CAPSTONE PROJECT

Business Report - Customer Churn Prediction

Rohan R. Khade
DSBA Mar-2022 Batch

Table of Contents

List of Tables

List of Figures

Chapter 1. Introduction & Initial Data Report

1.1 Defining problem statement

A DTH service provider is facing a lot of competition in the current market and it has become a challenge to retain the existing customers in the current situation. Hence, the company wants to develop a model through which they can do churn prediction of the accounts and provide segmented offers to the potential churners. In this company, account churn is a major thing because 1 account can have multiple customers. hence by losing one account the company might be losing more than one customer.

This project is aimed at developing a churn prediction model for this company and to provide business recommendations on a campaign focused on customer retention. The campaign recommendation should be such that it does not entail a huge cost for retention of customers and should remain within a budget earmarked for this purpose.

1.2 Need of the study/project

A subscription/service-based company's highest cost is the acquisition cost of a new customer. To mitigate this cost and convert a customer into a profitable one, customer retention is also important as longer tenure of the customer means higher revenue post cost of acquisition is recovered. So, conducting this study important to understand the factors leading to churning or not churning of a customer.

DTH service providers are under pressure to widen their customer base to maintain and improve their profitability as most of them have a fixed broadcaster/content provider fee irrespective of the number of customers in their customer base. So, more the number of customers, greater their profitability. As a result, it becomes very important to not only increase the customer base but also protect the current customer base.

Being a service-based entity, the DTH company earns majority of its revenue from its package subscribers, may it be topline or bottom-line customers, retention of these customers is of high priority. Providing all customers with offers to retain them would make a dent in the profitability and hence it is very important to focus only on select set of customers who are at a higher risk of churning.

Acquiring a new customer can cost five times more than retaining an existing customer. Increasing customer retention by 5% can increase profits from 25-95%. As customer churn directly impacts both the top-line and bottom-line revenue of the business, existing customer base needs to be protected. Providing all customers with offers to retain them would make a dent in the profitability and hence it is very important to focus only on select set of customers who are at a higher risk of churning.

1.3 Understanding business/social opportunity

This a case study of a DTH company where in they have customers assigned with the unique account ID and a single account ID can hold many customers (like a family plan) across gender and marital status, customers get flexibility in terms of the mode of payment they want to opt for. Customers are again segmented across various types of plans they opt for as per their usage which is also based on the device they use (computer or mobile) moreover, they earn cashback on bill payments.

The overall business runs on customer loyalty and stickiness which in turn comes from providing quality and value-added services. Also, running various promotional and festival offers may help the organization in getting new customers and also retaining the old ones.

We can conclude that a customer retained is a regular income for the organization, a customer added is a new income for the organization and a customer lost will be a negative impact as a single account ID holds multiple numbers of customers i.e.; closure of one account ID means losing multiple customers. It's a great opportunity for the company as it's a need of almost every individual in the family to have a DTH connection which in turn also leads to an increase and competition.

The question arises how can a company creates a difference when compared to other competitors, and what parameter plays a vital role in having customers' loyalty and making them stay? All these social responsibilities will decide the best player in the market.

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

- The dataset includes account level data for the current and churned customers as indicated by the presence of **"churn"** variable.
- The customers data seems to have been collected through random sampling and the **number count of customers** chosen at random is **11,260**
- There are variables such as **agent score, service score, days since last contacted to customer care, etc.** which indicate that the **data has been considered for the past 12 months**
- At the same time, variable such as **year-on-year revenue growth suggests** we have customers **revenue earned per month for past 13 - 24 months as well**
- The data categorizes the customers in to five different account segments, what kind of rating they have given, their revenue per month, their marital status, what city tier they belong to, and other details such as gender, cashback offered to them, and coupons used during transactions

1.5 Variable Details

Table 1 Variable Information

Variable	Details and description
AccountID	This variable represents a unique ID which represents a unique customer. This is of integer data type and there are no null values present in this.
Churn	This is the target variable, which represents if customer has churned or not. This is categorical in nature will no null values. **"0" represents "NO"** and **"1" represents "YES"**.
Tenure	Tenure variable basically tells us the number of months a customer has completed since the account was created
City_Tier	It is a categorical variable, where the customer is bifurcated into 3 different tiers on the basis of city.
CC_Contacted_L1 2m	The variable represents that how many times a customer account has contacted customer care in past 12 months
Payment	The variable provides information on the preferred payment mode by the customers of the company
Gender	The variable provides information about the gender mentioned while opening the account
Service_Score	Service score basically represents the satisfactory scores given by the customers to the services provided by the DTH company
Account_user_count	Account user count shows that how many users are enjoying the services by the company under one account
account_segment	The dataset divides the customers into five different segments namely regular, regular plus, super, super plus, and HNI
CC_Agent_Score	Service score basically represents the scores given by the customers to the agents present on customer care call
Marital_Status	This variable bifurcates the customers into single, married, and divorced based on their marital status
rev_per_month	Monthly average revenue generated by the customers in past 12 months
Complain_l12m	Complaints raised by the customers in last 12 months
rev_growth_yoy	The revenue growth of the accounts based on 12 months revenue vs 24 to 13 months
coupon_used_l12m	Number of times coupons used while transacting in last 12 months
Day_Since_CC_connect	Number days since the user has contacted the customer care
cashback_l12m	Average monthly cashback received by the customers in the last 12 months
Login_device	Customers' preferred login device while using the DTH services provided by the company

1.6 Visual Inspection

Figure 1. Columns and Data Description

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

	count	unique	top	freq	mean	std	min	25%	50%	75%	max
AccountID	11260.0	NaN	NaN	NaN	25629.5	3250.63	20000.0	22814.75	25629.5	28444.25	31259.0
Churn	11260.0	NaN	NaN	NaN	0.17	0.37	0.0	0.0	0.0	0.0	1.0
Tenure	11158.0	38.0	1.0	1351.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN
City_Tier	11148.0	NaN	NaN	NaN	1.65	0.92	1.0	1.0	1.0	3.0	3.0
CC_Contacted_LY	11158.0	NaN	NaN	NaN	17.87	8.85	4.0	11.0	16.0	23.0	132.0
Payment	11151	5	Debit Card	4587	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Gender	11152	4	Male	6328	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Service_Score	11162.0	NaN	NaN	NaN	2.9	0.73	0.0	2.0	3.0	3.0	5.0
Account_user_count	11148.0	7.0	4.0	4569.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN
account_segment	11163	7	Super	4062	NaN	NaN	NaN	NaN	NaN	NaN	NaN
CC_Agent_Score	11144.0	NaN	NaN	NaN	3.07	1.38	1.0	2.0	3.0	4.0	5.0
Marital_Status	11048	3	Married	5860	NaN	NaN	NaN	NaN	NaN	NaN	NaN
rev_per_month	11158.0	59.0	3.0	1746.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Complain_ly	10903.0	NaN	NaN	NaN	0.29	0.45	0.0	0.0	0.0	1.0	1.0
rev_growth_yoy	11260.0	20.0	14.0	1524.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN
coupon_used_for_payment	11260.0	20.0	1.0	4373.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Day_Since_CC_connect	10903.0	24.0	3.0	1816.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN
cashback	10789.0	5693.0	155.62	10.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Login_device	11039	3	Mobile	7482	NaN	NaN	NaN	NaN	NaN	NaN	NaN

1.7 Variable Transformation

Table 2 Bad Data & Null Values

Variable Name	Variable Count	Null Value Count	NaN Treatment Method	Bad Data Presence	Bad Data Treatment
AccountID	11260	0	No Nulls Present	None	N/A
Churn	11260	0	No Nulls Present	None	N/A
Tenure	11158	102	Median	Yes - #	Replaced it with NaN
City_Tier	11148	112	Median	None	N/A
CC_Contacted_LY	11158	102	Median	None	N/A
Payment	11151	109	Mode	None	N/A
Gender	11152	108	Mode	Yes - M,F	Replaced M with Male and F with Female to avoid two different entries for the same segment
Service_Score	11162	98	Mode	None	N/A
Account_user_count	11148	112	Median	Yes - @	Replaced it with NaN
account_segment	11163	97	Mode	"Yes-Regular +	Replaced regular + with regular plus and super + with super plus
CC_Agent_Score	11144	116	Mode	None	N/A
Marital_Status	11048	212	Mode	None	N/A
rev_per_month	11158	102	Median	Yes - +	Replaced it with NaN
Complain_ly	10903	357	Mode	None	N/A
rev_growth_yoy	11260	0	Median	Yes - $	Replaced it with NaN
coupon_used_for_payment	11260	0	Median	Yes - $, *, #	Replaced them with NaN
Day_Since_CC_connect	10903	357	Median	Yes - $	Replaced it with NaN
Cashback	10789	471	Median	Yes - $	Replaced it with NaN
Login_device	11039	221	Mode	Yes - &&&&	Replaced it with NaN

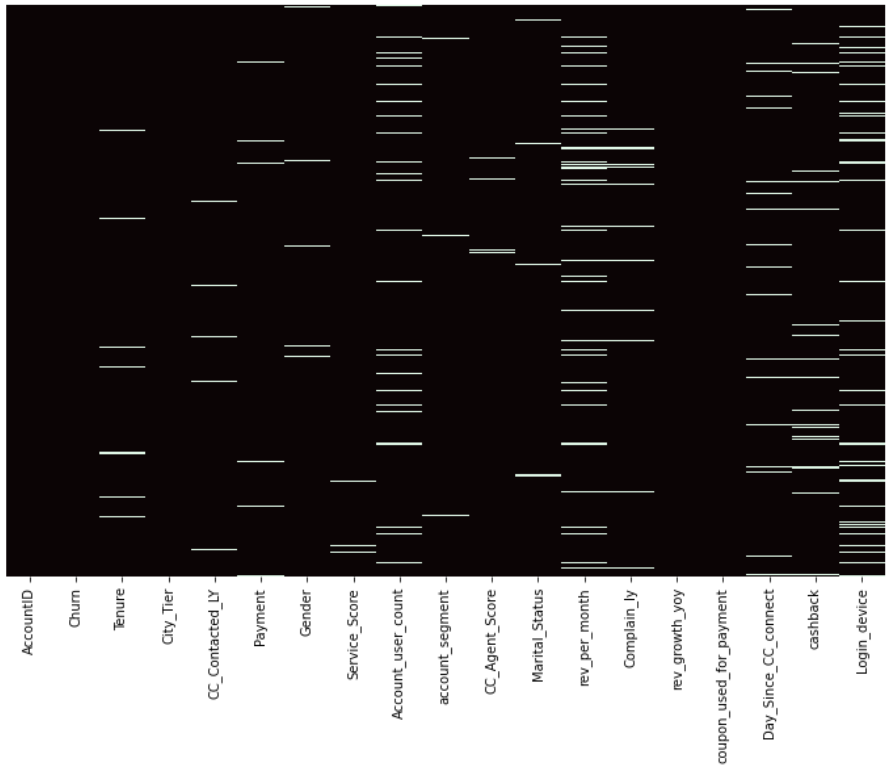
We have used “Median” to impute null values where variable is continuous in nature as Median is less prone to outliers when compared with mean or mode. However, we have used mode to replace null values in the categorical variables. Since we are treating numerical and categorical variables separately, we have treated the null values of each variable individually.

Chapter 2. Expolratory Data Analysis

2.1 Null Value Treatment

Figure 2. Dataframe (df): Percentage of Nulls per Variable (%)

rev_per_month	7.02
Login_device	6.75
cashback	4.20
Account_user_count	3.94
Day_Since_CC_connect	3.18
Complain_ly	3.17
Tenure	1.94
Marital_Status	1.88
CC_Agent_Score	1.03
City_Tier	0.99
Payment	0.97
Gender	0.96
CC_Contacted_LY	0.91
Service_Score	0.87
account_segment	0.86
coupon_used_for_payment	0.03
rev_growth_yoy	0.03
Churn	0.00
AccountID	0.00
dtype: float64	

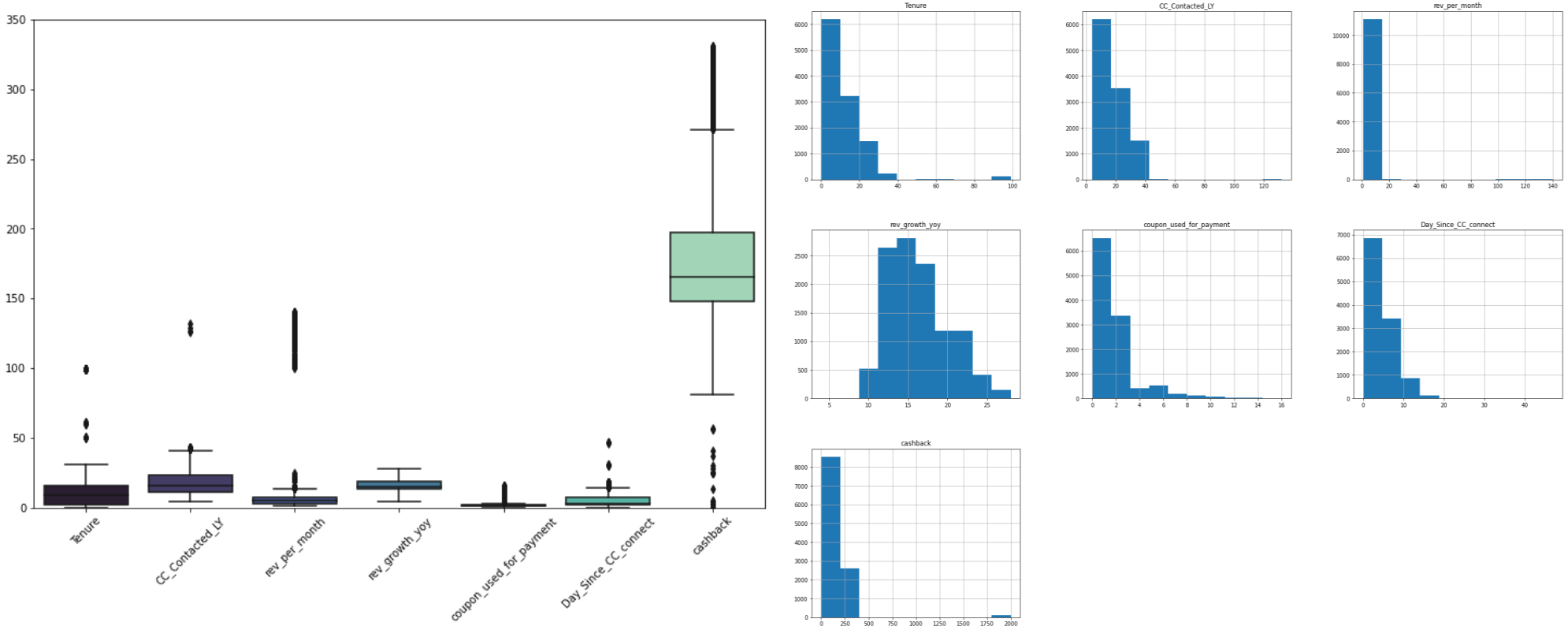


2.2 Removal of Unwanted Variables

We have removed the “AccountID” column before further analysis as the variable does not seem to contribute in accurate inferences and business insights. As the variable helps in understanding the only account ID given to the customer during the opening of the account, it will not provide any insights particularly as name is not associated with any account and as we are trying to predict the churning of customers, account ID will not play any role if a customer is going to churn or not. As a result, we have decided to drop the variable, before performing exploratory data analysis.

2.3 Univariate Analysis

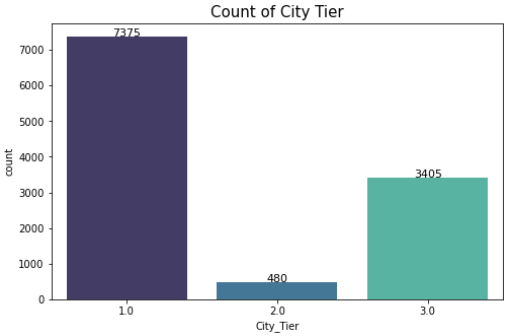
Figure 3. Histograms and boxplot for continuous variables

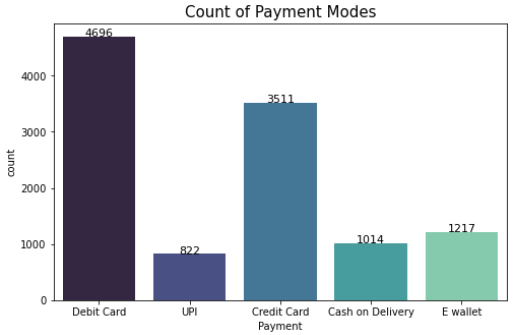


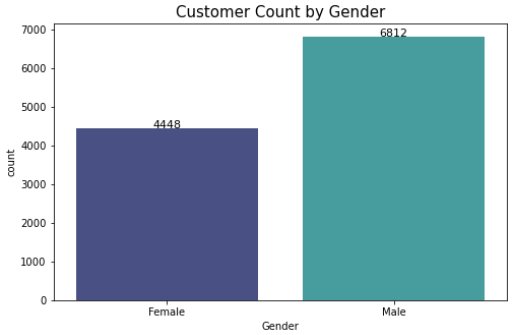
With the exception of rev growth yoy, all numerical variables contain outliers. In contrast to a set of outliers that are well outside the whisker with no in-between values, some outliers for particular variables are closer to the whisker. In the case of rev_per_month, the large gap between 30 and 100 indicates that there are no values in that range. These extreme value outliers are unrelated to equivalent outliers in the cashback field. These anomalies cannot be ruled out as being inaccurate figures because they could be associated with large hotels rooms.

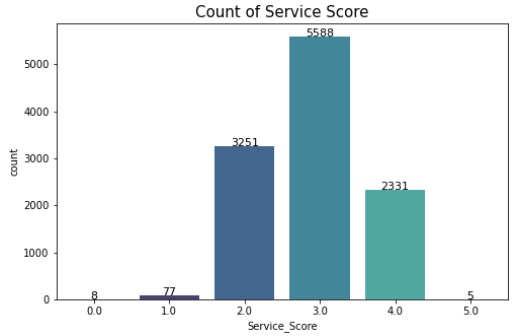
However, models like logistic regression are susceptible to outliers and might not perform well if outliers are not dealt with.Thus, we will employ two modelling strategies: one that treats outliers for outlier-sensitive models and another that does not (leaves outliers as-is) for outlier-resistant models like Random Forest. The highly constrained range of Coupon_used_for_Payment is 0 to 16. Thus, the outliers won't be handled for the purposes of this analysis (similar to categorical variables).

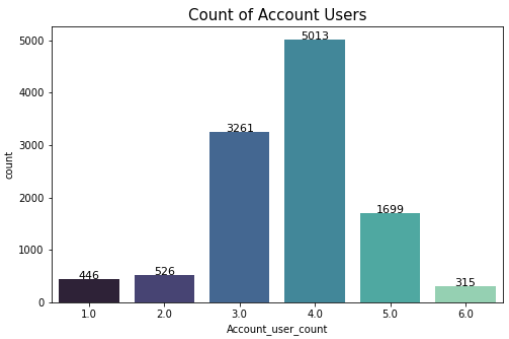
Figure 4. Count plots for categorical variables

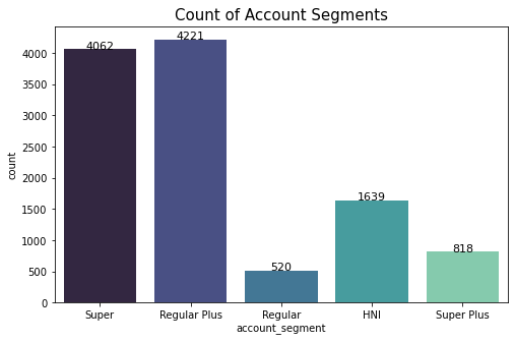


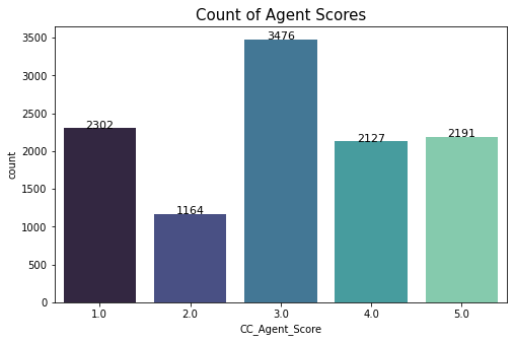


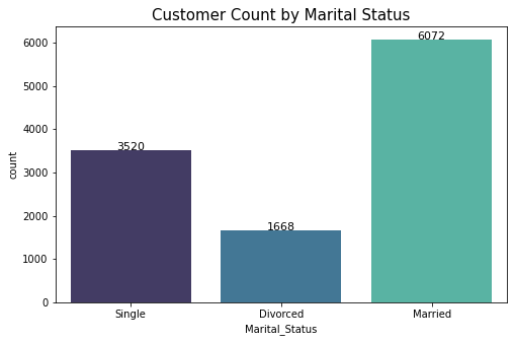


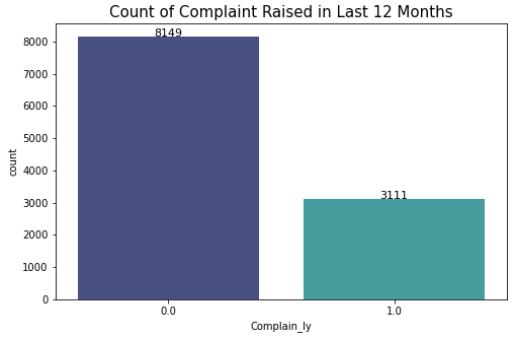


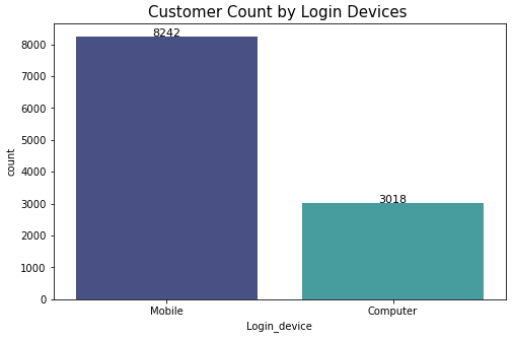












2.4 Bivariate Analysis

Figure 5. Bivariate Analysis – Variables vs Churn

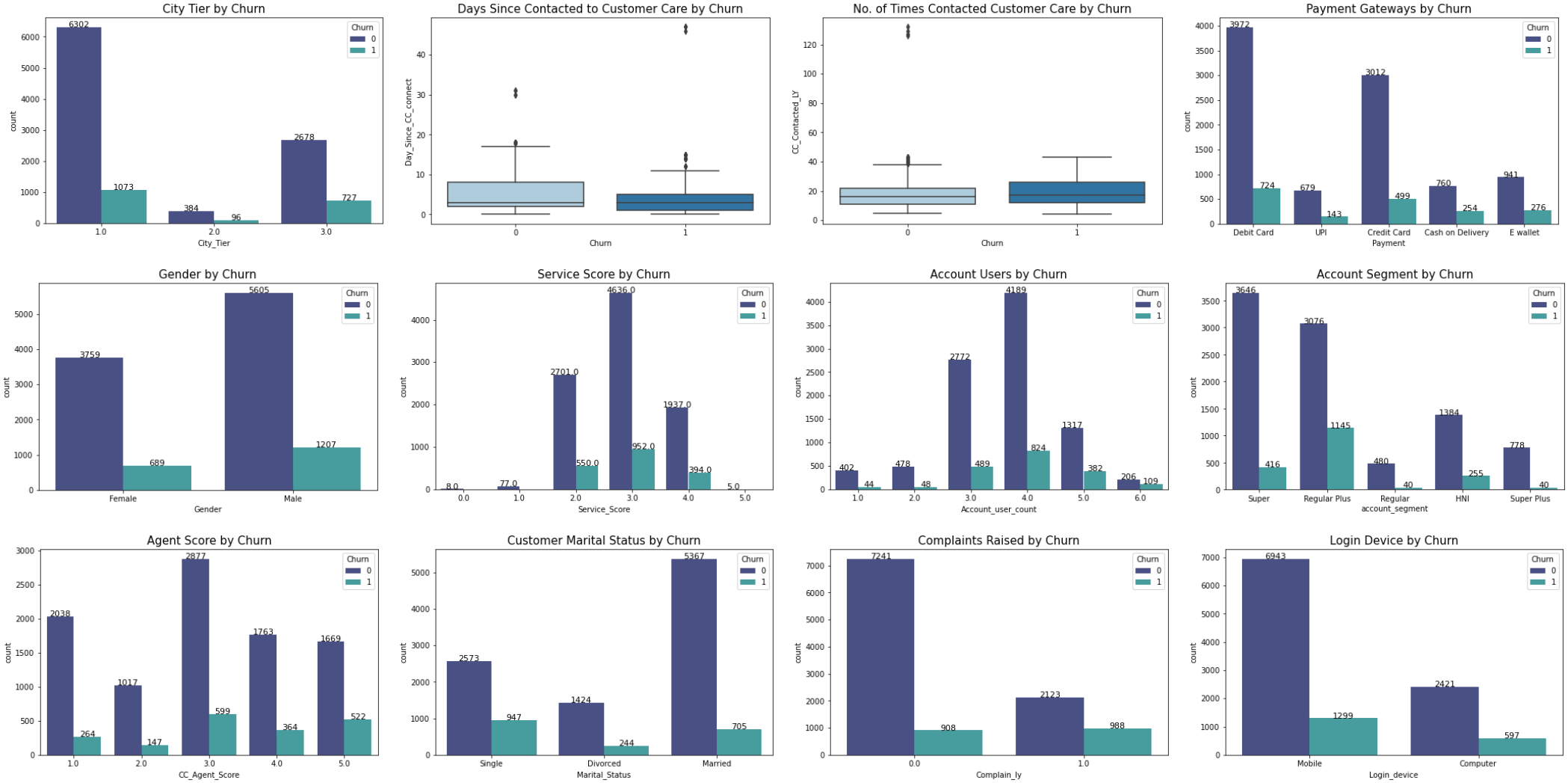
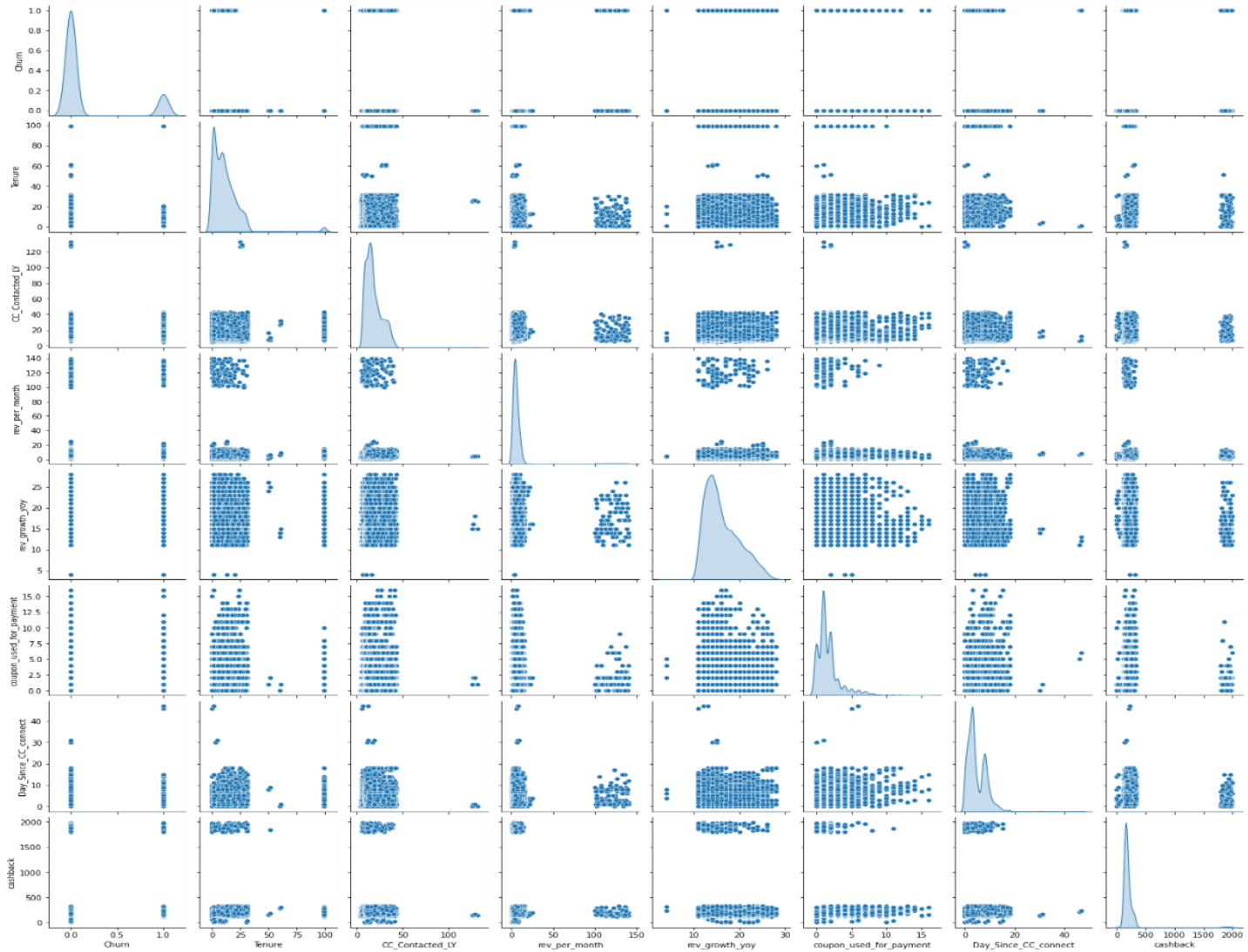


Figure 6. Pairplot



CHI SQUARE TEST

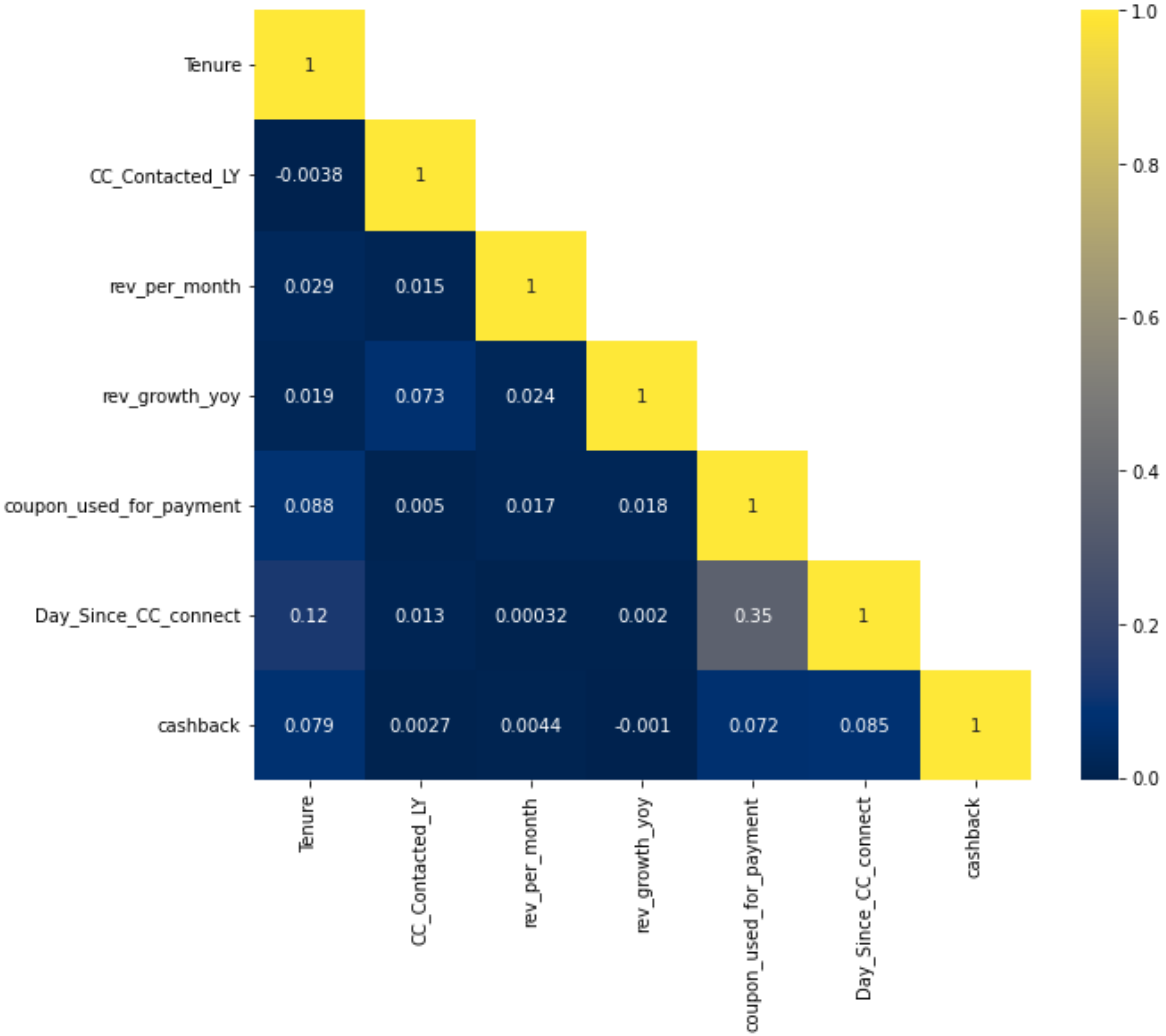
We have conducted Chi Square Analysis at a later stage of the jupyter codebook, wherein we have checked the dependency of categorical variables and we can see that null hypothesis is rejected which states that there is no dependency of the variable, which is incorrect and we cannot drop anu categorical variable as they are relevant.

Variable	chi2	p-value	chi2_output
Gender	9.39	0.00	Reject Ho; Dependent
Service_Score	18.40	0.00	Reject Ho; Dependent
City_Tier	80.54	0.00	Reject Ho; Dependent
Payment	102.71	0.00	Reject Ho; Dependent
account_segment	561.67	0.00	Reject Ho; Dependent
CC_Agent_Score	139.01	0.00	Reject Ho; Dependent
Marital_Status	378.98	0.00	Reject Ho; Dependent
Complain_LY	681.88	0.00	Reject Ho; Dependent

BARTLETT'S & KMO TEST

We have also conducted Bartlett's and KMO test to check the multicollinearity and we can see that it is conclusive that we do not need to remove any variable for model building.

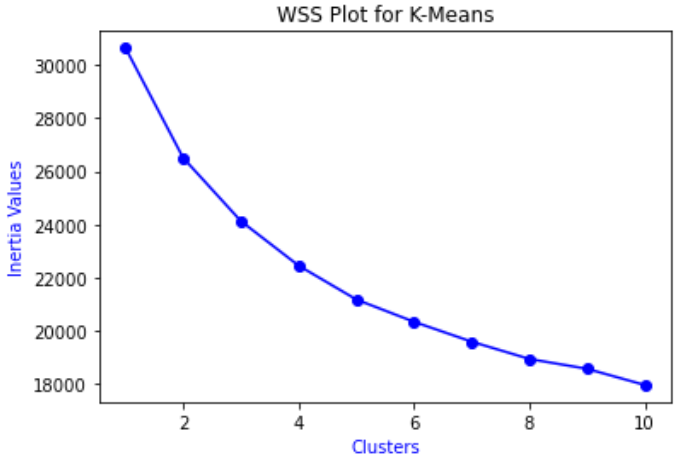
Figure 7. Correlation Heatmap



Note: We have not rounded off the heatmap readings to 2 decimals to understand the correlation between variables accurately.

CLUSTERING

We have used KMeans clustering as part of the clustering analysis for the given dataset, wherein we have tried to analyze if forming clusters can help in understanding the dataset and derive business insights from the same:



Based on our calculations, we have provided silhouette scores for 2, 3, 4, 5, and 6 clusters. Wherein we have realized that forming 4 clusters would be our best option and can provide us with optimum level outcomes. However, we cannot derive any specific insights from clustering and our best option would be to build other models such as logistic regression, LDA, boosting models, etc. to better analyze and derive at accurate business insights.

Furthermore, clustering is the method for dealing with unsurpvised data and based on the aforementioned plot, we cannot derive conclusions from the 3 clusters that we have chosen using elbow method.

2.5 Observations from Univariate and Bivariate Analysis

Connection between Account category and Churn: Regular Plus plan customers appear to churn at a higher rate. To determine whether the plan or pricing needs to be changed, one might compare this plan to rivals' plans that offer the same features and fall within the same price range. To determine why there is increased churn in this account segment, it is also possible to collect and evaluate customer feedback for consumers on this plan. Days since customer care connect and Churn have a lower inverse relationship for churned customers than for active customers.

Data demonstrates that churn occurred soon after customers contacted customer service. In terms of Customer Care Contacted Last Year and Churn, in comparison to active customers, churned customers contacted customer care more often last year. Furthermore, Last Year's Complaints vs. Churn; the proportion of churned customers who complained is much higher than that of active customers.

Customer feedback: 78% of consumers gave the service a rating of 3 or lower (out of a scale of 5). Similarly, 61% of clients gave customer service representatives a score of 3 or lower (out of a scale of 5). This shows that consumer feedback either expresses unhappiness with the service or only bare satisfaction with the customer care contact. The connection between tenure and churn: The churn is particularly high for low tenures, as can be seen in the bivariate histogram for Tenure vs. Churn. Customers have churned at a rate of about 51.85% for tenures of 0 to 1. The root cause of this churn must be identified and remedied. The histogram demonstrates that the proportion of churn reduces as tenure rises.

Monthly revenue and Churn Relationship: High revenue customers churn at a somewhat higher rate than low revenue customers. The fact that not only is there higher turnover, but also more high revenue clients are churning, should worry the DTH provider. If we look at User count vs Churn; A higher percentage of accounts with user counts of 5 and 6 belong to churned customers than active ones. Looking at Payment method and Churn, compared to active customers, churned customers have a higher percentage of consumers who paid with an e-wallet or cash on delivery.

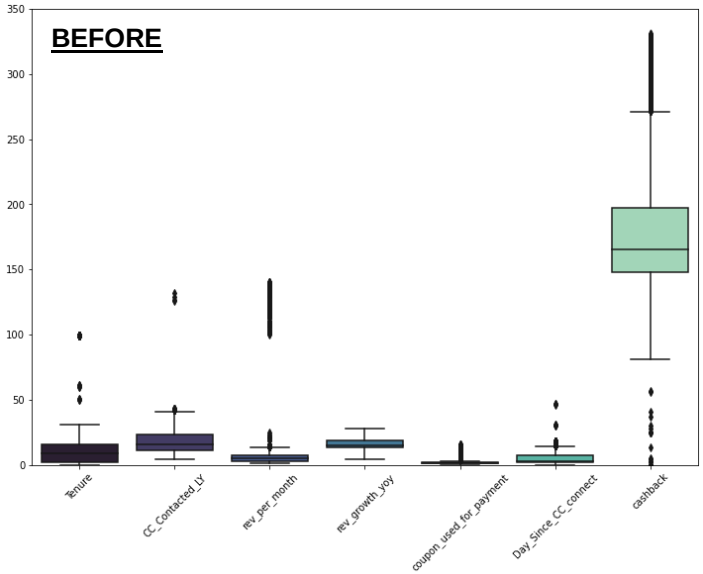
The proportion of churned customers who live in Tier 3 cities is higher than the proportion of active consumers in those cities. The DTH provider needs to investigate whether there is greater competition in those cities than in Tier-1 cities. In addition, the connection between Marital Status and Churn; the customers who are single have churned more frequently than those who are married or divorced.

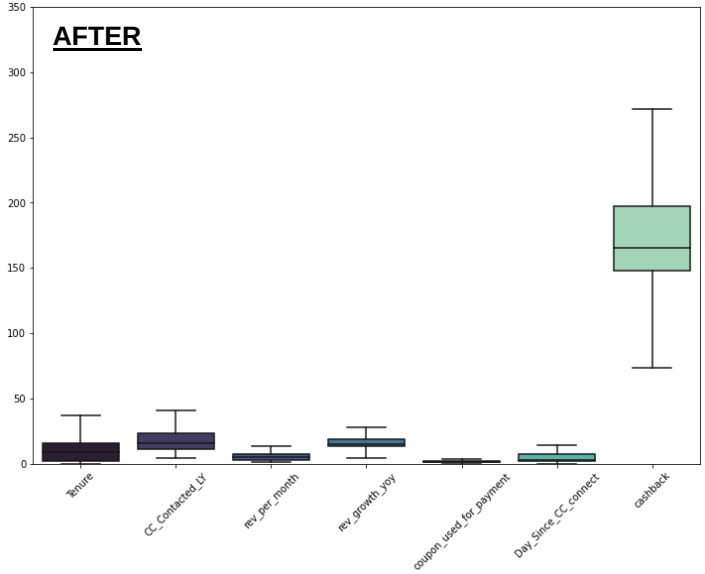
2.6 Outlier Treatment, Encoding, and Scaling

	Outlier Proportion (%)
Account_user_count	6.76
CC_Agent_Score	0.00
CC_Contacted_LY	0.37
Churn	16.84
City_Tier	0.00
Complain_ly	0.00
Day_Since_CC_connect	1.15
Gender	0.00
Login_device	0.00
Marital_Status	0.00
Payment	0.00
Service_Score	0.12
Tenure	1.23
account_segment	0.00
cashback	8.76
coupon_used_for_payment	12.26
rev_growth_yoy	0.00
rev_per_month	1.64

We have used the inter quartile range method as it will provide us the best results. For calculating IQR, we need the 75th percentile and 25th percentile, where IQR is the difference between the 75th and 25th Quartile. Then we treat the upper limit and lower limit by multiplying IQR with 1.5 and then subtracting the value from Q1 for lower limit and add for upper limit.

We have treated the outliers for few variables instead of all based on the above plot. We have treated the outliers for "Tenure", "CC_Contacted_LY", "rev_per_month", "Day_Since_CC_connect", "cashback", and "coupon_used_for_payment". We have not treated the outliers for categorical variables such as churn, account_user_count, etc.





ENCODING

We have used onehot encoding for nominal categorical variables as there is no order to the sub-categories which include payment, gender, marital status, and login device variables. However, we have manually replaced the ordinal categorical variables with numbers as they follow a specific ranking order.

We can understand from the city tier variable that tier 1 has the highest ranking followed by tier 2 and tier 3. As a result, we have given higher value to city tier 1 which is 3, and so on for other variables.

In the same way, we know that in the account segment variable, the ranking order from high to low is HNI (High Networth Individual), Super Plus, Super, Regular Plus, and Regular. As a result, have assigned the values to these sub-segments accordingly.

SCALING

We have used MinMax Scaler to bring all the variables at the level and so we can build models for accurate prediction.

We have used MinMax scaler as it transforms data by scaling features to a given range. It scales the values to a specific value range without changing the shape of the original distribution.

Chapter 3. Modelling Approach

3.1 Introduction

According to several news articles and industry journals, the customer churning rate in India is between 15% to 20%. As a result, statistically the data is imbalanced and the percentage of churned customers in the dataset Is approx. 16%. So, we don't need to balance the data using any tools such as SMOTE, etc.

In this chapter, we will move towards various model building after EDA and data cleaning performed earlier followed by model tuning and assessing the performance over different metrics Accuracy, F1 Score, Recall, Precession, ROC curve, AUC score, Confusion matrix and classification report. We will choose the model which does not underfit or overfit along with the best accuracy in place.

3.2 Splitting Data into Train and Test Dataset

Following the accepted market practice, we have divided data into Train and Test dataset into 70:30 ratio and building various models on training dataset and testing for accuracy over testing dataset.

3.3 Model Building & Approach

In this business case, the need is to predict whether a given customer would churn or not. This is a binary classification problem with only two prediction outcomes '0' means will not churn and '1' means will churn. Since there is a target variable 'Churn' to be predicted, this is a supervised learning problem. We have used following model building approach:

- Splitting Data into Train and Test (70:30 respectively)
- Performing Variance Inflation Factor (VIF) Analysis on Logistic Regression Model
- Analyzed the performance on Train & Test Dataset for Base Models for all the 8 alogorithms used (mentioned one-by-one below)
- Tuning the Models with Several Combinations of Hyperparameters ***(Please note that models mentioned other than base models in the algorithms are hyptertuned)***
- Selecting Best Model based on Evaluation Metrics

3.3.1 Logistic Regression & Linear Discriminant Analysis (LDA)

We have conducted **VIF analysis for Logistic Regression** after making the base model before doing the hypertuning to check the mode performances. We have also built stats model, which has turned out to be the best model with highest recall.

LOGISTIC REGRESSION

Base Model	Train		Test	
	0	1	0	1
Precision	0.89	0.74	0.89	0.75
Recall	0.97	0.43	0.97	0.43
f1 Score	0.93	0.54	0.93	0.54
Accuracy	0.88		0.88	
Model 1	**Train**		**Test**	
	0	1	0	1
Precision	0.89	0.73	0.90	0.74
Recall	0.97	0.44	0.97	0.44
f1 Score	0.93	0.55	0.93	0.55
Accuracy	0.88		0.88	
Model 2	**Train**		**Test**	
	0	1	0	1
Precision	0.89	0.73	0.90	0.74
Recall	0.97	0.44	0.97	0.44
f1 Score	0.93	0.55	0.93	0.55
Accuracy	0.88		0.88	
Model 3	**Train**		**Test**	
	0	1	0	1
Precision	0.89	0.74	0.89	0.74
Recall	0.97	0.44	0.97	0.44
f1 Score	0.93	0.55	0.93	0.55
Accuracy	0.88		0.88	
Model 4	**Train**		**Test**	
	0	1	0	1
Precision	0.90	0.73	0.89	0.74
Recall	0.97	0.44	0.97	0.44
f1 Score	0.93	0.55	0.93	0.55
Accuracy	0.88		0.88	
Model 5 (Stats Model)	**Train**		**Test**	
	0	1	0	1
Precision	0.89	0.74	0.89	0.74
Recall	0.97	0.44	0.97	0.44
f1 Score	0.93	0.55	0.93	0.55
Accuracy	0.88		0.88	
Model 6 (Stats Model)	**Train**		**Test**	
	0	1	0	1
Precision	0.91	0.66	0.92	0.68
Recall	0.94	0.56	0.95	0.58
f1 Score	0.93	0.61	0.93	0.63
Accuracy	0.88		0.88	

LDA

Base Model	Train		Test	
	0	1	0	1
Precision	0.89	0.72	0.89	0.75
Recall	0.97	0.41	0.97	0.42
f1 Score	0.93	0.52	0.93	0.54
Accuracy	0.87		0.88	
Model 1	**Train**		**Test**	
	0	1	0	1
Precision	0.89	0.73	0.90	0.76
Recall	0.97	0.41	0.97	0.42
f1 Score	0.93	0.52	0.93	0.54
Accuracy	0.87		0.88	
Model 2	**Train**		**Test**	
	0	1	0	1
Precision	0.89	0.73	0.89	0.76
Recall	0.97	0.41	0.97	0.42
f1 Score	0.93	0.52	0.93	0.54
Accuracy	0.87		0.88	
Model 3	**Train**		**Test**	
	0	1	0	1
Precision	0.89	0.73	0.89	0.76
Recall	0.97	0.41	0.97	0.42
f1 Score	0.93	0.52	0.93	0.54
Accuracy	0.87		0.88	

Models highlighted in red are the best models across the algorithms

LOGISTIC REGRESSION: INTERPRETATION

The following variables have a positive correlation with Churn, i.e, as the variable increases, churn increases: City tier, User count, Customer care score, Revenue per month, Complaint last year, Regular customer segment and Single marital status.

The following variables have a negative correlation with Churn, i.e, as the variable increases, churn decreases: Tenure, Credit/debit/UPI payment, Super segment, Mobile login device.

From the above hypertuned model, we can conclude that the data is neither “Overfit” nor “Underfit” in nature. And we can also inference that the model built using Stats Model is best optimized considering the best parameters obtained. However, the accuracy scores along with recall, precision, F1 values, ROC curve and AUC score are not that significant. However, as this is the best model, we will compare it with the best models from other model types.

LDA: INTERPRETATION

Using GridSearchCV function we tried finding the best parameters to further tune-in the above model for better accuracy and we have found that shrinkage as “auto”, solver as “lsqr” and tol value of “0.001” gives the best model considering accuracy, precision, recall, F1, ROC curve and AUC score.

3.3.2 Artifical Neural Network (ANN) and K-Nearest Neighbour (KNN)

ANN

Base Model	Train		Test	
	0	1	0	1
Precision	0.99	0.98	0.97	0.93
Recall	1.00	0.95	0.99	0.86
f1 Score	0.99	0.97	0.98	0.90
Accuracy	0.99		0.87	
Model 1	Train		Test	
	0	1	0	1
Precision	1.00	1.00	0.98	0.95
Recall	1.00	0.99	0.99	0.92
f1 Score	1.00	0.99	0.99	0.93
Accuracy	1.00		0.98	
Model 2	Train		Test	
	0	1	0	1
Precision	1.00	1.00	0.98	0.97
Recall	1.00	1.00	0.99	0.91
f1 Score	1.00	1.00	0.99	0.94
Accuracy	1.00		0.98	
Model 3	Train		Test	
	0	1	0	1
Precision	0.99	1.00	0.97	0.95
Recall	1.00	0.95	0.99	0.87
f1 Score	0.99	0.97	0.98	0.91
Accuracy	0.99		0.97	

ARTIFICIAL NEURAL NETWORK: INTERPRETATION

Maximum precision score for 1s. The problem statement states that Revenue assurance team do not want to give unnecessary freebies. If precision for 1s (churned customers) is high, then the actual churned customers out of the model's predicted churned customers would be high and hence the revenue assurance team's criteria would be satisfied.

High F1-score (to ensure recall is also good) for churned customers. High precision should not come at the cost of recall. The model should be able to get a good part of the actual churns as predicted churned customers so that this prediction exercise is meaningful and the DTH provider can actually address their churn problem. Hence combination of precision and recall to get the F1-score is important.

We have received the best model performance with ANN model 1, and we will use it as benchmark for this algorithm and compare with the best models built using other alogorithms.

KNN

Base Model	Train		Test	
	0	1	0	1
Precision	0.97	0.95	0.95	0.88
Recall	0.99	0.87	0.98	0.76
f1 Score	0.98	0.91	0.97	0.81
Accuracy	0.97		0.94	
Model 1	Train		Test	
	0	1	0	1
Precision	1.00	1.00	0.98	0.94
Recall	1.00	1.00	0.99	0.88
f1 Score	1.00	1.00	0.98	0.91
Accuracy	1.00		0.97	

K-NEAREST NEIGHBOUR (KNN): INTERPRETATION

KNN classifier works by looking at K-Nearest Neighbours to the given datapoint. It decides the target value based on its neighbours. KNN works on a principle assuming every data point falling near to each other is falling in the same class. It is also a black box model and lacks interpretability. Since it is non-parametric, it may be computationally expensive and require more memory to store training data. It also has a tendency to overfit. Although this model was tried on the given data and tuned extensively, due to the above said reasons, it has been decided not to select this as best model even if model performance is good.

3.3.3 Random Forest and ADA Booster

RANDOM FOREST

Base Model	Train		Test	
	0	1	0	1
Precision	1.00	1.00	0.97	0.97
Recall	1.00	1.00	1.00	0.83
f1 Score	1.00	1.00	0.98	0.89
Accuracy	1.00		0.97	
Model 1	**Train**		**Test**	
	0	1	0	1
Precision	0.93	0.91	0.92	0.85
Recall	0.99	0.62	0.98	0.60
f1 Score	0.96	0.73	0.95	0.70
Accuracy	0.92		0.92	
Model 2	**Train**		**Test**	
	0	1	0	1
Precision	0.94	0.90	0.93	0.84
Recall	0.99	0.67	0.98	0.63
f1 Score	0.96	0.77	0.95	0.72
Accuracy	0.93		0.92	
Model 3	**Train**		**Test**	
	0	1	0	1
Precision	0.94	0.95	0.94	0.88
Recall	0.98	0.79	0.98	0.71
f1 Score	0.96	0.87	0.96	0.78
Accuracy	0.96		0.93	
Model 4	**Train**		**Test**	
	0	1	0	1
Precision	0.98	0.98	0.96	0.92
Recall	1.00	0.89	0.99	0.78
f1 Score	0.99	0.93	0.97	0.85
Accuracy	0.98		0.95	

Model 5	Train		Test	
	0	1	0	1
Precision	0.99	1.00	0.96	0.95
Recall	1.00	0.95	0.99	0.82
f1 Score	1.00	0.98	0.98	0.88
Accuracy	0.99		0.96	
Model 6	**Train**		**Test**	
	0	1	0	1
Precision	0.99	1.00	0.97	0.95
Recall	1.00	0.96	0.99	0.83
f1 Score	1.00	0.98	0.98	0.88
Accuracy	0.99		0.96	
Model 7	**Train**		**Test**	
	0	1	0	1
Precision	0.96	0.95	0.97	0.95
Recall	0.99	0.79	0.99	0.84
f1 Score	0.98	0.87	0.98	0.89
Accuracy	0.96		0.97	
Model 8	**Train**		**Test**	
	0	1	0	1
Precision	1.00	1.00	0.97	0.95
Recall	1.00	1.00	0.99	0.85
f1 Score	1.00	1.00	0.98	0.90
Accuracy	1.00		0.97	

RANDOM FOREST: INTERPRETATION

The best model for Random Forest model provides high accuracy; however, the train model seems overfitted with respect to recall as the different between the train and test recall score is more than 10%, owing to which, we cannot consider it as the best model. However, we will use it as benchmark to compare it with best models from other algorithms.

ADA BOOSTER: INTERPRETATION

The ADA booster models, base as well as hypertuned one have lower accuracy in terms of performance as compared to random forest and ANN. However, as this the baset ADA booster model, we will use it for comparison with other models.

ADA BOOSTER

Base Model	Train		Test	
	0	1	0	1
Precision	0.92	0.74	0.92	0.76
Recall	0.96	0.57	0.96	0.61
f1 Score	0.94	0.65	0.94	0.68
Accuracy	0.89		0.9	
Model 1	**Train**		**Test**	
	0	1	0	1
Precision	0.92	0.76	0.93	0.76
Recall	0.96	0.59	0.96	0.62
f1 Score	0.94	0.66	0.94	0.68
Accuracy	0.90		0.90	

3.3.4 Gradient Boosting and SVM

GRADIENT BOOSTING

Base Model	Train		Test	
	0	1	0	1
Precision	0.92	0.83	0.92	0.79
Recall	0.98	0.57	0.97	0.57
f1 Score	0.95	0.68	0.94	0.66
Accuracy	0.91		0.90	

Model 1	Train		Test	
	0	1	0	1
Precision	1.00	1.00	0.98	0.98
Recall	1.00	1.00	1.00	0.91
f1 Score	1.00	1.00	0.99	0.94
Accuracy	1.00		0.98	

Model 2	Train		Test	
	0	1	0	1
Precision	1.00	1.00	0.99	0.98
Recall	1.00	1.00	1.00	0.93
f1 Score	1.00	1.00	0.99	0.95
Accuracy	1.00		0.98	

Model 3	Train		Test	
	0	1	0	1
Precision	1.00	1.00	0.99	0.98
Recall	1.00	1.00	1.00	0.93
f1 Score	1.00	1.00	0.99	0.95
Accuracy	1.00		0.99	

Model 4	Train		Test	
	0	1	0	1
Precision	1.00	1.00	0.98	0.97
Recall	1.00	1.00	0.99	0.92
f1 Score	1.00	1.00	0.99	0.95
Accuracy	1.00		0.98	

Model 5	Train		Test	
	0	1	0	1
Precision	1.00	1.00	0.98	0.98
Recall	1.00	1.00	1.00	0.92
f1 Score	1.00	1.00	0.99	0.95
Accuracy	1.00		0.98	

Model 6	Train		Test	
	0	1	0	1
Precision	1.00	1.00	0.98	0.98
Recall	1.00	1.00	1.00	0.92
f1 Score	1.00	1.00	0.99	0.95
Accuracy	1.00		0.98	

Model 7	Train		Test	
	0	1	0	1
Precision	1.00	1.00	0.98	0.98
Recall	1.00	1.00	1.00	0.91
f1 Score	1.00	1.00	0.99	0.94
Accuracy	1.00		0.97	

GRADIENT BOOSTING: INTERPRETATION

Since the tuned model has shown 1s in all the train data performance parameters, it seems to have learnt all the noise in training dataset well. The test data shows a performance that is within 10% of train data performance. The model is robust and the best performance as compared to other algorithms.

SVM: INTERPRETATION

SVM has given a reasonably good performance on train and test datasets. There has been no overfitting or underfitting of the model as the train and test dataset performances have been comparable. The model performance is much better than several other algorithms and model has better precision, recall, f1 score, and accuracy. This model will be compared with other models algorithms.

SUPPORT VECTOR MACHINE (SVM)

Base Model	Train		Test	
	0	1	0	1
Precision	0.92	0.92	0.91	0.89
Recall	0.99	0.58	0.99	0.54
f1 Score	0.95	0.71	0.95	0.68
Accuracy	0.92		0.91	

Model 1	Train		Test	
	0	1	0	1
Precision	1.00	0.99	0.98	0.90
Recall	1.00	1.00	0.98	0.91
f1 Score	1.00	0.99	0.98	0.90
Accuracy	1.00		0.97	

3.4 Model Validation

- All models were trained only on the train dataset. The trained model was used to predict train dataset target variable. The performance metrics such as Accuracy, F1-score, Precision, Recall, Confusion Matrix, ROC curve and AUC was observed and recorded on train dataset.
- The trained model was then used to predict target variable on test dataset. All the above said performance metrics were observed and recorded for the test data performance as well.
- As a feasibility test, we have conducted 5-fold and 10-fold cross validation for the models which gave an output of 1.00 for train dataset and test dataset had less than 10% difference

3.5 Criteria for Best Model Selection

First Criteria:

1 being the minority class or churned customers, we will analyze the precision, recall, and f1 score of the models for the said minority class, which will help us in determining the performance of the model and whether the model is able to identify or correct predict the churned customers with actual churns. As a result, this is the first criteria to identify the best through its performance.

Recall: It is defined as True Positives / (True Positives + False Negatives). Out of all actual churning customers, how many does the model correctly identify as churners. This is very important as the purpose of the project is to identify as many churners as possible in order to give special offers in order to retain them. For the DTH provider, customer acquisition cost is very high and hence retention is of utmost importance. This is the whole reason why the project exists and hence Recall for 1s is also important in this case.

Precision: It is defined as True Positives / (True Positives + False Positives). Precision as the name suggests means how correctly the model predicts the churned customers as compared to the actual churned. The problem statement states that the revenue assurance team is very stringent about providing freebies where it is not required. Translated into metric, this would mean that the precision for 1's/churns should be highest.

Second Criteria:

Accuracy: This is a classification problem and the dataset has class imbalance. That is, the proportion of churn and non-churn customers are not equal. With imbalanced classes, it's easy to get a high accuracy without actually making useful predictions. So, accuracy as an evaluation metric makes sense only if the class labels are uniformly distributed. We are concerned with correct prediction of churn customers (class 1). Hence 'Accuracy' is not a correct metric to compare various models but for the sake of completeness and to ensure that 0s (majority class) are not overlooked, it is still recorded in the comparison matrix.

AUROC: In addition to the above metrics, the Area under curve of ROC curve is also used to evaluate model performance. An ROC curve (or receiver operating characteristic curve) is a plot that summarizes the performance of a binary classification model on the positive class. It is a curve that is constructed by evaluating true positives and false positives for different threshold values. As visualizing ROC curve is difficult for actual comparison, the Area Under Curve (AUC) metric helps with a numeric comparison. The closer the AUC is to 1, the better the model. However, like accuracy this also works well for balanced dataset4. For the sake of completeness, this is also recorded in comparison matrix.

3.6 Model Comparison & Interpretation

We will compare the best models for all the algorithms and analyze their performances:

Table 3 Model Comparison

Logistic Regression	Train		Test	
	0	1	0	1
Precision	0.91	0.66	0.92	0.68
Recall	0.94	0.56	0.95	0.58
f1 Score	0.93	0.61	0.93	0.63
Accuracy	0.88		0.88	
Linear Discriminant Analysis (LDA)	**Train**		**Test**	
	0	1	0	1
Precision	0.89	0.73	0.90	0.76
Recall	0.97	0.41	0.97	0.42
f1 Score	0.93	0.52	0.93	0.54
Accuracy	0.87		0.88	
Artificial Neural Network (ANN)	**Train**		**Test**	
	0	1	0	1
Precision	1.00	1.00	0.98	0.97
Recall	1.00	1.00	0.99	0.91
f1 Score	1.00	1.00	0.99	0.94
Accuracy	1.00		0.98	
K-Nearest Neighbour (KNN)	**Train**		**Test**	
	0	1	0	1
Precision	1.00	1.00	0.98	0.94
Recall	1.00	1.00	0.99	0.88
f1 Score	1.00	1.00	0.98	0.91
Accuracy	1.00		0.97	

Random Forest	Train		Test	
	0	1	0	1
Precision	1.00	1.00	0.97	0.95
Recall	1.00	1.00	0.99	0.85
f1 Score	1.00	1.00	0.98	0.90
Accuracy	1.00		0.97	
ADA Booster	**Train**		**Test**	
	0	1	0	1
Precision	0.92	0.76	0.93	0.76
Recall	0.96	0.59	0.96	0.62
f1 Score	0.94	0.66	0.94	0.68
Accuracy	0.9		0.9	
Gradient Boosting	**Train**		**Test**	
	0	1	0	1
Precision	**1.00**	**1.00**	**0.99**	**0.98**
Recall	**1.00**	**1.00**	**1.00**	**0.93**
f1 Score	**1.00**	**1.00**	**0.99**	**0.95**
Accuracy	**1.00**		**0.99**	
Support Vector Machine (SVM)	**Train**		**Test**	
	0	1	0	1
Precision	1.00	0.99	0.98	0.9
Recall	1.00	1.00	0.98	0.91
f1 Score	1.00	0.99	0.98	0.9
Accuracy	1.00		0.97	

MODEL COMPARISON

When compared to the best models from all the algorithms, Gradient Boosting model has best performance with better precision, recall, and f1 score. The model is cleal not overfitting as the difference between train dataset and test dataset for 1, which is the minority class, is less than 10%. However, we have also performed 5-fold and 10-fold cross validation and idenfied that the model is not overfitting (Kindly refer jupyter notebook 1 of 2 to see the cross-validation scores or appendix below).

3.6.1 Best Model Interpretation: Gradient Boosting

Figure 8. Gradient Boosting: Best Model Confusion Matrix

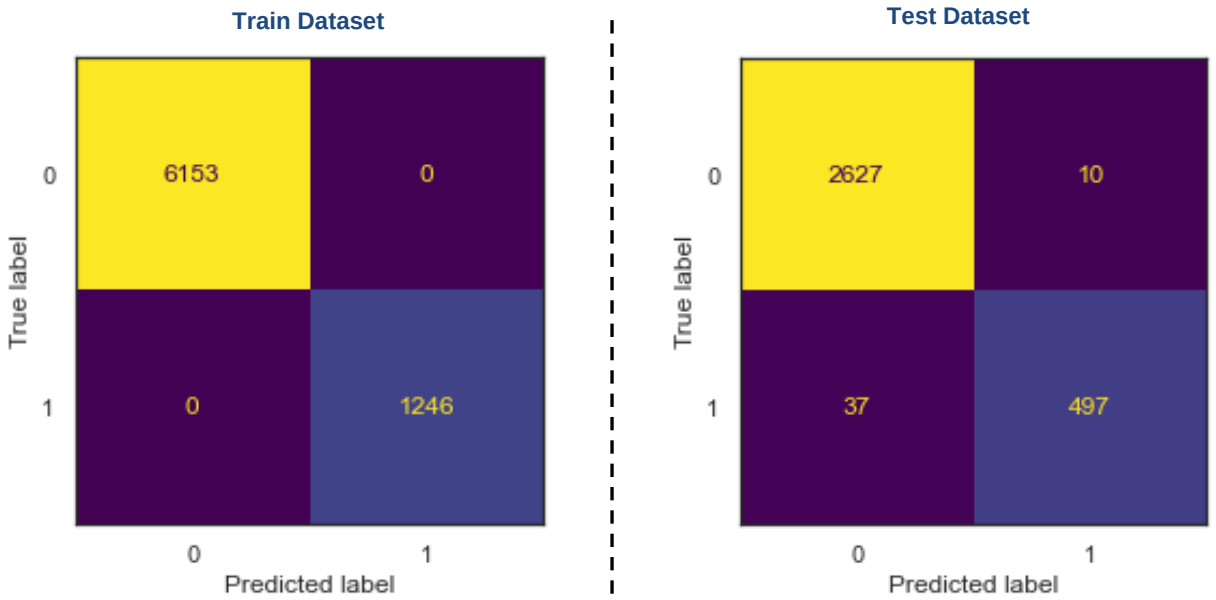


Figure 9. Gradient Boosting: Best Model Classification Report

Train Dataset

	precision	recall	f1-score	support
0	1.00	1.00	1.00	6153
1	1.00	1.00	1.00	1246
accuracy			1.00	7399
macro avg	1.00	1.00	1.00	7399
weighted avg	1.00	1.00	1.00	7399

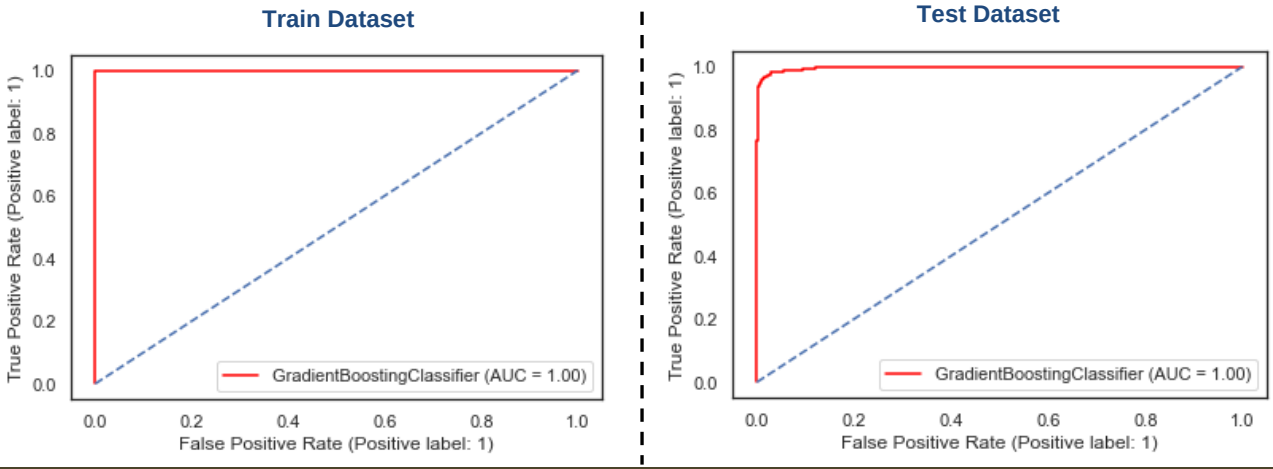
AUC: 1.000

Test Dataset

	precision	recall	f1-score	support
0	0.99	1.00	0.99	2637
1	0.98	0.93	0.95	534
accuracy			0.99	3171
macro avg	0.98	0.96	0.97	3171
weighted avg	0.99	0.99	0.99	3171

AUC: 0.997

Figure 10. Gradient Boosting: Best Model ROC Curves



- A precision of 0.98, or 98%, means that 98 of the 100 consumers the model has predicted will churn will do so, while only two will not. Only 2/100 of these customers would be mistakenly labeled as having churned, therefore any marketing budget set up for a targeted campaign to keep these consumers would be used to the best possible effect.
- If 100 customers actually churn, the model would have correctly classified 93 of them as churners and 7 as non-churners, according to a recall of 0.93 or 93%. This would imply that the campaign would focus on these 93 clients, with the potential to keep them while losing 7 others. Businesses can utilize the aforementioned to project based on the consumer base for which a forecast needs to be made.

3.6.2 Individual Model Interpretation

The model interpretation of the three best models from three different algorithm. We have provided feature importance plot and identified the key common features among all. We have provided plot for following best models:

- ANN
- Random Forest
- Gradient Boosting

Figure 11. Feature Importance: ANN

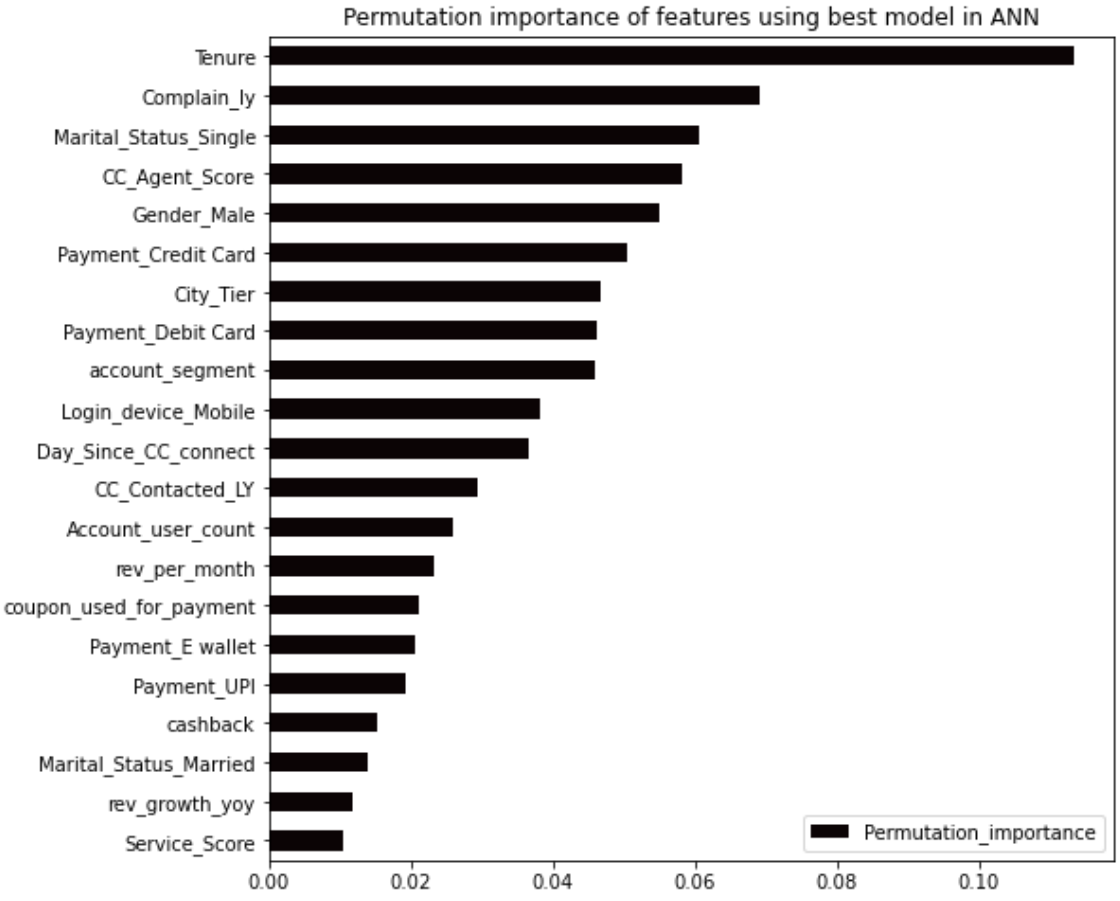


Figure 12. Feature Importance: Random Forest and Gradient Boosting

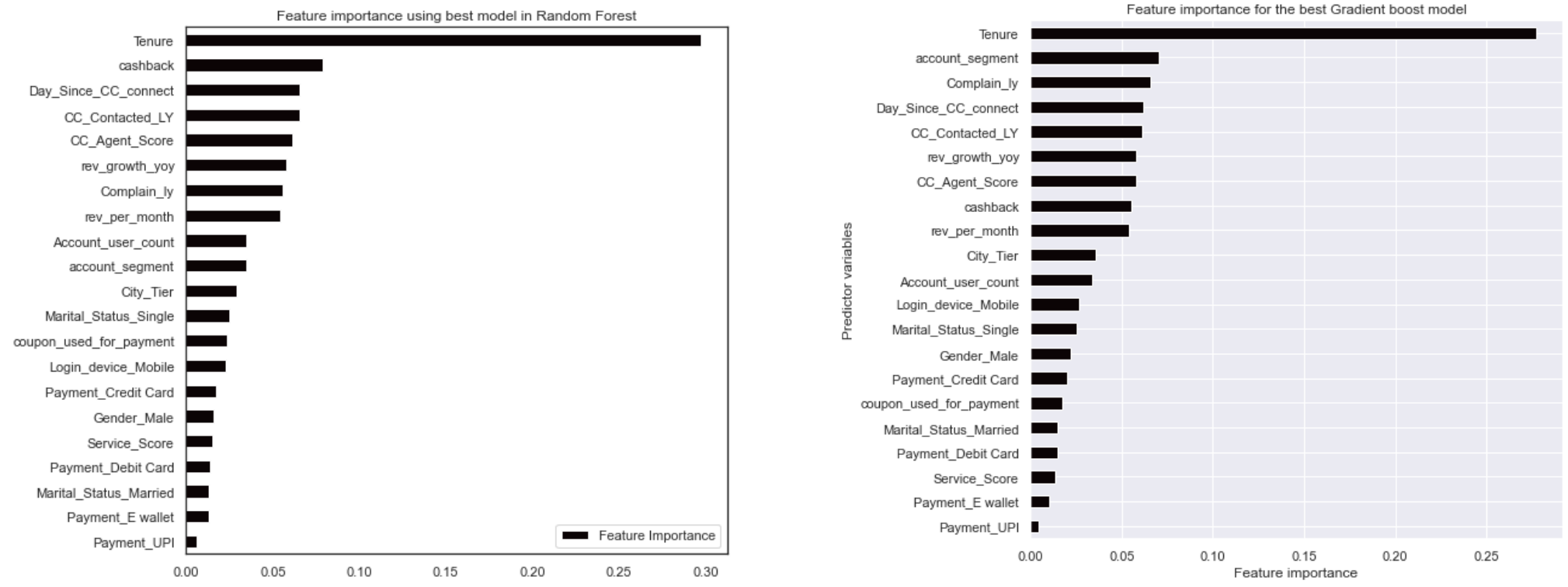


Table 4 Comparison of Feature Importance

Features	ANN	Random Forest	Gradient Boosting
1	Tenure	Tenure	Tenure
2	Complain_ly	cashback	Account_segment
3	Marital_Status_Single	Days_Since_CC_Connect	Complain_ly
4	CC_Agent_Score	CC_Contacted_LY	Days_Since_CC_Connect
5	Gender_Male	CC_Agent_Score	CC_Contacted_LY
6	Payment_Credit_Card	rev_growth_yoy	rev_growth_yoy

The top 5 features that have influenced Gradient boost model are Tenure, Days since last customer connect, number of times customer contacted last year, complaint last year and customer care score. Together, they add up to almost 66% of the total feature importance.

Chapter 4. Business Implications

4.1 Low tenure: High churn

Bad initial experience, trial periods, and prepaid accounts that automatically expire after a set amount of time have passed without a top-up. It's crucial to choose between the two reasons mentioned above. High customer service calls, recorded complaints, and low cashback and incentives for short-tenured consumers all point to the first cause.

Recommendations:

- An activation team can be formed and this team could extend support beyond the initial setup until customers settle down with the service. The team can engage with customers for the first 2-3 months.
- The DTH company should come up with a solution that will engage customers for a longer tenure with gift cards & coupons, at the same time making sure that the company doesn't incur any losses

4.2 Provide offers for retention

The company should provide offers such as cashback coupons and if the customers are ready to pay for the complete year, then they will only have to pay for 11 months such schemes can be used as the current retention programs do not seem to be focusing on the customers with higher risk of churn.

4.3 Higher times contact to customer care: High churn

Based on the EDA and model building Days_Since_CC_Connect is a highly crucial variable for churning of the customers. The customers who have contacted the customer care more often have churned at a higher rate. As a result, a matrix should be made as to what were the majority of the customers complaining about and why they contacted customer care.

4.4 Emergence of OTT platform

DTH  **OTT**

The interactive TV service is a current trend emerging in Direct-To-Home (DTH) services.

The interactive services can be anything that can be accessed for movie-on-demand, video conferencing, e-mail or any other similar activity.

As a result, aggregation with OTT platforms is the major factor to improve the overall revenue of the company and it will aid in acquiring and retaining more customers.

Offers like subscription to 12 months of DTH services will provide a subscription of 12 months of popular OTT platforms will attract customers.

4.5 Other business recommendations

- The company can train their employees and agents to improve the service score and agent score given during the customer care conversations.
- The company can provide some kind of reward points system to encourage customer payments/transactions through e-wallet facility offered by the company
- If the user count of 1 and 2 can be converted into 4 or 5, it will easier for the company to improve gains as even if 1 or 2 customers leave post increasing the user count, these accounts with low number users won't be dead accounts before recovering the acquisition cost
- We have the gender variable which represents the data filled during the opening of the account. However, if the company can find out or determine the ratio based on the user count it might help them introduce channels for males or females more accurately
- We can also see several bad entries or null entries in the data, if the company can train the employees and standardize the account filling process during account opening, it would be help to analyze the data better.
- Analyze customer feedback. Perform Sentiment analysis of the feedback (if any) that went along with scores. Identify top reasons that have resulted in low scores; if subjective feedback not captured, capture that as well.
- The customers with higher average revenue per month have higher market share in terms of churn. The DTH company needs to look into, why the high spenders are churning and what can be done to stop it.
- As we are tracking the revenue, timeline of last 12 months, and frequency of calls and scores, we can also perform RFM analysis, which might help us get some more insights pertaining to the monetary benefits that can be extracted from the customers

Chapter 5. Appendix

5.1 EDA

Figure 13. Count: Payment

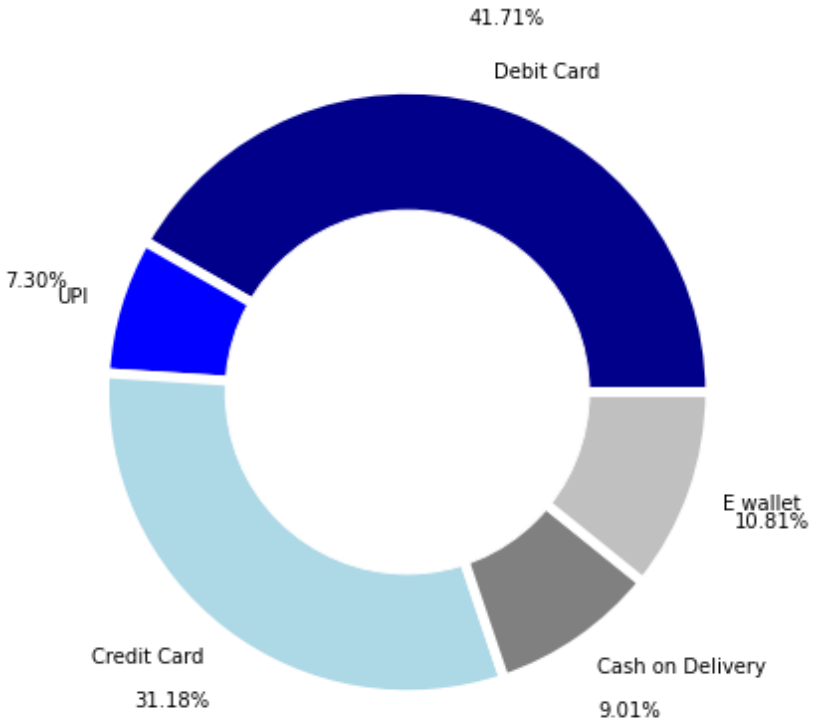


Figure 14. Count: Gender

Figure 15. Count: Account Segments

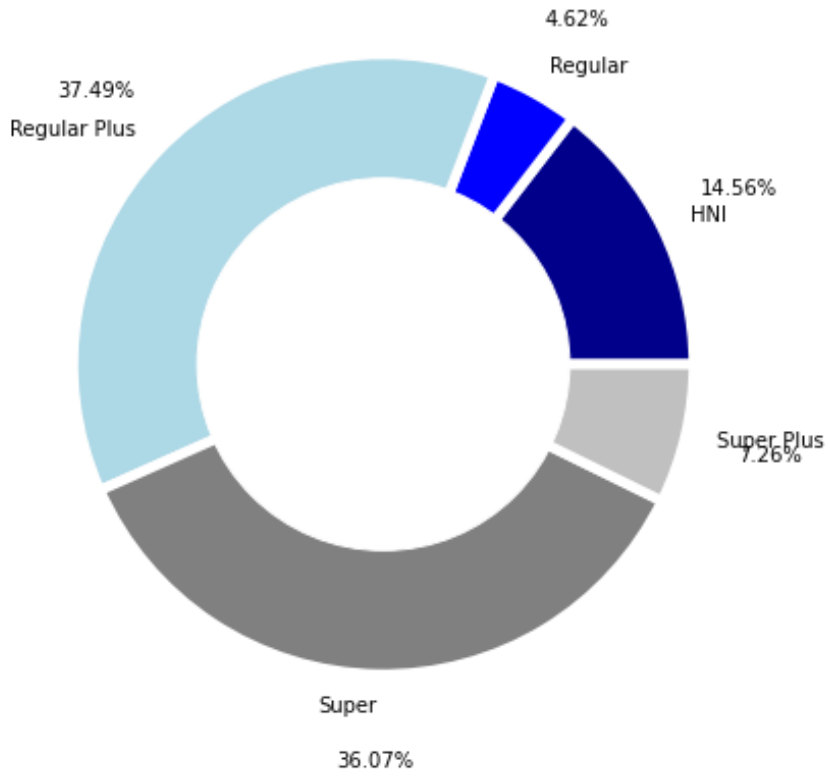


Figure 16. Count: Marital Status

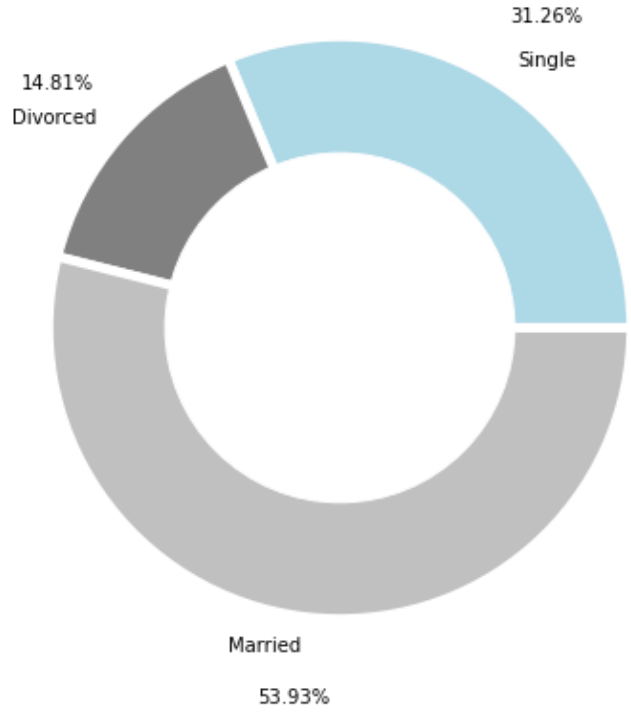
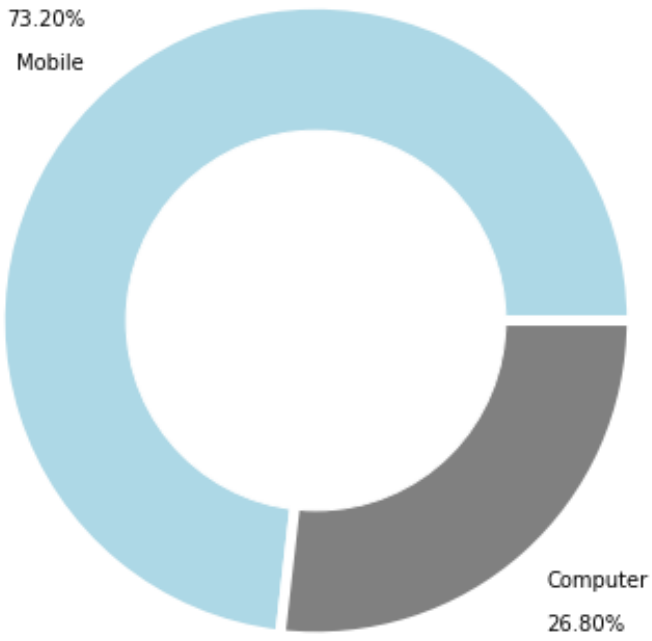


Figure 17. Count: Login Devices



5.1.1 Encoding

Table 5 Glimpse of Encoded Data

	Churn	Tenure	City_Tier	CC_Contacted_LY	Service_Score	Account_user_count	account_segment	CC_Agent_Score	rev_per_month	Complain_ly	...
0	1	0.11	1.0	0.05	0.6	0.4	0.50	0.25	0.67	1.0	...
1	1	0.00	0.0	0.11	0.6	0.6	0.25	0.50	0.50	1.0	...
2	1	0.00	0.0	0.70	0.4	0.6	0.25	0.50	0.42	1.0	...
3	1	0.00	1.0	0.30	0.4	0.6	0.50	1.00	0.58	0.0	...
4	1	0.00	0.0	0.22	0.4	0.4	0.25	1.00	0.17	0.0	...

5 rows × 22 columns

5.1.1 Scaling

Table 6 Glimpse of Scaled Data

	Churn	Tenure	City_Tier	CC_Contacted_LY	Service_Score	Account_user_count	account_segment	CC_Agent_Score	rev_per_month	Complain_ly	...
0	1	0.11	1.0	0.05	0.6	0.4	0.50	0.25	0.67	1.0	...
1	1	0.00	0.0	0.11	0.6	0.6	0.25	0.50	0.50	1.0	...
2	1	0.00	0.0	0.70	0.4	0.6	0.25	0.50	0.42	1.0	...
3	1	0.00	1.0	0.30	0.4	0.6	0.50	1.00	0.58	0.0	...
4	1	0.00	0.0	0.22	0.4	0.4	0.25	1.00	0.17	0.0	...

5 rows × 22 columns

5.2 Modelling

5.2.1 Train and Test dataset

Figure 18. Shape: Train and Test dataset

```
X_train (7399, 21)
X_test (3171, 21)
y_train (7399,)
y_test (3171,)
```

5.2.2 Logistic Regression

5.2.2.1 Base Model

Accuracy of the training dataset: 0.8786322476010272

Accuracy of the testing dataset: 0.8789025543992431

Figure 19. Logistic Regression: Base Model Confusion Matrix

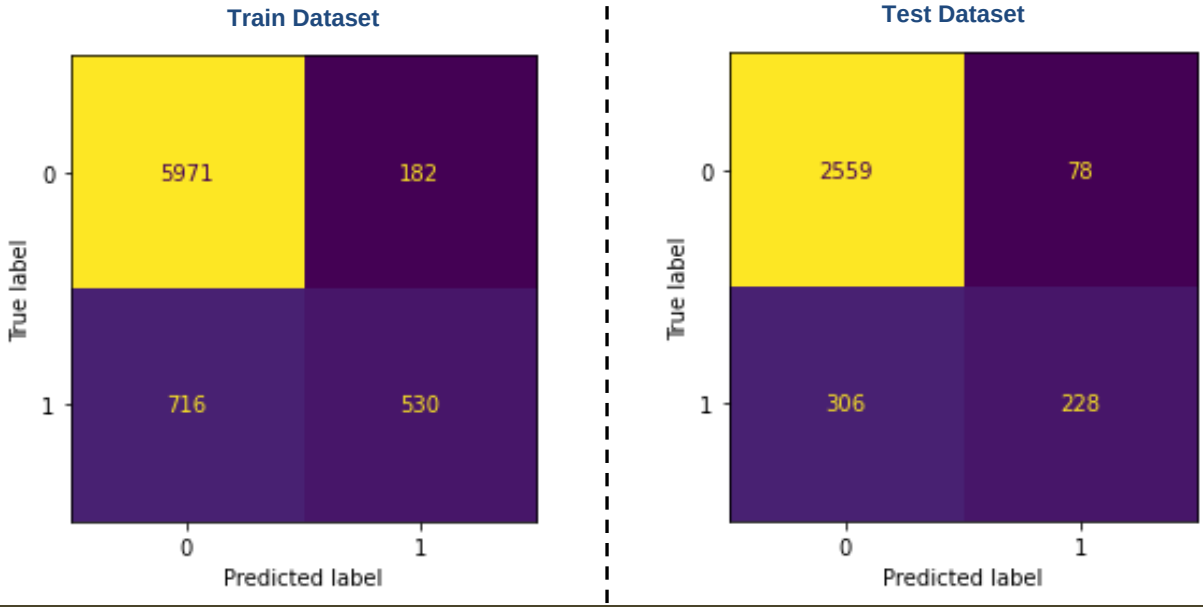


Figure 20. Logistic Regression: Base Model Classification Report

Train Dataset

	precision	recall	f1-score	support
0	0.89	0.97	0.93	6153
1	0.74	0.43	0.54	1246
accuracy			0.88	7399
macro avg	0.82	0.70	0.74	7399
weighted avg	0.87	0.88	0.86	7399

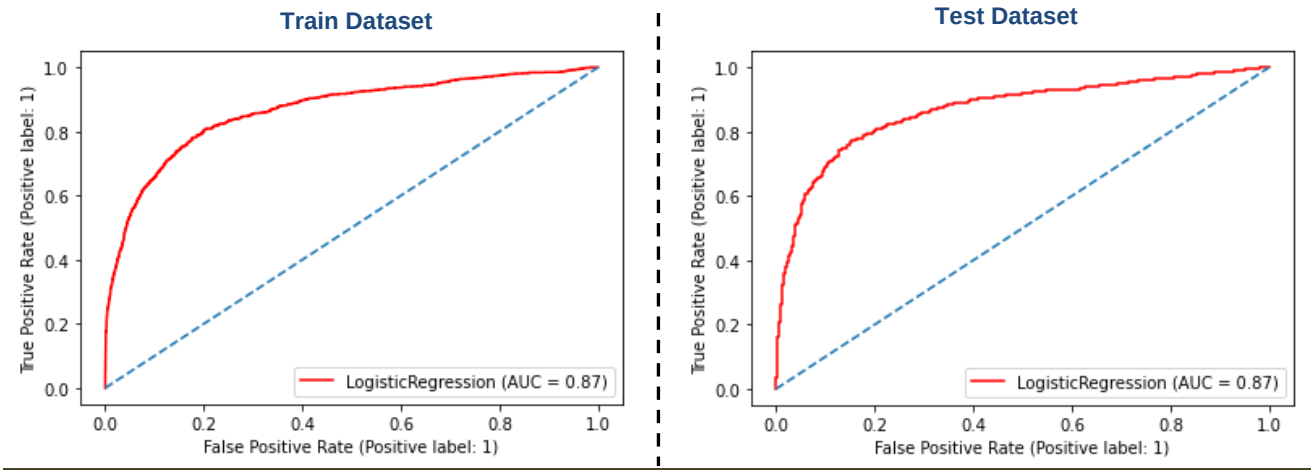
AUC: 0.866

Test Dataset

	precision	recall	f1-score	support
0	0.89	0.97	0.93	2637
1	0.75	0.43	0.54	534
accuracy			0.88	3171
macro avg	0.82	0.70	0.74	3171
weighted avg	0.87	0.88	0.86	3171

AUC: 0.868

Figure 21. Logistic Regression: Base Model ROC Curves



5.2.2.2 Best Model – Stats Model (Model 6)

We have considered multiple hyperparameters to hypertune the model to improve precision, recall, and accuracy. Below are the accuracy scores obtained from this hypertuned model:

Figure 22. Logistic Regression: Best Model Confusion Matrix

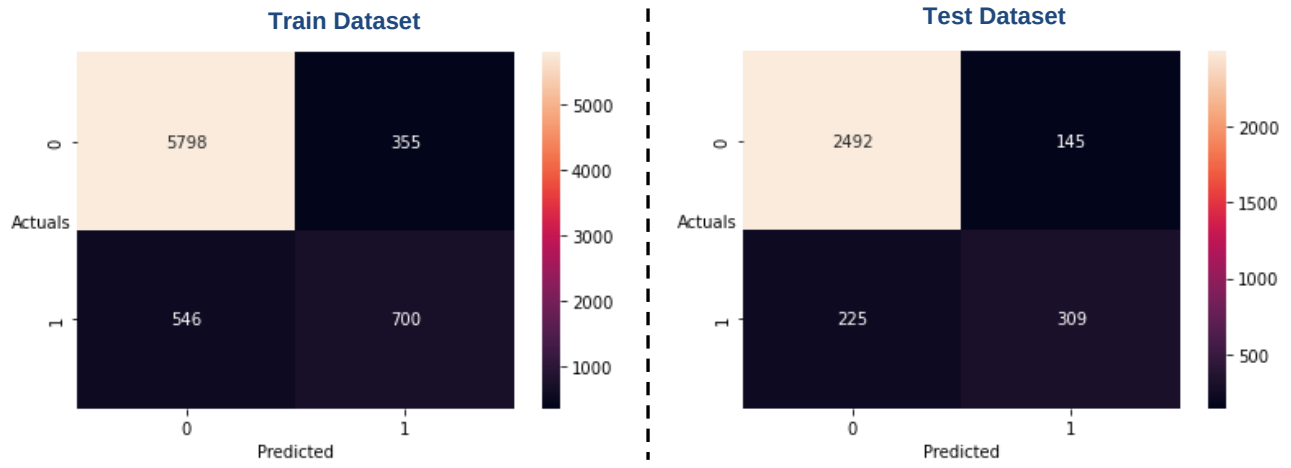


Figure 23. Logistic Regression: Best Model Classification Report

Train Dataset

	precision	recall	f1-score	support
0	0.91	0.94	0.93	6153
1	0.66	0.56	0.61	1246
accuracy			0.88	7399
macro avg	0.79	0.75	0.77	7399
weighted avg	0.87	0.88	0.87	7399

Test Dataset

	precision	recall	f1-score	support
0	0.92	0.95	0.93	2637
1	0.68	0.58	0.63	534
accuracy			0.88	3171
macro avg	0.80	0.76	0.78	3171
weighted avg	0.88	0.88	0.88	3171

5.2.3 Linear Discriminant Analysis (LDA)

5.2.3.1 Base Model

We fitted linear discriminant analysis model into training dataset and performed prediction on training and testing dataset using the same model. We made the first model with default hyperparameters and below are the accuracy scores obtained from this base model:

Accuracy of the training dataset: 0.8734964184349236

Accuracy of the testing dataset: 0.8795332702617471

Figure 24. Linear Discriminant Analysis (LDA): Base Model Confusion Matrix

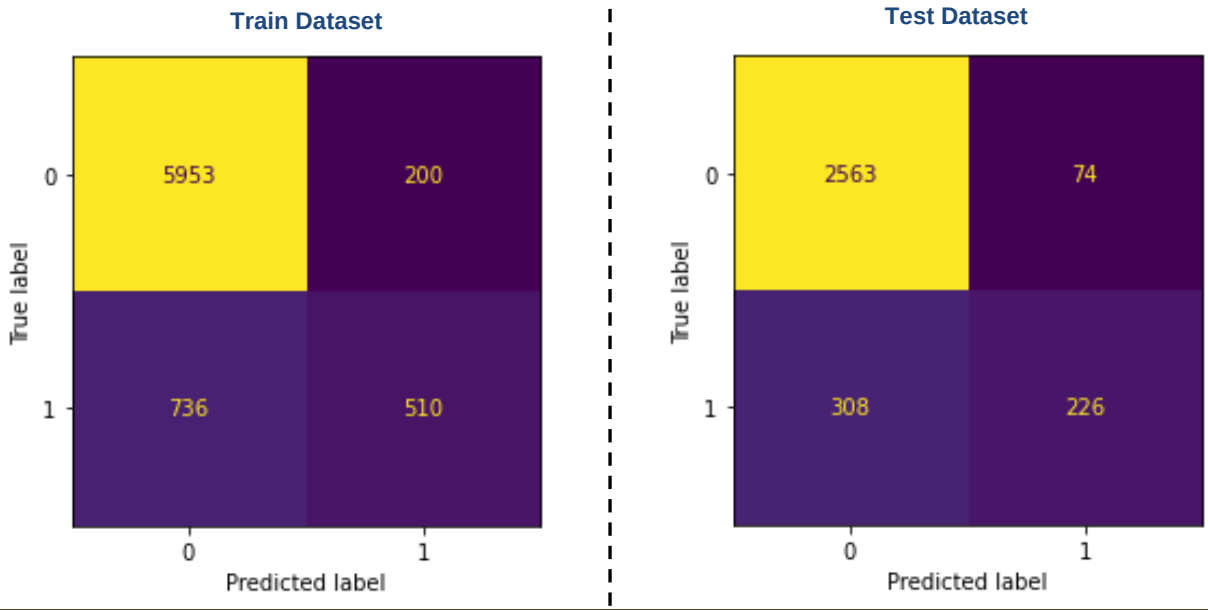


Figure 25. Linear Discriminant Analysis (LDA): Base Model Classification Report

Train Dataset

	precision	recall	f1-score	support
0	0.89	0.97	0.93	6153
1	0.72	0.41	0.52	1246
accuracy			0.87	7399
macro avg	0.80	0.69	0.72	7399
weighted avg	0.86	0.87	0.86	7399

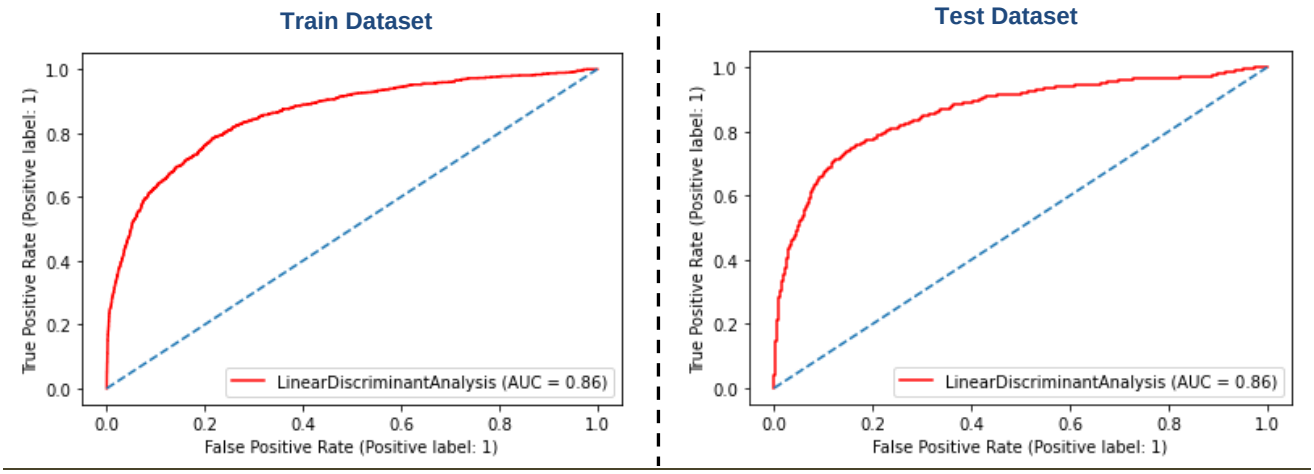
AUC: 0.857

Test Dataset

	precision	recall	f1-score	support
0	0.89	0.97	0.93	2637
1	0.75	0.42	0.54	534
accuracy			0.88	3171
macro avg	0.82	0.70	0.74	3171
weighted avg	0.87	0.88	0.87	3171

AUC: 0.861

Figure 26. Linear Discriminant Analysis (LDA): Base Model ROC Curves



5.2.3.2 Hypertuned Model

We have considered multiple hyperparameters to hypertune the model to improve precision, recall, and accuracy. Below are the accuracy scores obtained from this hypertuned model:

Accuracy of the training dataset: 0.8743073388295716

Accuracy of the testing dataset: 0.879848628192999

Figure 27. Linear Discriminant Analysis (LDA): Hypertuned Model Confusion Matrix

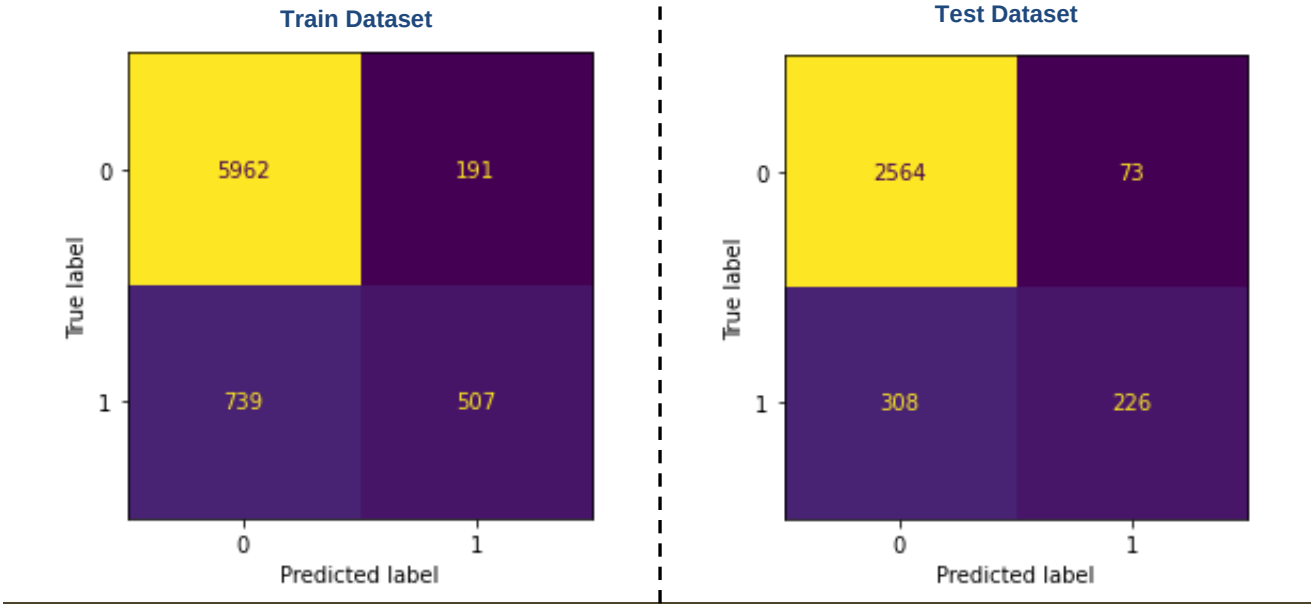


Figure 28. Linear Discriminant Analysis (LDA): Hypertuned Model Classification Report

Train Dataset

	precision	recall	f1-score	support
0	0.89	0.97	0.93	6153
1	0.73	0.41	0.52	1246
accuracy			0.87	7399
macro avg	0.81	0.69	0.72	7399
weighted avg	0.86	0.87	0.86	7399

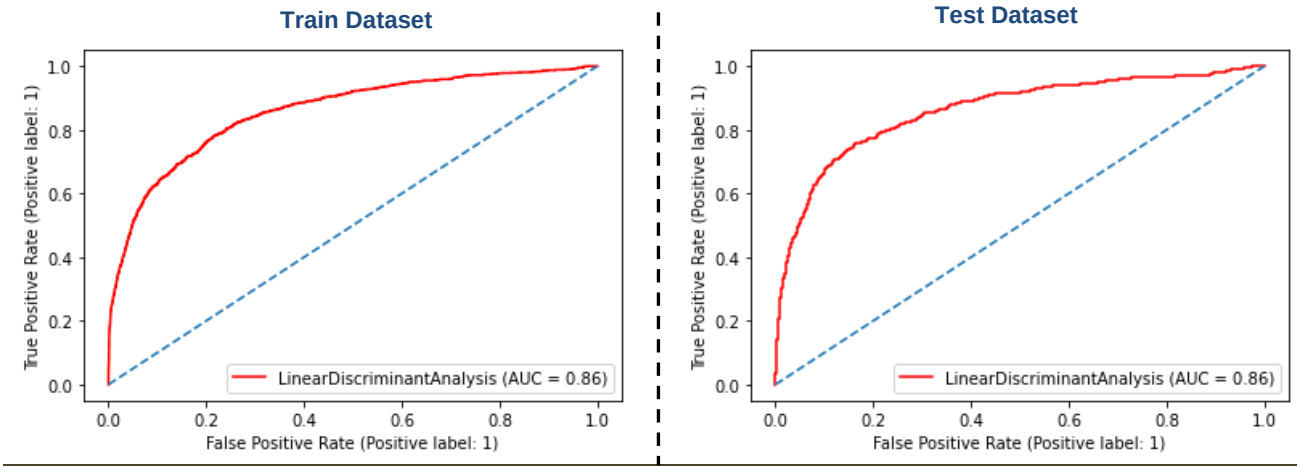
AUC: 0.857

Test Dataset

	precision	recall	f1-score	support
0	0.89	0.97	0.93	2637
1	0.76	0.42	0.54	534
accuracy			0.88	3171
macro avg	0.82	0.70	0.74	3171
weighted avg	0.87	0.88	0.87	3171

AUC: 0.861

Figure 29. Linear Discriminant Analysis (LDA): Hypertuned Model ROC Curves



5.2.4 Artificial Neural Network (ANN)

5.2.4.1 Base Model

We fitted ANN model into training dataset and performed prediction on training and testing dataset using the same model. We made the first model with default hyperparameters and below are the accuracy scores obtained from this base model:

Accuracy of the training dataset: 0.989458034869577

Accuracy of the testing dataset: 0.9662567013560391

Figure 30. Artificial Neural Network (ANN): Base Model Confusion Matrix

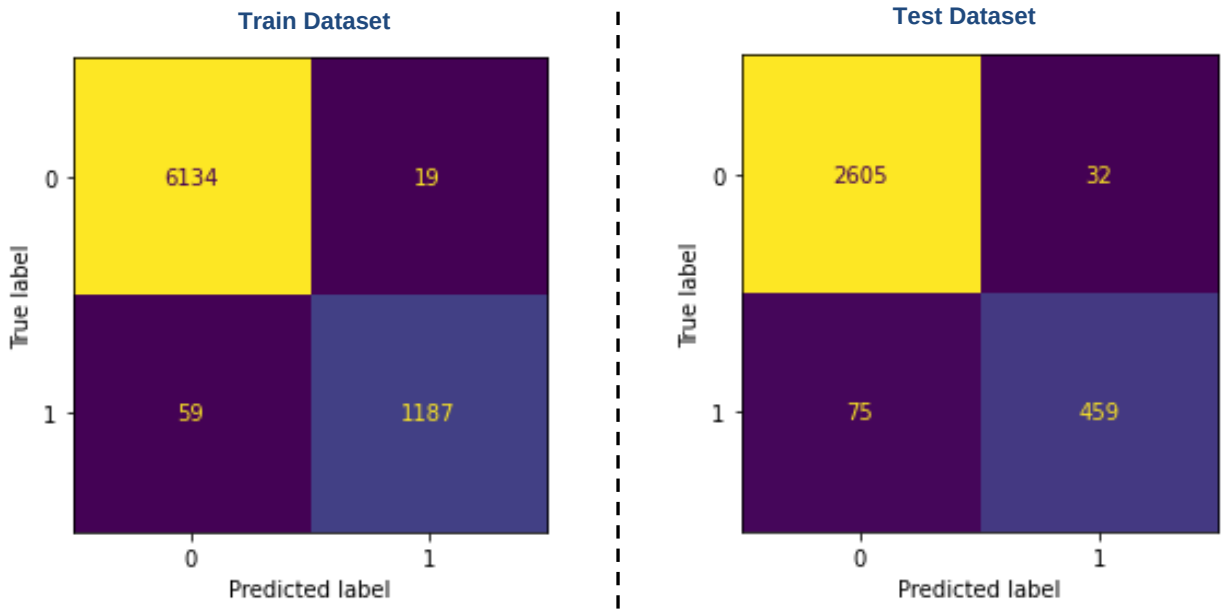


Figure 31. Artificial Neural Network (ANN): Base Model Classification Report

Train Dataset

	precision	recall	f1-score	support
0	0.99	1.00	0.99	6153
1	0.98	0.95	0.97	1246
accuracy			0.99	7399
macro avg	0.99	0.97	0.98	7399
weighted avg	0.99	0.99	0.99	7399

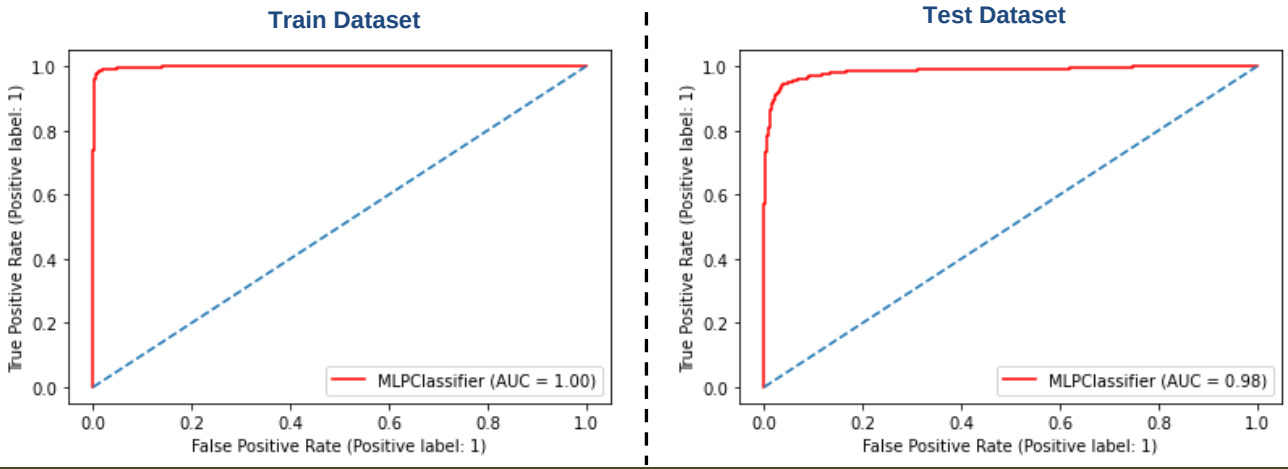
AUC: 0.999

Test Dataset

	precision	recall	f1-score	support
0	0.97	0.99	0.98	2637
1	0.93	0.86	0.90	534
accuracy			0.97	3171
macro avg	0.95	0.92	0.94	3171
weighted avg	0.97	0.97	0.97	3171

AUC: 0.985

Figure 32. Artificial Neural Network (ANN): Base Model ROC Curves



5.2.4.2 Hypertuned Model

ANN_2 model marked in red is the best model as it is providing the best scores for precision, recall, f1 and accuracy on the test dataset for 1, which represents the churned customers in the dataset.

5.2.4.2.1 ANN2: Hyperparameters

```
param_grid = {
    'hidden_layer_sizes': [100, 150],
    'max_iter': [1500],
    'solver': ['sgd','adam','lbfgs'],
    'tol': [0.01,0.001,0.0001],
    'activation': ['identity', 'logistic', 'tanh', 'relu'],
    'alpha': [0.0001,0.001,0.01]
}
```

5.2.4.2.2 ANN3: Hyperparameters

```
param_grid = {
    'hidden_layer_sizes': [90,100,150],
    'max_iter': [2000, 2500],
    'solver': ['sgd','adam','lbfgs'],
    'tol': [0.01,0.001,0.0001],
    'activation': ['identity', 'logistic', 'tanh', 'relu'],
    'alpha': [0.0001,0.001,0.01]
}
```

5.2.4.2.3 ANN4: Hyperparameters

```
param_grid = {
    'hidden_layer_sizes': [80],
    'max_iter': [1000],
    'solver': ['sgd','adam','lbfgs'],
    'tol': [0.01,0.001],
    'activation': ['identity', 'logistic', 'tanh', 'relu'],
    'alpha': [0.001,0.01]
}
```

Figure 33. Artificial Neural Network (ANN): Best Model Confusion Matrix

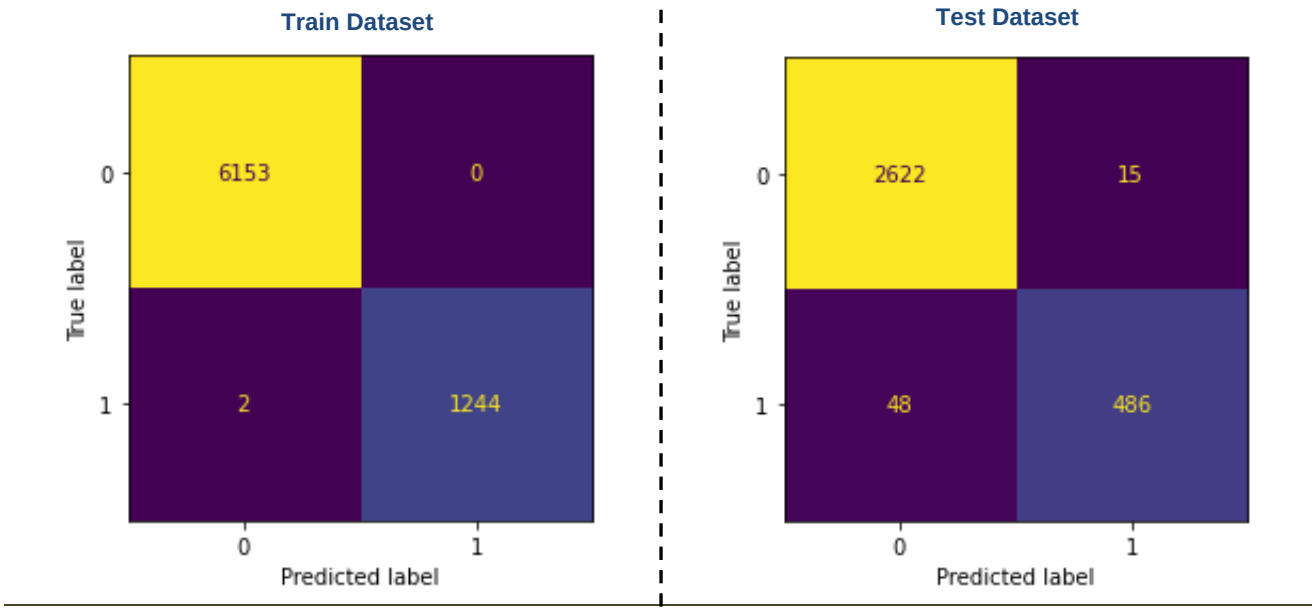


Figure 34. Artificial Neural Network (ANN): Best Model Classification Report

Train Dataset

	precision	recall	f1-score	support
0	1.00	1.00	1.00	6153
1	1.00	1.00	1.00	1246
accuracy			1.00	7399
macro avg	1.00	1.00	1.00	7399
weighted avg	1.00	1.00	1.00	7399

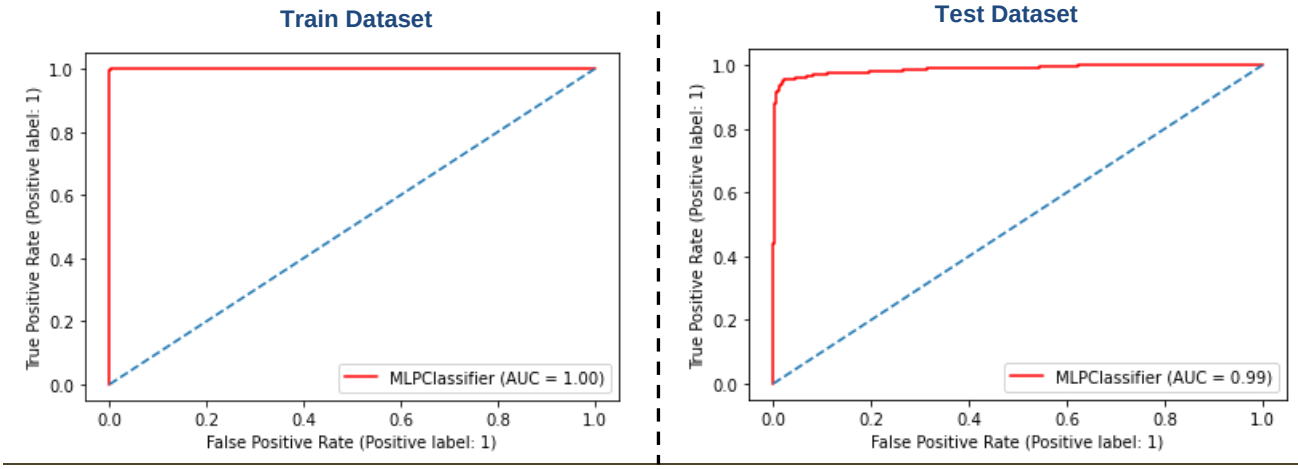
AUC: 1.000

Test Dataset

	precision	recall	f1-score	support
0	0.98	0.99	0.99	2637
1	0.97	0.91	0.94	534
accuracy			0.98	3171
macro avg	0.98	0.95	0.96	3171
weighted avg	0.98	0.98	0.98	3171

AUC: 0.988

Figure 35. Artificial Neural Network (ANN): Best Model ROC Curves



5.2.5 K-Nearest Neighbour (KNN)

5.2.5.1 Base Model

For KNN we have built a model with default parameters and then a hypertuned model with manual entered hyperparameters and tune the model to improve the performance from the base model. We made the first model with default hyperparameters and below are the accuracy scores obtained from this base model:

Accuracy of the training dataset: 0.9716177861873226

Accuracy of the testing dataset: 0.9416587827183853

Figure 36. K-Nearest Neighbour (KNN): Base Model Confusion Matrix

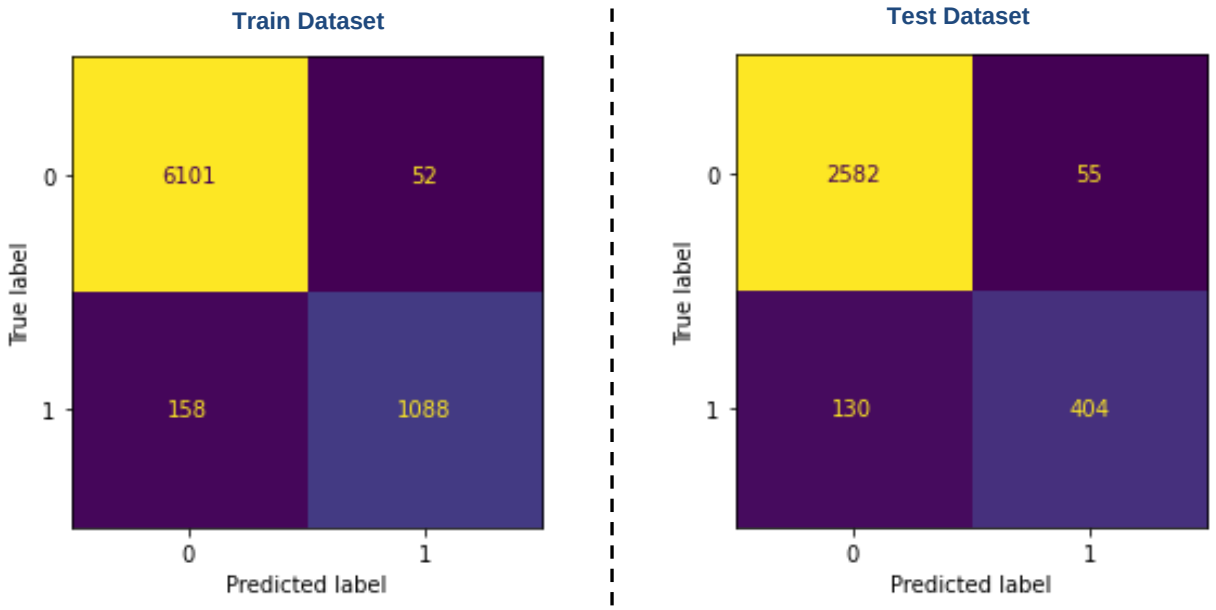


Figure 37. K-Nearest Neighbour (KNN): Base Model Classification Report

Train Dataset

	precision	recall	f1-score	support
0	0.97	0.99	0.98	6153
1	0.95	0.87	0.91	1246
accuracy			0.97	7399
macro avg	0.96	0.93	0.95	7399
weighted avg	0.97	0.97	0.97	7399

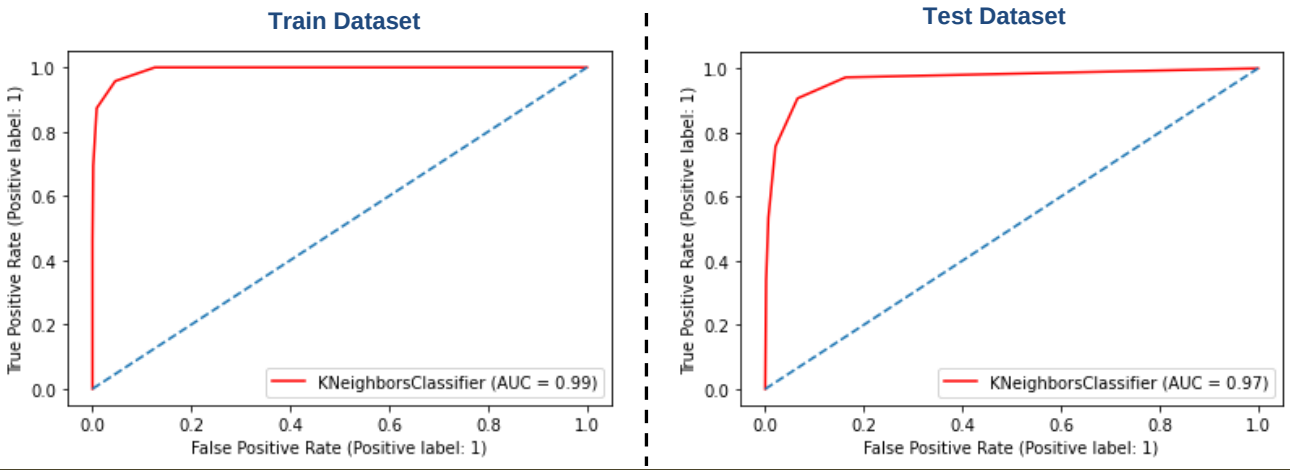
AUC: 0.993

Test Dataset

	precision	recall	f1-score	support
0	0.95	0.98	0.97	2637
1	0.88	0.76	0.81	534
accuracy			0.94	3171
macro avg	0.92	0.87	0.89	3171
weighted avg	0.94	0.94	0.94	3171

AUC: 0.966

Figure 38. K-Nearest Neighbour (KNN): Base Model ROC Curves



5.2.5.2 Hypertuned Model

We have considered multiple hyperparameters to hypertune the model to improve precision, recall, and accuracy. The best hyperparameters output using GridSearch CV were:

'algorithm': 'auto', **'metric'**: 'minkowski', **'p'**: 1, **'weights'**: 'distance'

Below are the accuracy scores obtained from this hypertuned model:

Accuracy of the training dataset: 1.00

Accuracy of the testing dataset: 0.9697256385998108

Figure 39. K-Nearest Neighbour (KNN): Hypertuned Model Confusion Matrix

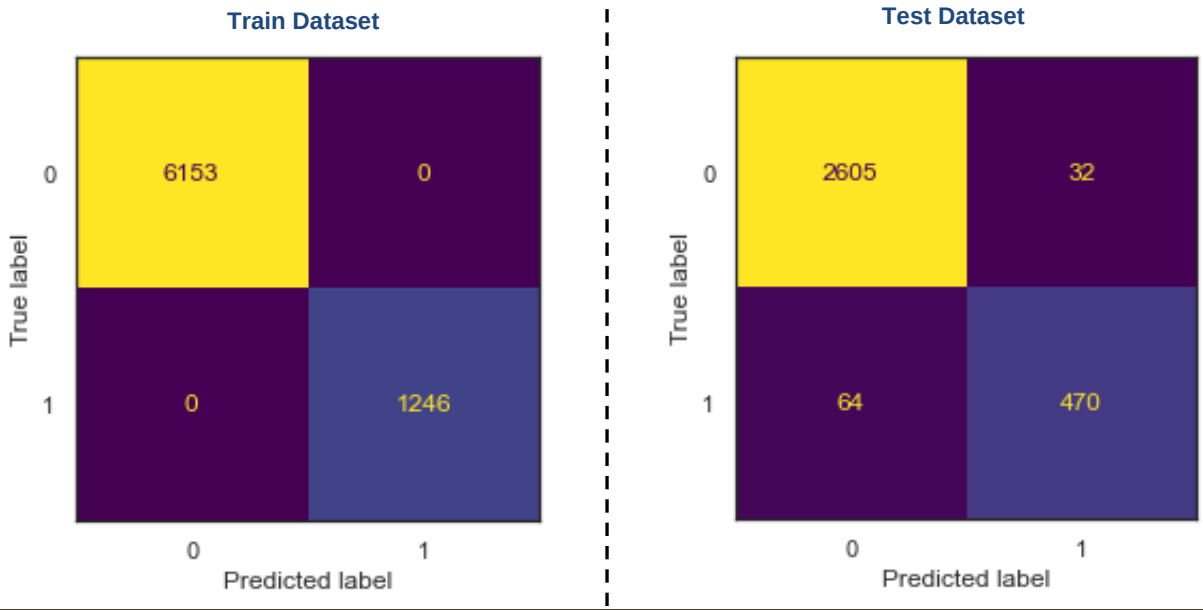


Figure 40. K-Nearest Neighbour (KNN): Hypertuned Model Classification Report

Train Dataset

	precision	recall	f1-score	support
0	1.00	1.00	1.00	6153
1	1.00	1.00	1.00	1246
accuracy			1.00	7399
macro avg	1.00	1.00	1.00	7399
weighted avg	1.00	1.00	1.00	7399

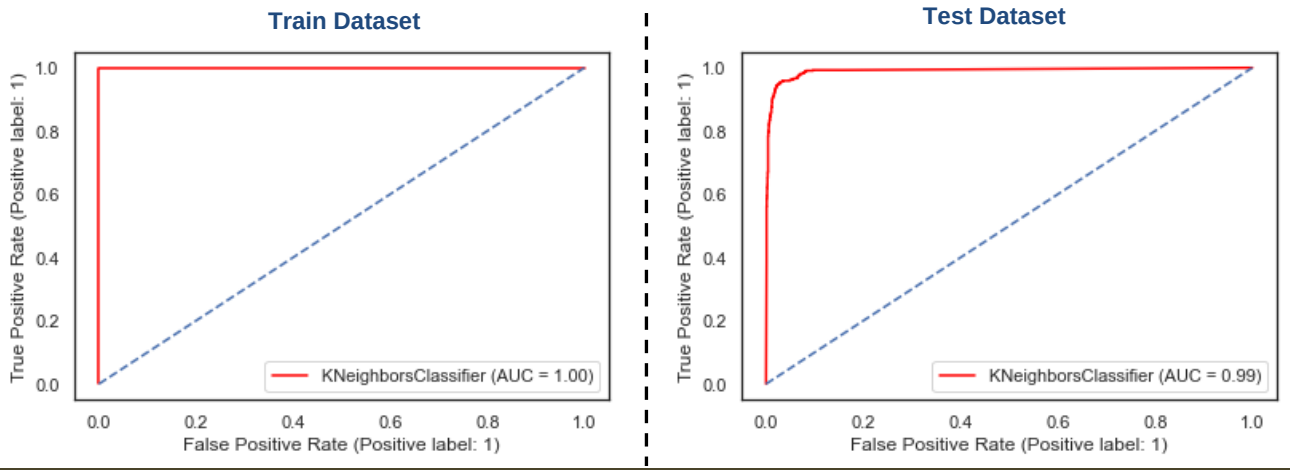
AUC: 1.000

Test Dataset

	precision	recall	f1-score	support
0	0.98	0.99	0.98	2637
1	0.94	0.88	0.91	534
accuracy			0.97	3171
macro avg	0.96	0.93	0.94	3171
weighted avg	0.97	0.97	0.97	3171

AUC: 0.990

Figure 41. K-Nearest Neighbour (KNN): Hypertuned Model ROC Curves



5.2.6 Random Forest

5.2.6.1 Base Model

We fitted Random Forest model into training dataset and performed prediction on training and testing dataset using the same model. We made the first model with default hyperparameters and below are the accuracy scores obtained from this base model:

Accuracy of the training dataset: 1.0

Accuracy of the testing dataset: 0.967202775149795

Figure 42. Random Forest: Base Model Confusion Matrix

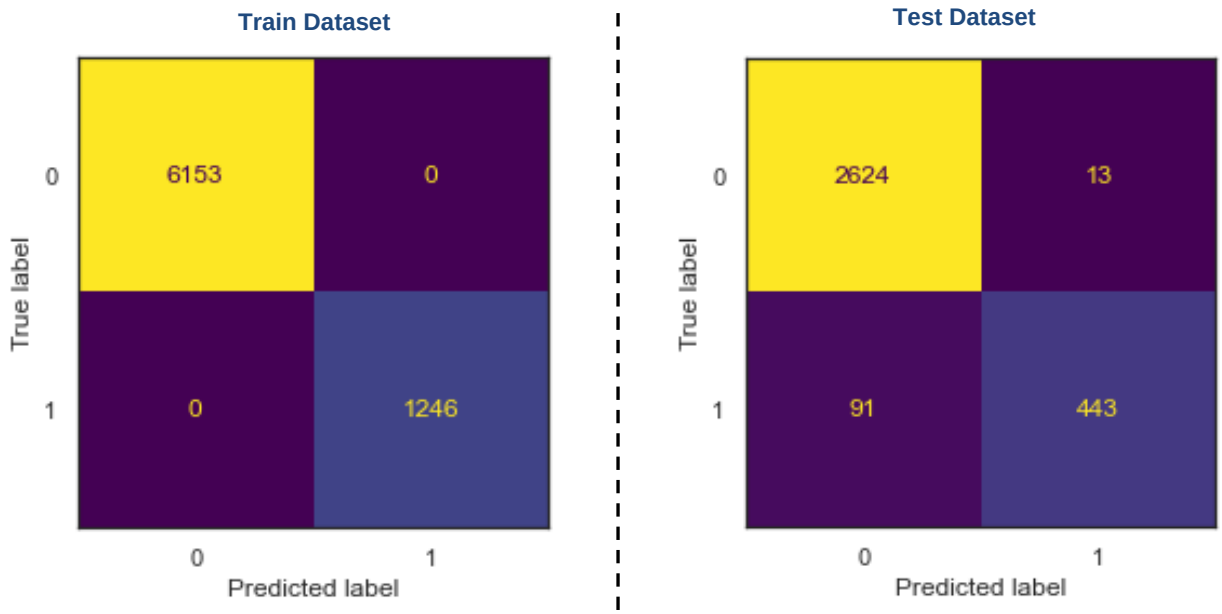


Figure 43. Random Forest: Base Model Classification Report

Train Dataset

	precision	recall	f1-score	support
0	1.00	1.00	1.00	6153
1	1.00	1.00	1.00	1246
accuracy			1.00	7399
macro avg	1.00	1.00	1.00	7399
weighted avg	1.00	1.00	1.00	7399

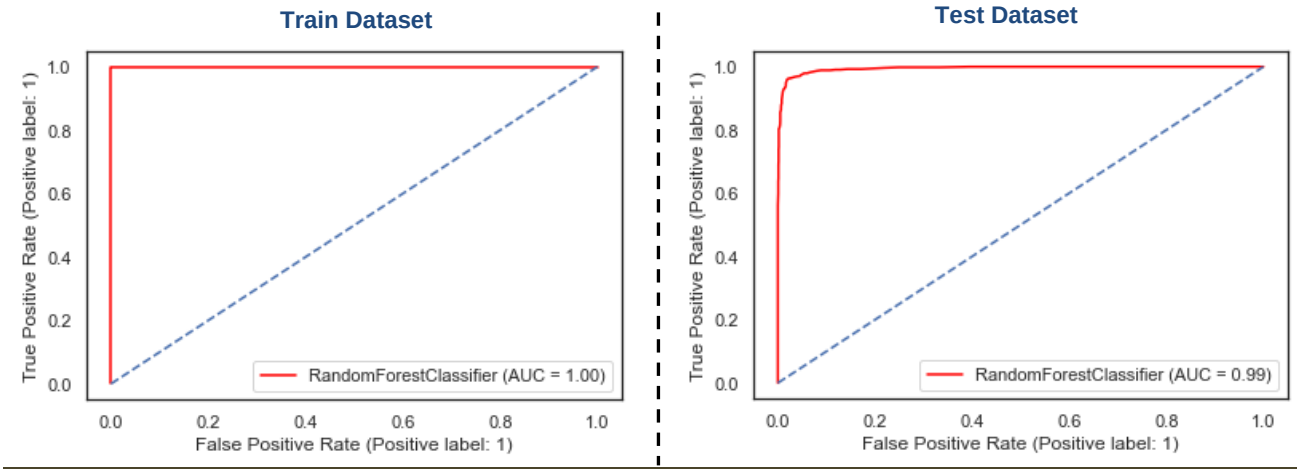
AUC: 1.000

Test Dataset

	precision	recall	f1-score	support
0	0.97	1.00	0.98	2637
1	0.97	0.83	0.89	534
accuracy			0.97	3171
macro avg	0.97	0.91	0.94	3171
weighted avg	0.97	0.97	0.97	3171

AUC: 0.995

Figure 44. Random Forest: Base Model ROC Curves



5.2.6.2 Hypertuned Models

5.2.6.2.1 Random Forest: Hyperparameters

```
param_grid1 = {
    'max_depth': [5,7,10],
    'max_features': [4,5,6],
    'min_samples_leaf': [5,10],
    'min_samples_split': [50,100],
    'n_estimators': [101,201,301]
}

param_grid2 = {
    'max_depth': [11],
    'max_features': [8,9,10],
    'min_samples_leaf': [4],
    'min_samples_split': [40],
    'n_estimators': [101,151]
}

param_grid3 = {
    'max_depth': [13,14,15],
    'max_features': [15,16],
    'min_samples_leaf': [3],
    'min_samples_split': [30,20],
    'n_estimators': [101]
}
```

```
param_grid4 = {
    'max_depth': [15,16,17],
    'max_features': [16,17],
    'min_samples_leaf': [3],
    'min_samples_split': [20,10],
    'n_estimators': [101]
}

param_grid5 = {
    'max_depth': [17,18,19],
    'max_features': [16,17],
    'min_samples_leaf': [2,3],
    'min_samples_split': [8,7,6],
    'n_estimators': [101,151]
}

param_grid6 = {
    'max_depth': [19,20,21],
    'max_features': [16],
    'min_samples_leaf': [2],
    'min_samples_split': [6],
    'n_estimators': [121,151,171]
}

param_grid7 = {
    'max_depth': [19,20],
    'max_features': [16,17],
    'min_samples_leaf': [2],
    'min_samples_split': [2,3,4,5],
    'n_estimators': [121]
}
```

```
param_grid8 = {
    'max_depth': [20,21,22],
    'max_features': [16,17,18],
    'min_samples_leaf': [1,2,3],
    'min_samples_split': [2,3],
    'n_estimators': [121,151]
}
```

RF8 model marked in red is the best model as it is providing the best scores for precision, recall, f1 and accuracy on the test dataset for 1, which represents the churned customers in the dataset.

Figure 45. Random Forest: Best Model Confusion Matrix

Figure 46. Random Forest: Best Model Classification Report

Train Dataset

	precision	recall	f1-score	support
0	1.00	1.00	1.00	6153
1	1.00	1.00	1.00	1246
accuracy			1.00	7399
macro avg	1.00	1.00	1.00	7399
weighted avg	1.00	1.00	1.00	7399

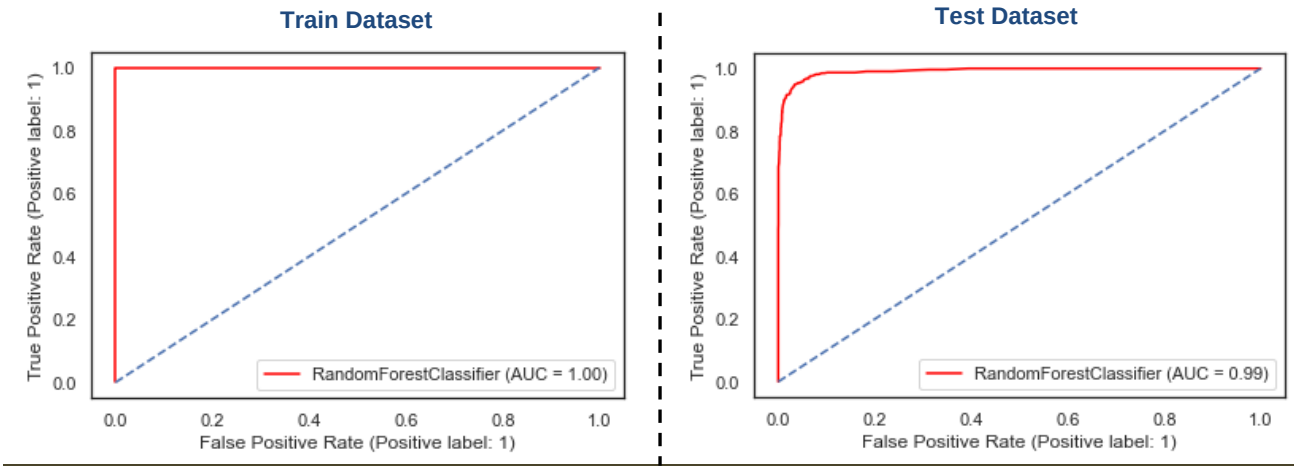
AUC: 1.000

Test Dataset

	precision	recall	f1-score	support
0	0.97	0.99	0.98	2637
1	0.96	0.85	0.90	534
accuracy			0.97	3171
macro avg	0.96	0.92	0.94	3171
weighted avg	0.97	0.97	0.97	3171

AUC: 0.992

Figure 47. Random Forest: Best Model ROC Curves



5.2.7 Adaptive Boosting (ADA Booster)

5.2.7.1 Base Model

An AdaBoost classifier is a meta-estimator that begins by fitting a classifier on the original dataset and then fits additional copies of the classifier on the same dataset but where the weights of incorrectly classified instances are adjusted such that subsequent classifiers focus more on difficult cases.We made the first model with default hyperparameters and below are the accuracy scores obtained from this base model:

Accuracy of the training dataset: 0.8940397350993378

Accuracy of the testing dataset: 0.9016083254493851

Figure 48. ADA Booster: Base Model Confusion Matrix

Figure 49. ADA Booster: Base Model Classification Report

Train Dataset

	precision	recall	f1-score	support
0	0.92	0.96	0.94	6153
1	0.74	0.57	0.65	1246
accuracy			0.89	7399
macro avg	0.83	0.77	0.79	7399
weighted avg	0.89	0.89	0.89	7399

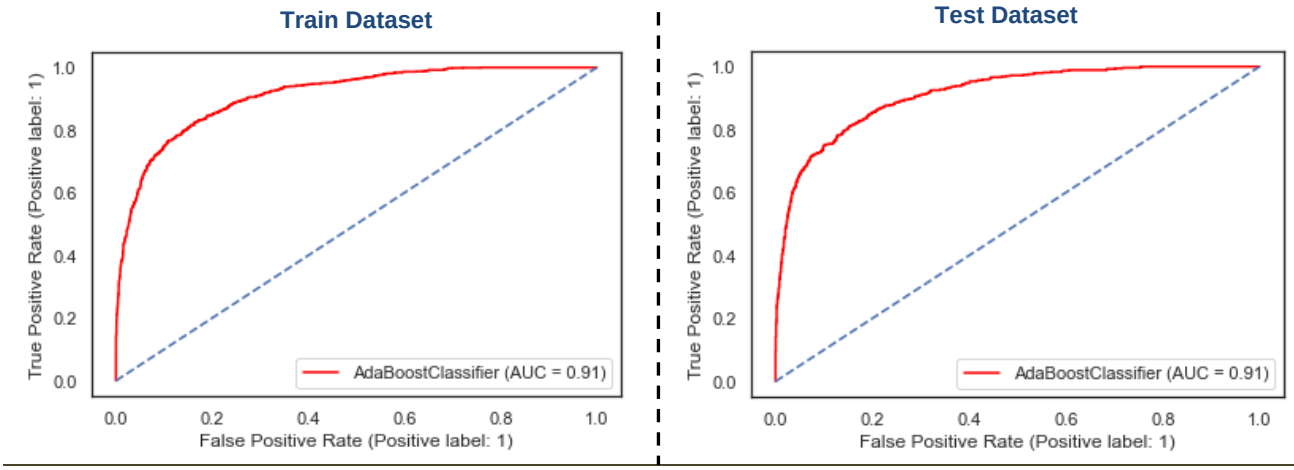
AUC: 0.912

Test Dataset

	precision	recall	f1-score	support
0	0.92	0.96	0.94	2637
1	0.76	0.61	0.68	534
accuracy			0.90	3171
macro avg	0.84	0.79	0.81	3171
weighted avg	0.90	0.90	0.90	3171

AUC: 0.914

Figure 50. ADA Booster: Base Model ROC Curves



5.2.7.2 Hypertuned Model

We have considered multiple hyperparameters to hypertune the model to improve precision, recall, and accuracy. The best hyperparameters output using GridSearch CV were:

algorithm = 'SAMME.R'**, learning_rate =** 1, **n_estimators** = 151

Below are the accuracy scores obtained from this hypertuned model:

Accuracy of the training dataset: 0.8979591836734694

Accuracy of the testing dataset: 0.8997161778618732

Figure 51. ADA Booster: Hypertuned Model Confusion Matrix

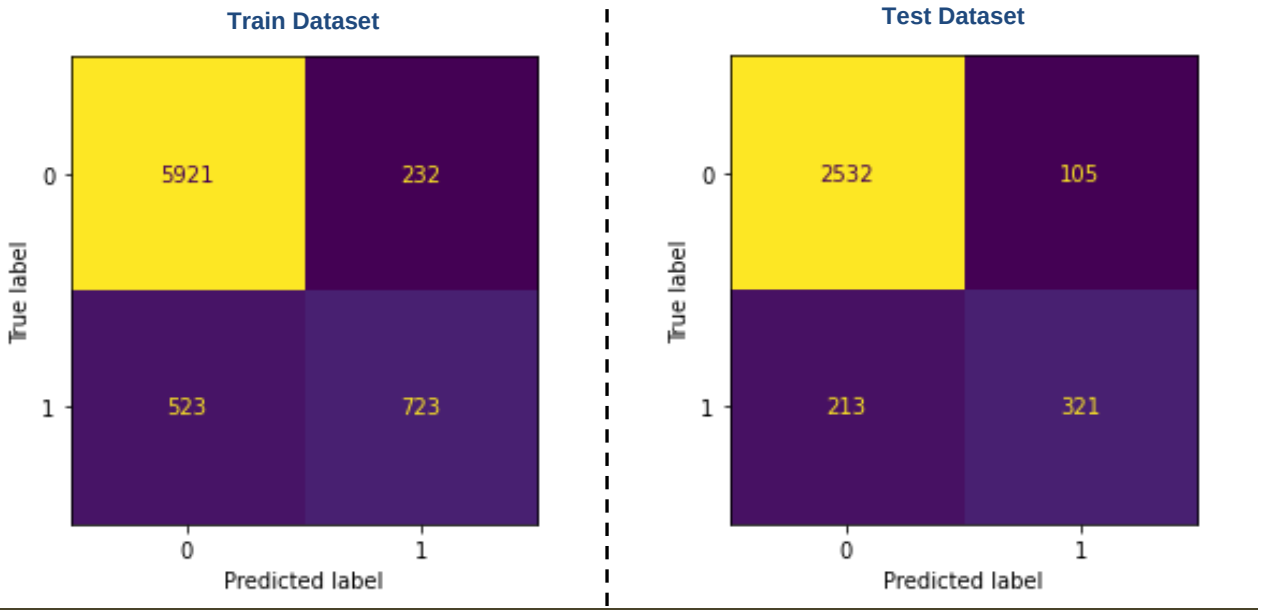


Figure 52. ADA Booster: Hypertuned Model Classification Report

Train Dataset

	precision	recall	f1-score	support
0	0.92	0.96	0.94	6153
1	0.76	0.58	0.66	1246
accuracy			0.90	7399
macro avg	0.84	0.77	0.80	7399
weighted avg	0.89	0.90	0.89	7399

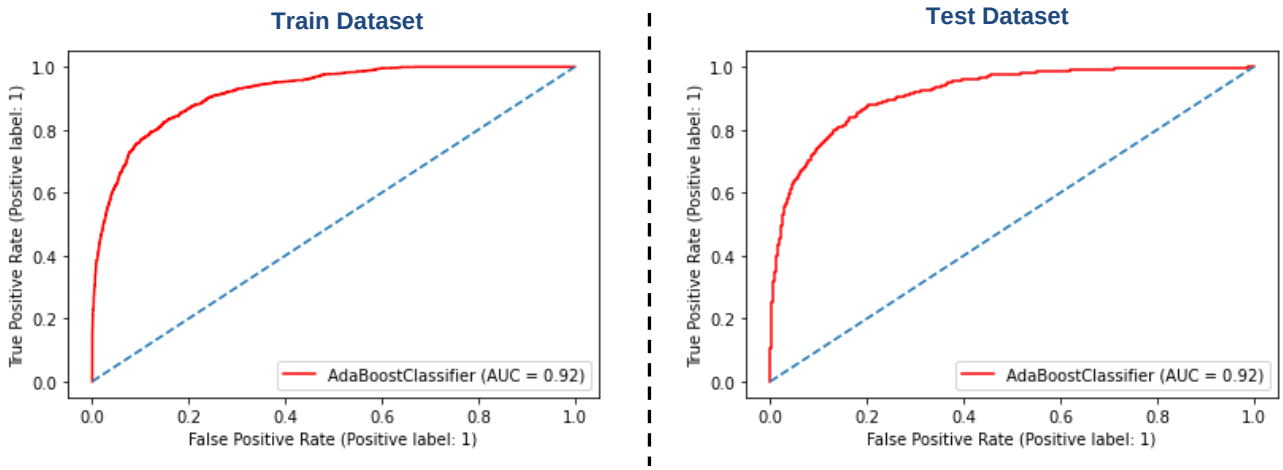
AUC: 0.921

Test Dataset

	precision	recall	f1-score	support
0	0.92	0.96	0.94	2637
1	0.75	0.60	0.67	534
accuracy			0.90	3171
macro avg	0.84	0.78	0.80	3171
weighted avg	0.89	0.90	0.90	3171

AUC: 0.917

Figure 53. ADA Booster: Hypertuned Model ROC Curves



5.2.8 Gradient Boosting

5.2.8.1 Base Model

We fitted gradient boosting classifier model into training dataset and performed prediction on training and testing dataset using the same model. We made the first model with default hyperparameters and below are the accuracy scores obtained from this base model:

Accuracy of the training dataset: 0.9085011488038924

Accuracy of the testing dataset: 0.9016083254493851

Figure 54. Gradient Boosting: Base Model Confusion Matrix

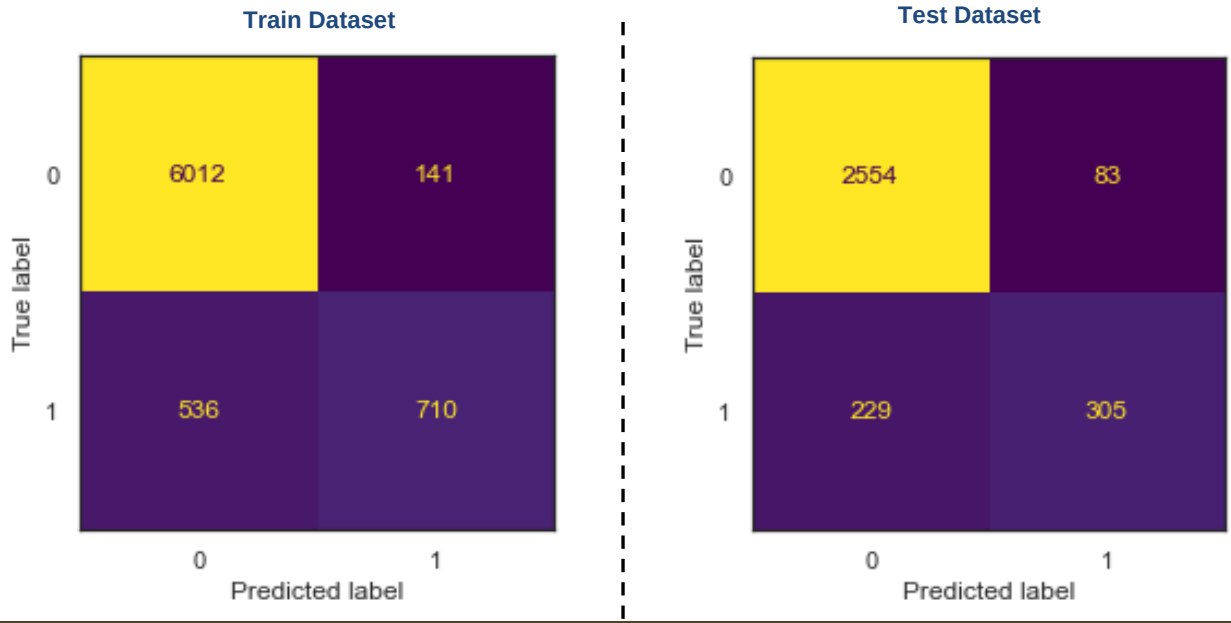


Figure 55. Gradient Boosting: Base Model Classification Report

Train Dataset

	precision	recall	f1-score	support
0	0.92	0.98	0.95	6153
1	0.83	0.57	0.68	1246
accuracy			0.91	7399
macro avg	0.88	0.77	0.81	7399
weighted avg	0.90	0.91	0.90	7399

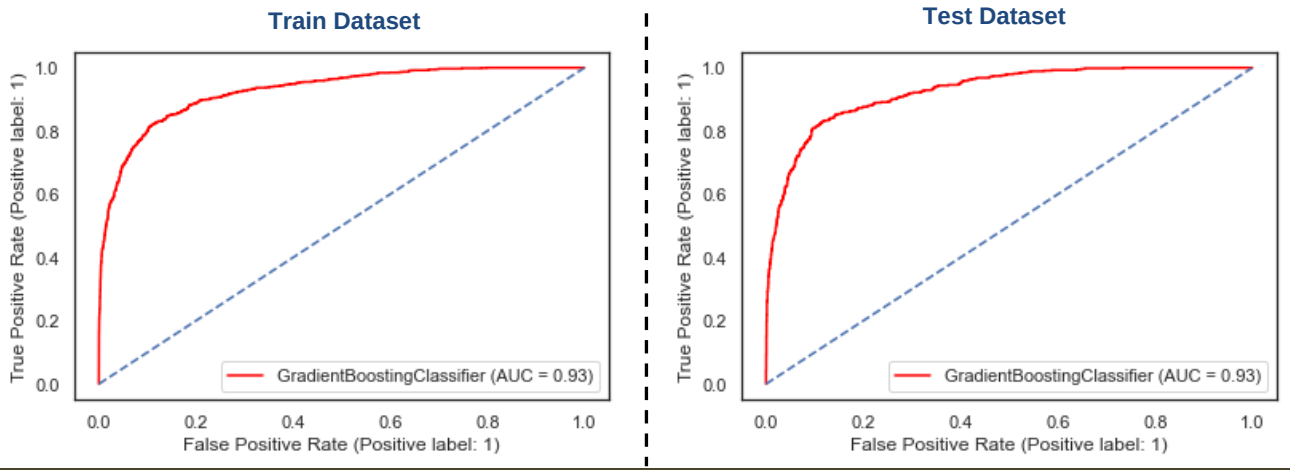
AUC: 0.927

Test Dataset

	precision	recall	f1-score	support
0	0.92	0.97	0.94	2637
1	0.79	0.57	0.66	534
accuracy			0.90	3171
macro avg	0.85	0.77	0.80	3171
weighted avg	0.90	0.90	0.90	3171

AUC: 0.925

Figure 56. Gradient Boosting: Base Model ROC Curves



5.2.8.2 Hypertuned Models

Gradient boost is an ensemble machine learning algorithm that trains underlying models in a gradual, additive and sequential manner.

In this modelling exercise, SKlearn's Gradient Boost classifier function was used for modelling:

- The base model was run with default hyperparameters and the performance metrics noted. Tuning was later done using GridSearchCV.
- Model performance metrics for base model and best model have been provided in the below sections

The details of this algorithm, its tuning and performance metrics have been provided in this section. The reasons why the tuned Gradient boost was selected as best model and model interpretation.

GB4 model marked in red is the best model as it is providing the best scores for precision, recall, f1 and accuracy on the test dataset for 1, which represents the churned customers in the dataset.

5.2.8.2.1 Gradient Boosting: Hyperparameters

```
param_grid1 = {
    'loss': ['deviance', 'exponential'],
    'learning_rate': [0.1,0.5],
    'n_estimators': [51,101,151],
    'criterion': ['friedman_mse', 'mse', 'mae'],
    'min_samples_split': [20,60,100],
    'min_samples_leaf': [2,6,10],
    'max_depth':[3,6,9],
    'max_features':[7,10]
}

param_grid2 = {
    'loss': ['deviance','exponential'],
    'learning_rate': [0.5],
    'n_estimators': [101],
    'criterion': ['mse'], #
    'min_samples_split': [20],
    'min_samples_leaf': [6],
    'max_depth':[9],
    'max_features':[10]
}

param_grid4 = {
    'loss': ['deviance'],
    'learning_rate': [0.5],
    'n_estimators': [101,151,201],
    'criterion': ['mse'],
    'min_samples_split': [20],
    'min_samples_leaf': [6],
    'max_depth':[9],
    'max_features':[10]
}
```

```
param_grid5 = {
     'loss': ['deviance'],
     'learning_rate': [0.5],
     'n_estimators': [201,401,601],
     'criterion': ['mse'],
     'min_samples_split': [20],
     'min_samples_leaf': [6],
     'max_depth':[9],
     'max_features':[10]
}

param_grid6 = {
     'loss': ['deviance'],
     'learning_rate': [0.5],
     'n_estimators': [201],
     'criterion': ['mse'],
     'min_samples_split': [20],
     'min_samples_leaf': [6],
     'max_depth':[9,12,15],
     'max_features':[10,11,12]
}

param_grid7 = {
     'loss': ['deviance'],
     'learning_rate': [0.5],
     'n_estimators': [201],
     'criterion': ['mse'],
     'min_samples_split': [20,15],
     'min_samples_leaf': [4,6],
     'max_depth':[9,11],
     'max_features':[10,11]
}

param_grid8 = {
     'loss': ['deviance'],
     'learning_rate': [0.1, 0.5, 1],
     'n_estimators': [201],
     'criterion': ['mse'],
     'min_samples_split': [15],
     'min_samples_leaf': [6],
     'max_depth':[9],
     'max_features':[11]
}
```

5.2.9 Gradient Boosting: 5-Fold Cross Validation & Mean Scores (Best Model)

```
(array([0.9394387 , 0.94736842, 0.97428571, 0.96115108, 0.89665211]),
 0.9437792050497313)
```

5.2.10 Gradient Boosting: 10-Fold Cross Validation & Mean Scores (Best Model)

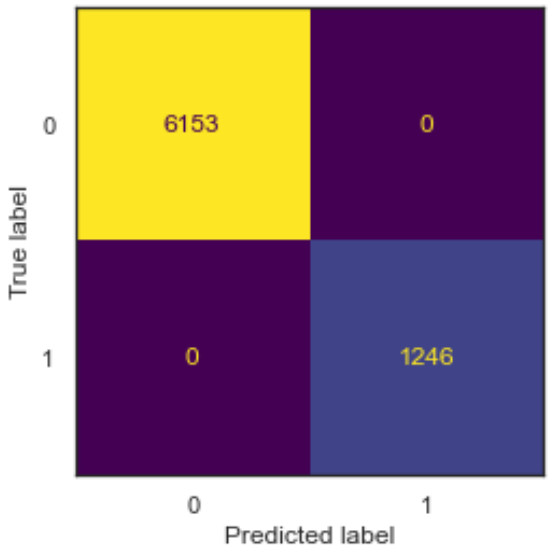
```
(array([0.97421203, 0.97421203, 0.97701149, 0.94736842, 0.95930233,
        0.98016997, 0.96571429, 0.94767442, 0.95428571, 0.95953757]),
 0.9639488272185183)
```

5.2.11 Gradient Boosting: Feature Importance Table (Best Model)

	importance
Tenure	0.277573
account_segment	0.070482
Complain_ly	0.065553
Day_Since_CC_connect	0.061621
CC_Contacted_LY	0.061232
rev_growth_yoy	0.057713
CC_Agent_Score	0.057630
cashback	0.055381
rev_per_month	0.054191
City_Tier	0.035733
Account_user_count	0.033901
Login_device_Mobile	0.026697
Marital_Status_Single	0.025183
Gender_Male	0.021796
Payment_Credit Card	0.020110
coupon_used_for_payment	0.017183
Marital_Status_Married	0.014897
Payment_Debit Card	0.014814
Service_Score	0.013672
Payment_E wallet	0.010198
Payment_UPI	0.004437

Figure 57. Gradient Boosting: Best Model Confusion Matrix

Train Dataset



Test Dataset

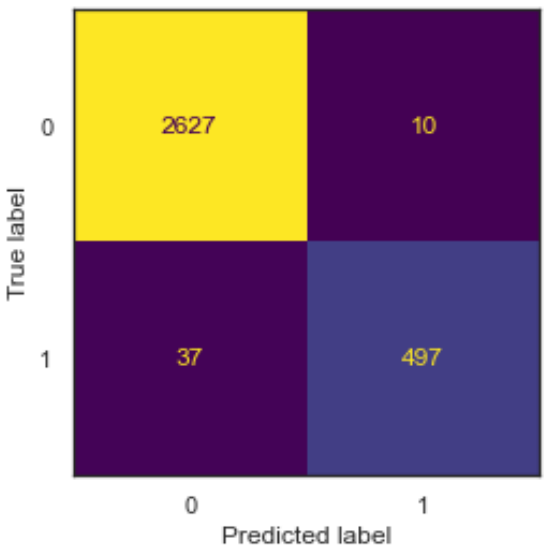


Figure 58. Gradient Boosting: Best Model Classification Report

Train Dataset

	precision	recall	f1-score	support
0	1.00	1.00	1.00	6153
1	1.00	1.00	1.00	1246
accuracy			1.00	7399
macro avg	1.00	1.00	1.00	7399
weighted avg	1.00	1.00	1.00	7399

AUC: 1.000

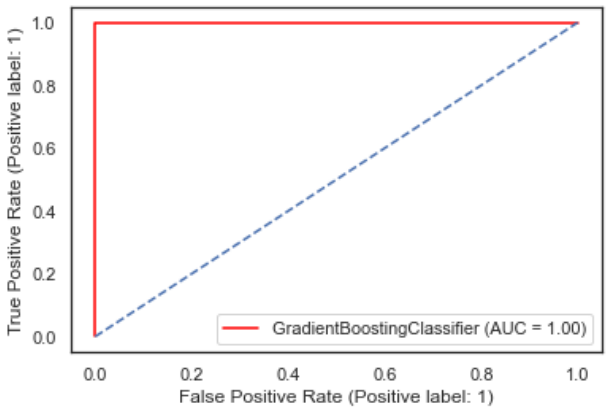
Test Dataset

	precision	recall	f1-score	support
0	0.99	1.00	0.99	2637
1	0.98	0.93	0.95	534
accuracy			0.99	3171
macro avg	0.98	0.96	0.97	3171
weighted avg	0.99	0.99	0.99	3171

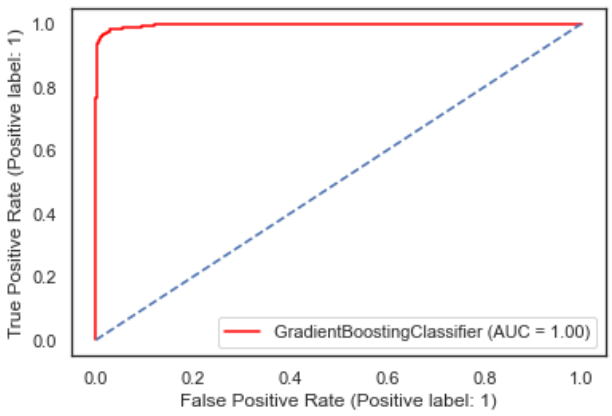
AUC: 0.997

Figure 59. Gradient Boosting: Best Model ROC Curves

Train Dataset



Test Dataset



5.2.12 Support Vector Machine (SVM)

5.2.12.1 Base Model

Support vector machine (SVM) is a popular machine learning algorithm that can be used for classification as well as regression. It works well when there are higher dimensions as well. It is very versatile as there are different kernel functions that can be specified to work well with the given data. In this modelling exercise, SKlearn's Support Vector machine function was used for modelling:

- The model requires scaled data so scaling was done using Sklearn's Standard Scaler.
- The base model was run with default hyperparameters with outlier treated unscaled dataset and the performance metrics noted. Tuning was later done using GridSearchCV.
- Model performance metrics for base model and best model have been provided in the below sections.

We made the first model with default hyperparameters and below are the accuracy scores obtained from this base model:

Accuracy of the training dataset: 0.9199891877280714

Accuracy of the testing dataset: 0.9116997792494481

Figure 60. Support Vector Machine (SVM): Base Model Confusion Matrix

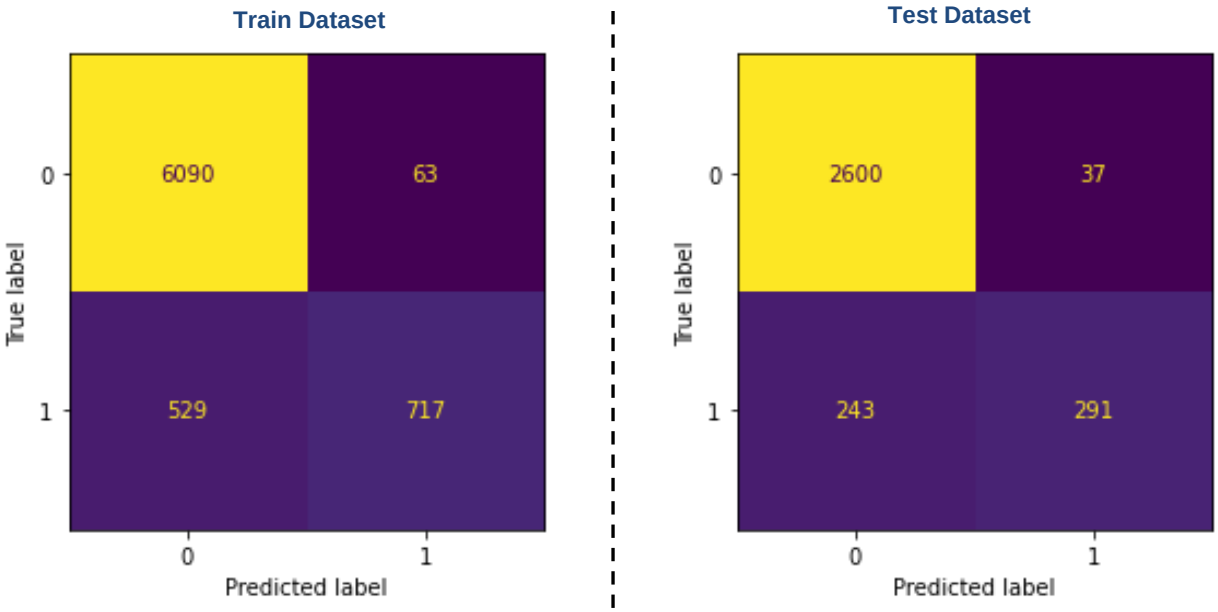


Figure 61. Support Vector Machine (SVM): Base Model Classification Report

Train Dataset

	precision	recall	f1-score	support
0	0.92	0.99	0.95	6153
1	0.92	0.58	0.71	1246
accuracy			0.92	7399
macro avg	0.92	0.78	0.83	7399
weighted avg	0.92	0.92	0.91	7399

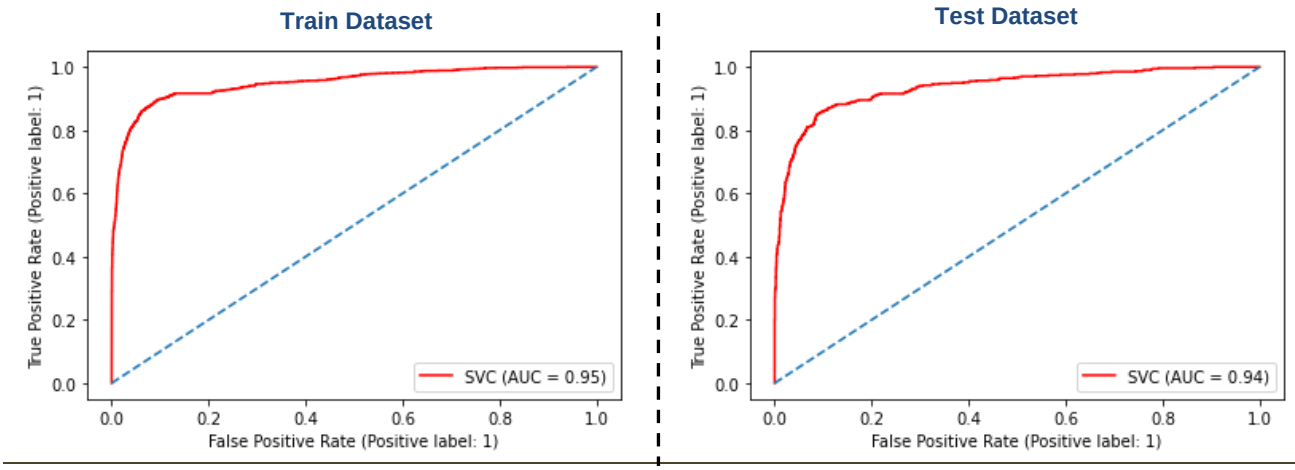
AUC: 0.950

Test Dataset

	precision	recall	f1-score	support
0	0.91	0.99	0.95	2637
1	0.89	0.54	0.68	534
accuracy			0.91	3171
macro avg	0.90	0.77	0.81	3171
weighted avg	0.91	0.91	0.90	3171

AUC: 0.936

Figure 62. Support Vector Machine (SVM): Base Model ROC Curves



5.2.12.2 Hypertuned Model

We have considered multiple hyperparameters to hypertune the model to improve precision, recall, and accuracy. The best hyperparameters output using GridSearch **CV** were:

C = 9**, class_weight** = 'balanced'**, degree** = 5**, gamma** = 'scale'**, kernel** = 'poly'**, tol** = 0.001

Below are the accuracy scores obtained from this hypertuned model:

Accuracy of the training dataset: 0.9982430058115962

Accuracy of the testing dataset: 0.967833491012299

Figure 63. Support Vector Machine (SVM): Hypertuned Model Confusion Matrix

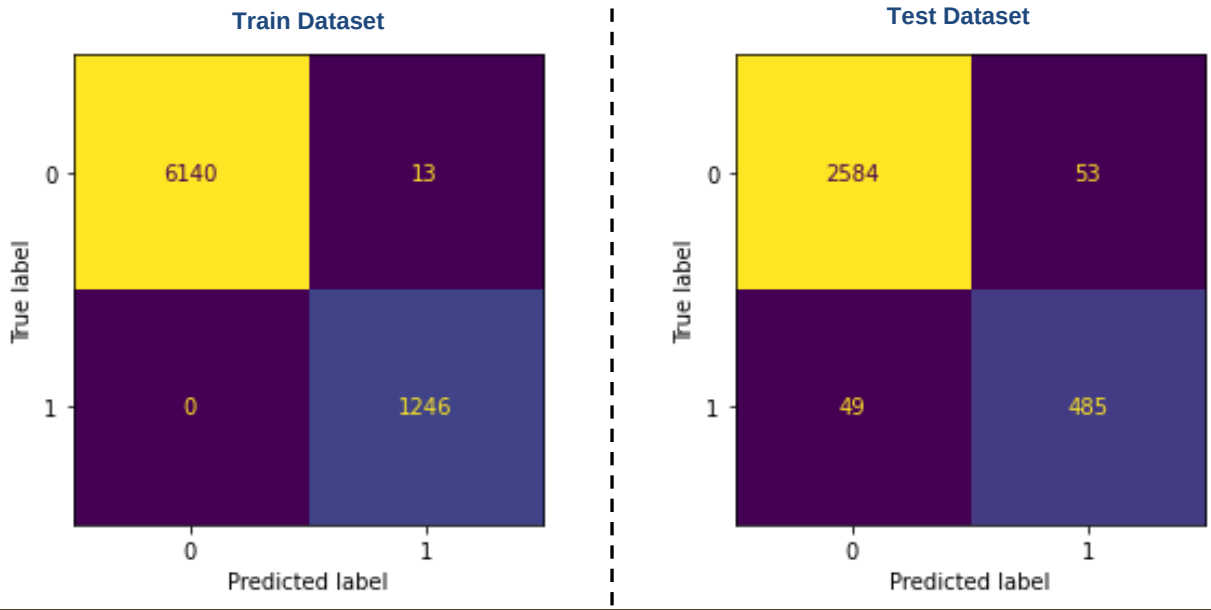


Figure 64. Support Vector Machine (SVM): Hypertuned Model Classification Report

Train Dataset

	precision	recall	f1-score	support
0	1.00	1.00	1.00	6153
1	0.99	1.00	0.99	1246
accuracy			1.00	7399
macro avg	0.99	1.00	1.00	7399
weighted avg	1.00	1.00	1.00	7399

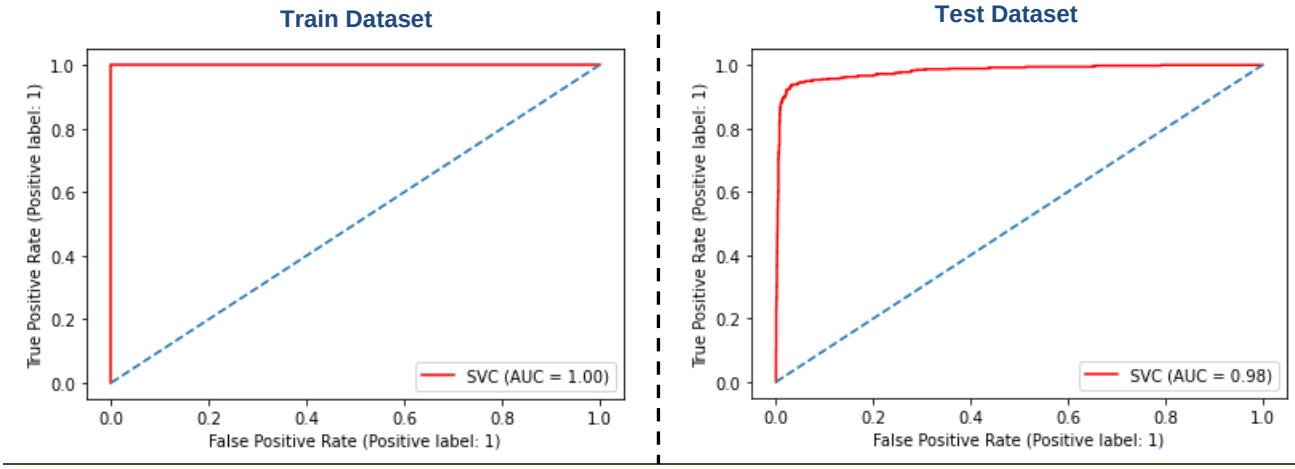
AUC: 1.000

Test Dataset

	precision	recall	f1-score	support
0	0.98	0.98	0.98	2637
1	0.90	0.91	0.90	534
accuracy			0.97	3171
macro avg	0.94	0.94	0.94	3171
weighted avg	0.97	0.97	0.97	3171

AUC: 0.981

Figure 65. Support Vector Machine (SVM): Hypertuned Model ROC Curves



The End