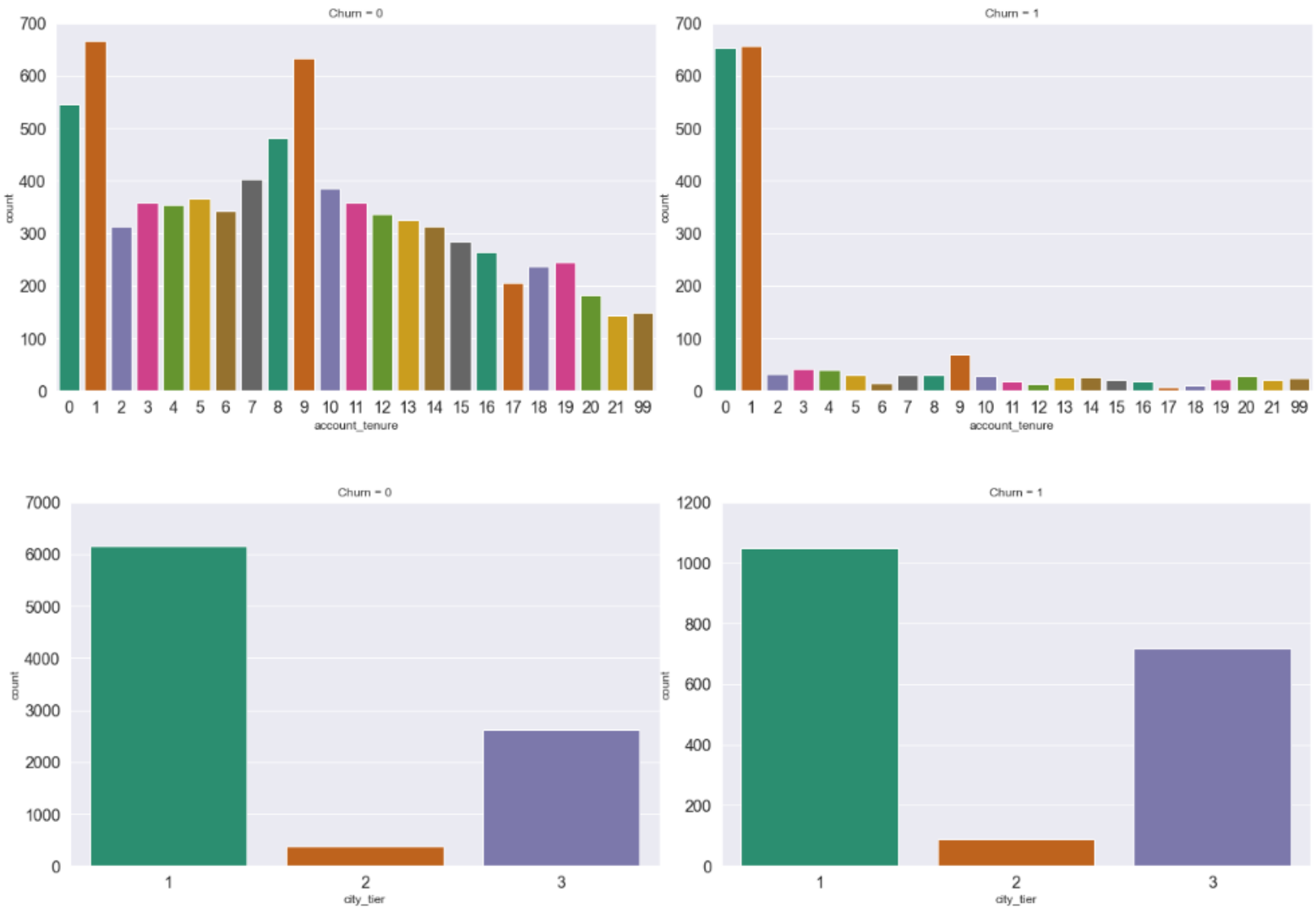
Fig.9 Count plot of categorical features

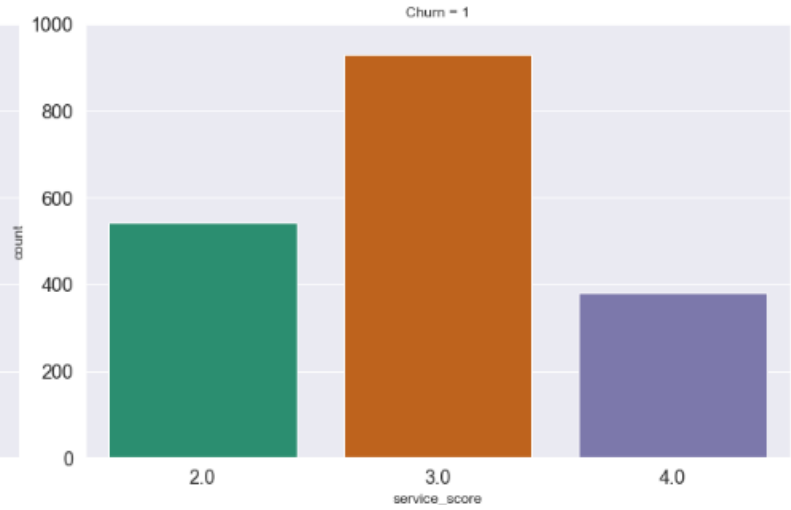
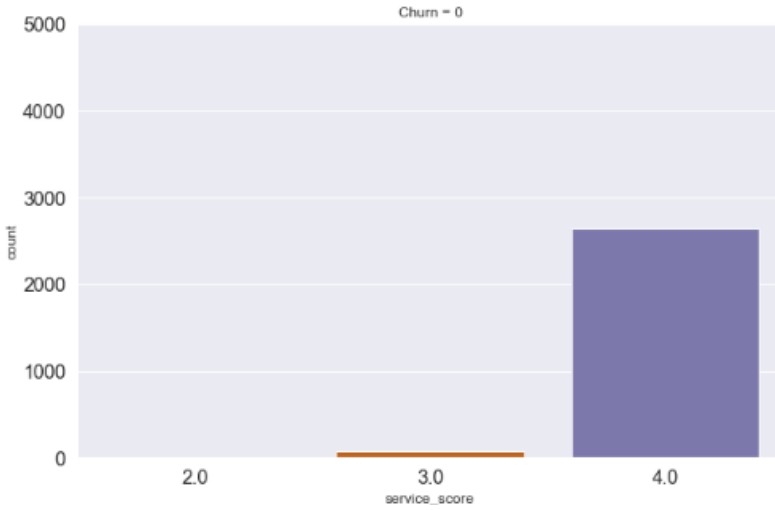
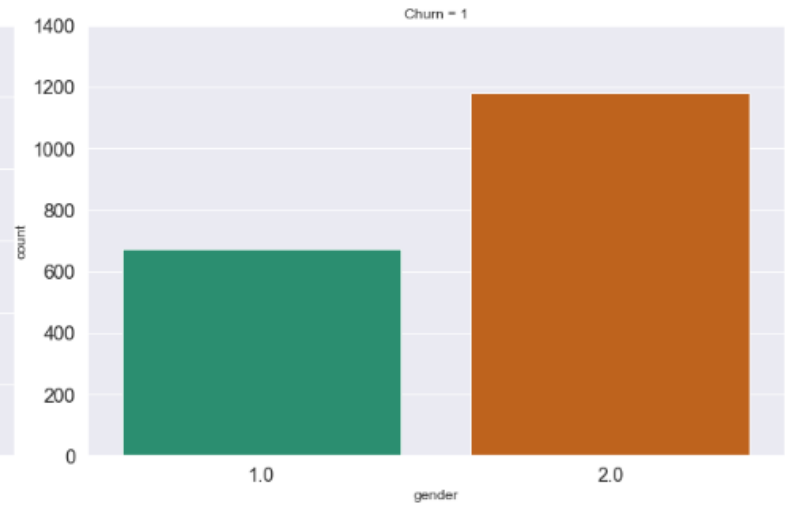
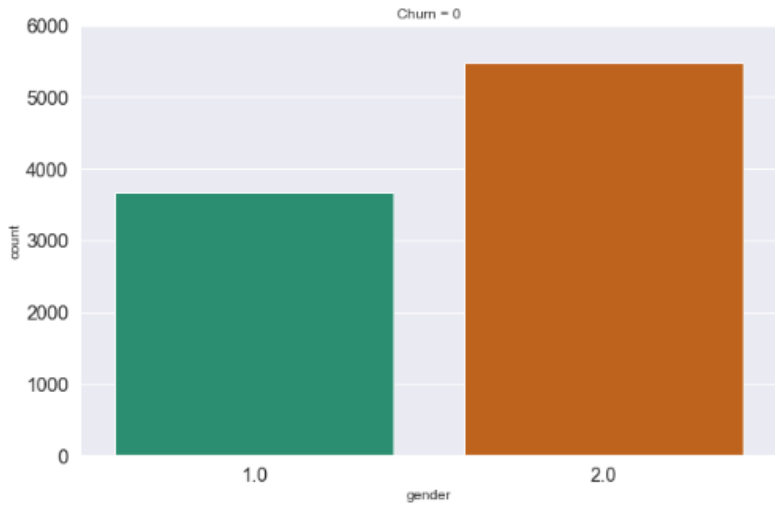
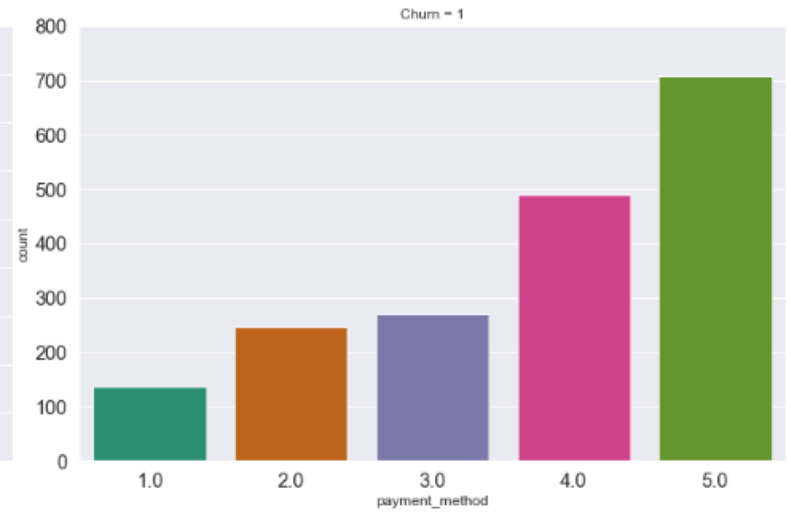
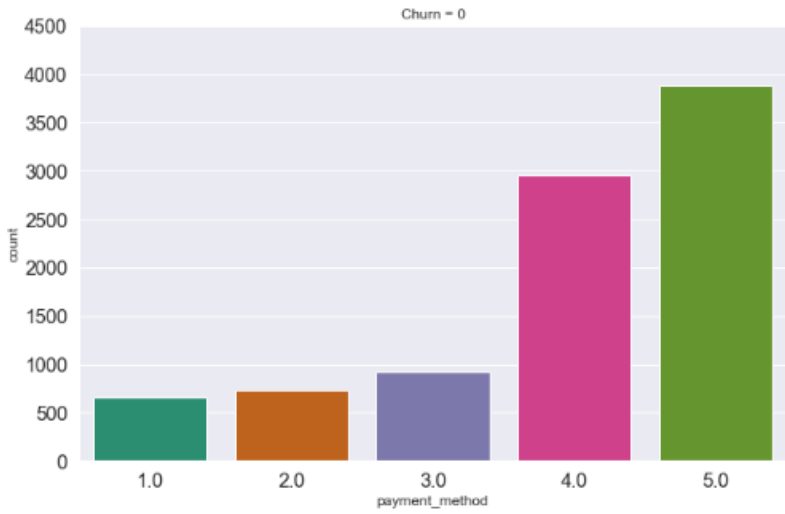
Insights:

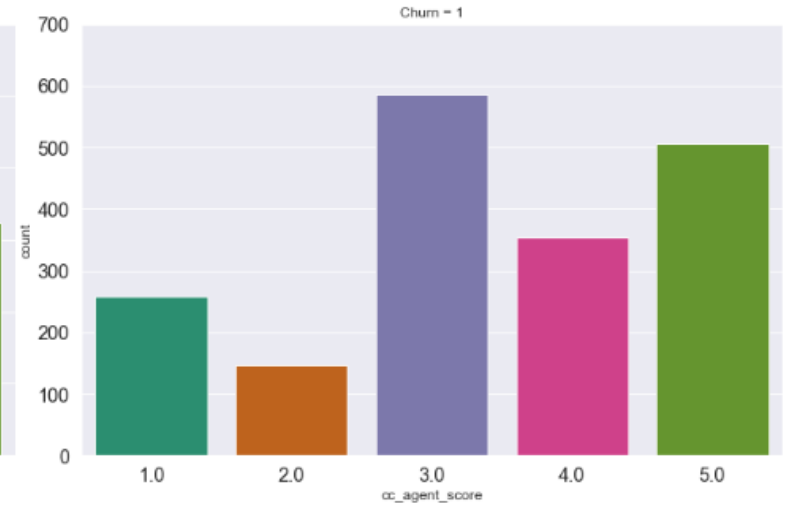
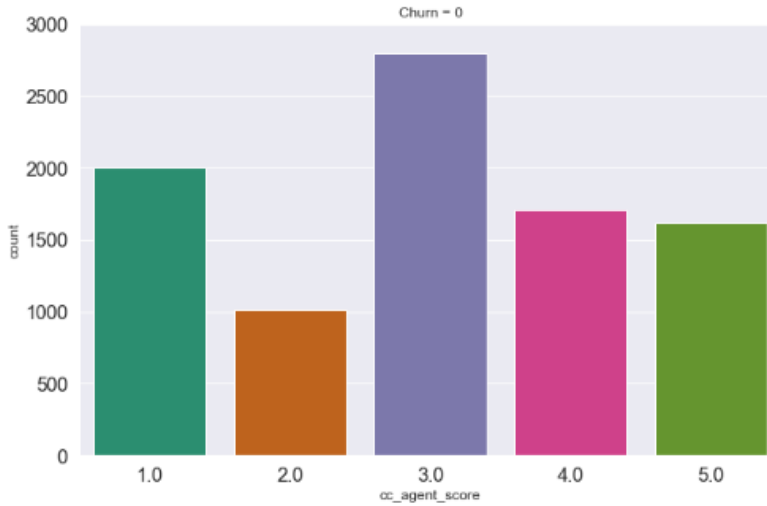
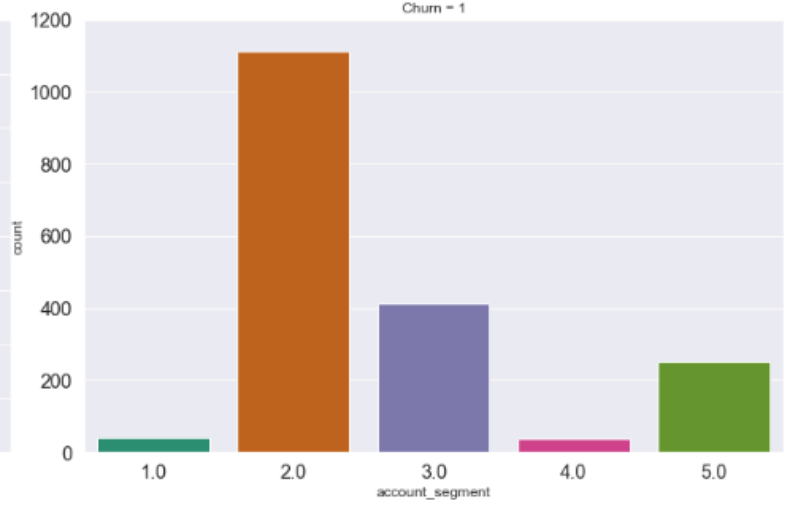
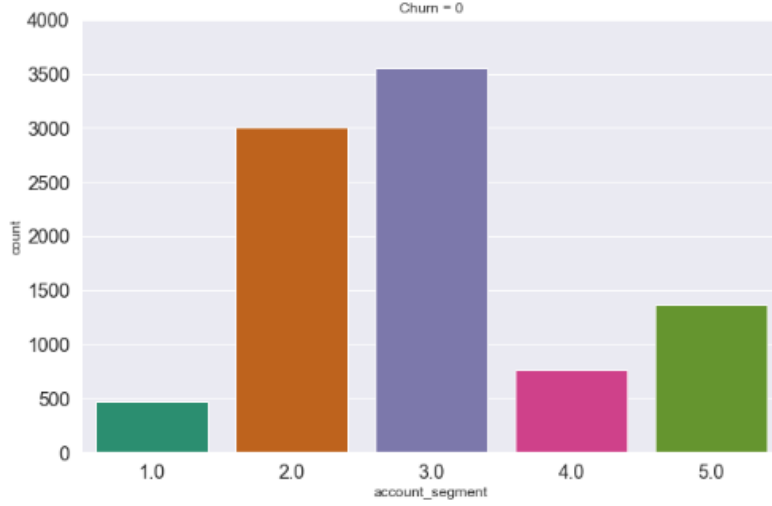
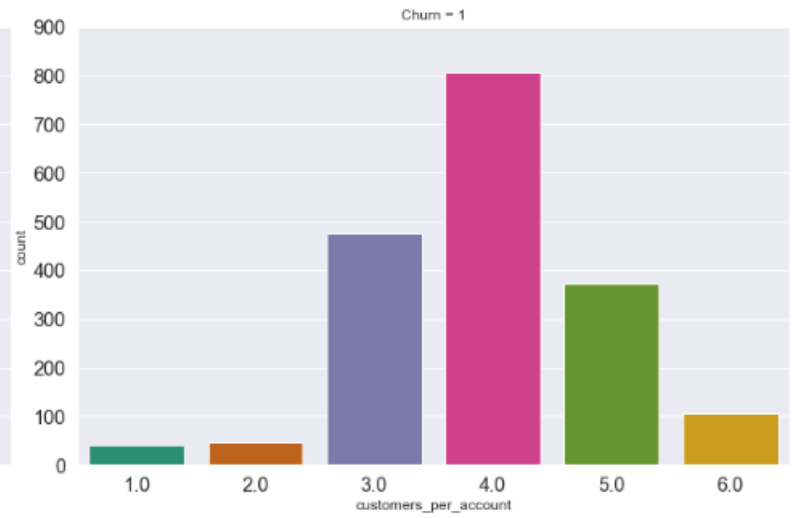
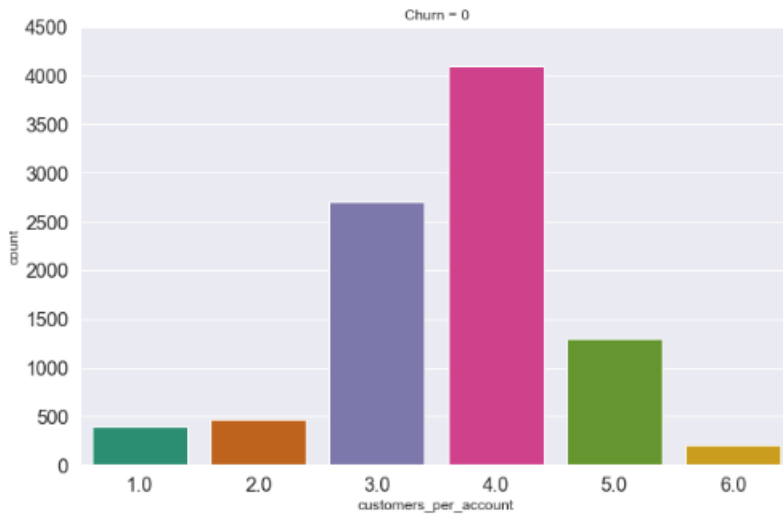
- Most numeric variables in the dataset contain outliers, except for `rev_growth_yoy`. Some of these outliers are located closer to the whisker, while others are far beyond the whisker with no in-between values. Additionally, all numeric variables, except for `rev_growth_yoy`, exhibit high positive skewness.
- The majority of customers have accounts that are 1 or 0 months old and there is a nearly equal distribution for the rest of the tenure categories. However, the account tenure feature contains a "#" category which needs to be pre-processed before modeling
- Most customers come from tier 1 and tier 3 cities, while only a small percentage is from tier 2 cities
- Most customers have contacted customer support between 6 to 17 times, with an average frequency of 18 and 75% of the accounts have contacted customer support less than 23 times
- The most popular payment method among customers is Debit card, followed by a nearly equal proportion of customers using Cash on Delivery and E-wallet options
- Male customers are the majority compared to female customers
- The average customer rating for the service provided is 2.9, with only 25% of customers giving a rating of 3 or higher on a scale of 0 to 5. The majority of customers have given a rating of 2, 3, or 4
- On average, an account has 4 customers, followed by 3 and 5 customers per account. However, there are some miscellaneous characters such as "@" that need to be replaced
- The largest segment of accounts belongs to the "Super" category, followed by "Regular Plus" out of seven segments

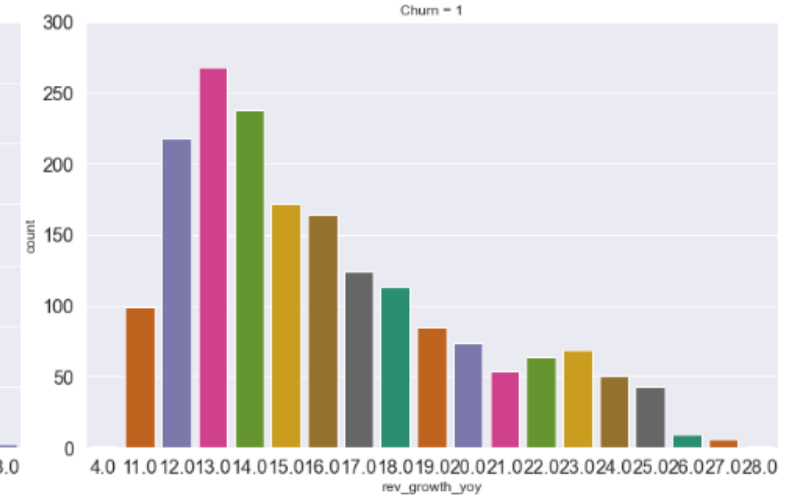
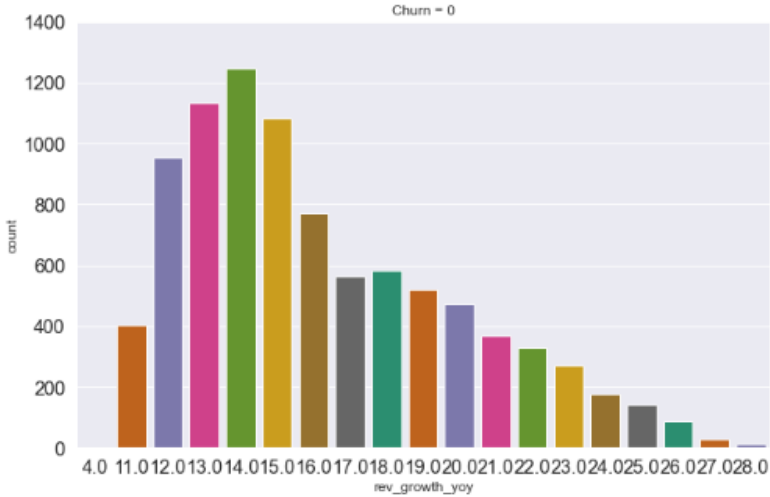
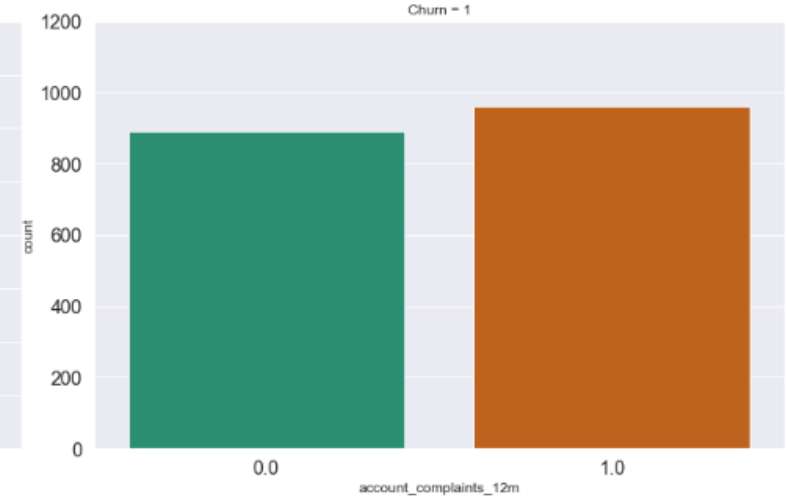
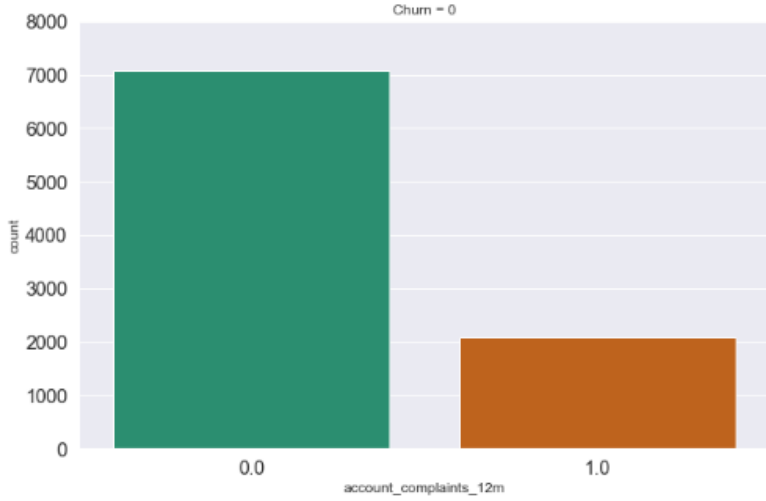
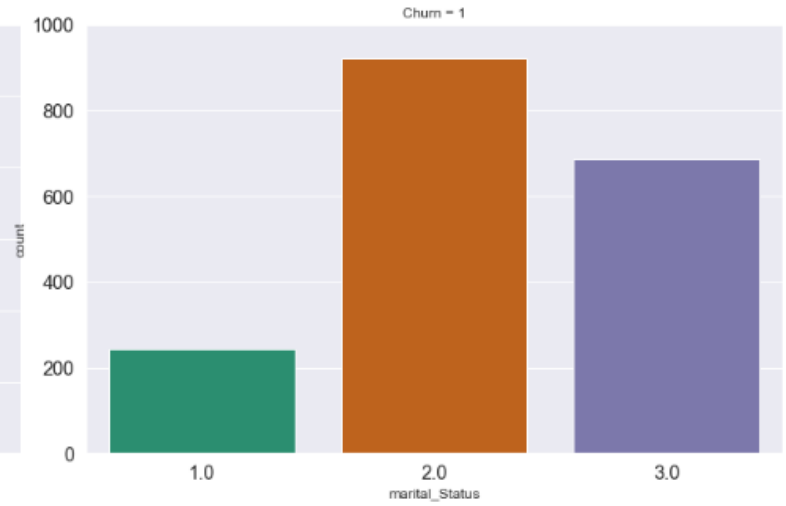
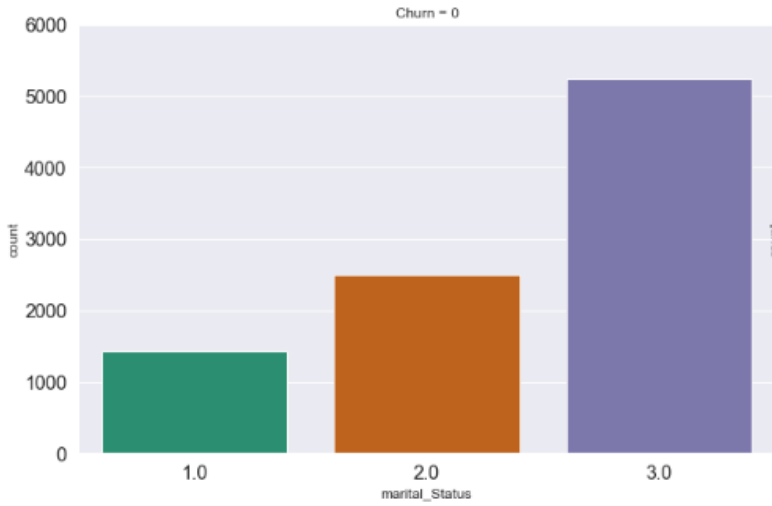
- The average rating for customer service agents is 3, with only 25% of customers giving a score of 4 or higher
- Married customers are the majority, followed by Single and Divorced customers
- The majority of accounts have an average revenue per month in the range of 1-10 Rs
- Very few complaints have been reported, with only 25% of accounts having submitted at least one complaint
- There is a higher number of accounts with 12-17% revenue growth year on year.
- The majority of accounts have used at least one or two coupons, with a good number of accounts still having unused coupons
- The most preferred way of login among accounts is using mobile, compared to using a computer.
- A dataset is considered balanced if the distribution of the target variable (Churn) across its categorical values is equal. In such a dataset, the model can predict both classes with equal accuracy since there are equal observations in each class. However, in the case of customer churn, most businesses have a smaller number of churned customers compared to active ones. This dataset also reflects this reality, with approximately 17% churned customers. While this is similar to real-life churn data, it can pose a challenge when building a model.

Bivariate Analysis w.r.t Target Variable









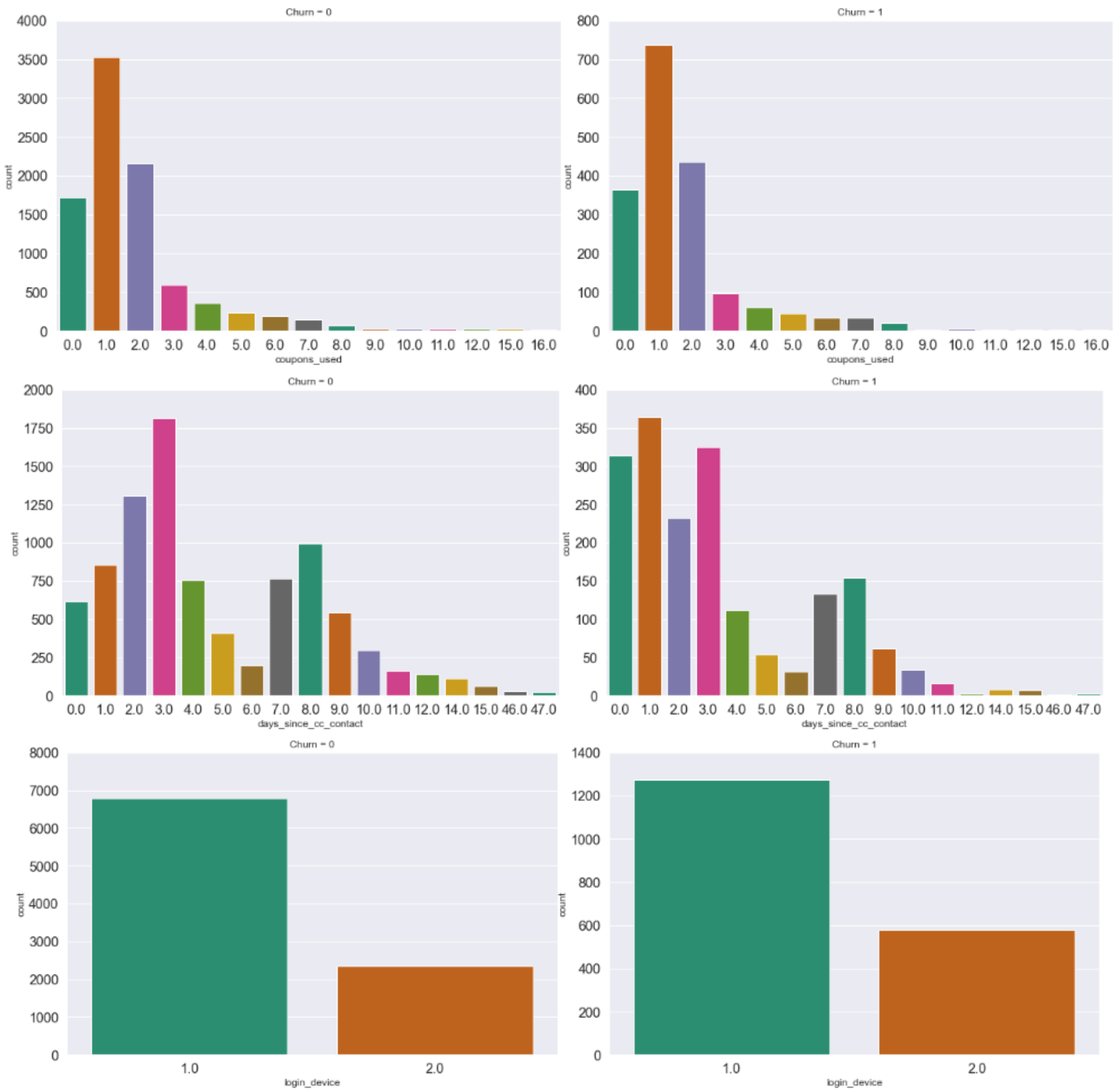


Fig.10 Count plot of features vs Target column

Insights:

Account Tenure

- The majority of customers who have churned have a tenure of 0 or 1 month.
- Conversely, customers with a tenure of more than 4 months are less likely to churn.

City Tier

- The city tier does not seem to have a significant impact on churn, although customers from tier 3 cities are more likely to churn compared to those from tier 1 or 2 cities.

Payment Method

- The analysis reveals that customers who have used Debit Card and Credit Card have a higher churn rate compared to those who have used other payment methods.
- On the other hand, customers who have used UPI or Cash On Delivery are less likely to churn.

Gender

- The churn rate is higher among male customers compared to female customers.

Service Score

- Surprisingly, customers who have given a service score of 3 are more likely to churn.
- This may indicate that these customers had a neutral feeling towards the service provided by the company.

Customers per Account

- Accounts with 5 customers are more likely to churn compared to those with other numbers of customers.

Account Segment

- Customers in the regular segment are less likely to churn compared to those in other segments.

CC Agent Score

- Most customers who have rated the customer service agent with a score of 5 have churned, while those who have given a score of 1 have not churned as much.
- Customers who have given a score of 3 have an equal chance of churning.

Marital Status

- Customers who are divorced are less likely to churn compared to those who are single or married.

Account Complaints in the last 12 Months

- The majority of customers who have registered a complaint have churned.

Login Device

- Customers who log in using a computer are more likely to churn compared to those who log in using a mobile device.

Multivariate Analysis



Fig.11 Pair plot of all the features

In order to perform multivariate analysis, we have used pair plot and correlation heat map to distinguish and analyze the relationship between the variables present in the dataset.

Pair plot shows the relationship between the variables in the form of scatterplot and the distribution of the variable in the form of histogram. The histogram shows that virtually all of the variables are normally distributed only few are moderately left skewed. Let us also confirm this with a heat map

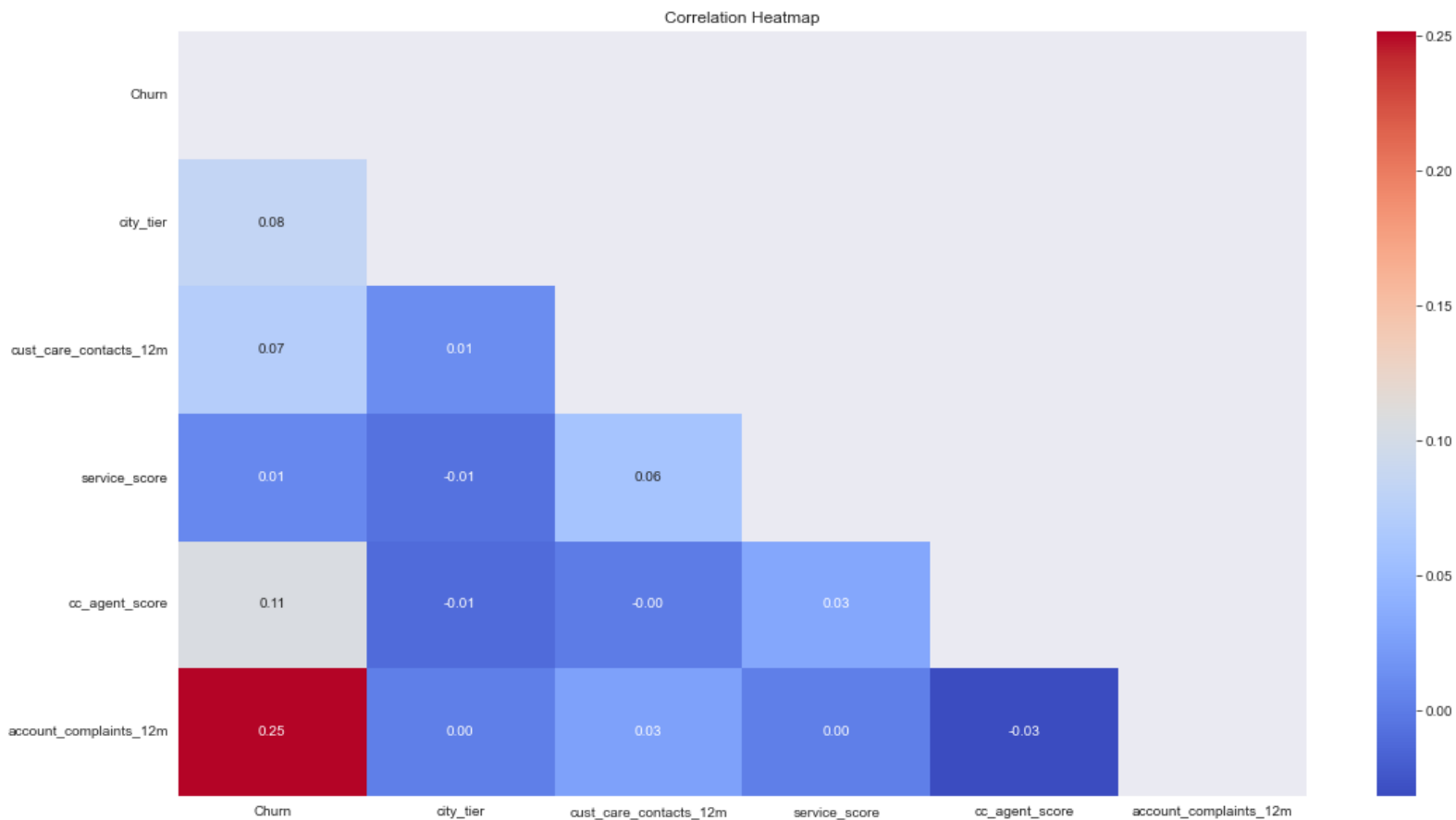


Fig.12 Correlation Heatmap

Correlation is a statistical measure that indicates how strongly two variables are related. The correlation value ranges from -1 to +1.

Inference:

- From the correlation plot, we can see that few **attributes of the data are moderately correlated** to each other. Correlation values near to 1 or -1 are highly positively correlated and

highly negatively correlated respectively. Correlation values near to 0 are not correlated to each other.

- The length of time someone has had an account is slightly negatively related to the target variable.
- The relationship between the following features is moderately positive:
 - Cashback and account segment
 - Number of customers per account and service score
 - The number of days since the last credit card contact and the number of coupons used

Business insights using clustering

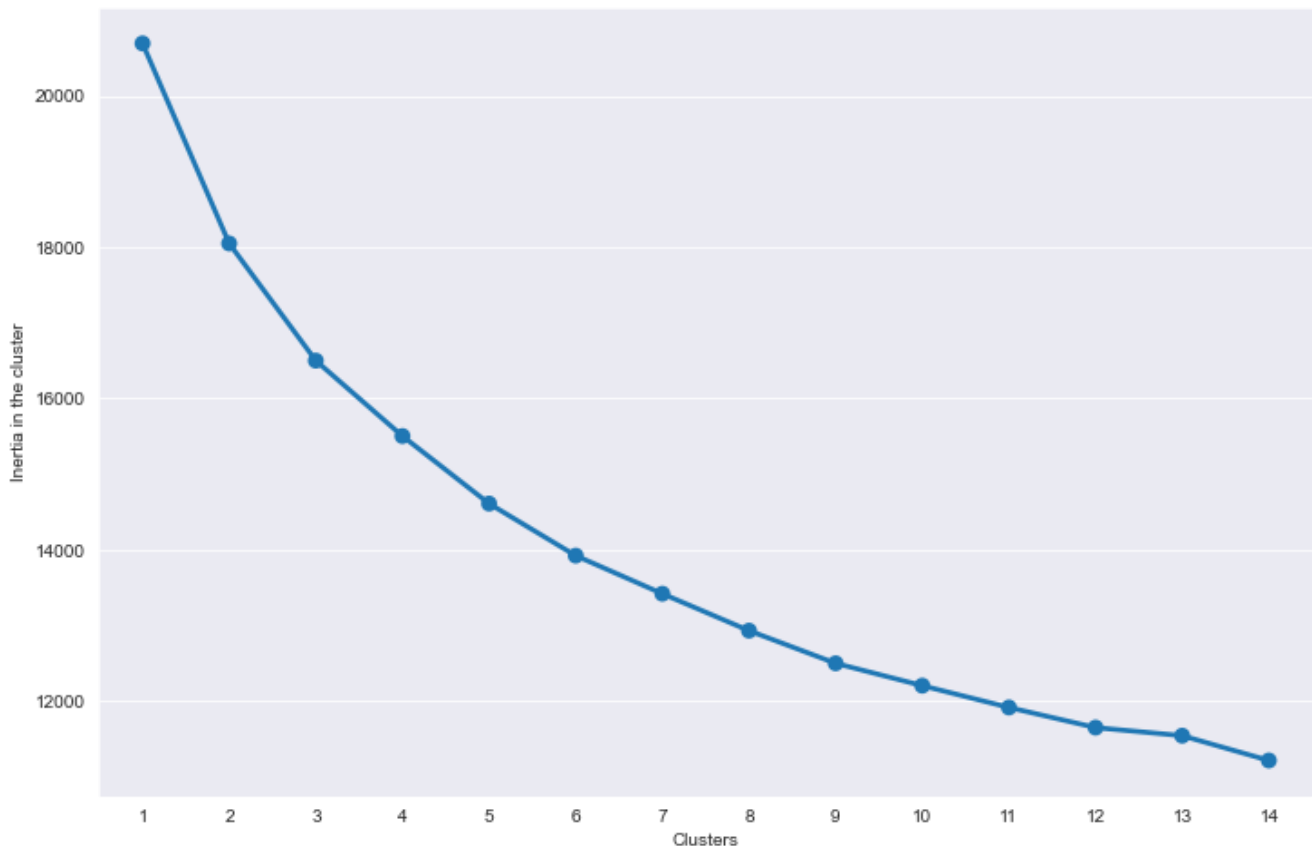


Fig.13 WSS Plot

The process of K-means clustering was employed to divide customers into three separate groups or clusters. The choice to create three clusters was based on the analysis of the inertia value obtained from the within-sum-of-squares (WSS) plots. It was found that the largest number of customers were in the second cluster and the smallest in the third cluster.

Cluster	0	1	2
Churn	1723	4030	5248
account_tenure	1723	4030	5248
city_tier	1723	4030	5248
cust_care_contacts_12m	1723	4030	5248
payment_method	1723	4030	5248
gender	1723	4030	5248
service_score	1723	4030	5248
customers_per_account	1723	4030	5248
account_segment	1723	4030	5248
cc_agent_score	1723	4030	5248
marital_Status	1723	4030	5248
revenue_per_month	1723	4030	5248
account_complaints_12m	1723	4030	5248
rev_growth_yoy	1723	4030	5248
coupons_used	1723	4030	5248
days_since_cc_contact	1723	4030	5248
cashback	1723	4030	5248
login_device	1723	4030	5248

Fig.14 Clustered Profile – 3 segments

However, despite these observations, the results of the clustering analysis did not provide a significant amount of useful information for further analysis

3. Data Cleaning & Pre-Processing

3.1 Removal of unwanted variables

Before the modeling exercise, various checks were performed to determine whether any columns can be removed from the dataset. The following checks were performed:

- Variables with unique values for each observation were dropped, as they would not contribute to the model (AccountID)
- Variables that remain constant for all or most observations were removed, as they do not add any strength to the prediction.
- Variables with null values in more than 25 to 30% of observations were considered for removal, but none of the columns had significant proportion of null values and hence no column was removed.
- Predictor variables with strong correlation with other predictor variables were evaluated, and one of the variables was dropped if necessary. However, there were no strong correlations observed, so no variable was dropped.

3.2 Addition of new features

The analysis conducted on the dataset did not reveal any significant gaps or issues that require the creation of new variables. Therefore, it has been determined that there is no need to generate any new variables for the purpose of modelling. The existing variables in the dataset contain sufficient information for prediction and decision making, and there is no indication that additional variables would significantly improve the accuracy or efficacy of the models. As a result, the focus will remain on analyzing and utilizing the existing variables in the dataset.

3.3 Missing value & Null value treatment

Checking null values in the dataset

```
Churn      0
account_tenure  102
city_tier   112
cust_care_contacts_12m  102
payment_method  109
gender      108
service_score  98
customers_per_account  112
account_segment  97
cc_agent_score  116
marital_Status  212
revenue_per_month  102
account_complaints_12m  357
rev_growth_yoy  0
coupons_used  0
days_since_cc_contact  357
cashback    471
login_device  221
dtype: int64
```

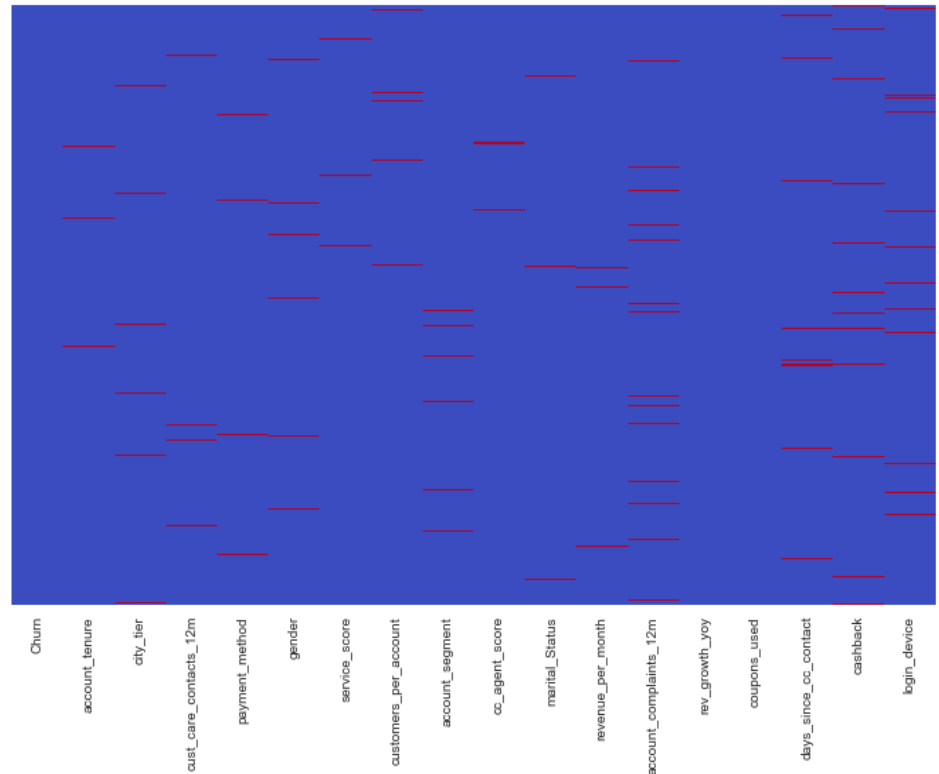


Fig.15 Null values in the dataset and its visualization

As can be seen from the above figure, apart from columns Churn, rev_growth_yoy and coupons_used, all the remaining columns contain null values and it must be treated

Each bar in the graph represents a column, and when there are no missing values, the bar appears fully colored in blue. The above graph indicates there are missing values in the dataset

Checking duplicate values in the dataset

Number of duplicate rows: 259

	Churn	account_tenure	city_tier	cust_care_contacts_12m	payment_method	gender	service_score	customers_per_account	account_segment	cc_agent_s
1347	1	0	1.0	8.0	UPI	Male	3.0	4	Regular Plus	
1395	1	11	1.0	6.0	Debit Card	Male	3.0	4	HNI	
1456	1	0	1.0	13.0	Debit Card	Male	3.0	5	Super	
1485	1	0	1.0	15.0	Credit Card	Female	2.0	3	Regular Plus	
1498	0	18	1.0	15.0	Debit Card	Male	2.0	3	Super	
1514	0	5	1.0	12.0	Debit Card	Male	2.0	3	Regular Plus	
1516	1	0	3.0	6.0	Cash on Delivery	Female	3.0	5	Regular Plus	
1542	0	15	1.0	27.0	Credit Card	Female	2.0	3	Super	
1547	0	7	1.0	16.0	Credit Card	Male	2.0	3	Regular Plus	
1568	0	11	3.0	9.0	Debit Card	Male	3.0	3	HNI	
1607	0	5	1.0	8.0	Cash on Delivery	Male	2.0	4	Super	
1619	0	0	1.0	9.0	Credit Card	Male	3.0	4	Regular Plus	

Fig.16 Duplicate values in the dataset

```
Percentage % Distribution of target variable among the duplicated records
0      83.011583
1      16.988417
Name: Churn, dtype: float64
```

Fig.17 Distribution of target variable in duplicate records

It has been found that there are 259 duplicate records in the data, but the distribution of the target variable in both the duplicate records and original records is nearly the same. As a result, removing the duplicates will not alter the distribution. The duplicates will be removed after Exploratory Data Analysis.

Removing Duplicates

Number of duplicate rows: 259

	Churn	account_tenure	city_tier	cust_care_contacts_12m	payment_method	gender	service_score	customers_per_account	account_segment	cc_ag
1347	1	0	1.0	8.0	UPI	Male	3.0	4	Regular Plus	
1395	1	11	1.0	6.0	Debit Card	Male	3.0	4	HNI	
1456	1	0	1.0	13.0	Debit Card	Male	3.0	5	Super	
1485	1	0	1.0	15.0	Credit Card	Female	2.0	3	Regular Plus	
1498	0	18	1.0	15.0	Debit Card	Male	2.0	3	Super	
1514	0	5	1.0	12.0	Debit Card	Male	2.0	3	Regular Plus	
1516	1	0	3.0	6.0	Cash on Delivery	Female	3.0	5	Regular Plus	
1542	0	15	1.0	27.0	Credit Card	Female	2.0	3	Super	
1547	0	7	1.0	16.0	Credit Card	Male	2.0	3	Regular Plus	

Before removing the duplicate rows (11260, 18)

After removing the duplicate rows (11001, 18)

Number of duplicate rows = 0

Fig.18 Duplicate Rows

There were 259 duplicate records found, but after analyzing the distribution of the target variable, it was concluded that removing these duplicates would not result in a significant change in the distribution of the target variable as the distribution is nearly the same in both the duplicates and original records.

Missing Value Treatment

```

Number of missing values in each column
Churn                                0
account_tenure                       102
city_tier                            112
cust_care_contacts_12m              102
payment_method                       109
gender                              108
service_score                        98
customers_per_account               112
account_segment                     97
cc_agent_score                      116
marital_Status                      211
revenue_per_month                   102
account_complaints_12m              357
rev_growth_yoy                      0
coupons_used                        0
days_since_cc_contact               357
cashback                            471
login_device                        221
dtype: int64

```

Fig.19 Null Values before pre-processing

Our data has anomalies in 17 out of 19 variables as per the EDA and missing values in 15 variables as per the above figure. To mitigate the effect of outliers, we used the median instead of the mean for continuous variables. For categorical variables, the mode was used to fill in the missing values. We treated each variable differently while handling missing values, taking into account its unique characteristics.

Column Name	Data Cleanup Needed	Description
AccountID	None	
Churn	None	
Tenure	Yes	Presence of "#" character
City_Tier	None	
CC_Contacted_LY	None	
Payment	None	
Gender	Yes	Presence of "M" & "F" as categories
Service_Score	None	
Account_user_count	Yes	Presence of "@" character
account_segment	Yes	Presence of "Regular +" & "Super +" as categories
CC_Agent_Score	None	
Marital_Status	None	
rev_per_month	Yes	Presence of "+" character
Complain_ly	None	
rev_growth_yoy	Yes \$	Presence of "\$" character
coupon_used_for_payment	Yes-\$, -, *, #	Presence of "\$, *, #" characters
Day_Since_CC_connect	Yes	Presence of "\$" character
Cashback	Yes	Presence of "\$" character

Fig.20 Data Cleanup Table

Upon examination of the individual observations of each column, we can observe that the some of the columns having miscellaneous characters or categories.

By replacing the miscellaneous characters with "nan" and then substituting "nan" with the median of the variable in case of numeric or mode in case of categorical, all incorrect data and null values are no longer present. The columns were then converted to their respective data types as future modeling processes require it to be in right format for pre-processing/encoding.

Outlier Treatment

	Outlier %		Outlier %
Churn	16.83	Churn	16.83
account_segment	14.68	account_segment	14.68
coupons_used	12.42	service_score	0.12
cashback	8.59	marital_Status	0.00
customers_per_account	6.73	cashback	0.00
revenue_per_month	1.68	days_since_cc_contact	0.00
account_tenure	1.26	coupons_used	0.00
days_since_cc_contact	1.16	rev_growth_yoy	0.00
cust_care_contacts_12m	0.38	account_complaints_12m	0.00
service_score	0.12	revenue_per_month	0.00
account_complaints_12m	0.00	cc_agent_score	0.00
rev_growth_yoy	0.00	account_tenure	0.00
cc_agent_score	0.00	customers_per_account	0.00
marital_Status	0.00	gender	0.00
gender	0.00	payment_method	0.00
payment_method	0.00	cust_care_contacts_12m	0.00
city_tier	0.00	city_tier	0.00
login_device	0.00	login_device	0.00

Fig.21 Outlier Proportion in each column before & after treatment

These columns have outliers: 'account_tenure', 'cust_care_contacts_12m', 'customers_per_account', 'revenue_per_month', 'coupons_used', 'days_since_cc_contact', and 'cashback', and they need to be handled

There's no need to handle the "account_segment" and "service_score" columns as they have categorical properties

Variable transformation

This difference in scale can impact the performance of some machine learning models, particularly those that are sensitive to the scale of the input features. To mitigate this issue, it's common to normalize or standardize the data so that all features are represented on a similar scale. There are various techniques for normalization and standardization, such as Min-Max scaling, Z-score normalization, and others. Choosing the right technique depends on the distribution of the data and the requirements of the specific machine learning model being used.

	min	max	std
Churn	0.000	1.000	0.374192
account_tenure	0.000	37.000	8.910558
city_tier	1.000	3.000	0.913608
cust_care_contacts_12m	4.000	41.000	8.574801
payment_method	1.000	5.000	1.234744
gender	1.000	2.000	0.488865
service_score	0.000	5.000	0.722796
customers_per_account	1.500	5.500	0.924405
account_segment	1.000	5.000	1.099918
cc_agent_score	1.000	5.000	1.373116
marital_Status	1.000	3.000	0.735143
revenue_per_month	1.000	13.000	2.880470
account_complaints_12m	0.000	1.000	0.447297
rev_growth_yoy	4.000	28.000	3.759624
coupons_used	0.000	3.500	1.104993
days_since_cc_contact	0.000	14.500	3.492314
cashback	72.995	273.195	43.944101
login_device	1.000	2.000	0.442208

	min	max	std
Churn	-0.449918	2.222626	1.000045
account_tenure	-1.153115	2.999451	1.000045
city_tier	-0.709243	1.479979	1.000045
cust_care_contacts_12m	-1.613778	2.701387	1.000045
payment_method	-2.360488	0.879200	1.000045
gender	-1.237690	0.807957	1.000045
service_score	-4.014781	2.903113	1.000045
customers_per_account	-2.391847	1.935456	1.000045
account_segment	-1.727489	1.909312	1.000045
cc_agent_score	-1.498316	1.414899	1.000045
marital_Status	-1.886251	0.834432	1.000045
revenue_per_month	-1.482048	2.684131	1.000045
account_complaints_12m	-0.618230	1.617520	1.000045
rev_growth_yoy	-3.248961	3.136945	1.000045
coupons_used	-1.341042	1.826544	1.000045
days_since_cc_contact	-1.305301	2.846863	1.000045
cashback	-2.382687	2.173308	1.000045
login_device	-0.602939	1.658544	1.000045

Fig.22 Descriptive summary before & after scaling

	Churn	account_tenure	city_tier	cust_care_contacts_12m	payment_method	gender	service_score	customers_per_account	account_segment	cc_agen
0	2.222626	-0.704189	1.479979	-1.380526	0.879200	-1.237690	0.135956	-0.769108	0.090912	-0
1	2.222626	-1.153115	-0.709243	-1.147274	-2.380488	0.807957	0.135956	0.312718	-0.818288	-0
2	2.222626	-1.153115	-0.709243	1.418500	0.879200	0.807957	-1.247623	0.312718	-0.818288	-0
3	2.222626	-1.153115	1.479979	-0.330891	0.879200	0.807957	-1.247623	0.312718	0.090912	1
4	2.222626	-1.153115	-0.709243	-0.680769	0.089279	0.807957	-1.247623	-0.769108	-0.818288	1

Fig.23 Sample rows of scaled data frame

Assumptions:

The following assumptions have been made for the encoding of categorical variables:

Gender:

Female: 1

Male: 2

Payment Method

Debit Card: 5

Credit Card: 4

E Wallet: 3

Cash on Delivery: 2

UPI: 1

Account Segment:

Regular: 1

Regular Plus: 2

Super: 3

Super Plus: 4

HNI: 5

Marital Status:

Divorced: 1

Single: 2

Married: 3

Login Devices:

Mobile: 1

Computer: 2

The following assumptions have been made for units of certain features:

Account Tenure: Represented in months

Revenue per Month: Average monthly revenue, measured in units of Rupees

Year-over-Year Revenue Growth: Measured in percentage (%)

Cashback: Measured in Rupees

4. Model Building

4.1 Algorithms applicable for the given problem

The need in this business case is to predict whether or not a given customer would churn. There are just two possible results for this binary classification problem. This is a supervised learning problem since a target variable, 'Churn,' must be predicted

Many algorithms can be employed to solve classification problems like these

- Logistic Regression, Linear Discriminant Analysis are examples of linear classification methods
- Non-linear classification algorithms include Decision trees, KNN, and Artificial Neural Networks
- Ensemble models like Random Forest, Adaboost, Gradient Boost
- These algorithms make assumptions about the data they are fitting. Algorithms will perform well or poorly depending on the nature of the data

4.2 Evaluation metrics for model comparison

The optimal model is determined by comparing the evaluation metrics for all of the models' train and test data.

When predicting customer churn, it's important to consider the cost associated with false positives (predicting a customer will churn when they actually won't) and false negatives (predicting a customer won't churn when they actually will). The optimal model should balance the trade-off

between precision (the proportion of predicted positives that are actually positive) and recall (the proportion of actual positives that are correctly predicted).

In this case, precision measures the proportion of customers who are predicted to churn and actually do churn. A high precision value indicates that the model is identifying true positives effectively and minimizing false positives. False positives are costly for the DTH company, as they may result in unnecessary retention efforts for customers who would not have churned otherwise.

Recall measures the proportion of customers who actually churn and are correctly identified by the model. A high recall value indicates that the model is effectively identifying true positives and minimizing false negatives. False negatives are also costly for the DTH company, as they may result in missed opportunities for retention efforts.

Therefore, **the optimal model should focus on maximizing recall first and then precision**. The **F1-score**, which is the harmonic mean of precision and recall, is a **useful metric to evaluate the trade-off between these two metrics and to select the best model**. Based on recall values, optimum model should be finalized.

Splitting of data into train and test sets

The dataset is divided into a train and test dataset using a **ratio of 0.3**. This means that only **30%** of the dataset is utilized for **testing**, with the remaining **70%** being used to **train the model**.

```

-----
X_train      : (7700, 17)
X_test       : (3301, 17)
Y_train      : (7700,)
Y_test       : (3301,)
Total Observations : 11001
-----

```

Fig.24 Splitting of train and test data

The data set presented has an imbalance. The target variable, "Churn," has a large disparity in its categorical count with 84% for "0" and 17% for "1."

To address this imbalance, we can apply the **SMOTE (Synthetic Minority Over-sampling Technique)** method, which generates additional data points to balance the distribution.

It's important to note that SMOTE should only be applied to the training dataset and not the test dataset. The data was divided into a training and testing dataset in a 70:30 ratio, but this ratio may be adjusted later as needed.

===== Class= 0.0, n=6406 (83.19%) Class= 1.0, n=1294 (16.81%) =====	===== Class= 0.0, n=6406 (50.00%) Class= 1.0, n=6406 (50.00%) =====
--	--

Fig.25 Distribution of Target Variable in Train Dataset before & after SMOTE

For certain models, data obtained using SMOTE is used to check the model performance

4.3 Improving model performance

The model's performance was improved by trying different algorithms such as linear, non-linear, and ensemble methods. The initial model for each algorithm was set as the base model with default hyperparameters. The hyperparameters were then tuned using Grid Search CV, which tries out multiple parameter options for each hyperparameter to achieve better performance. Different data treatments were applied such as Smote resampled/not resampled and outlier treated/not treated, and their effects on model performance were observed for some of the models. Ensemble methods such as Random Forest, Adaboost, and Gradient Boost were also used, and their hyperparameters were tuned. The tables show all the algorithms tried, various model tuning trials done, and their performance in both train and test datasets. The best model for each algorithm is highlighted in yellow.

4.4 Comparison of metrics for all the models

Summary of the performance metrics of all models

Model	Train Data					Test Data				
	Accuracy	Precision	Recall	F1	AUC	Accuracy	Precision	Recall	F1	AUC
Logistic Regression with Smote	0.78	0.42	0.82	0.56	0.87	0.78	0.42	0.82	0.55	0.85
Logistic Regression without Smote	0.88	0.75	0.41	0.53	0.87	0.88	0.75	0.44	0.56	0.85
Logistic Regression - Tuned	0.88	0.74	0.43	0.54	0.87	0.88	0.74	0.46	0.57	0.85
LDA	0.87	0.73	0.39	0.51	0.86	0.87	0.73	0.41	0.53	0.84
Ada Boost	0.9	0.76	0.58	0.66	0.91	0.9	0.74	0.6	0.66	0.90
Ada Boost - Tuned	1.00	1.00	1.00	1.00	1.00	0.97	0.96	0.86	0.91	0.93
Gradient Boost	0.92	0.85	0.63	0.72	1.0	0.91	0.79	0.59	0.68	0.93
Gradient Boost - Tuned	1.0	1.0	0.99	1.0	1.0	0.96	0.93	0.83	0.88	0.99
XG Boost	1.0	1.0	1.0	1.0	1.0	0.96	0.91	0.86	0.88	0.99
XG Boost - Tuned	1.0	1.0	1.0	1.0	1.0	0.98	0.95	0.91	0.93	0.99
Random Forest	1.0	1.0	1.0	1.0	1.0	0.96	0.96	0.82	0.88	0.99
Random Forest - Tuned	0.74	0.9	0.63	0.74	0.95	0.69	0.83	0.59	0.69	0.93
ANN	0.94	0.85	0.79	0.82	0.97	0.92	0.78	0.74	0.76	0.95
ANN - Tuned	0.98	0.98	0.98	0.98	0.99	0.88	0.87	0.89	0.88	0.98
KNN	0.99	0.97	0.94	0.96	0.99	0.96	0.89	0.85	0.87	0.96
KNN - Tuned	0.99	0.97	0.94	0.96	0.99	0.96	0.89	0.85	0.87	0.97
Bagging with Decision Tree	1.0	1.0	1.0	1.0	1.0	0.97	0.94	0.85	0.89	0.98

Fig.26 Performance metrics summary for all models

For various train and test data sets, we have obtained accuracy, model parameters including recall, precision, f1 score, ROC curve, and AUC curve based on the aforementioned model implementations. Here, it is necessary to compare the predictions' results on the train and test sets in order to identify the most optimal/best model

	Model	Train_Accuracy	Test_Accuracy	Train_Recall	Test_Recall	Train_Precision	Test_Precision	Train_f1	Test_f1
9	XGBoost-Tuned	1.00	0.98	1.00	0.91	1.00	0.95	1.00	0.93
5	AdaBoost-Tuned	1.00	0.97	1.00	0.86	1.00	0.96	1.00	0.91
16	Bagging with Decision Tree	1.00	0.97	1.00	0.85	1.00	0.94	1.00	0.89
7	Gradient Boosting-Tuned	1.00	0.96	0.99	0.83	1.00	0.93	1.00	0.88
13	ANN-Tuned	0.98	0.88	0.98	0.89	0.98	0.87	0.98	0.88
10	Random Forest	1.00	0.96	1.00	0.82	1.00	0.96	1.00	0.88
8	XGBoost	1.00	0.96	1.00	0.86	1.00	0.91	1.00	0.88
14	KNN	0.99	0.96	0.94	0.85	0.97	0.89	0.96	0.87
15	KNN-Tuned	0.99	0.96	0.94	0.85	0.97	0.89	0.96	0.87
12	ANN	0.94	0.92	0.79	0.74	0.85	0.78	0.82	0.76
11	Random Forest-Tuned	0.74	0.69	0.63	0.59	0.90	0.83	0.74	0.69
6	Gradient Boost	0.92	0.91	0.63	0.59	0.85	0.79	0.72	0.68
4	AdaBoost	0.90	0.90	0.58	0.60	0.76	0.74	0.66	0.66
2	Logistic Regression-Tuned	0.88	0.88	0.43	0.46	0.74	0.74	0.54	0.57
1	Logistic Regression Without Smote	0.88	0.88	0.41	0.44	0.75	0.75	0.53	0.56
0	Logistic Regression with Smote	0.78	0.78	0.82	0.82	0.42	0.42	0.56	0.55
3	LDA	0.87	0.87	0.39	0.41	0.73	0.73	0.51	0.53

Fig.27 Performance metrics – Sorted by Test F1-Score

	Model	Train_Accuracy	Test_Accuracy	Train_Recall	Test_Recall	Train_Precision	Test_Precision	Train_f1	Test_f1
9	XGBoost-Tuned	1.00	0.98	1.00	0.91	1.00	0.95	1.00	0.93
13	ANN-Tuned	0.98	0.88	0.98	0.89	0.98	0.87	0.98	0.88
8	XGBoost	1.00	0.96	1.00	0.86	1.00	0.91	1.00	0.88
5	AdaBoost-Tuned	1.00	0.97	1.00	0.86	1.00	0.96	1.00	0.91
15	KNN-Tuned	0.99	0.96	0.94	0.85	0.97	0.89	0.96	0.87
14	KNN	0.99	0.96	0.94	0.85	0.97	0.89	0.96	0.87
16	Bagging with Decision Tree	1.00	0.97	1.00	0.85	1.00	0.94	1.00	0.89
7	Gradient Boosting-Tuned	1.00	0.96	0.99	0.83	1.00	0.93	1.00	0.88
10	Random Forest	1.00	0.96	1.00	0.82	1.00	0.96	1.00	0.88
0	Logistic Regression with Smote	0.78	0.78	0.82	0.82	0.42	0.42	0.56	0.55
12	ANN	0.94	0.92	0.79	0.74	0.85	0.78	0.82	0.76
4	AdaBoost	0.90	0.90	0.58	0.60	0.76	0.74	0.66	0.66
6	Gradient Boost	0.92	0.91	0.63	0.59	0.85	0.79	0.72	0.68
11	Random Forest-Tuned	0.74	0.69	0.63	0.59	0.90	0.83	0.74	0.69
2	Logistic Regression-Tuned	0.88	0.88	0.43	0.46	0.74	0.74	0.54	0.57
1	Logistic Regression Without Smote	0.88	0.88	0.41	0.44	0.75	0.75	0.53	0.56
3	LDA	0.87	0.87	0.39	0.41	0.73	0.73	0.51	0.53

Fig.28 Performance metrics – Sorted by Test Recall

	Model	Train_Accuracy	Test_Accuracy	Train_Recall	Test_Recall	Train_Precision	Test_Precision	Train_f1	Test_f1
5	AdaBoost-Tuned	1.00	0.97	1.00	0.86	1.00	0.96	1.00	0.91
10	Random Forest	1.00	0.96	1.00	0.82	1.00	0.96	1.00	0.88
9	XGBoost-Tuned	1.00	0.98	1.00	0.91	1.00	0.95	1.00	0.93
16	Bagging with Decision Tree	1.00	0.97	1.00	0.85	1.00	0.94	1.00	0.89
7	Gradient Boosting-Tuned	1.00	0.96	0.99	0.83	1.00	0.93	1.00	0.88
8	XGBoost	1.00	0.96	1.00	0.86	1.00	0.91	1.00	0.88
14	KNN	0.99	0.96	0.94	0.85	0.97	0.89	0.96	0.87
15	KNN-Tuned	0.99	0.96	0.94	0.85	0.97	0.89	0.96	0.87
13	ANN-Tuned	0.98	0.88	0.98	0.89	0.98	0.87	0.98	0.88
11	Random Forest-Tuned	0.74	0.69	0.63	0.59	0.90	0.83	0.74	0.69
6	Gradient Boost	0.92	0.91	0.63	0.59	0.85	0.79	0.72	0.68
12	ANN	0.94	0.92	0.79	0.74	0.85	0.78	0.82	0.76
1	Logistic Regression Without Smote	0.88	0.88	0.41	0.44	0.75	0.75	0.53	0.56
4	AdaBoost	0.90	0.90	0.58	0.60	0.76	0.74	0.66	0.66
2	Logistic Regression-Tuned	0.88	0.88	0.43	0.46	0.74	0.74	0.54	0.57
3	LDA	0.87	0.87	0.39	0.41	0.73	0.73	0.51	0.53
0	Logistic Regression with Smote	0.78	0.78	0.82	0.82	0.42	0.42	0.56	0.55

Fig.29 Performance metrics – Sorted by Test Precision

Inference:

- In customer churn prediction, the **first priority is generally recall**, as the goal is to identify as many customers who are likely to churn as possible, even if there are some false positives (customers who are predicted to churn but actually do not). This is because the cost of losing a customer is usually higher than the cost of retaining a customer who was predicted to churn but did not.
- That being said, **precision is also important**, as it measures the proportion of correctly predicted churners out of all predicted churners. A high precision means that the company can be confident that the customers identified as likely to churn are indeed likely to churn, which can help them to take appropriate actions to retain these customers.
- Therefore, while recall is generally the first priority in customer churn prediction, it is important to strike a balance between recall and precision to ensure that the model is effective in predicting customer churn and helping the company to retain valuable customers.
- **Based on the above criteria, tuned XG Boost is the best model for our problem statement since it has the highest f1-score, recall, and precision.**

Overall, it has the best mix of recall and precision and will be suitable for use in production.

5. Model Validation

- The dataset was divided into two sets: the training set (70%) and the test set (30%) for validation
- All the models were trained on the training set and evaluated using various performance metrics such as accuracy, F1-score, precision, recall, confusion matrix, ROC curve, and AUC
- These metrics were recorded for both the training and test sets. If the model was found to be overfitting on the training set and the difference in performance between the training and test sets was not greater than 10%, then a validation of the test scores was done using 5-fold on the entire dataset. The scores obtained from cross-validation were compared to the test set scores to ensure that the model had not overfitted on the training set.

5.1 Criteria for best performing model

- The primary criteria for evaluating model performance in this case study are Precision, Recall, and F1-score for the minority class (churners), as the dataset has a class imbalance with only 16.8% churns.
- Precision measures the proportion of actual churners among the customers identified as churners by the model, while Recall measures the proportion of actual churners that the model correctly identifies.
- Recall is the primary criteria for evaluation followed by precision. The F1-score is the harmonic mean of Precision and Recall and is used as a single metric to optimize both Precision and Recall.
- The secondary criteria for evaluation are Accuracy and AUC-ROC, although they are not as important as Precision, Recall, and F1-score.
- Accuracy is not a good metric to evaluate models with imbalanced classes, but it is still recorded for completeness.
- AUROC is used to evaluate model performance based on the area under the ROC curve, which summarizes the model's performance on the positive class.

5.2 Why Tuned XG Boost Model is the Optimal Model?

Tuned XG Boost Model

Model	Train Data					Test Data				
	Accuracy	Precision	Recall	F1	AUC	Accuracy	Precision	Recall	F1	AUC
XG Boost - Tuned	1.0	1.0	1.0	1.0	1.0	0.98	0.95	0.91	0.93	0.99

Fig.30 Performance metrics of Optimal Model

The model's hyperparameters were tuned using GridSearchCV function from Sklearn library and model constructed using the best parameters selected. The tuning parameters for Gridsearch and individual scoring metrics for each parameter grid have been provided in Appendix – Annexure A.

Model Interpretations:

- Accuracy:** Accuracy measures the proportion of correct predictions among all predictions made. It is calculated as the number of correct predictions divided by the total number of predictions made. In this case, the accuracy on the training set is 1.0, indicating that the model correctly predicted all instances in the training set. The accuracy on the test set is 0.976, which is also very high and **indicates that the model is performing well on unseen data.**
- Recall:** Recall measures the proportion of true positives that were correctly identified by the model among all actual positive instances in the dataset. In other words, it measures the ability of the model to identify all positive instances. In this case, the recall on the training set is 1.0, indicating that the model correctly identified all instances of customer churn in the training set. The recall on the test set is 0.907, which is also quite high and **indicates that the model is able to identify a large proportion of actual positive instances.**
- Precision:** Precision measures the proportion of true positives that were correctly identified by the model among all instances that the model predicted as positive. In other words, it measures the ability of the model to avoid false positives. In this case, the precision on the

training set is 1.0, indicating that the model did not make any false positive predictions in the training set. The precision on the test set is 0.948, which is also quite high and **indicates that the model is able to avoid false positive predictions to a large extent.**

- **F1 Score:** F1 score is the harmonic mean of precision and recall. It provides a balance between precision and recall and is a useful metric when the classes are imbalanced. In this case, the F1 score on the training set is 1.0, indicating that the model is performing very well in terms of precision and recall. The F1 score on the test set is 0.927, which is also quite high and indicates that the **model is able to balance precision and recall to a large extent.**
- **AUC:** AUC (Area Under the Curve) is a metric used to evaluate the performance of a binary classification model. It measures the ability of the model to distinguish between positive and negative instances. A higher AUC indicates that the model is better at distinguishing between the two classes. In this case, the AUC on the training set is 1.0, indicating that the model is able to distinguish between positive and negative instances perfectly. The AUC on the test set is 0.99, which is also very high and indicates that the **model is able to distinguish between positive and negative instances with a high degree of accuracy.**

Overall, the performance metrics of the tuned XG boost model are very good, indicating that **it is a strong model for predicting customer churn.** The high accuracy, recall, precision, and F1 score, as well as the high AUC, indicate that the model is able to make accurate predictions and avoid false positives and false negatives. This is very important in the context of customer churn prediction, as false positives can lead to unnecessary customer retention efforts and false negatives can result in the loss of valuable customers.

5.3 Business Implications

Using the tuned XG boost model to predict customer churn can have significant business implications for a company. The model has high accuracy, recall, precision, and F1 scores, indicating that it is **highly effective at predicting customer churn.** Here are some of the business implications of using this model:

- **Increase customer retention:** By identifying customers who are likely to churn, a company can take proactive steps to retain them. The **model can help the company identify customers**

who are at risk of leaving and take steps to address their concerns. This could include providing targeted offers or incentives, improving customer service, or addressing product or service issues.

- **Reduce customer acquisition costs:** Customer acquisition is expensive, and retaining existing customers is often more cost-effective than acquiring new ones. By using the model to identify customers who are at risk of leaving, the company can take steps to retain them, reducing the need to acquire new customers.
- **Improve customer satisfaction:** By identifying and addressing issues that are causing customers to churn, the company can improve overall customer satisfaction. This could lead to increased loyalty and repeat business, as well as positive word-of-mouth referrals.
- **Improve marketing effectiveness:** The model can also help the company improve the effectiveness of its marketing efforts. By identifying the characteristics of customers who are most likely to churn, the company can tailor its marketing campaigns to target these customers with specific messaging and offers.
- **Increase revenue:** Retaining customers who are at risk of churning can help the company increase revenue. These customers may be more likely to purchase additional products or services or to remain loyal to the company over the long term.

Overall, the tuned XG Boost model is a powerful tool that can help a company improve customer retention, reduce customer acquisition costs, improve customer satisfaction, and increase revenue. By leveraging the insights provided by the model, a company can take proactive steps to retain customers and build long-term relationships that drive business success.

5.4 How Can Business Use these metrics?

Here's how to explain these metrics to a business person:

- A precision of 0.95 implies that out of 100 customers that the model has identified as churned, 95 would actually churn and 5 would not. Any marketing budget allocated for a targeted campaign for retention of these customers would be most optimally utilized as only 5/100 customers would be incorrectly identified as churned.

- A recall of 0.91 implies that for 100 customers who actually churn, the model would have identified 91 as churn and 9 as not-churn. This would mean that the campaign would target these 91 customers and there is scope for retaining these customers and missing out on 9 customers.
- Based on the customer base for which prediction needs to be done, business can use the above to project the numbers of customers who would churn and how many the model would identify and miss.
- That could be further used to come up with per customer budget (if total budget for retention campaign is known) so that appropriate offers can be designed for each customer.
- If per customer budget is known, cost projections for retention campaign can be calculated.
- If a limited budget is available and not all customers can be covered, the model can also provide the probability of churning so that high probability customers can be targeted first. Also, a different perspective to this problem would be to do a segmentation and target the high-value customer segment.

5.5 Model Interpretations from Feature Importance

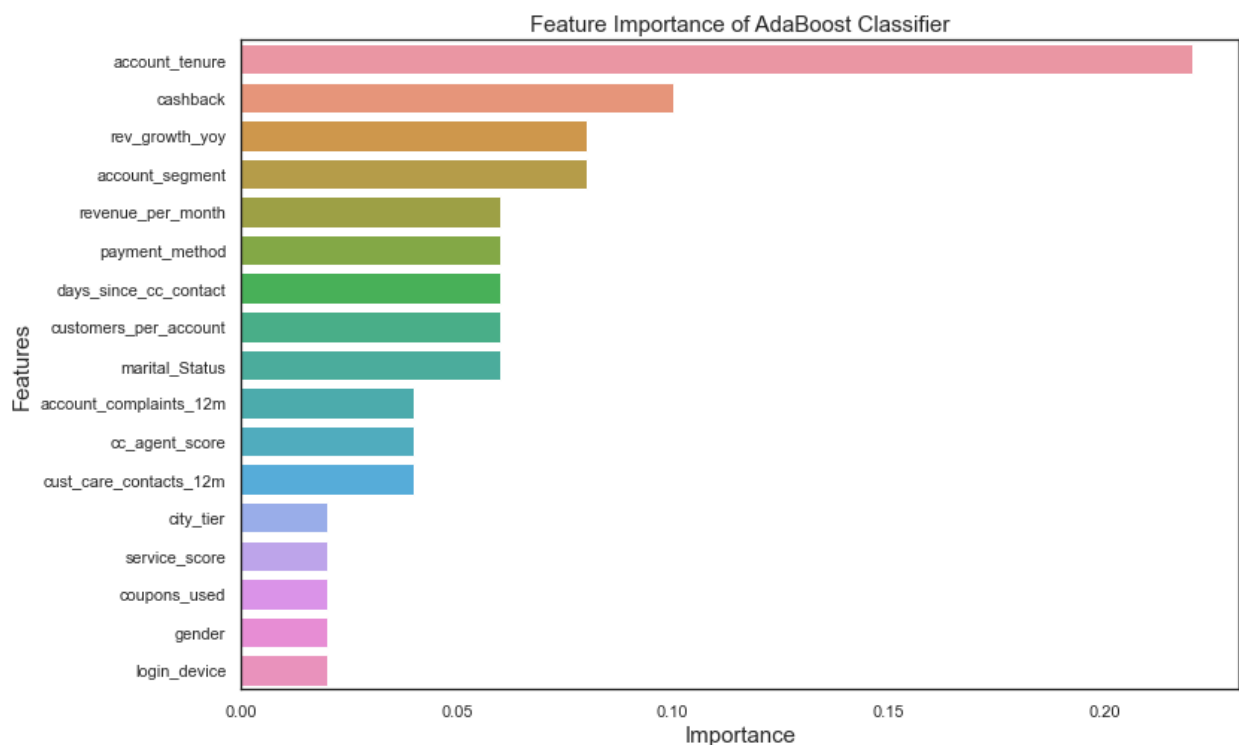


Fig.31 Ada Boost Feature Importance

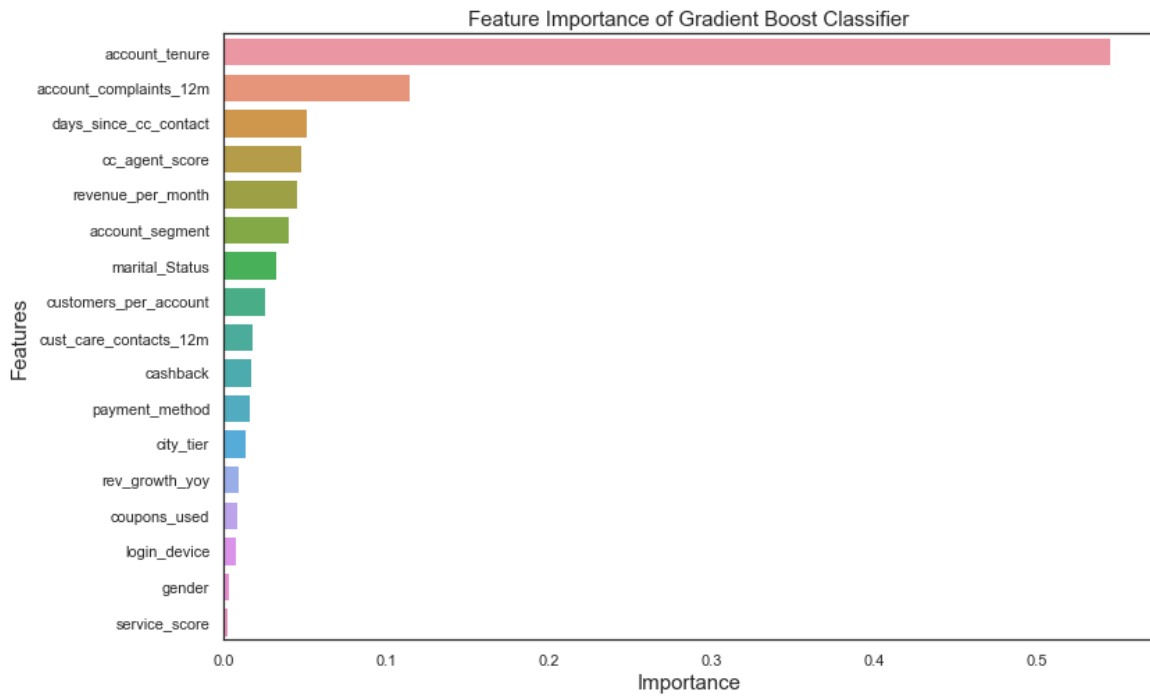


Fig.32 Gradient Boost Feature Importance

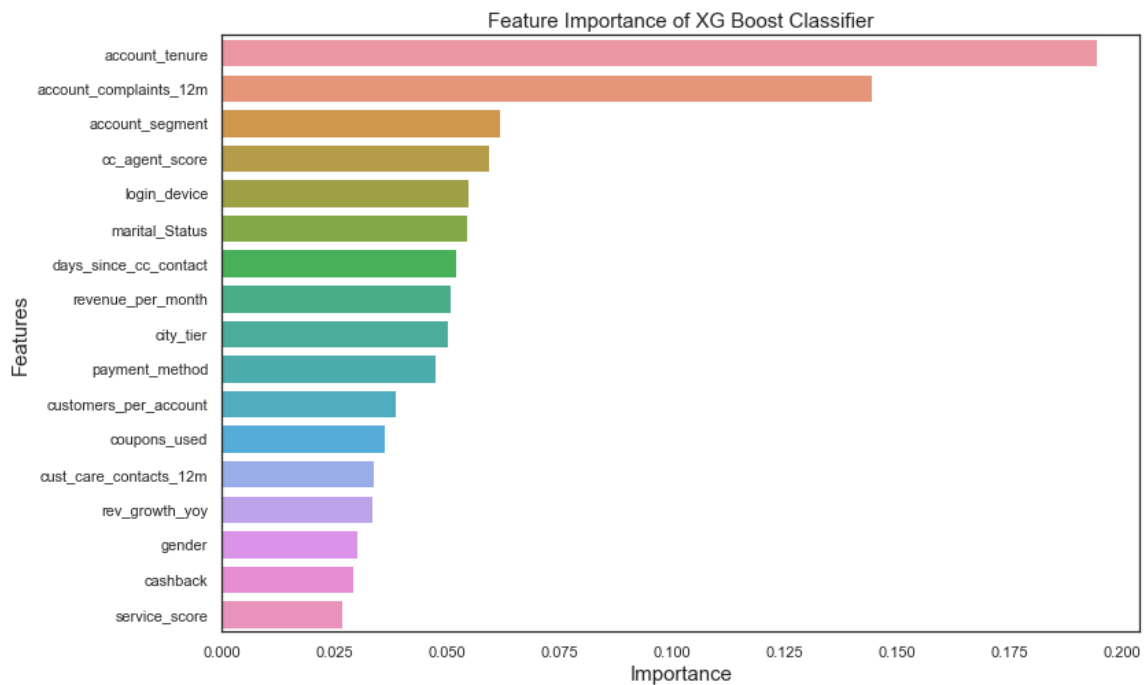


Fig.33 XG Boost Feature Importance

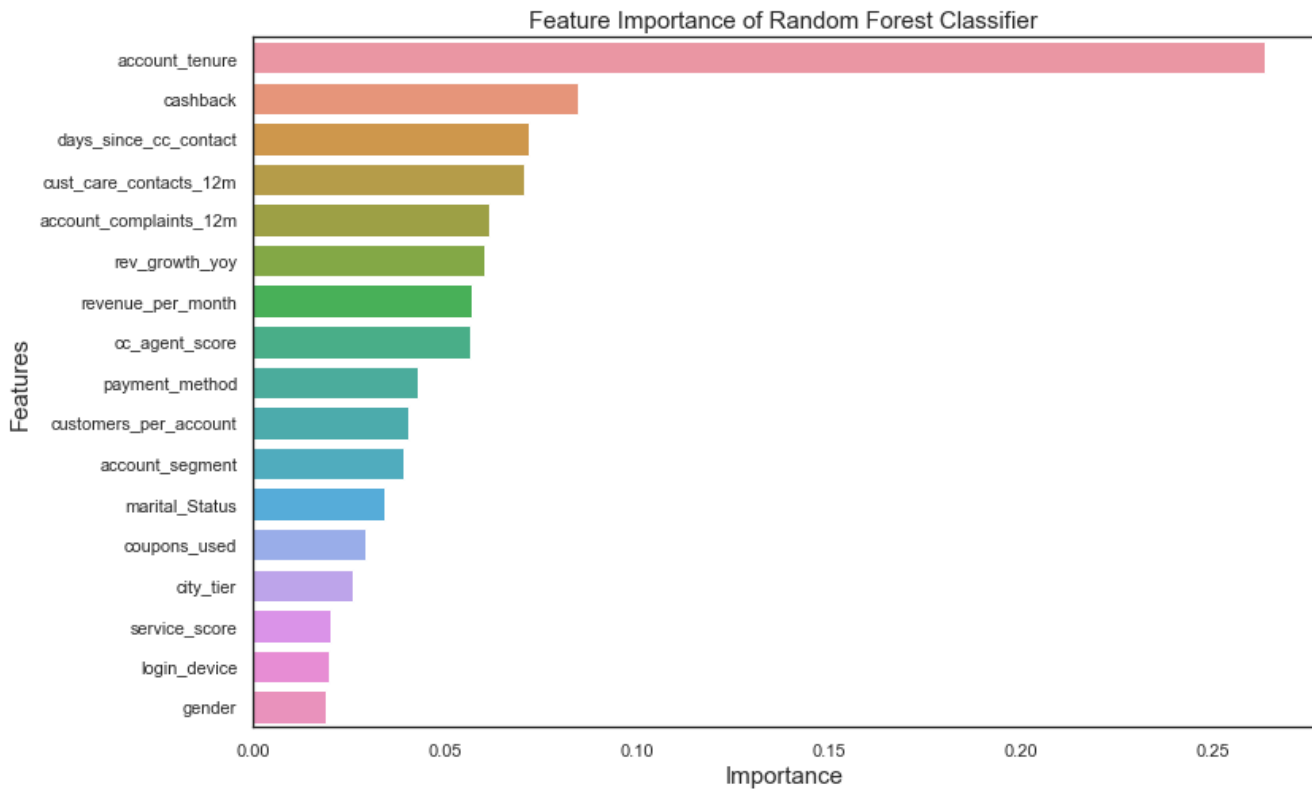


Fig.34 Random Forest Feature Importance

Inference:

The feature importance graphs from the above models suggests that the top 5 features affecting customer churn in the DTH company are account tenure, account complaints in last 12 months, account segmentation, customer satisfaction score with customer care service, and preferred login device.

- The first feature, **account tenure**, indicates how long the customer has been with the company. This feature is important because it suggests that customers who have been with the company for a longer period are less likely to churn. The business can use this information to identify customers who have been with the company for a shorter period and may be at a higher risk of churning. The company can then focus on providing better services and incentives to retain these customers.

- The second feature, **account complaints in the last 12 months**, indicates the number of times all customers in the account have contacted customer care in the last year. This feature suggests that customers who have had more complaints in the last year are more likely to churn. The company can use this information to identify accounts with a high number of complaints and take corrective measures to improve customer service.
- The third feature, **account segmentation**, suggests that customers in different segments based on spend have different churn rates. The company can use this information to segment their customers based on their spend and offer customized plans and services to retain customers in each segment.
- The fourth feature, **customer satisfaction score with customer care service**, suggests that customers who are more satisfied with customer care service are less likely to churn. The company can use this information to focus on improving customer care service, ensuring quick resolution of customer complaints, and increasing customer satisfaction levels.
- The fifth feature, **preferred login device**, suggests that customers who prefer a specific login device are less likely to churn. The company can use this information to offer a seamless experience across all devices and ensure that customers can access their services easily from their preferred device.

Overall, the feature importance analysis can help the business identify the most important factors affecting customer churn and take proactive steps to improve customer retention.

6. Business Recommendations & Insights

Higher Churn in New Customers

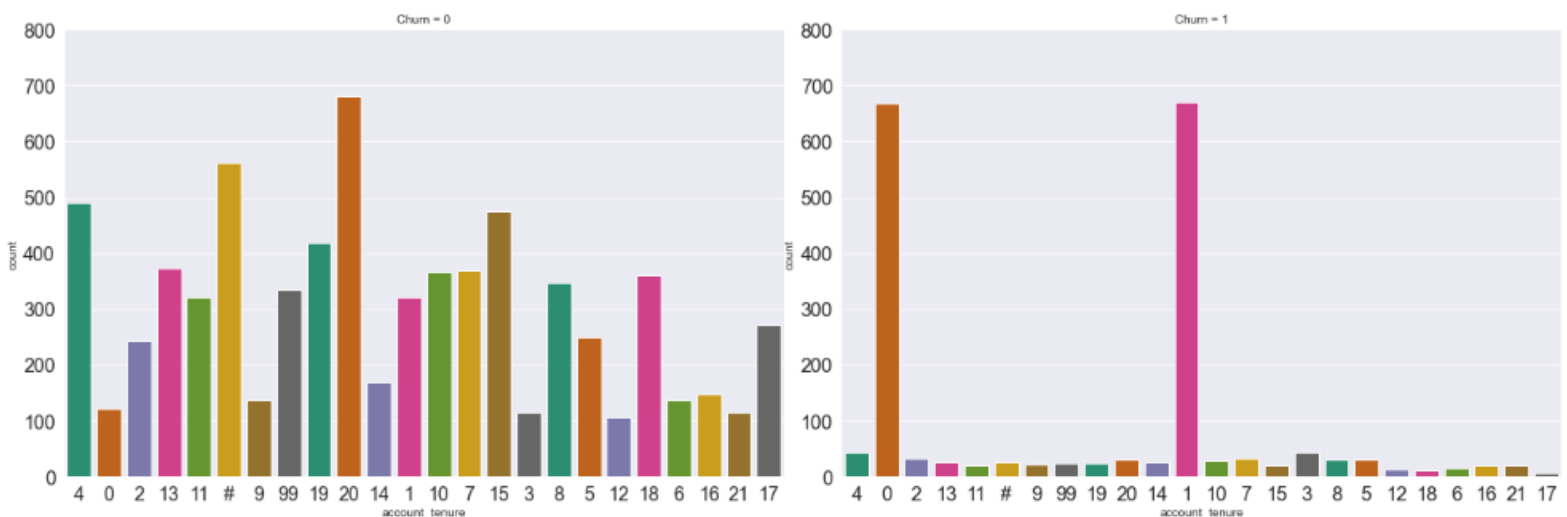


Fig.35 Random Forest Feature Importance

Insight: Churn is highest during the initial period of 0 to 2 month, it indicates that customers are not satisfied with the service or the overall experience during this period

Recommendation:

- Improve the onboarding process
- Ensure that the service quality during the initial period is of high quality
- Collect feedback from customers during the initial period to identify common issues and address them proactively

Payment Method

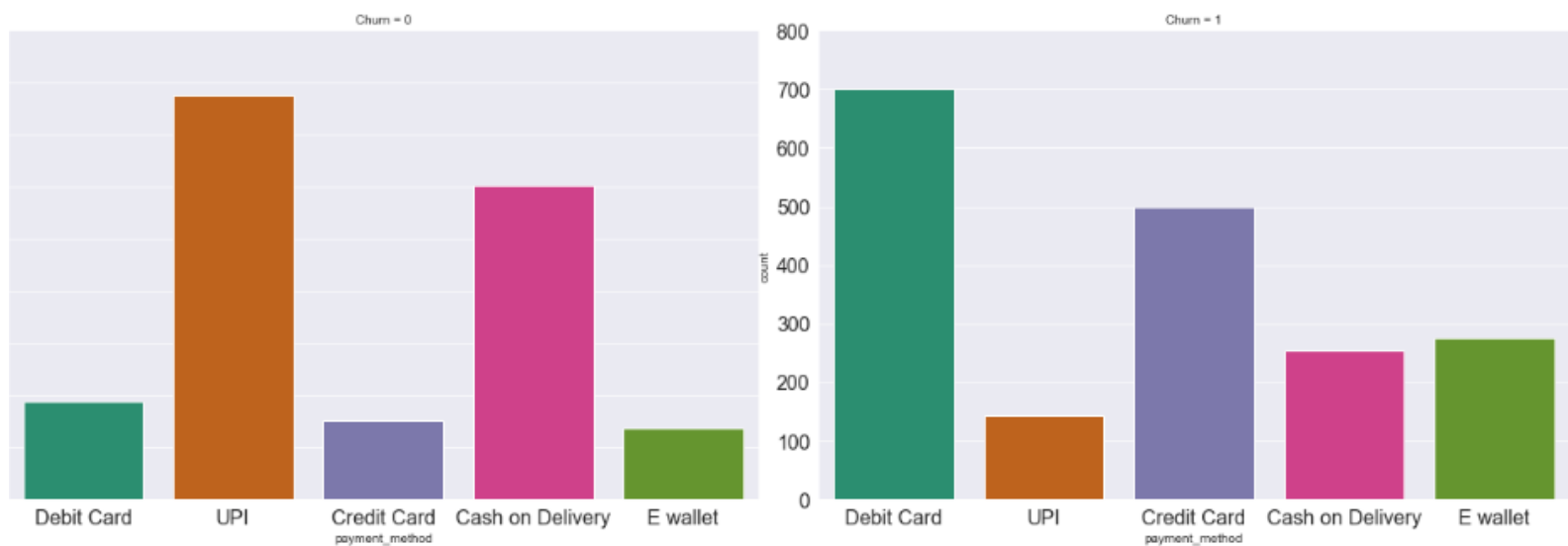


Fig.36 Random Forest Feature Importance

Insight: Using Debit Card and Credit Card have a higher churn rate than those using other payment methods. Conversely, customers using UPI or Cash on Delivery are less likely to churn

Recommendation:

- Asses payment security for Debit Card & Credit Card
- Simplify payment process
- Offer cashback or discounts
- Continue to offer UPI & COD payment methods

Coupons Used

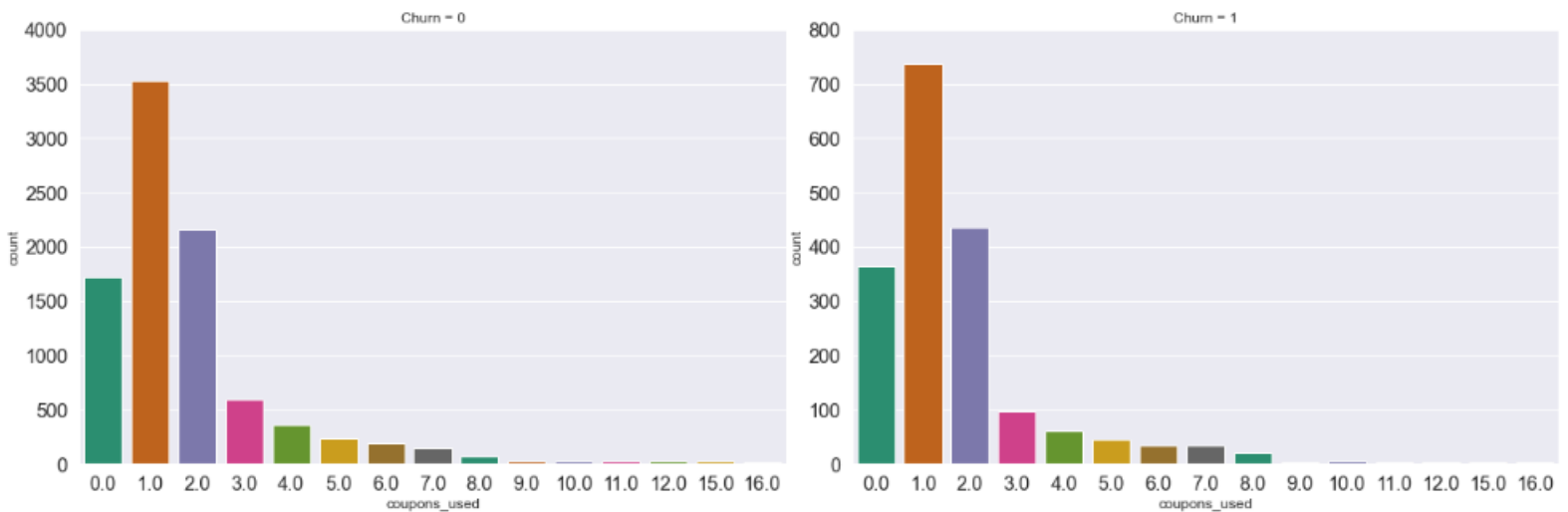


Fig.37 Random Forest Feature Importance

Insight: Customers who were not provided any coupons or received less than 2 coupons are more inclined towards churning

Recommendation:

- Increase coupon distribution
- Conduct A/B testing to determine which types of coupons are most effective in reducing churn. Use this information to optimize coupon distribution.

Insights from other features

Gender:

- The churn rate is higher among male customers compared to female customers

Service Score:

- Surprisingly, customers who have given a service score of 3 are more likely to churn

- This may indicate that these customers had a neutral feeling towards the service provided by the company

City Tier:

- The city tier does not seem to have a significant impact on churn, although customers from tier 3 cities are more likely to churn compared to those from tier 1 or 2 cities.

Customers per Account:

- Accounts with 3 customers or more are likely to churn compared to those with other numbers of customers

Login Device:

- Customers who log in using a mobile are more likely to churn compared to those who log in using a computer

Account Segment:

- Customers in the regular plus segment are more likely to churn compared to those in other segments

CC Agent Score:

- Most customers who have rated the customer service agent with a score of 5 have churned, while those who have given a score of 1 have not churned as much
- Customers who have given a score of 3 have an equal chance of churning

Marital Status:

- Customers who are divorced are less likely to churn compared to those who are single or married

Account Complaints in the last 12 Months:

- The majority of customers who have registered a complaint have churned

Business Recommendations

- The DTH company can **provide better customer service to male customers** to increase their satisfaction levels. This can include training customer service representatives to address specific concerns that male customers might have. Company should **continue to provide excellent service to all customers, regardless of their gender, to maintain high levels of customer satisfaction and reduce churn overall**
- **Improve the customer service training** and address complaints in a timely and effective manner. **Train customer service representatives** and equip them with the necessary tools
- Analyze customer feedback. **If possible, perform sentiment analysis on the given feedbacks. Invest in technology to streamline customer service processes** for faster and more efficient service
- Company must **ensure that accounts with 3 or more customers receive excellent customer service** to increase their satisfaction levels. Offer loyalty programs to incentivize them to continue using the service.
- Should focus on **improving the user experience** on their website, especially **for customers who log in using a mobile**. This could involve optimizing the website for faster loading times, simplifying the navigation, or improving the design. Provide excellent customer support for customers who log in using a computer. Consider using **mobile-specific engagement tactics**, such as push notifications or in-app messaging, to keep mobile users engaged and connected to your brand
- The company can also **focus on cross-selling and up-selling to existing customers, especially those in the regular plus segment**, to increase their loyalty and lifetime value. This can involve offering relevant services or products that complement their existing subscriptions
- The company may also want to **investigate why customers who give a rating of 1 to Customer Care Agent Service are less likely to churn** and try to replicate that positive experience for other customers.

- **Target the single and married customers** with specific offers or promotions to increase their loyalty towards the company. **Offer family plans for married customers** to increase satisfaction and loyalty. Explore the possibility of offering specialized products or services for divorced customers
- Conduct further analysis to **identify common reasons for customer complaints** and take **proactive measures to address those issues** before they lead to customer dissatisfaction
- **Continuously monitor customer feedback** and implement improvements to enhance customer experience
- **Encourage customers from tier 2 cities** to use their services by offering them incentives
- **Special attention should be given to new customers during the first two months of their account setup** to prevent them from leaving and to establish a long-term relationship

Based on spending and loyalty, here is one potential way to segment customers into four categories:

High spenders, high loyalty: These are customers who spend a lot of money with the company and have been loyal customers for a long time. They may be eligible for exclusive rewards and benefits.

High spenders, low loyalty: These are customers who spend a lot of money with the company but have not been loyal customers for a long time. They may need additional incentives to continue shopping with the company.

Low spenders, high loyalty: These are customers who do not spend a lot of money with the company but have been loyal customers for a long time. They may appreciate rewards and incentives to encourage them to spend more.

Low spenders, low loyalty: These are customers who do not spend a lot of money with the company and have not been loyal customers for a long time. They may need more attention and engagement to keep them interested in the company.

Appendix

Annexure – A

Hyperparameter tuning for XG Boost Model

Code:

```

param_grid={'learning_rate':[0.1,0.05,0.2],
            'n_estimators':[1000,2000],
            'max_depth':[5,7,9]}

xgb = XGBClassifier(random_state=1)

grid_search_xgb = GridSearchCV(estimator = xgb, param_grid = param_grid, cv = 3,n_jobs=-1,scoring='f1')

grid_search_xgb.fit(X_train, Y_train)

print(grid_search_xgb.best_params_,'\n')
print(grid_search_xgb.best_estimator_)

best_model_xgb = grid_search_xgb.best_estimator_
best_model_xgb

#Using above defined function to get accuracy, recall and precision on train and test set
best_model_xgb_score=get_metrics_score(best_model_xgb)

important_features = pd.DataFrame({'Features': X_train.columns,
                                  'Importance': best_model_xgb.feature_importances_})

important_features = important_features.sort_values('Importance', ascending = False)

plt.figure(figsize=(12,8))
sns.barplot(x = 'Importance', y = 'Features', data = important_features)

plt.title('Feature Importance of Tuned XG Boost Classifier', fontsize = 15)
plt.xlabel('Importance', fontsize = 15)
plt.ylabel('Features', fontsize = 15)

plt.show()

```


The resulting model is built by using grid search and the best params are used

```

GridSearchCV
GridSearchCV(cv=3,
              estimator=XGBClassifier(base_score=None, booster=None,
                                      callbacks=None, colsample_bylevel=None,
                                      colsample_bynode=None, colsample_bytree=None,
                                      early_stopping_rounds=None,
                                      enable_categorical=False, eval_metric=None,
                                      gamma=None, gpu_id=None, grow_policy=None,
                                      importance_type=None,
                                      interaction_constraints=None,
                                      ...))
  estimator: XGBClassifier
XGBClassifier(base_score=None, booster=None, callbacks=None,
              colsample_bylevel=None, colsample_bynode=None,
              colsample_bytree=None, early_stopping_rounds=None,
              enable_categorical=False, eval_metric=None, gamma=None,
              gpu_id=None, grow_policy=None, importance_type=None,
              interaction_constraints=None, learning_rate=None, max_bin=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, n_estimators=100, n_jobs=None,
              num_parallel_tree=None, predictor=None, random_state=1,
              ...))
  XGBClassifier
  colsample_bylevel=None, colsample_bynode=None,
  colsample_bytree=None, early_stopping_rounds=None,
  enable_categorical=False, eval_metric=None, gamma=None,
  gpu_id=None, grow_policy=None, importance_type=None,
  interaction_constraints=None, learning_rate=None, max_bin=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, n_estimators=100, n_jobs=None,
  num_parallel_tree=None, predictor=None, random_state=1,
  reg_alpha=None, reg_lambda=None, ...)

```

Fig.38 Tuned XG Boost Model

```

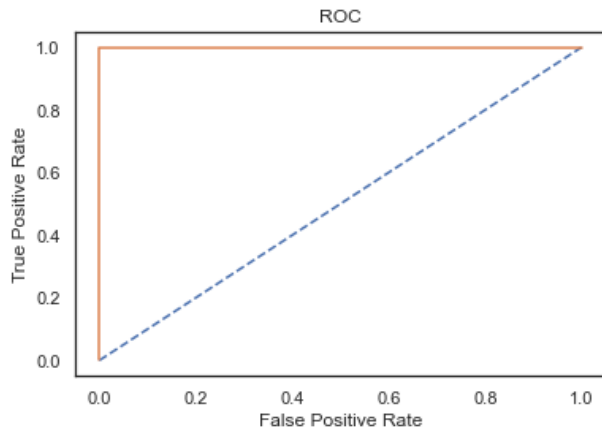
XGBClassifier
XGBClassifier(base_score=0.5, booster='gbtree', callbacks=None,
              colsample_bylevel=1, colsample_bynode=1, colsample_bytree=1,
              early_stopping_rounds=None, enable_categorical=False,
              eval_metric=None, gamma=0, gpu_id=-1, grow_policy='depthwise',
              importance_type=None, interaction_constraints='',
              learning_rate=0.1, max_bin=256, max_cat_to_onehot=4,
              max_delta_step=0, max_depth=9, max_leaves=0, min_child_weight=1,
              missing=nan, monotone_constraints=(), n_estimators=1000,
              n_jobs=0, num_parallel_tree=1, predictor='auto', random_state=1,
              reg_alpha=0, reg_lambda=1, ...)

```

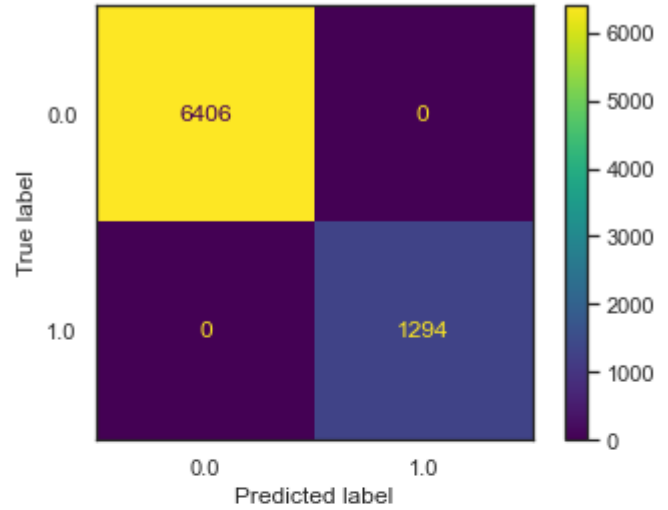
Fig.39 Best parameters of Tuned XG Boost Model

AUC & ROC Graph of Training Data

Area under the curve: 1.000

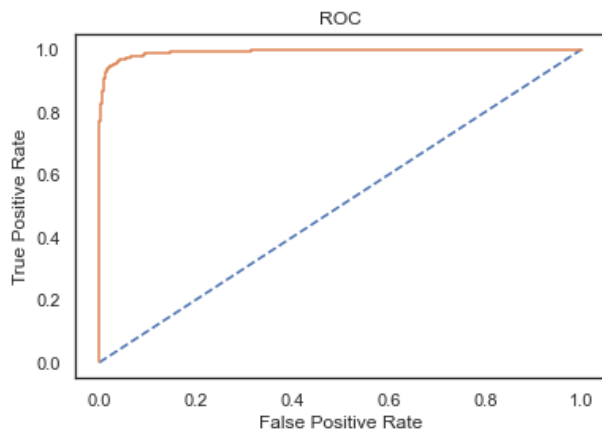


Confusion Matrix for Training Data



AUC & ROC Graph of Testing Data

Area under the curve: 0.993



Confusion Matrix for Testing Data

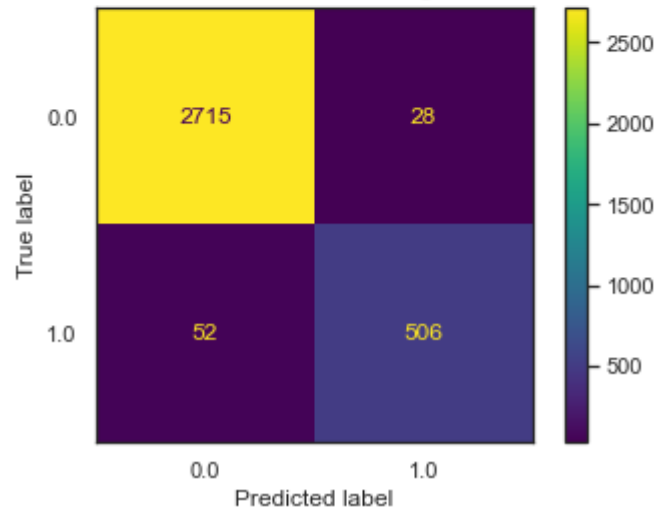


Fig.40 Tuned XG Boost – AUC & ROC curve

Fig.41 Tuned XG Boost – Confusion Matrix

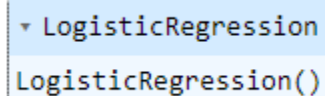
Annexure – B

Model building and tuning for all 8 models

Model 1 - Logistic Regression with Smote

When the data is linearly separable, the logistic regression model performs well. It is assumed that the goal and predictor variables are linear. Because it is a parametric model, it can be faster than KNN. It also includes coefficients that aid in model interpretation.

The logistic regression model is build using the SMOTE data using default parameters



```
LogisticRegression
LogisticRegression()
```

Fig.42 Logistic Regression Model with Smote

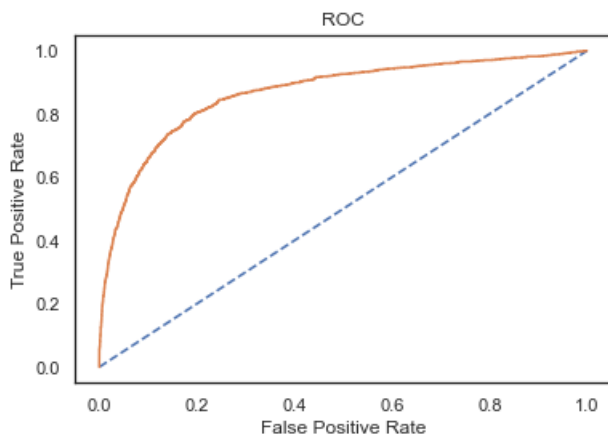
Inference:

- We are taking into account the positive class (class 1) of the predicted variable, which represents customers who are predicted to churn, in the current business situation because we are interested in identifying and targeting such customers with retention efforts.
- The model shows moderate performance in predicting customer churn. Both the training and test set accuracy, recall, and F1-score are above 0.5, indicating that the model is able to identify some of the customers who are likely to churn.
- However, the precision of the model is relatively low for both the training and test sets. This means that the model tends to predict a higher number of false positives (customers who are predicted to churn but do not actually churn). This can be costly for the DTH company, as it may result in unnecessary retention efforts for customers who would not have churned otherwise.
- The difference between the training and test set metrics is relatively small, which indicates that the model is not overfitting or underfitting.
- The F1-score of the test set is slightly lower than that of the training set, which suggests that the model may not generalize very well to new data.

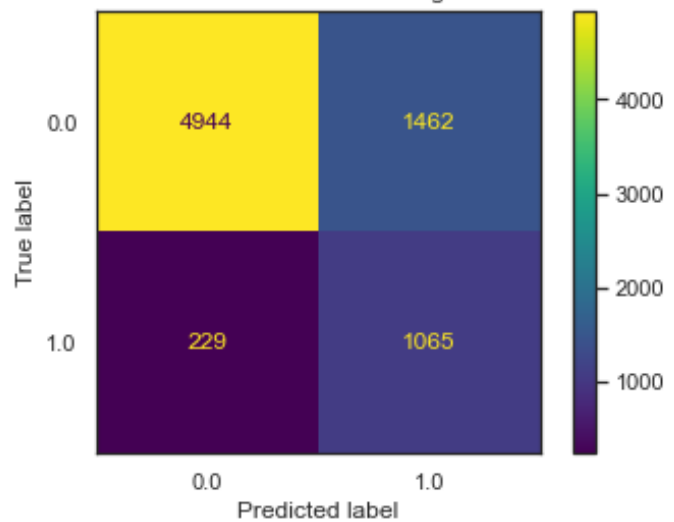
Overall, the model has room for improvement, especially in terms of improving precision, to reduce false positives and make the model more cost-effective for the DTH company. The model is neither overfitting nor underfitting, which is a good sign, but it may not be generalizing well to new data.

AUC & ROC Graph of Training Data

Area under the curve: 0.867

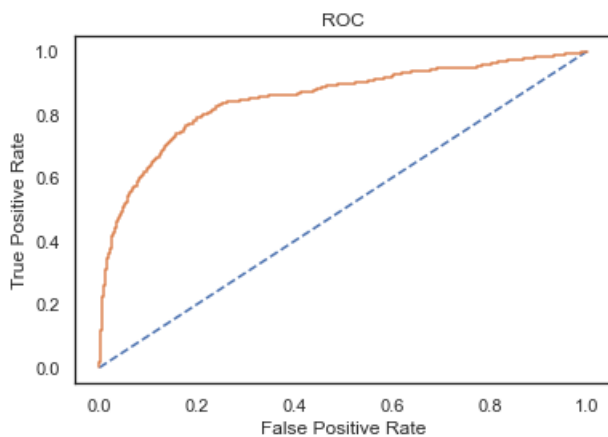


Confusion Matrix for Training Data



AUC & ROC Graph of Testing Data

Area under the curve: 0.850



Confusion Matrix for Testing Data

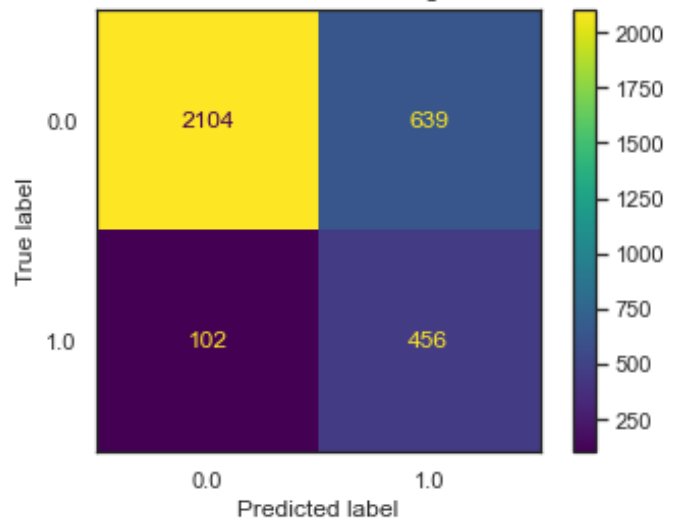


Fig.43 LR Smote – AUC & ROC curve

Fig.44 LR Smote – Confusion Matrix

Classification Report of Training Data					
	precision	recall	f1-score	support	
0.0	0.96	0.77	0.85	6406	
1.0	0.42	0.82	0.56	1294	
accuracy			0.78	7700	
macro avg	0.69	0.80	0.71	7700	
weighted avg	0.87	0.78	0.80	7700	

Classification Report of Testing Data					
	precision	recall	f1-score	support	
0.0	0.95	0.77	0.85	2743	
1.0	0.42	0.82	0.55	558	
accuracy			0.78	3301	
macro avg	0.69	0.79	0.70	3301	
weighted avg	0.86	0.78	0.80	3301	

Fig.45 LR Smote – Classification Report

Logistic Regression: Metrics Summary

```

Accuracy on training set : 0.78
Accuracy on test set : 0.776
Recall on training set : 0.823
Recall on test set : 0.817
Precision on training set : 0.421
Precision on test set : 0.416
F1 on training set : 0.557
F1 on test set : 0.552

```

Fig.46 LR Smote – Metrics Summary

Model 2 - Logistic Regression without Smote

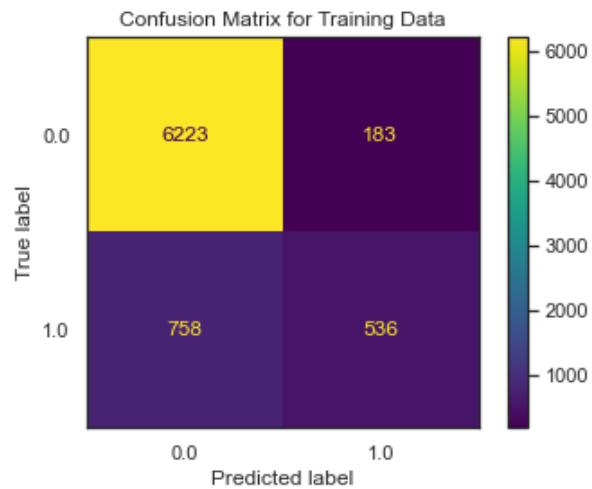
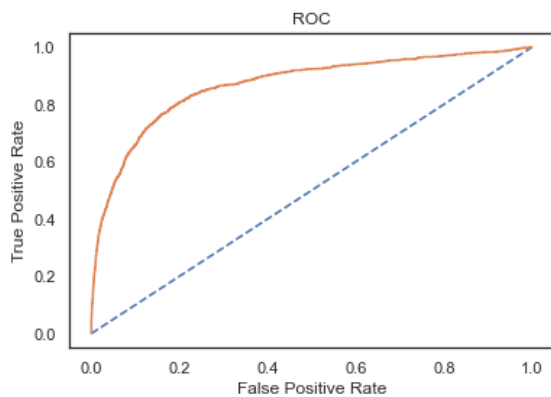
The logistic regression model is build using the original data using default parameters

```
LogisticRegression
LogisticRegression()
```

Fig.47 Logistic Regression Model

AUC & ROC Graph of Training Data

Area under the curve: 0.867



AUC & ROC Graph of Testing Data

Area under the curve: 0.851

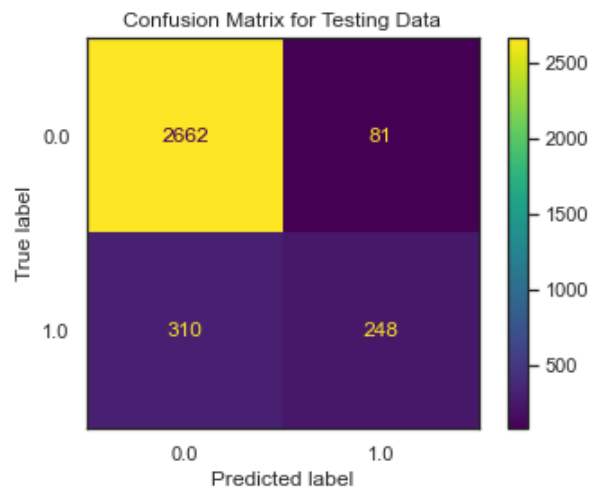
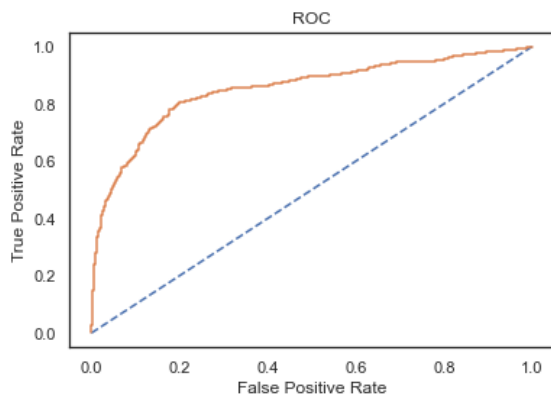


Fig.48 LR – AUC & ROC curve

Fig.49 LR – Confusion Matrix

Classification Report of Training Data					
	precision	recall	f1-score	support	
0.0	0.89	0.97	0.93	6406	
1.0	0.75	0.41	0.53	1294	
accuracy			0.88	7700	
macro avg	0.82	0.69	0.73	7700	
weighted avg	0.87	0.88	0.86	7700	

Classification Report of Testing Data					
	precision	recall	f1-score	support	
0.0	0.90	0.97	0.93	2743	
1.0	0.75	0.44	0.56	558	
accuracy			0.88	3301	
macro avg	0.82	0.71	0.75	3301	
weighted avg	0.87	0.88	0.87	3301	

Fig.50 LR – Classification Report

Logistic Regression: Metrics Summary

Accuracy on training set : 0.878
 Accuracy on test set : 0.882
 Recall on training set : 0.414
 Recall on test set : 0.444
 Precision on training set : 0.745
 Precision on test set : 0.754
 F1 on training set : 0.533
 F1 on test set : 0.559

Fig.52 LR – Metrics Summary

Inference:

- The model shows moderate to good performance in predicting customer churn. Both the training and test set accuracy are above 0.8, indicating that the model is able to identify a good number of the customers who are likely to churn.
- The precision of the model is relatively high for both the training and test sets, which is a good sign. This means that the model tends to predict fewer false positives (customers who are predicted to churn but do not actually churn).
- The recall of the model is relatively low for both the training and test sets. This means that the model tends to miss a good number of the customers who are likely to churn.
- The difference between the training and test set metrics is relatively small, which indicates that the model is not overfitting or underfitting.
- The F1-score of the test set is slightly higher than that of the training set, which is a good sign. This suggests that the model is generalizing well to new data.

Overall, the model has good performance in terms of accuracy and precision, but it needs improvement in terms of recall to capture more of the customers who are likely to churn. The model is neither overfitting nor underfitting, which is a good sign.

Model 3 – Tuned Logistic Regression Model

The resulting model is built by using grid search and the best params are used

```

GridSearchCV
GridSearchCV(cv=3, estimator=LogisticRegression(max_iter=100000, n_jobs=2),
             n_jobs=-1,
             param_grid={'penalty': ['l1', 'l2', 'none'],
                          'solver': ['lbfgs', 'liblinear'],
                          'tol': [0.0001, 0.001, 0.01]},
             scoring='f1')
  estimator: LogisticRegression
  LogisticRegression(max_iter=100000, n_jobs=2)
    LogisticRegression
    LogisticRegression(max_iter=100000, n_jobs=2)

```

Fig.53 Tuned Logistic Regression Model

```

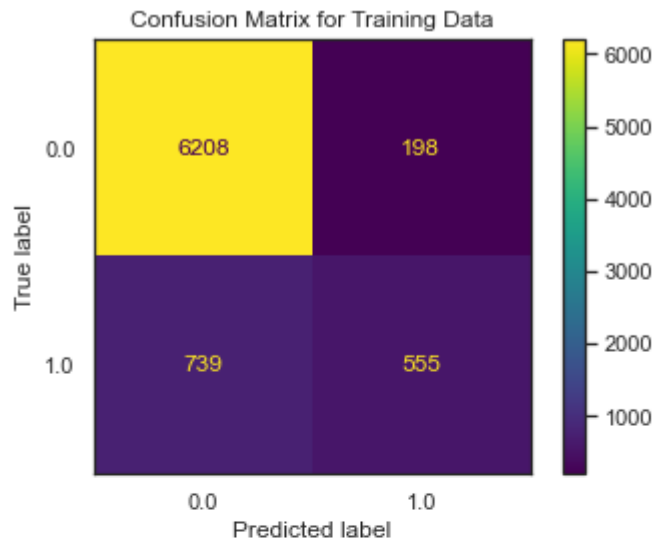
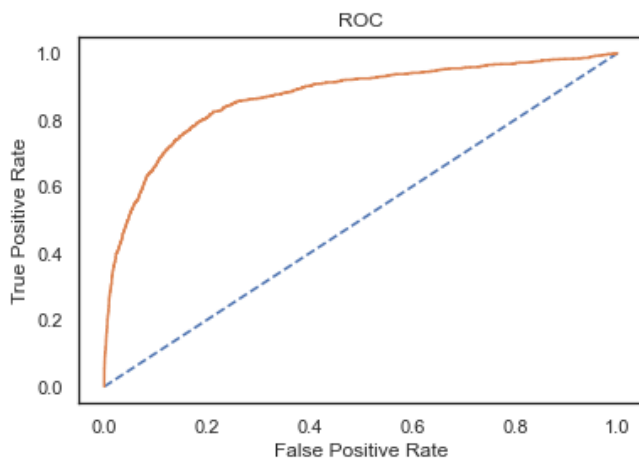
LogisticRegression
LogisticRegression(max_iter=100000, n_jobs=2, penalty='none')

```

Fig.54 Best parameters of Tuned Logistic Regression Model

AUC & ROC Graph of Training Data

Area under the curve: 0.868



AUC & ROC Graph of Testing Data

Area under the curve: 0.852

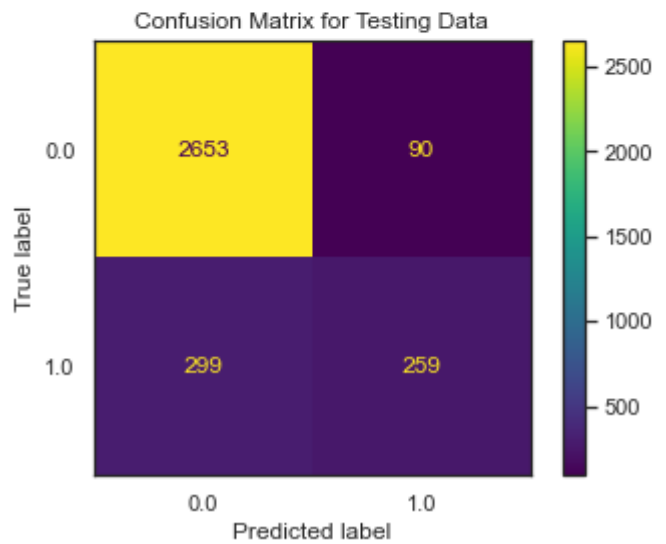
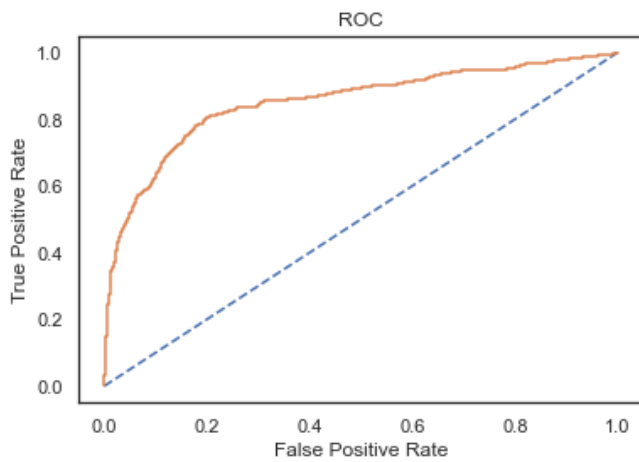


Fig.55 Tuned LR – AUC & ROC curve

Fig.56 Tuned LR – Confusion Matrix

Classification Report of Training Data					
	precision	recall	f1-score	support	
0.0	0.89	0.97	0.93	6406	
1.0	0.74	0.43	0.54	1294	
accuracy			0.88	7700	
macro avg	0.82	0.70	0.74	7700	
weighted avg	0.87	0.88	0.86	7700	

Classification Report of Testing Data					
	precision	recall	f1-score	support	
0.0	0.90	0.97	0.93	2743	
1.0	0.74	0.46	0.57	558	
accuracy			0.88	3301	
macro avg	0.82	0.72	0.75	3301	
weighted avg	0.87	0.88	0.87	3301	

Fig.57 Tuned LR – Classification Report

Tuned Logistic Regression: Metrics Summary

Accuracy on training set : 0.878
 Accuracy on test set : 0.882
 Recall on training set : 0.429
 Recall on test set : 0.464
 Precision on training set : 0.737
 Precision on test set : 0.742
 F1 on training set : 0.542
 F1 on test set : 0.571

Fig.58 Tuned LR – Metrics Summary

Inference:

- The model has good performance in terms of accuracy on both the training and test sets, indicating that it is able to identify a good number of the customers who are likely to churn.
- The precision of the model is relatively high for both the training and test sets, which is a good sign. This means that the model tends to predict fewer false positives (customers who are predicted to churn but do not actually churn).
- The recall of the model has improved slightly compared to the previous model for both the training and test sets. This means that the model is able to identify more of the customers who are likely to churn.
- The difference between the training and test set metrics is relatively small, which indicates that the model is not overfitting or underfitting.
- The F1-score of the test set has improved significantly compared to the previous model. This suggests that the model is generalizing well to new data and is performing better overall.

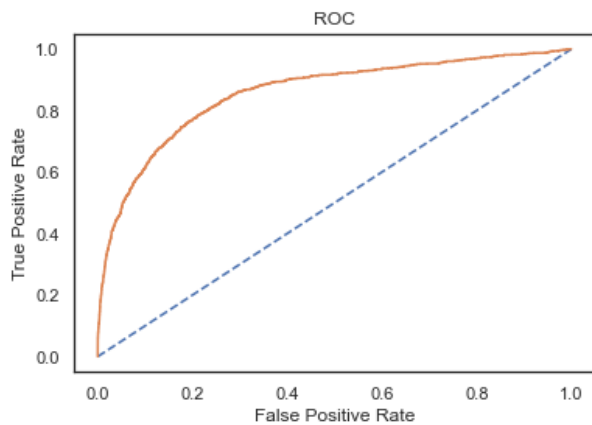
Overall, the model has good performance in terms of accuracy, precision, and recall. The F1-score has improved significantly, indicating that the model is better at balancing precision and recall. The model is not overfitting or underfitting, which is a good sign.

Model 4 – Linear Discriminant Analysis Model

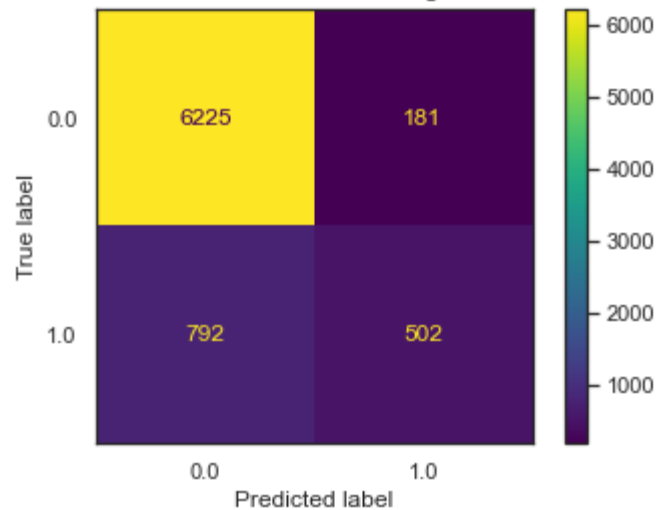
Another machine learning classifier is Linear Discriminant Analysis (LDA). It works well when the data has linearly separable classes. It is based on the assumption that the underlying data has a gaussian distribution, however it can perform well even if these assumptions are broken

AUC & ROC Graph of Training Data

Area under the curve: 0.856

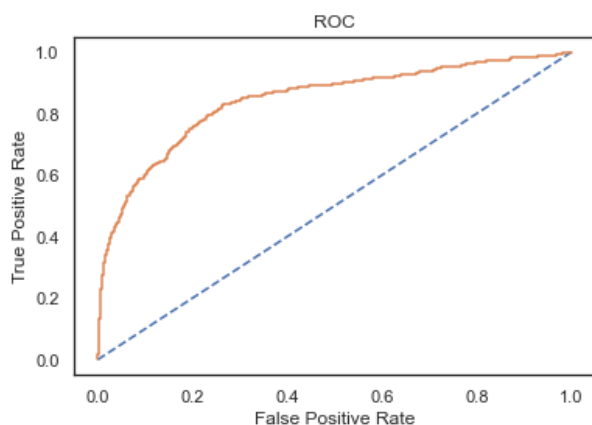


Confusion Matrix for Training Data



AUC & ROC Graph of Testing Data

Area under the curve: 0.843



Confusion Matrix for Testing Data

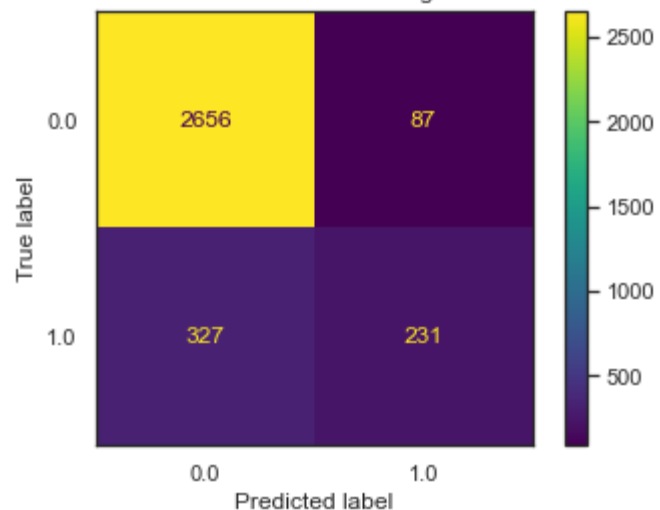


Fig.59 LDA – AUC & ROC curve

Fig.60 LDA – Confusion Matrix

Classification Report of Training Data				
	precision	recall	f1-score	support
0.0	0.89	0.97	0.93	6406
1.0	0.73	0.39	0.51	1294
accuracy			0.87	7700
macro avg	0.81	0.68	0.72	7700
weighted avg	0.86	0.87	0.86	7700

Classification Report of Testing Data				
	precision	recall	f1-score	support
0.0	0.89	0.97	0.93	2743
1.0	0.73	0.41	0.53	558
accuracy			0.87	3301
macro avg	0.81	0.69	0.73	3301
weighted avg	0.86	0.87	0.86	3301

Fig.61 LDA – Classification Report

LDA: Metrics Summary

Accuracy on training set : 0.874
 Accuracy on test set : 0.875
 Recall on training set : 0.388
 Recall on test set : 0.414
 Precision on training set : 0.735
 Precision on test set : 0.726
 F1 on training set : 0.508
 F1 on test set : 0.527

Fig.62 LDA – Metrics Summary

Inference:

- The model is not overfitting or underfitting since the training and test set accuracies are similar and not significantly different.
- The recall on both training and test sets is low, indicating that the model may have difficulty identifying true positives and has a relatively high false-negative rate.
- The precision on both training and test sets is moderate, indicating that the model does not have a high false-positive rate but also has room for improvement.
- The F1 score on both training and test sets is low, indicating that there is a trade-off between precision and recall, and the model can benefit from further optimization.
- Overall, while the model performs adequately, there is room for improvement in terms of identifying more true positives and achieving a better balance between precision and recall.

Model 5 – Ada Boost Classifier

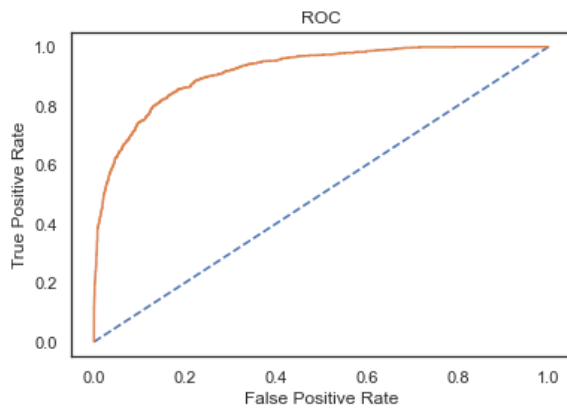
The resulting model is built by default parameters

```
AdaBoostClassifier
AdaBoostClassifier(random_state=1)
```

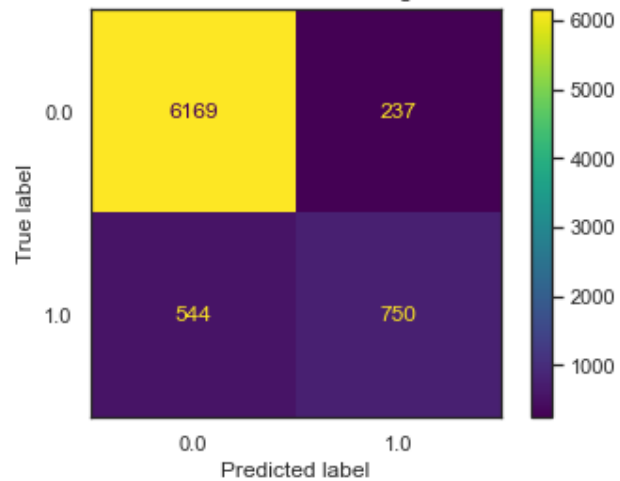
Fig.63 Ada Boost Model

AUC & ROC Graph of Training Data

Area under the curve: 0.917

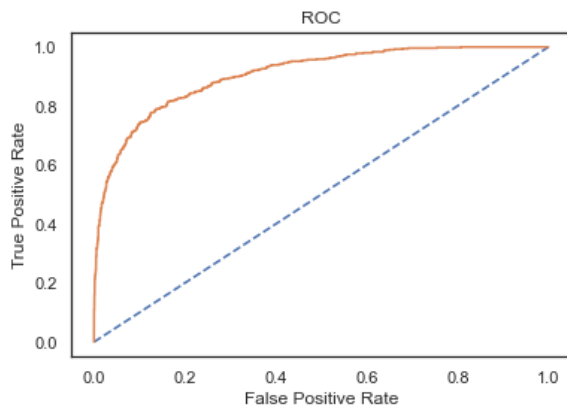


Confusion Matrix for Training Data



AUC & ROC Graph of Testing Data

Area under the curve: 0.906



Confusion Matrix for Testing Data

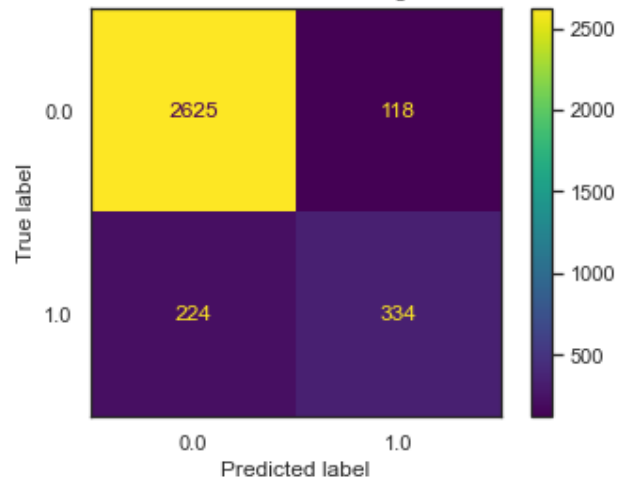


Fig.64 Ada Boost – AUC & ROC curve

Fig.65 Ada Boost – Confusion Matrix

Classification Report of Training Data					
	precision	recall	f1-score	support	
0.0	0.92	0.96	0.94	6406	
1.0	0.76	0.58	0.66	1294	
accuracy			0.90	7700	
macro avg	0.84	0.77	0.80	7700	
weighted avg	0.89	0.90	0.89	7700	

Classification Report of Testing Data					
	precision	recall	f1-score	support	
0.0	0.92	0.96	0.94	2743	
1.0	0.74	0.60	0.66	558	
accuracy			0.90	3301	
macro avg	0.83	0.78	0.80	3301	
weighted avg	0.89	0.90	0.89	3301	

Fig.66 Ada Boost – Classification Report

Ada Boost: Metrics Summary

Accuracy on training set : 0.899
 Accuracy on test set : 0.896
 Recall on training set : 0.58
 Recall on test set : 0.599
 Precision on training set : 0.76
 Precision on test set : 0.739
 F1 on training set : 0.658
 F1 on test set : 0.661

Fig.69 Ada Boost – Metrics Summary

Inference:

- The accuracy on both the training and test sets are relatively high, indicating that the model is able to classify the target variable with good accuracy.
- The recall scores are also reasonably high, indicating that the model is able to identify a good proportion of the positive class (i.e., the class we are interested in predicting).
- The precision score on the test set is slightly lower than that of the training set, which indicates that the model may be identifying some false positives in the test set. However, both precision scores are still relatively high, which is a good sign.
- The F1 score on the test set is similar to that of the training set, indicating that the model is performing consistently on both sets of data.
- Overall, the evaluation metrics suggest that the model is performing well and is not overfitting or underfitting the data, as the performance on the test set is similar to that of the training set.

Model 6 – Gradient Boost Classifier

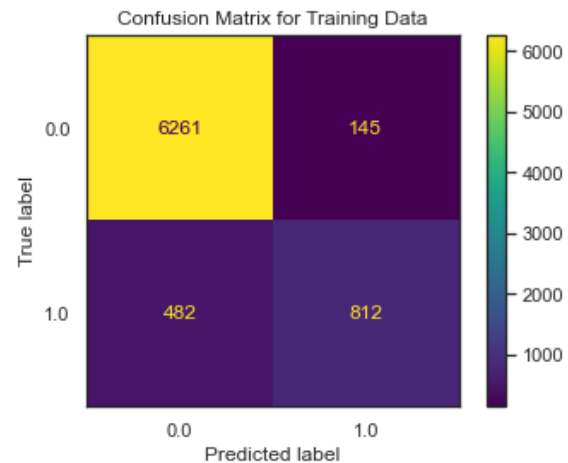
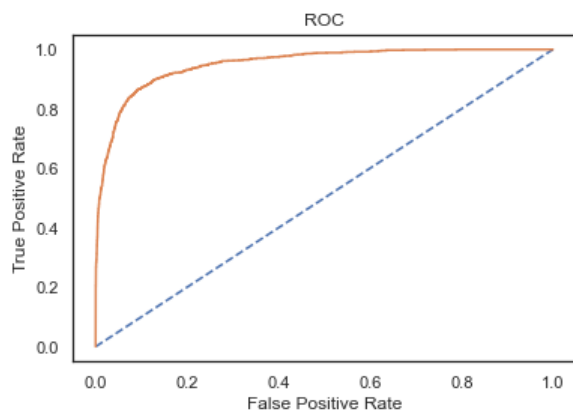
The resulting model is built by default parameters

```
GradientBoostingClassifier
GradientBoostingClassifier(random_state=1)
```

Fig.70 Gradient Boost Model

AUC & ROC Graph of Training Data

Area under the curve: 0.952



AUC & ROC Graph of Testing Data

Area under the curve: 0.934

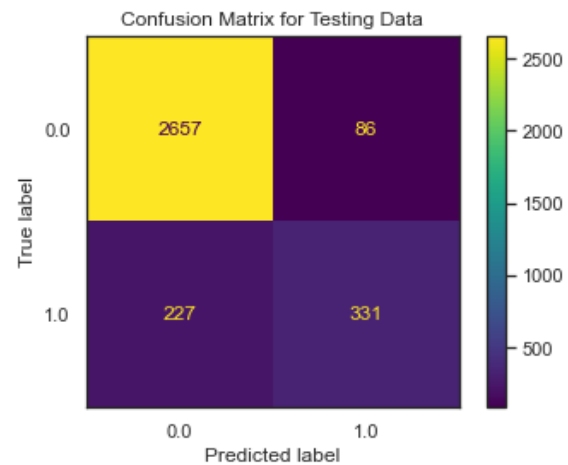
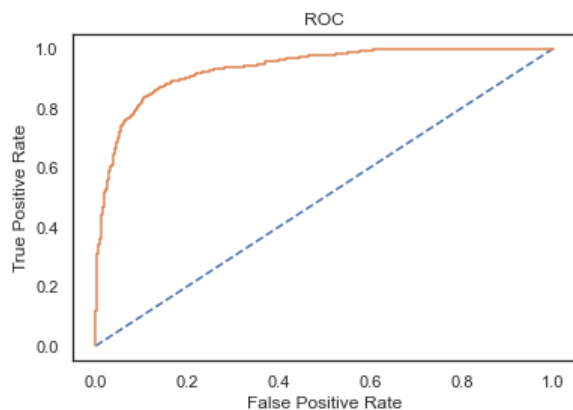


Fig.71 Gradient Boost – AUC & ROC curve

Fig.72 Gradient Boost – Confusion Matrix

Classification Report of Training Data					
	precision	recall	f1-score	support	
0.0	0.93	0.98	0.95	6406	
1.0	0.85	0.63	0.72	1294	
accuracy			0.92	7700	
macro avg	0.89	0.80	0.84	7700	
weighted avg	0.92	0.92	0.91	7700	

Classification Report of Testing Data					
	precision	recall	f1-score	support	
0.0	0.92	0.97	0.94	2743	
1.0	0.79	0.59	0.68	558	
accuracy			0.91	3301	
macro avg	0.86	0.78	0.81	3301	
weighted avg	0.90	0.91	0.90	3301	

Fig.73 Gradient Boost – Classification Report

Gradient Boost: Metrics Summary

```

Accuracy on training set : 0.919
Accuracy on test set : 0.905
Recall on training set : 0.628
Recall on test set : 0.593
Precision on training set : 0.848
Precision on test set : 0.794
F1 on training set : 0.721
F1 on test set : 0.679

```

Fig.74 Gradient Boost – Metrics Summary

Inference:

- The tuned Gradient Boost Model has high accuracy scores on both the training and test sets, with an accuracy of 0.919 on the training set and 0.905 on the test set. The precision and recall scores are also relatively high, with a precision score of 0.848 and 0.794 on the training and test sets respectively, and a recall score of 0.628 on the training set and 0.593 on the test set.
- However, there is a slight drop in performance from the training set to the test set, which may indicate some overfitting. The F1 scores are also relatively high, with a score of 0.721 on the training set and 0.679 on the test set, indicating good balance between precision and recall. Overall, the model appears to be performing well and may be a good candidate for deployment.

Model 7 – XG Boost Classifier

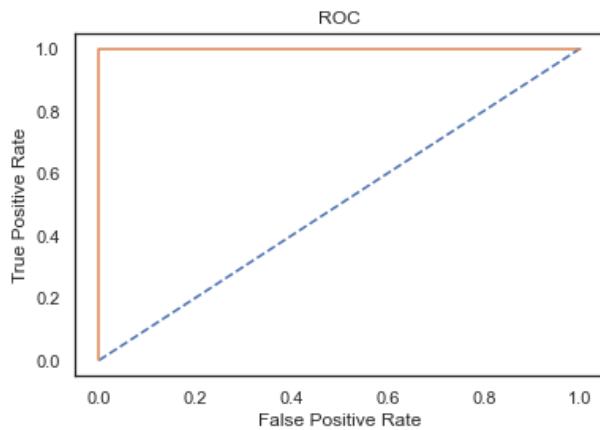
The resulting model is built by default parameters

```
XGBClassifier
XGBClassifier(base_score=0.5, booster='gbtree', callbacks=None,
               colsample_bylevel=1, colsample_bynode=1, colsample_bytree=1,
               early_stopping_rounds=None, enable_categorical=False,
               eval_metric=None, gamma=0, gpu_id=-1, grow_policy='depthwise',
               importance_type=None, interaction_constraints='',
               learning_rate=0.300000012, max_bin=256, max_cat_to_onehot=4,
               max_delta_step=0, max_depth=6, max_leaves=0, min_child_weight=1,
               missing=nan, monotone_constraints='()', n_estimators=100,
               n_jobs=0, num_parallel_tree=1, predictor='auto', random_state=1,
               reg_alpha=0, reg_lambda=1, ...)
```

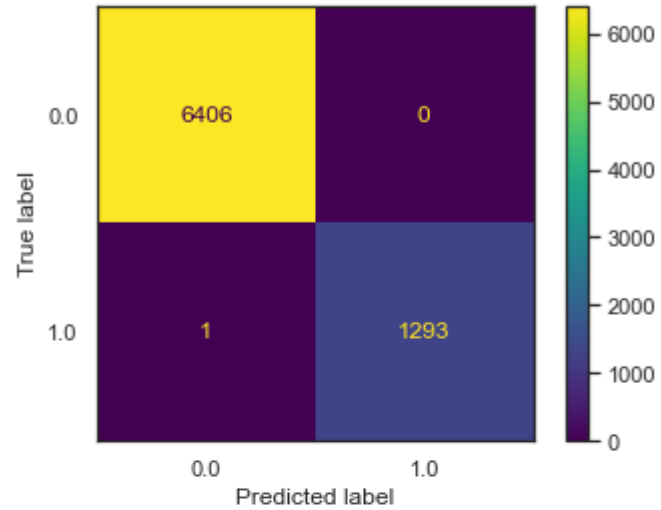
Fig.75 XG Boost Model

AUC & ROC Graph of Training Data

Area under the curve: 1.000

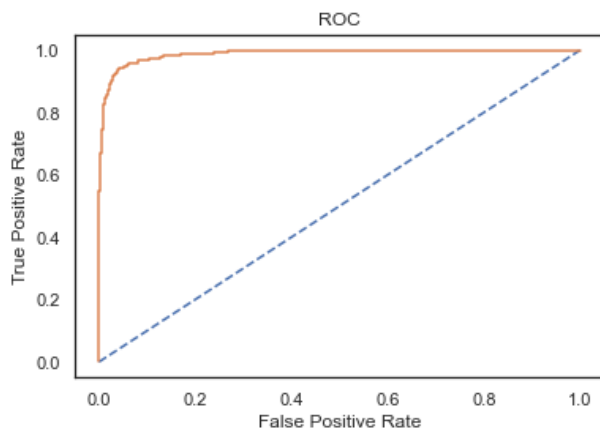


Confusion Matrix for Training Data



AUC & ROC Graph of Testing Data

Area under the curve: 0.988



Confusion Matrix for Testing Data

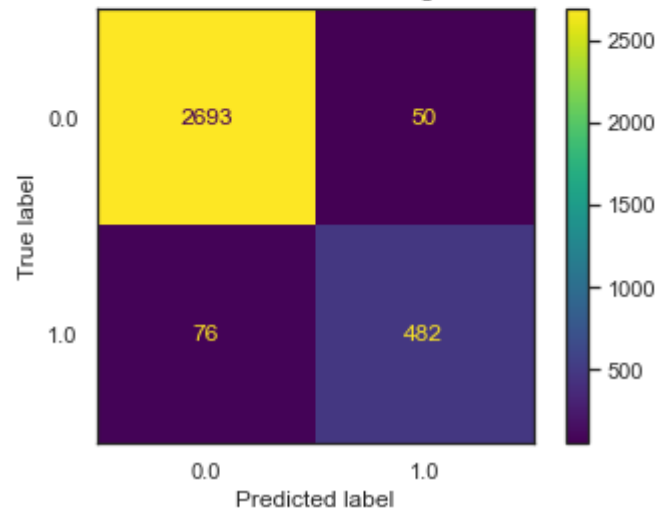


Fig.76 XG Boost – AUC & ROC curve

Fig.77 XG Boost – Confusion Matrix

Classification Report of Training Data					
	precision	recall	f1-score	support	
0.0	1.00	1.00	1.00	6406	
1.0	1.00	1.00	1.00	1294	
accuracy			1.00	7700	
macro avg	1.00	1.00	1.00	7700	
weighted avg	1.00	1.00	1.00	7700	

Classification Report of Testing Data					
	precision	recall	f1-score	support	
0.0	0.97	0.98	0.98	2743	
1.0	0.91	0.86	0.88	558	
accuracy			0.96	3301	
macro avg	0.94	0.92	0.93	3301	
weighted avg	0.96	0.96	0.96	3301	

Fig.78 XG Boost – Classification Report

XG Boost: Metrics Summary

Accuracy on training set : 1.0
 Accuracy on test set : 0.962
 Recall on training set : 0.999
 Recall on test set : 0.864
 Precision on training set : 1.0
 Precision on test set : 0.906
 F1 on training set : 1.0
 F1 on test set : 0.884

Fig.79 XG Boost – Metrics Summary

Inference:

- The XG Boost model has an accuracy of 1.0 on the training set and an accuracy of 0.962 on the test set, which suggests that the model may be overfitting to the training data.
- The recall, precision, and F1 score are relatively high for both the training and test sets, which suggests that the model is performing well in terms of predicting the positive class. However, the recall score on the test set is lower than on the training set, indicating that the model may not be generalizing well to new data. The precision and F1 scores are also slightly lower on the test set, which further supports the possibility of overfitting.
- Overall, it appears that the XG Boost model may be overfitting to the training data, and additional measures may be needed to improve its generalization performance on new data.

Model 9 – Tuned Ada Boost Model

The resulting model is built by using grid search and the best params are used

```
GridSearchCV
GridSearchCV(cv=3, estimator=AdaBoostClassifier(random_state=1), n_jobs=-1,
  param_grid={'base_estimator': [DecisionTreeClassifier(max_depth=10),
                                DecisionTreeClassifier(max_depth=20),
                                DecisionTreeClassifier(max_depth=30)],
              'learning_rate': array([0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1. , 1.1, 1.2, 1.3,
                                     1.4, 1.5, 1.6, 1.7, 1.8, 1.9]),
              'n_estimators': array([ 10,  20,  30,  40,  50,  60,  70,  80,  90, 100])},
  scoring='f1')
```

```
  estimator: AdaBoostClassifier
AdaBoostClassifier(random_state=1)
  AdaBoostClassifier
AdaBoostClassifier(random_state=1)
```

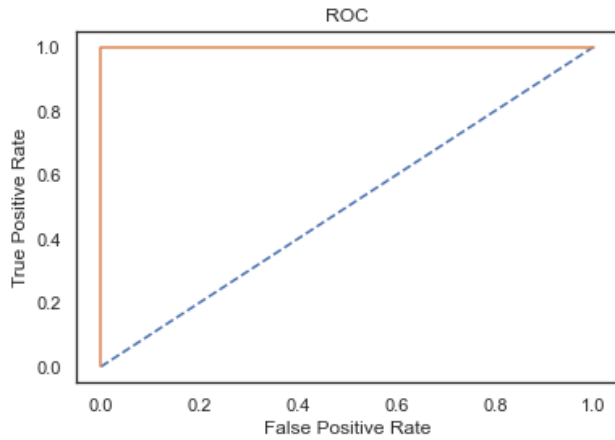
Fig.80 Tuned Ada Boost Model

```
AdaBoostClassifier
AdaBoostClassifier(base_estimator=DecisionTreeClassifier(max_depth=10),
  learning_rate=1.2000000000000002, n_estimators=100,
  random_state=1)
  base_estimator: DecisionTreeClassifier
DecisionTreeClassifier(max_depth=10)
  DecisionTreeClassifier
DecisionTreeClassifier(max_depth=10)
```

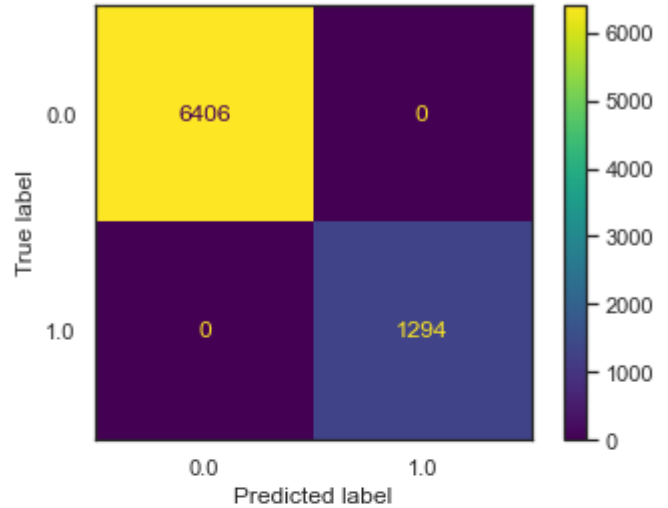
Fig.81 Best parameters of Tuned Ada Boost Model

AUC & ROC Graph of Training Data

Area under the curve: 1.000

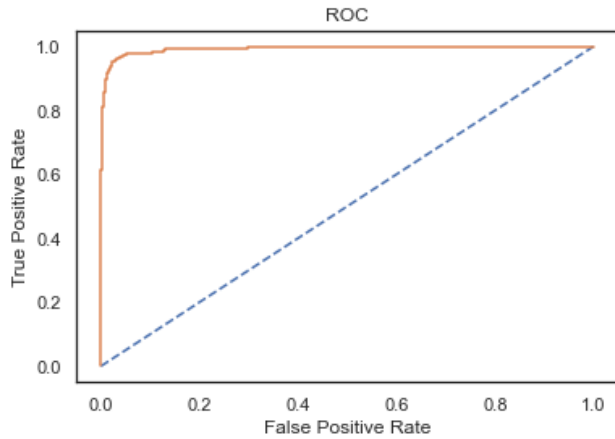


Confusion Matrix for Training Data



AUC & ROC Graph of Testing Data

Area under the curve: 0.993



Confusion Matrix for Testing Data

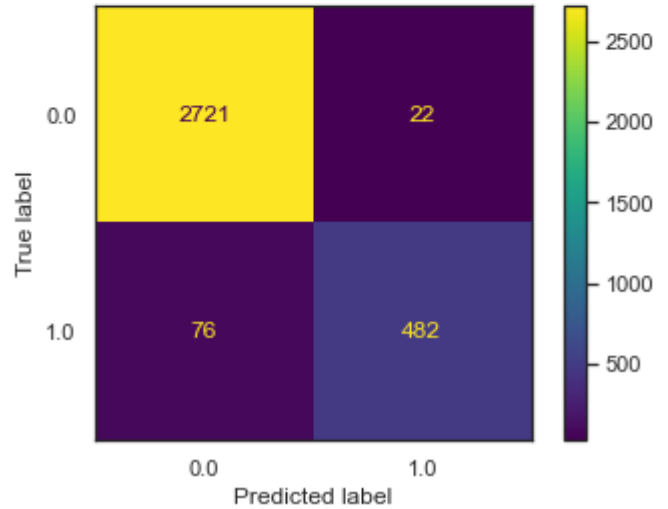


Fig.82 Tuned Ada Boost – AUC & ROC curve

Fig.83 Tuned Ada Boost – Confusion Matrix

Classification Report of Training Data					
	precision	recall	f1-score	support	
0.0	1.00	1.00	1.00	6406	
1.0	1.00	1.00	1.00	1294	
accuracy			1.00	7700	
macro avg	1.00	1.00	1.00	7700	
weighted avg	1.00	1.00	1.00	7700	

Classification Report of Testing Data					
	precision	recall	f1-score	support	
0.0	0.97	0.99	0.98	2743	
1.0	0.96	0.86	0.91	558	
accuracy			0.97	3301	
macro avg	0.96	0.93	0.95	3301	
weighted avg	0.97	0.97	0.97	3301	

Fig.84 Tuned Ada Boost – Classification Report

Tuned Ada Boost: Metrics Summary

Accuracy on training set : 1.0
 Accuracy on test set : 0.97
 Recall on training set : 1.0
 Recall on test set : 0.864
 Precision on training set : 1.0
 Precision on test set : 0.956
 F1 on training set : 1.0
 F1 on test set : 0.908

Fig.85 Tuned Ada Boost – Metrics Summary

Inference:

- Based on the provided evaluation metrics, it seems that the model is performing well on both the training and test sets. The accuracy and F1 score are high, indicating that the model is correctly classifying most of the instances. The precision score is also high, which means that when the model predicts a positive outcome, it is usually correct. However, the recall score on the test set is relatively low, which suggests that the model is not able to correctly identify all positive instances.
- Regarding overfitting or underfitting, the fact that the accuracy and F1 score are high on both the training and test sets suggests that the model is not significantly overfitting or underfitting. However, it's worth noting that the recall score on the test set is lower than on the training set, which could indicate some degree of overfitting.

Model 10 – Tuned Gradient Boost Model

The resulting model is built by using grid search and the best params are used

```

GridSearchCV
GridSearchCV(cv=3, estimator=GradientBoostingClassifier(random_state=1),
             n_jobs=-1,
             param_grid={'learning_rate': [0.1, 0.05, 0.01],
                          'max_depth': [4, 5, 7], 'max_features': [4, 5, 6],
                          'min_samples_leaf': [12, 15, 17],
                          'min_samples_split': [30, 50, 70],
                          'n_estimators': [50, 100, 150]},
             scoring='f1')
  estimator: GradientBoostingClassifier
    GradientBoostingClassifier(random_state=1)
      GradientBoostingClassifier
        GradientBoostingClassifier(random_state=1)

```

Fig.86 Tuned Gradient Boost Model

```

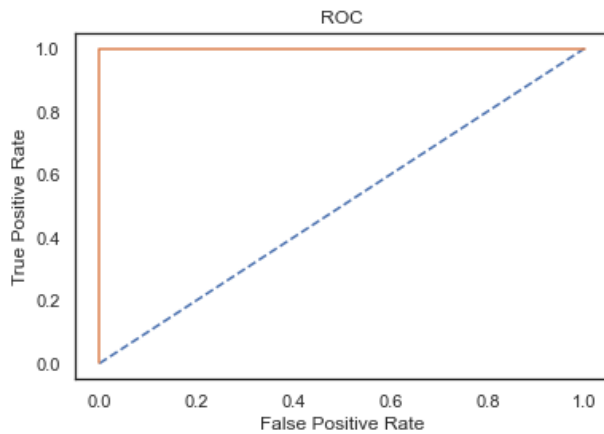
GradientBoostingClassifier
GradientBoostingClassifier(max_depth=7, max_features=4, min_samples_leaf=12,
                           min_samples_split=30, n_estimators=150,
                           random_state=1)

```

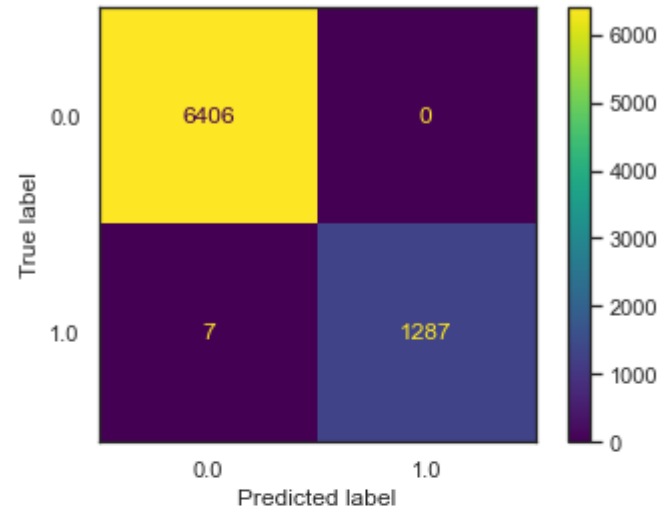
Fig.87 Best parameters of Tuned Gradient Boost Model

AUC & ROC Graph of Training Data

Area under the curve: 1.000

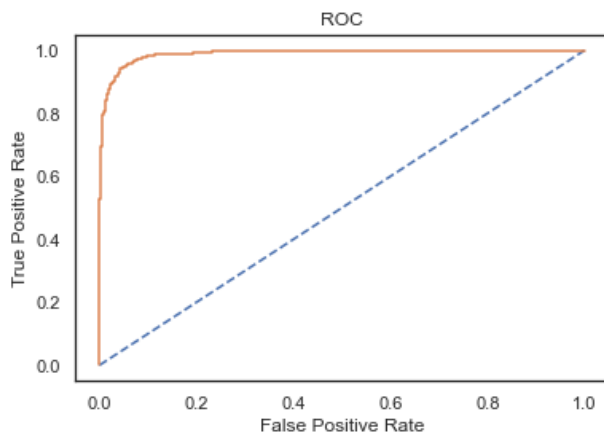


Confusion Matrix for Training Data

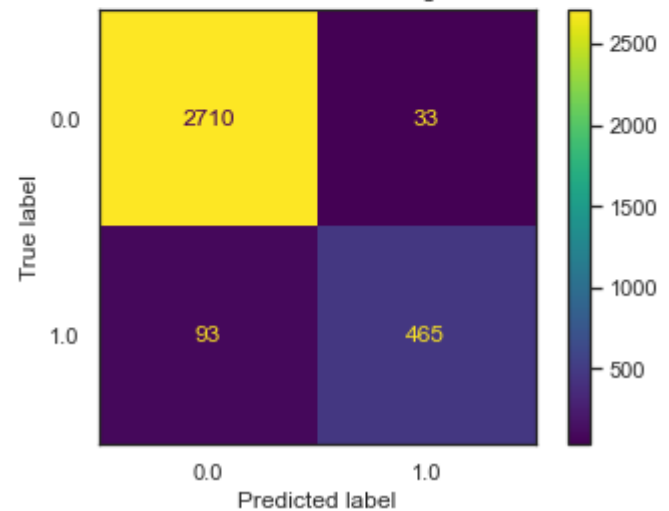


AUC & ROC Graph of Testing Data

Area under the curve: 0.990



Confusion Matrix for Testing Data



**Fig.88 Tuned Gradient Boost – AUC & ROC curve
Confusion Matrix**

Fig.89 Tuned Gradient Boost –

Classification Report of Training Data					
	precision	recall	f1-score	support	
0.0	1.00	1.00	1.00	6406	
1.0	1.00	0.99	1.00	1294	
accuracy			1.00	7700	
macro avg	1.00	1.00	1.00	7700	
weighted avg	1.00	1.00	1.00	7700	

Classification Report of Testing Data					
	precision	recall	f1-score	support	
0.0	0.97	0.99	0.98	2743	
1.0	0.93	0.83	0.88	558	
accuracy			0.96	3301	
macro avg	0.95	0.91	0.93	3301	
weighted avg	0.96	0.96	0.96	3301	

Fig.90 Tuned Gradient Boost – Classification Report

Tuned Gradient Boost: Metrics Summary

Accuracy on training set : 0.999
 Accuracy on test set : 0.962
 Recall on training set : 0.995
 Recall on test set : 0.833
 Precision on training set : 1.0
 Precision on test set : 0.934
 F1 on training set : 0.997
 F1 on test set : 0.881

Fig.91 Tuned Gradient Boost – Metrics Summary

Inference:

- The Gradient Boost model appears to be performing well, with high accuracy and precision scores on both the training and test sets. The recall scores are also decent but could be improved, particularly on the test set.
- However, the high accuracy and precision scores on the training set compared to the test set suggest that the model may be overfitting the training data. The precision score on the test set is notably lower than the training set, which is another indication of overfitting. It may be worth considering regularization techniques to reduce overfitting, such as reducing the model complexity or adding penalties to the loss function.

Model 11 – Random Forest Classifier

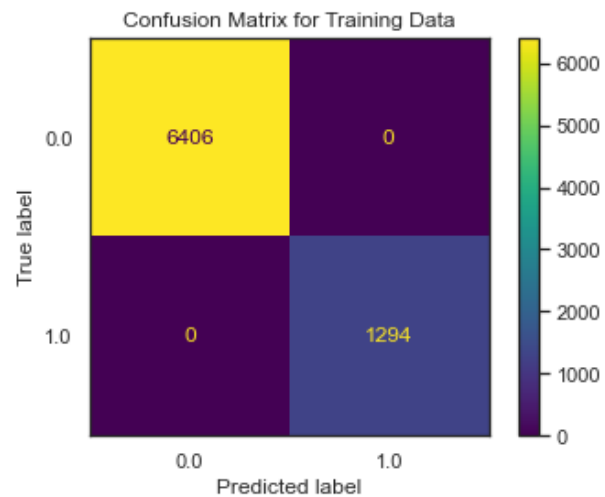
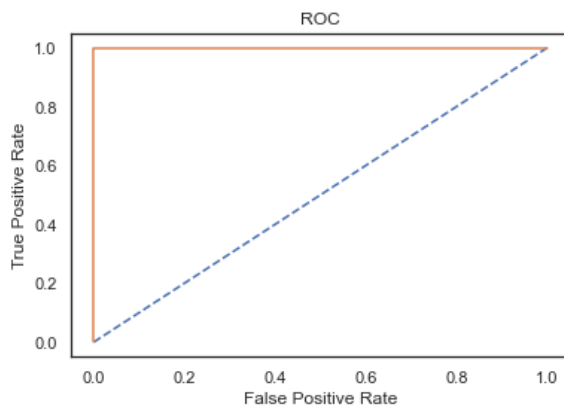
The resulting model is built by default parameters

```
RandomForestClassifier
RandomForestClassifier(random_state=1)
```

Fig.92 RF Model

AUC & ROC Graph of Training Data

Area under the curve: 1.000



AUC & ROC Graph of Testing Data

Area under the curve: 0.990

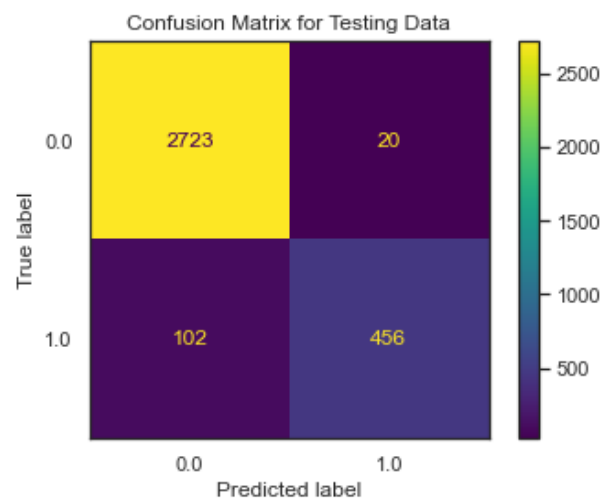
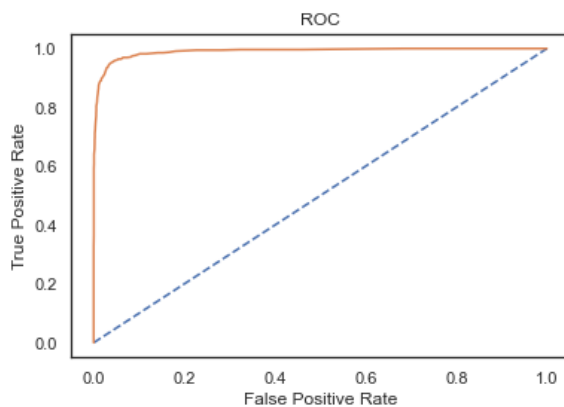


Fig.93 RF – AUC & ROC curve

Fig.94 RF – Confusion Matrix

Classification Report of Training Data				
	precision	recall	f1-score	support
0.0	1.00	1.00	1.00	6406
1.0	1.00	1.00	1.00	1294
accuracy			1.00	7700
macro avg	1.00	1.00	1.00	7700
weighted avg	1.00	1.00	1.00	7700

Classification Report of Testing Data				
	precision	recall	f1-score	support
0.0	0.96	0.99	0.98	2743
1.0	0.96	0.82	0.88	558
accuracy			0.96	3301
macro avg	0.96	0.90	0.93	3301
weighted avg	0.96	0.96	0.96	3301

Fig.95 RF – Classification Report

RF: Metrics Summary

```

Accuracy on training set : 1.0
Accuracy on test set : 0.963
Recall on training set : 1.0
Recall on test set : 0.817
Precision on training set : 1.0
Precision on test set : 0.958
F1 on training set : 1.0
F1 on test set : 0.882

```

Fig.96 RF – Metrics Summary

Inference:

- The model seems to be performing well as it has high accuracy on both the training and test sets. However, the recall on the test set is relatively low compared to the other metrics, which may indicate that the model is not performing as well at identifying positive cases. The precision on the test set is quite high, indicating that when the model does predict a positive case, it is usually correct.
- The F1 score on the test set is also quite good, although it is lower than the F1 score on the training set, which may suggest some overfitting.
- Overall, it seems that the model is performing well, but some additional tuning or regularization may be needed to reduce overfitting.

Model 12 – Tuned Random Forest Model

The resulting model is built by using grid search and the best params are used

```
GridSearchCV
GridSearchCV(cv=5, estimator=RandomForestClassifier(random_state=1), n_jobs=-1,
             param_grid={'criterion': ['gini', 'entropy'],
                          'max_depth': [7, 8, 9],
                          'max_features': ['sqrt', 'log2'],
                          'min_samples_leaf': [12, 15, 17],
                          'min_samples_split': [30, 50, 70],
                          'n_estimators': [100, 200, 300]},
             scoring='f1')
  estimator: RandomForestClassifier
RandomForestClassifier(random_state=1)
  RandomForestClassifier
RandomForestClassifier(random_state=1)
```

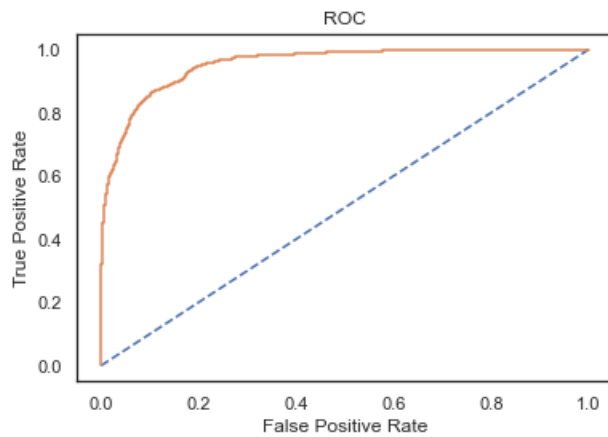
Fig.97 Tuned RF Model

```
RandomForestClassifier
RandomForestClassifier(criterion='entropy', max_depth=9, min_samples_leaf=12,
                      min_samples_split=30, n_estimators=200, random_state=1)
```

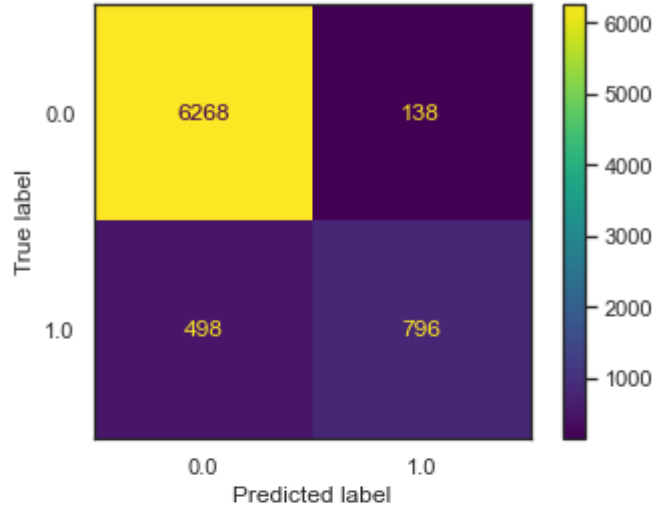
Fig.98 Best parameters of Tuned RF Model

AUC & ROC Graph of Training Data

Area under the curve: 0.954

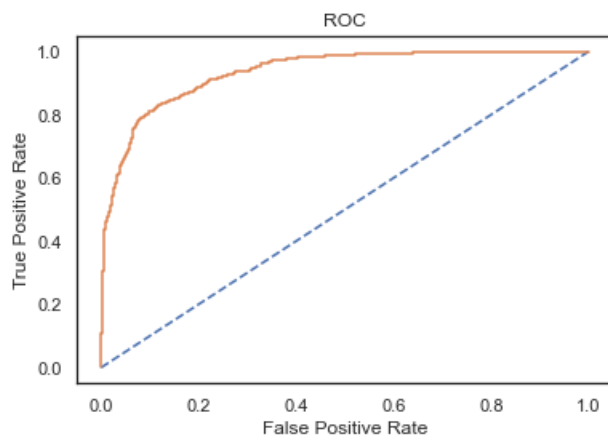


Confusion Matrix for Training Data



AUC & ROC Graph of Testing Data

Area under the curve: 0.935



Confusion Matrix for Testing Data

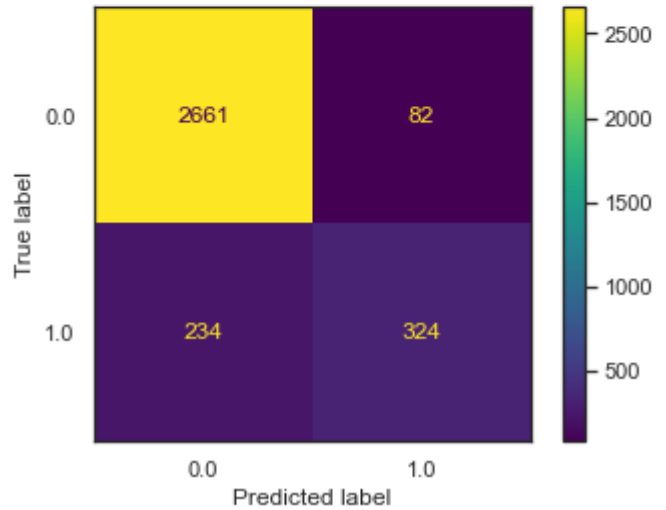


Fig.99 Tuned RF – AUC & ROC curve

Fig.100 Tuned RF – Confusion Matrix

Classification Report of Training Data					
	precision	recall	f1-score	support	
0.0	0.93	0.99	0.96	6406	
1.0	0.90	0.63	0.74	1294	
accuracy			0.93	7700	
macro avg	0.91	0.81	0.85	7700	
weighted avg	0.92	0.93	0.92	7700	

Classification Report of Testing Data					
	precision	recall	f1-score	support	
0.0	0.92	0.98	0.95	2743	
1.0	0.83	0.59	0.69	558	
accuracy			0.91	3301	
macro avg	0.88	0.78	0.82	3301	
weighted avg	0.91	0.91	0.90	3301	

Fig.101 Tuned RF – Classification Report

Tuned RF: Metrics Summary

Accuracy on training set : 0.742
 Accuracy on test set : 0.688
 Recall on training set : 0.631
 Recall on test set : 0.586
 Precision on training set : 0.9
 Precision on test set : 0.834
 F1 on training set : 0.742
 F1 on test set : 0.688

Fig.102 Tuned RF – Metrics Summary

Inference:

- From the given evaluation metrics, we can see that the accuracy on the training set is significantly higher than that on the test set. This is an indication that the model is overfitting to the training data.
- Additionally, the recall, precision, and F1 scores are also higher on the training set than on the test set. This further supports the fact that the model is overfitting.
- Overall, the model seems to have been tuned too much on the training data, and as a result, it is not generalizing well to new data.

Model 13 – Artificial Neural Network Classifier

The resulting model is built by default parameters

```

▼ MLPClassifier
MLPClassifier()
    
```

Fig.103 ANN Model

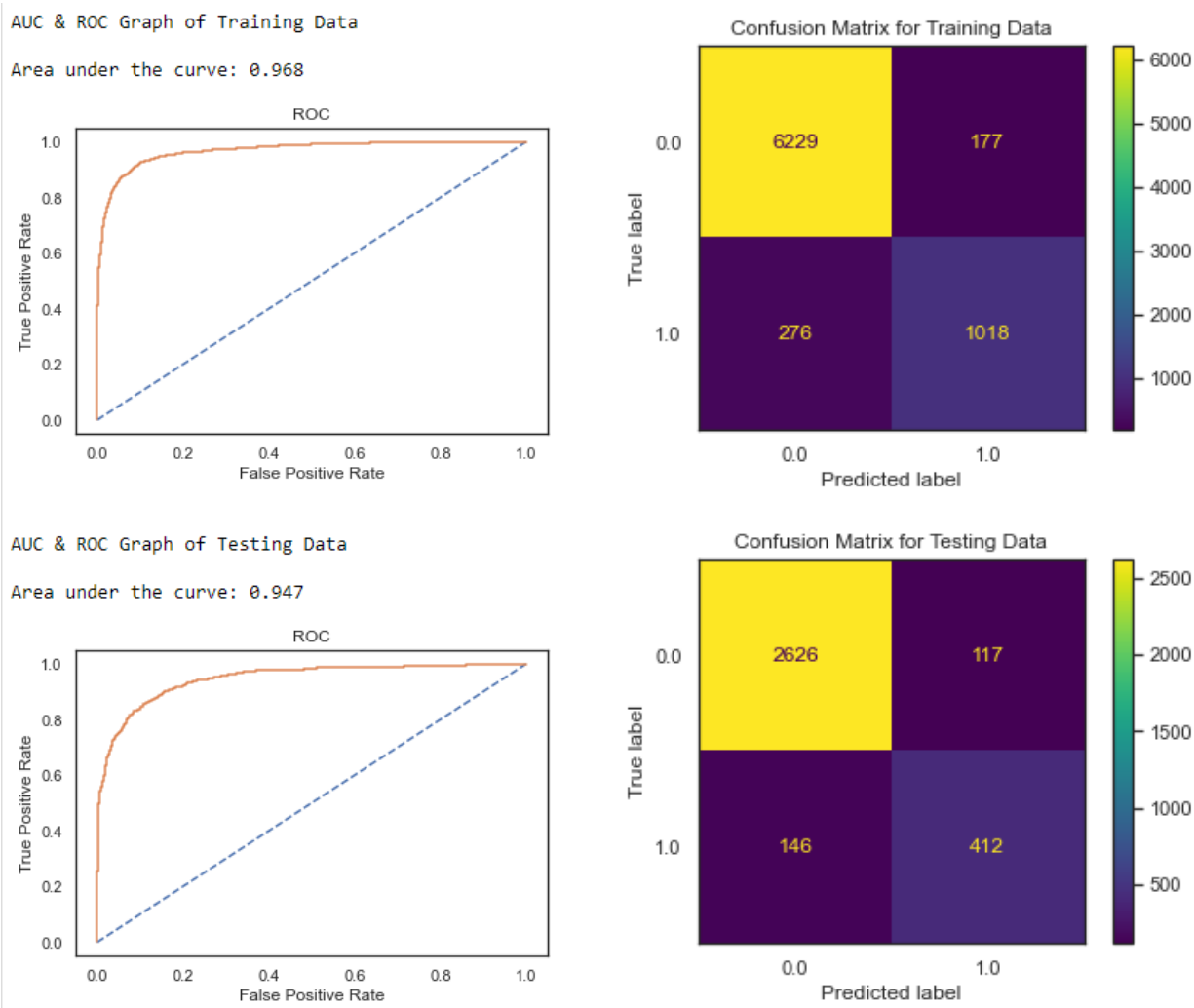


Fig.104 ANN – AUC & ROC curve

Fig.105 ANN – Confusion Matrix

Classification Report of Training Data					
	precision	recall	f1-score	support	
0.0	0.96	0.97	0.96	6406	
1.0	0.85	0.79	0.82	1294	
accuracy			0.94	7700	
macro avg	0.90	0.88	0.89	7700	
weighted avg	0.94	0.94	0.94	7700	

Classification Report of Testing Data					
	precision	recall	f1-score	support	
0.0	0.95	0.96	0.95	2743	
1.0	0.78	0.74	0.76	558	
accuracy			0.92	3301	
macro avg	0.86	0.85	0.86	3301	
weighted avg	0.92	0.92	0.92	3301	

Fig.106 ANN – Classification Report

ANN: Metrics Summary

Accuracy on training set : 0.941
 Accuracy on test set : 0.92
 Recall on training set : 0.787
 Recall on test set : 0.738
 Precision on training set : 0.852
 Precision on test set : 0.779
 F1 on training set : 0.818
 F1 on test set : 0.758

Fig.107 ANN – Metrics Summary

Inference:

- The model seems to be performing well as both the training and test set metrics are close to each other. The accuracy of the model on the training set is 0.941, and on the test set, it is 0.92, which suggests that the model is able to generalize well on unseen data.
- The recall, precision, and F1 score of the model on the test set are 0.738, 0.779, and 0.758, respectively, which indicates that the model is performing reasonably well in terms of predicting the positive class.
- Overall, the model seems to be neither overfitting nor underfitting, as the training and test set metrics are close to each other. However, the model could be further improved by fine-tuning hyperparameters or trying out different algorithms.

Model 14 – Tuned ANN Model

The resulting model is built by using grid search and the best params are used

```

GridSearchCV
GridSearchCV(cv=5,
              estimator=MLPClassifier(max_iter=5000, random_state=1,
                                      verbose=True),
              n_jobs=-1,
              param_grid={'activation': ['logistic', 'tanh', 'relu'],
                          'alpha': [0.001, 0.01], 'hidden_layer_sizes': [100],
                          'learning_rate': ['constant', 'adaptive'],
                          'solver': ['sgd', 'adam']},
              scoring='f1')
  estimator: MLPClassifier
MLPClassifier(max_iter=5000, random_state=1, verbose=True)
  MLPClassifier
MLPClassifier(max_iter=5000, random_state=1, verbose=True)

```

Fig.108 Tuned ANN Model

```

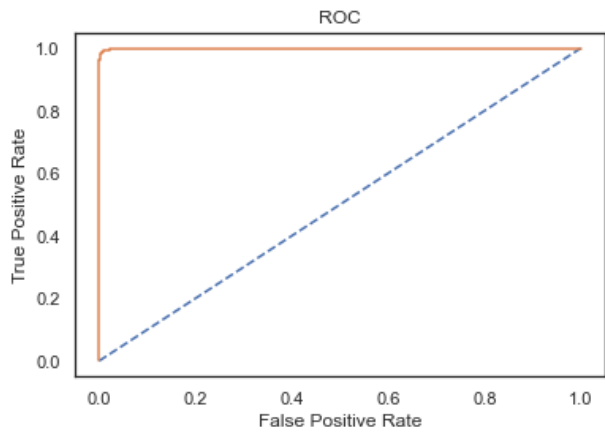
MLPClassifier
MLPClassifier(activation='tanh', alpha=0.001, hidden_layer_sizes=100,
              max_iter=5000, random_state=1, verbose=True)

```

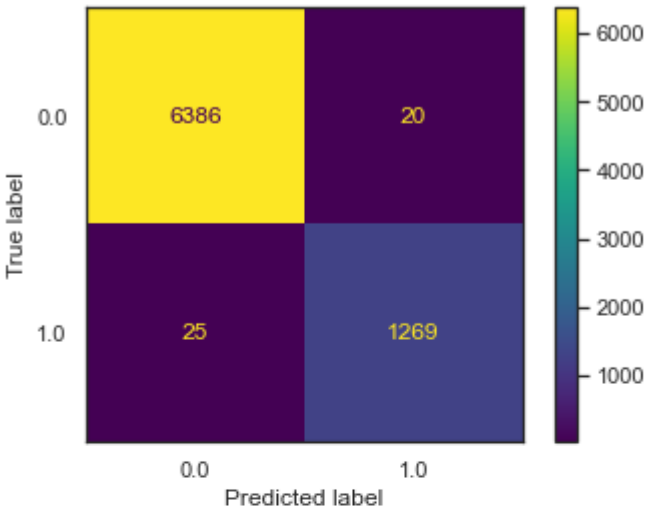
Fig.109 Best parameters of Tuned ANN Model

AUC & ROC Graph of Training Data

Area under the curve: 0.999

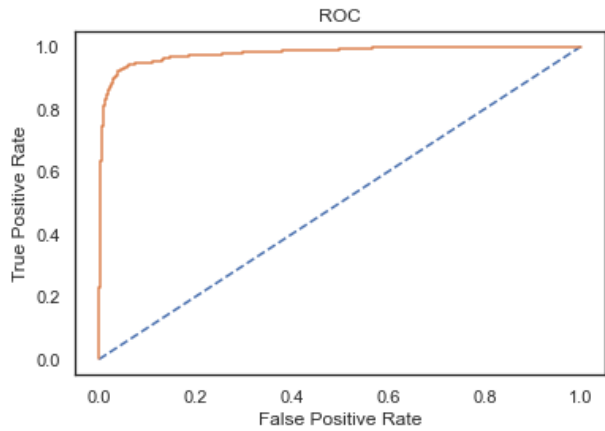


Confusion Matrix for Training Data



AUC & ROC Graph of Testing Data

Area under the curve: 0.979



Confusion Matrix for Testing Data

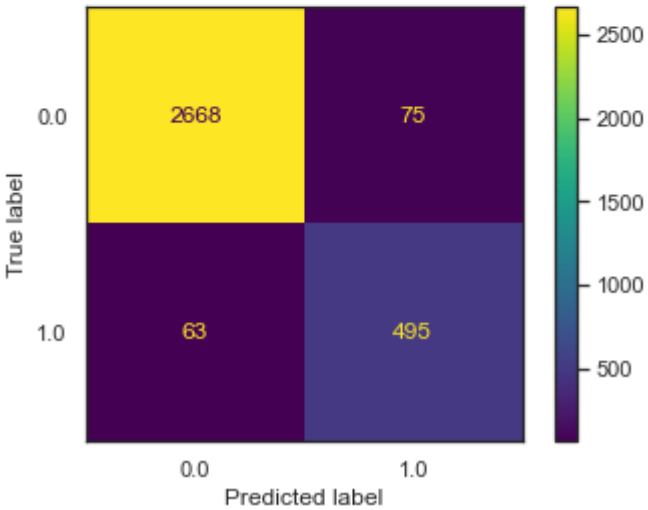


Fig.110 Tuned ANN – AUC & ROC curve

Fig.111 Tuned ANN – Confusion Matrix

Classification Report of Training Data				
	precision	recall	f1-score	support
0.0	1.00	1.00	1.00	6406
1.0	0.98	0.98	0.98	1294
accuracy			0.99	7700
macro avg	0.99	0.99	0.99	7700
weighted avg	0.99	0.99	0.99	7700

Classification Report of Testing Data				
	precision	recall	f1-score	support
0.0	0.98	0.97	0.97	2743
1.0	0.87	0.89	0.88	558
accuracy			0.96	3301
macro avg	0.92	0.93	0.93	3301
weighted avg	0.96	0.96	0.96	3301

Fig.112 Tuned ANN – Classification Report

Tuned ANN: Metrics Summary

Accuracy on training set : 0.983
 Accuracy on test set : 0.878
 Recall on training set : 0.981
 Recall on test set : 0.887
 Precision on training set : 0.984
 Precision on test set : 0.868
 F1 on training set : 0.983
 F1 on test set : 0.878

Fig.113 Tuned ANN – Metrics Summary

Inference:

- The accuracy on the training set is high (0.983) which suggests that the model is performing well on the training data. However, the accuracy on the test set is relatively lower (0.878), which suggests that the model may be overfitting to the training data.
- The recall on both the training and test sets is relatively high (0.981 and 0.887 respectively), indicating that the model is able to identify a high proportion of positive instances. Similarly, the precision on both sets is also relatively high (0.984 on the training set and 0.868 on the test set), which suggests that the model makes few false positive predictions.
- Overall, the F1 score on both sets is high (0.983 on the training set and 0.878 on the test set), which is a good indication that the model is performing well. However, the lower accuracy on the test set suggests that there may still be room for improvement in the model's ability to generalize to new data.

Model 15 – KNN Classifier

The resulting model is built by default parameters

```

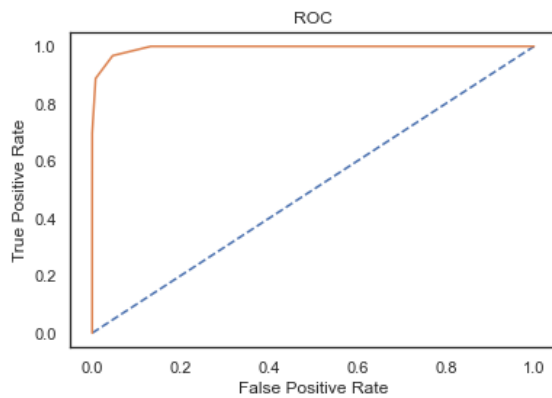
KNeighborsClassifier
KNeighborsClassifier()

```

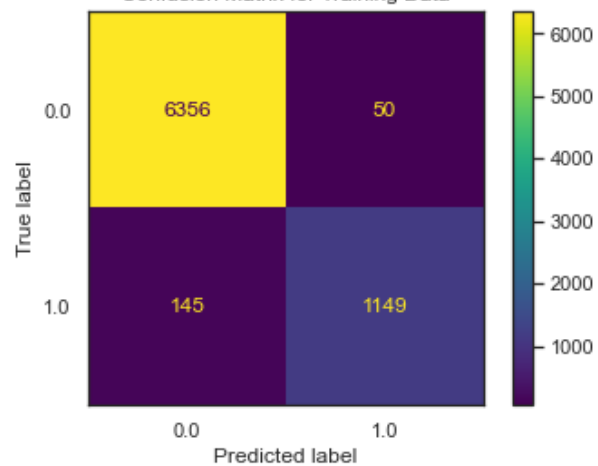
Fig.114 KNN Model

AUC & ROC Graph of Training Data

Area under the curve: 0.994

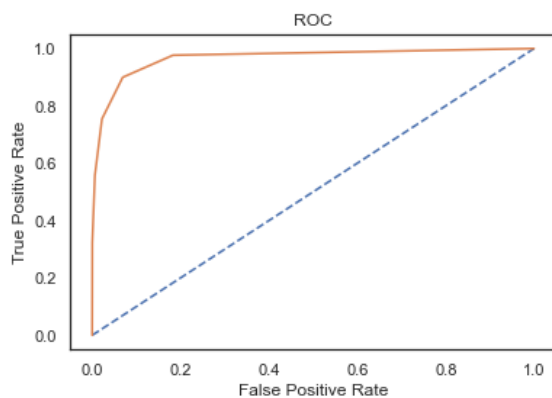


Confusion Matrix for Training Data



AUC & ROC Graph of Testing Data

Area under the curve: 0.966



Confusion Matrix for Testing Data

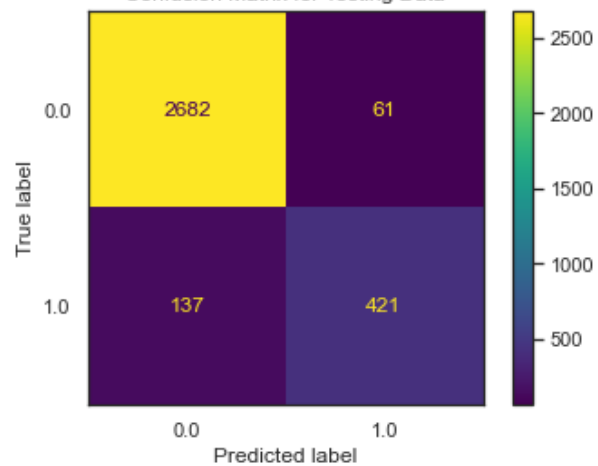


Fig.115 KNN – AUC & ROC curve

Fig.116 KNN – Confusion Matrix

Classification Report of Training Data				
	precision	recall	f1-score	support
0.0	0.99	1.00	0.99	6406
1.0	0.97	0.94	0.96	1294
accuracy			0.99	7700
macro avg	0.98	0.97	0.97	7700
weighted avg	0.99	0.99	0.99	7700

Classification Report of Testing Data				
	precision	recall	f1-score	support
0.0	0.97	0.98	0.97	2743
1.0	0.89	0.85	0.87	558
accuracy			0.96	3301
macro avg	0.93	0.91	0.92	3301
weighted avg	0.96	0.96	0.96	3301

Fig.117 KNN – Classification Report

KNN: Metrics Summary

Accuracy on training set : 0.986
 Accuracy on test set : 0.957
 Recall on training set : 0.941
 Recall on test set : 0.849
 Precision on training set : 0.974
 Precision on test set : 0.891
 F1 on training set : 0.958
 F1 on test set : 0.87

Fig.118 KNN – Metrics Summary

Inference:

- The KNN model has performed well on both training and test sets, with high accuracy and precision. However, the recall on the test set is relatively lower than the recall on the training set, indicating that the model may be overfitting to the training data.
- Additionally, the F1 score on the test set is lower than the F1 score on the training set, which is another indication of potential overfitting. Further analysis, such as cross-validation or adjusting the model's hyperparameters, may be necessary to address this overfitting issue.

Model 16 – Tuned KNN Classifier

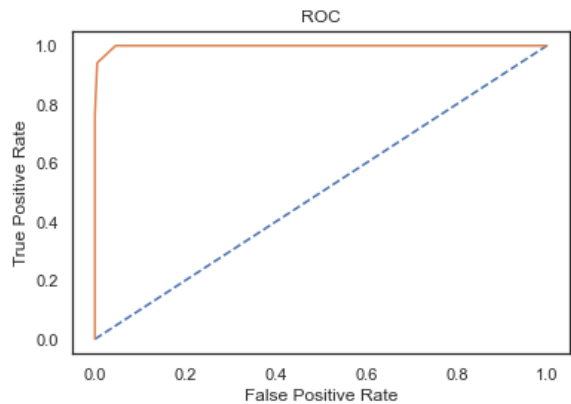
The resulting model is built by default parameters

```
▼ KNeighborsClassifier
KNeighborsClassifier()
```

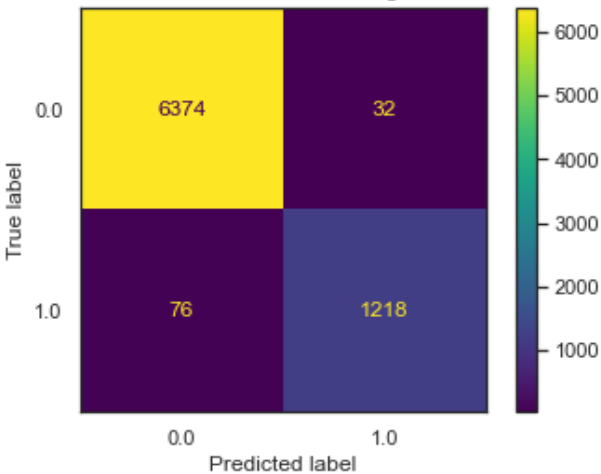
Fig.119 Tuned KNN Model

AUC & ROC Graph of Training Data

Area under the curve: 0.998

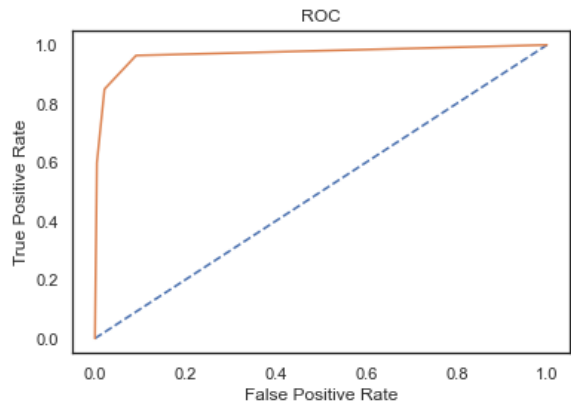


Confusion Matrix for Training Data



AUC & ROC Graph of Testing Data

Area under the curve: 0.969



Confusion Matrix for Testing Data

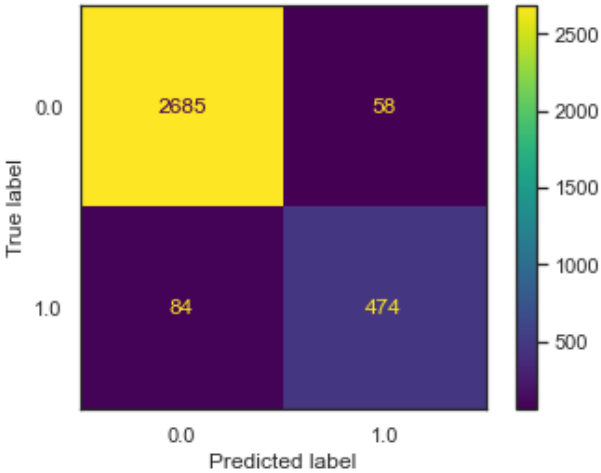


Fig.120 Tuned KNN – AUC & ROC curve

Fig.121 Tuned KNN – Confusion Matrix

Classification Report of Training Data				
	precision	recall	f1-score	support
0.0	1.00	1.00	1.00	6406
1.0	1.00	1.00	1.00	1294
accuracy			1.00	7700
macro avg	1.00	1.00	1.00	7700
weighted avg	1.00	1.00	1.00	7700

Classification Report of Testing Data				
	precision	recall	f1-score	support
0.0	0.97	0.99	0.98	2743
1.0	0.94	0.85	0.89	558
accuracy			0.97	3301
macro avg	0.95	0.92	0.94	3301
weighted avg	0.96	0.97	0.96	3301

Fig.122 Tuned KNN – Classification Report

Tuned KNN: Metrics Summary

```

Accuracy on training set : 1.0
Accuracy on test set : 0.965
Recall on training set : 1.0
Recall on test set : 0.853
Precision on training set : 1.0
Precision on test set : 0.935
F1 on training set : 1.0
F1 on test set : 0.892

```

Fig.123 Tuned KNN – Metrics Summary

Inference:

- The accuracy on the training set is 0.986 and on the test set is 0.957. The recall on the training set is 0.941 and on the test set is 0.849. The precision on the training set is 0.974 and on the test set is 0.891. The F1 score on the training set is 0.958 and on the test set is 0.87.
- The test set performance is reasonably good and the difference between the training and test set accuracy and F1 scores is not very large, indicating that the model is not overfitting. However, there is still some difference between the training and test set recall and precision scores, suggesting that there may be some overfitting present. Overall, the model seems to be performing well and can be considered for use in practical applications.

Model 17 – Bagging Classifier

The resulting model is built by default parameters

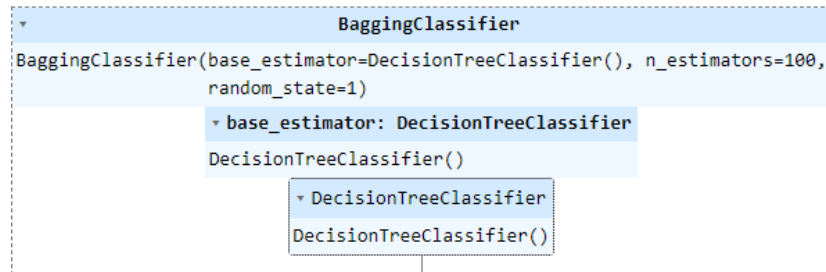
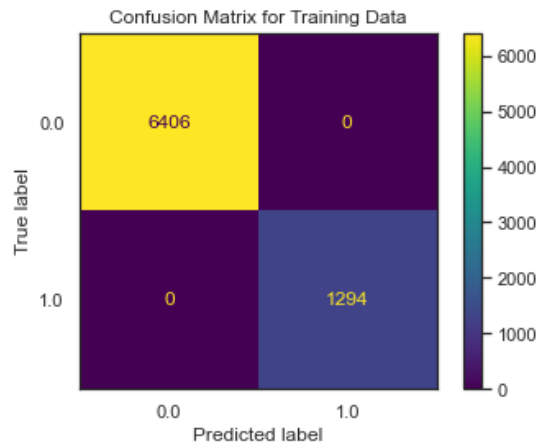
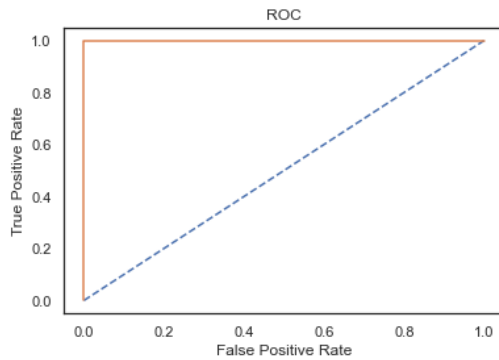


Fig.124 Bagging Model

AUC & ROC Graph of Training Data

Area under the curve: 1.000



AUC & ROC Graph of Testing Data

Area under the curve: 0.986

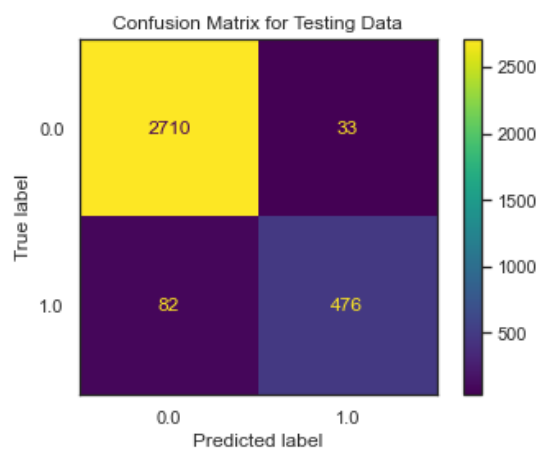
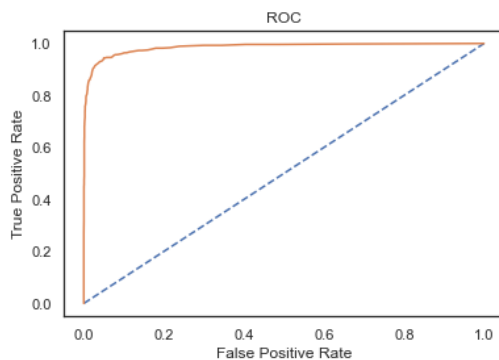


Fig.125 Bagging – AUC & ROC curve

Fig.126 Bagging – Confusion Matrix

Classification Report of Training Data				
	precision	recall	f1-score	support
0.0	1.00	1.00	1.00	6406
1.0	1.00	1.00	1.00	1294
accuracy			1.00	7700
macro avg	1.00	1.00	1.00	7700
weighted avg	1.00	1.00	1.00	7700

Classification Report of Testing Data				
	precision	recall	f1-score	support
0.0	0.97	0.99	0.98	2743
1.0	0.94	0.85	0.89	558
accuracy			0.97	3301
macro avg	0.95	0.92	0.94	3301
weighted avg	0.96	0.97	0.96	3301

Fig.127 Bagging – Classification Report

Bagging: Metrics Summary

```

Accuracy on training set : 1.0
Accuracy on test set : 0.965
Recall on training set : 1.0
Recall on test set : 0.853
Precision on training set : 1.0
Precision on test set : 0.935
F1 on training set : 1.0
F1 on test set : 0.892

```

Fig.128 Bagging – Metrics Summary

Inference:

- The bagging model appears to perform well, with high accuracy and precision scores on both the training and test sets. The recall scores are also quite good, though slightly lower than the precision scores on both sets. The F1 scores are also relatively high, indicating a good balance between precision and recall.

Since the accuracy and other metrics are high on both the training and test sets, and the difference between the training and test set performance is not significant, the model is not overfitting or underfitting.