FINAL REPORT
CAPSTONE PROJECT

THANUSRI A

11-08-2024

Table of Contents

1. Introduction - Business problem definition

1.1 Problem statement

A DTH 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 potential churners. In this company, account churn is a major issue because 1 account can have multiple customers. Hence by losing one account the company might be losing more than one customer. The current project is aimed at developing a churn prediction model for this company and to provide business recommendations on a campaign focused on retaining customers. 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 for the project

A DTH provider's biggest cost is cost of an acquisition of a new customer thus acquired will need to be retained for quite a few years so that the initial cost of acquisition is recovered back, and that account is profitable. Due to this reason, customer churn directly impacts the profitability of a DTH operator. DTH providers also are in a constant pressure to increase their customer base to maintain 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. Hence it becomes very important to not only increase the customer base but also protect the current customer base.
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. Project Objectives

1) This project aims at identifying customers who have a propensity to churn so that targeted campaign offers can be provided to them. This would ensure better top-line and bottom-line revenue for the business.
2) Other insights from Exploratory data analysis and best performing model(s) will also be used to come up with business recommendations.

2. Data report

2.1. Data collection

- The dataset contains Account level data which is master data along with features of account such as gender, marital status, city tier, account user count of primary account holder, whether the account is live or churned, segment the account belongs to. It also possibly contains certain derived features such as tenure (which probably could have been derived from account open date)
- The dataset also contains information taken/derived from transaction data and rolled up at Account level and given as a feature for the account – _for e.g., Number of days since none of the account holders contacted customer care, monthly average cashback for last 12 months, number of complaints made last year, number of times customer care contacted last year, revenue per month in last 12 months, how many times customers have used coupons to pay in last 12 months, satisfaction score and customer service score. Most of the transaction data roll-up at account level has been done for 12 months (previous year). However, the revenue growth percentage has been taken for last year in comparison with previous one year which implies that 24- months' worth of data has been used to calculate this
- As the dataset has been provided, the methodology used by customer to extract data is not known. The frequency at which this dataset is extracted has also not been specified.

2.2. Visual inspection of data

• The dataset has 11260 rows and 19 columns

• There are 5 columns of float type and 2 columns of integer type and 12 columns of object type.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 11260 entries, 0 to 11259
Data columns (total 18 columns):
 #   Column                      Non-Null Count  Dtype
---  ------                      --------------  -----
 0   Churn                       11260 non-null  int64
 1   Tenure                      11158 non-null  object
 2   City_Tier                   11148 non-null  float64
 3   CC_Contacted_LY             11158 non-null  float64
 4   Payment                     11151 non-null  object
 5   Gender                      11152 non-null  object
 6   Service_Score               11162 non-null  float64
 7   Account_user_count          11148 non-null  object
 8   account_segment             11163 non-null  object
 9   CC_Agent_Score              11144 non-null  float64
 10  Marital_Status              11048 non-null  object
 11  rev_per_month               11158 non-null  object
 12  Complain_ly                 10903 non-null  float64
 13  rev_growth_yoy              11260 non-null  object
 14  coupon_used_for_payment     11260 non-null  object
 15  Day_Since_CC_connect        10903 non-null  object
 16  cashback                    10789 non-null  object
 17  Login_device                11039 non-null  object
dtypes: float64(5), int64(1), object(12)
memory usage: 1.5+ MB
```

• There are several columns that are supposed to be read as numeric, instead they have been read as object type for e.g., Tenure is a numeric field but has been read as object. Those columns need to be checked for special characters and need to be treated before the column can be changed to numeric for further processing.

• There are no duplicate rows in the data set. Each account id has one unique row.

• Several columns have null values

• The following table shows number of rows containing nulls and special characters that require data cleaning. All special characters present in the data set were treated with nulls so that they can be imputed. Some columns such as Gender and account_segment contained multiple values to represent the same category for e.g., ‘M’, ‘Male’. Cleaning up of those values was also performed.

Column	Values present	% Rows with values present	Number of Nulls	% Rows with nulls	Data clean-up needed?	% Rows needing data cleaning
AccountID	11260	100.00%	0	0%	None	0.00%
Churn	11260	100.00%	0	0%	None	0.00%
Tenure	11158	99.09%	102	0.91%	Yes - #	1.03%
City_Tier	11148	99.01%	112	0.99%	None	0.00%
CC_Contacted_LY	11158	99.09%	102	0.91%	None	0.00%
Payment	11151	99.03%	109	0.97%	None	0.00%
Gender	11152	99.04%	108	0.96%	Yes - M,F	5.74%
Service_Score	11162	99.13%	98	0.87%	None	0.00%
Account_user_count	11148	99.01%	112	0.99%	Yes - @	2.95%
account_segment	11163	99.14%	97	0.86%	Yes-Regular + Super +	2.74%
CC_Agent_Score	11144	98.97%	116	1.03%	None	0.00%
Marital_Status	11048	98.12%	212	1.88%	None	0.00%
rev_per_month	11158	99.09%	102	0.91%	Yes - +	6.12%
Complain_ly	10903	96.83%	357	3.17%	None	0.00%
rev_growth_yoy	11260	100.00%	0	0.00%	Yes - $	0.03%
coupon_used_for_payment	11260	100.00%	0	0.00%	Yes - $, *, #	0.03%
Day_Since_CC_connect	10903	96.83%	357	3.17%	Yes - $	0.01%
Cashback	10789	95.82%	471	4.18%	Yes - $	0.02%
Login_device	11039	98.04%	221	1.96%	Yes - &&&&	4.79%

Table 2-2 Nulls and special characters in dataset

2.3. Understanding the attributes

The following table shows the attribute names, their description, and the kind of values that they contain. Although some of the variable names are slightly long, they do not have blanks or special characters in them. Hence, it has been decided to let the current column names stay as-is as they are self-explanatory and would be easy to understand and interpret when seen in the plots as part of univariate and bivariate analysis. The variable names would be changed later to shorten or make it uniform when one hot encoding is done in a later section.

S.no	Column	Column Description	Data description
1	AccountID	account unique identifier	Unique ID. Hence, it will not be used in modelling
2	Churn	account churn flag (Target)	Target variable. Contains 1 for churned and 0 for non-churned
3	Tenure	Tenure of account	Continuous field. Contains values ranging from 0 to 99
4	City_Tier	Tier of primary customer's city	Categorical ordinal - values 1,2,3
5	CC_Contacted_LY	How many times all the customers of the account have contacted customer care in last 12months	Continuous field. Contains values ranging from 4 to 132
6	Payment	Preferred Payment mode of the customers in the account	Categorical nominal - values Credit card, debit card, E wallet, UPI, Cash on Delivery

S.no	Column	Column Description	Data description
7	Gender	Gender of the primary customer	Categorical nominal - values Male, Female, M and F (M and F need to be converted to Male and Female)
8	Service_Score	Satisfaction score given by customers of the account on service provided by company	Categorical ordinal - values 0 to 5
9	Account_user_count	Number of customers tagged with this account	Limited range. Can be treated as categorical - values 1 to 6
10	account_segment	Account segmentation on the basis of spend	Categorical nominal - values HNI, Regular, Regular Plus, Super, Super plus and variations with +
11	CC_Agent_Score	Satisfaction score given on customer care service provided	Categorical ordinal - values 1 to 5
12	Marital_Status	Marital status of primary customer	Categorical nominal - contains values Married, Single and Divorced
13	rev_per_month	Monthly average revenue from account in last 12 months	Continuous field. Contains values ranging from 1 to 140
14	Complain_ly	Complaints raised by account in last 12 months	Categorical - 0 (for no) or 1 (for yes)
15	rev_growth_yoy	revenue growth percentage of the account (last 12 months vs last 24 to 13 month)	Continuous field. Contains values ranging from 4 to 28
16	coupon_used_for_payment	How many times customers have used coupons to do the payment in last 12 months	Continuous field, but with limited range. Contains values ranging from 0 to 16
17	Day_Since_CC_connect	Number of days since no customers in the account has contacted the customer care	Continuous field. Contains values ranging from 0 to 47
18	Cashback	Monthly average cashback generated by account in last 12 months	Continuous field. Contains values ranging from 0 to 1997
19	Login_device	Preferred login device of the customers in the account	Categorical nominal - contains values Mobile, Computer

Table 2-3 Attribute description

The following table shows a basic statistical description of the numeric columns after data clean-up. It contains a 5-point summary of the numeric fields – _minimum, maximum, 25th percentile, 50th percentile and 75th percentile. In addition, it contains count of values present in each column, mean and standard deviation.

	count	unique	top	freq	mean	std	min	25%	50%	75%	max
Churn	11260.0	2.0	0.0	9364.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Tenure	11042.0	NaN	NaN	NaN	11.025086	12.879782	0.0	2.0	9.0	16.0	99.0
City_Tier	11148.0	3.0	1.0	7263.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN
CC_Contacted_LY	11158.0	NaN	NaN	NaN	17.867091	8.853269	4.0	11.0	16.0	23.0	132.0
Payment	11151	5	Debit Card	4587	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Gender	11152	2	Male	6704	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Service_Score	11162.0	6.0	3.0	5490.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Account_user_count	10816.0	6.0	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	5.0	3.0	3360.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Marital_Status	11048	3	Married	5860	NaN	NaN	NaN	NaN	NaN	NaN	NaN
rev_per_month	10469.0	NaN	NaN	NaN	6.362594	11.909686	1.0	3.0	5.0	7.0	140.0
Complain_ly	10903.0	2.0	0.0	7792.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN
rev_growth_yoy	11257.0	NaN	NaN	NaN	16.193391	3.757721	4.0	13.0	15.0	19.0	28.0
coupon_used_for_payment	11257.0	NaN	NaN	NaN	1.790619	1.969551	0.0	1.0	1.0	2.0	16.0
Day_Since_CC_connect	10902.0	NaN	NaN	NaN	4.633187	3.697637	0.0	2.0	3.0	8.0	47.0
cashback	10787.0	NaN	NaN	NaN	196.23637	178.660514	0.0	147.21	165.25	200.01	1997.0
Login_device	10500	2	Mobile	7482	NaN	NaN	NaN	NaN	NaN	NaN	NaN

• Tenure seems to have a very huge range up to 99. It could be in months in which case those would be valid values.

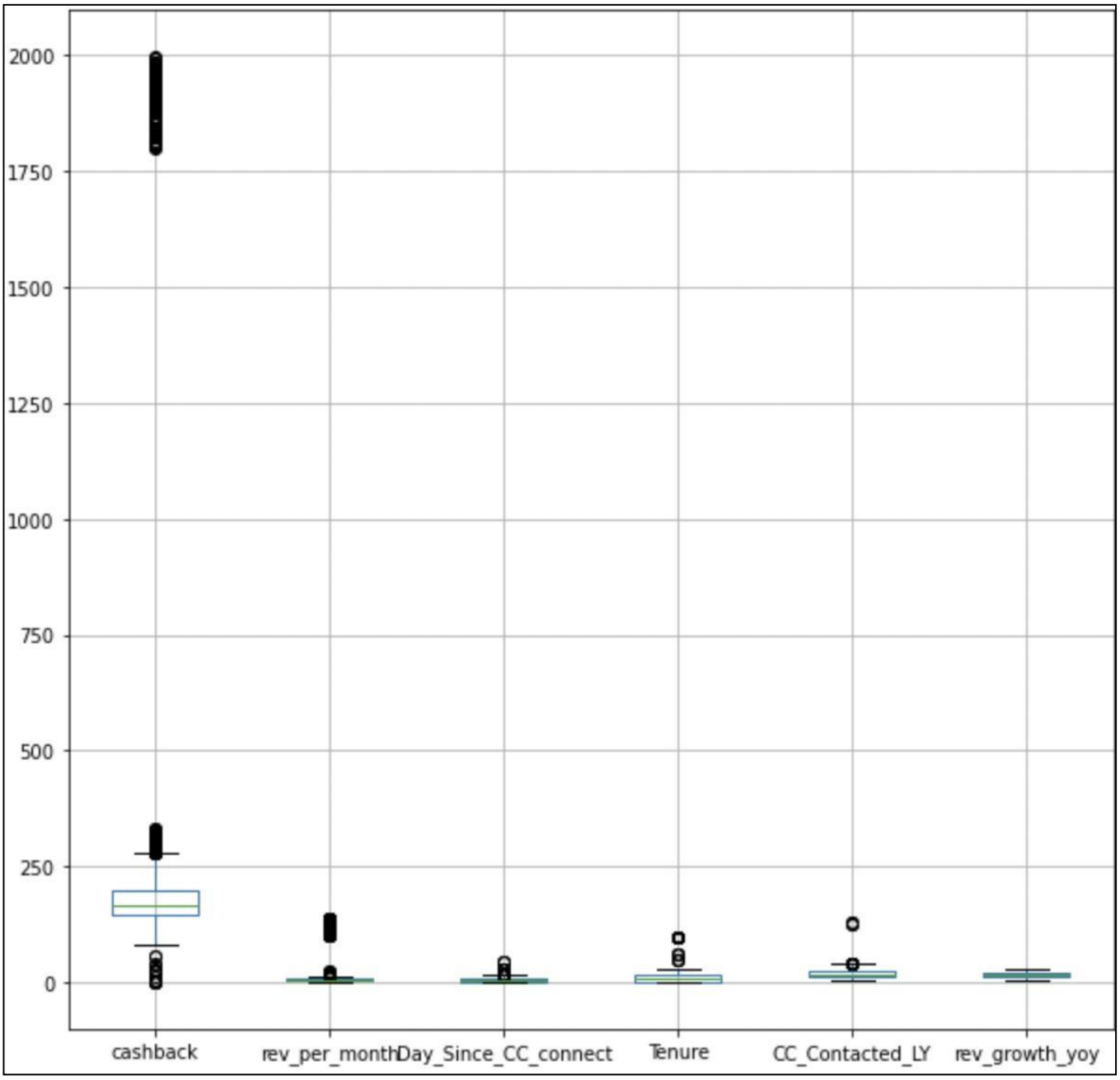
• The maximum limit for many of the variables seems to be very far apart from the 75th percentile for many variables such as cashback, revenue per month and customer contacted last year. There seems to be significant positive skew in these variables. A look at the boxplot and histogram will confirm the presence of outliers.

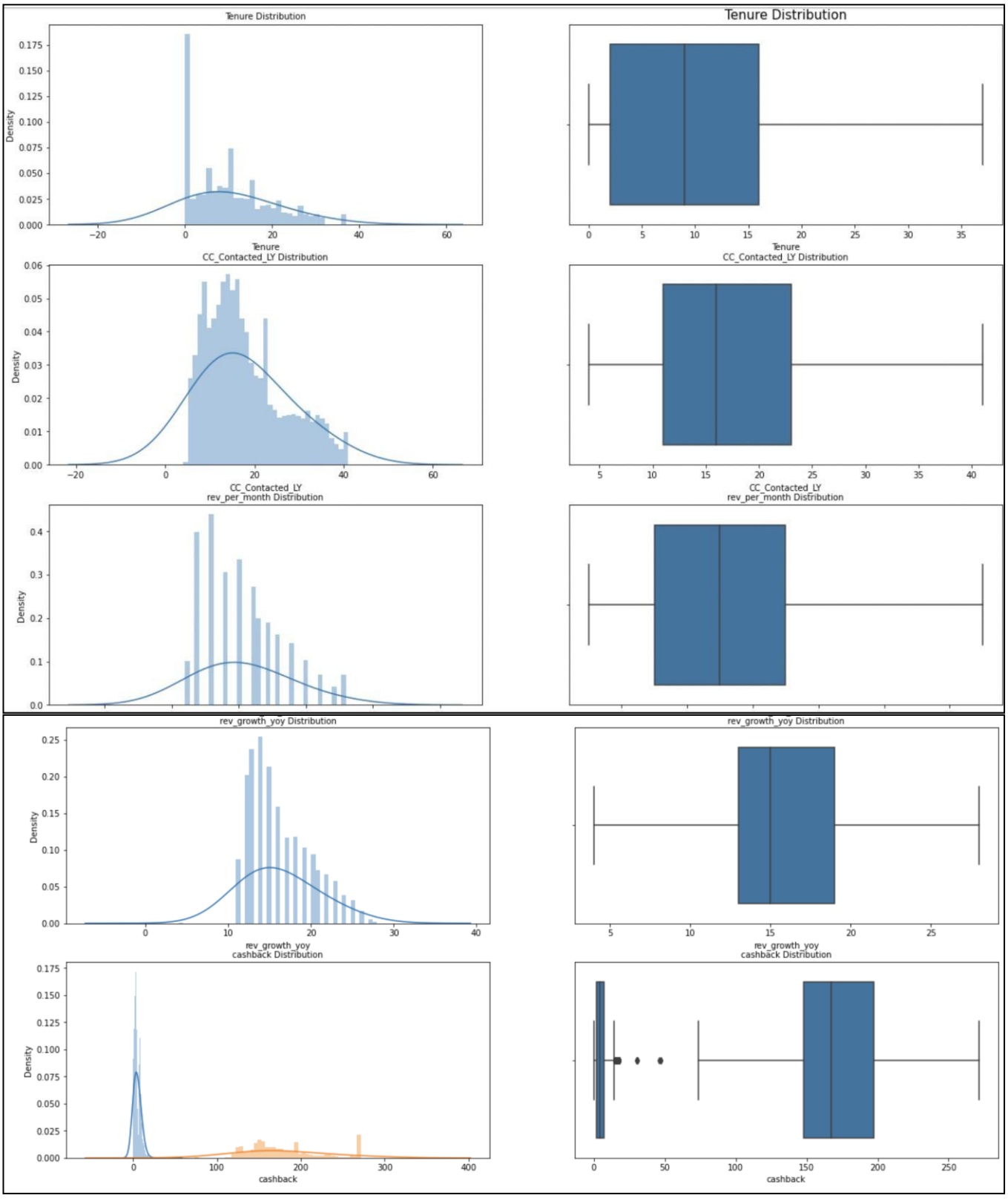
3. Exploratory data analysis

3.1. Univariate analysis

Univariate analysis is done for the purpose of observing distribution and spread for every continuous attribute and distribution of data in categories for categorical ones. It has been done by observing:

• Box plots and histograms for continuous variables
• Count plots for categorical variables


2000
1750
1500
1250
1000
750
500
250
0
cashback
rev_per_month
Day_Since_CC_connect
Tenure
CC_Contacted_LY
rev_growth_yoy



Churn	1.772606
Tenure	3.895707
City_Tier	0.737107
CC_Contacted_LY	1.422977
Service_Score	0.003891
Account_user_count	-0.393100
CC_Agent_Score	-0.142149
rev_per_month	9.093909
Complain_ly	0.950876
rev_growth_yoy	0.752474
coupon_used_for_payment	2.575199
Day_Since_CC_connect	1.273021
cashback	8.770766
dtype: float64	

Observations:

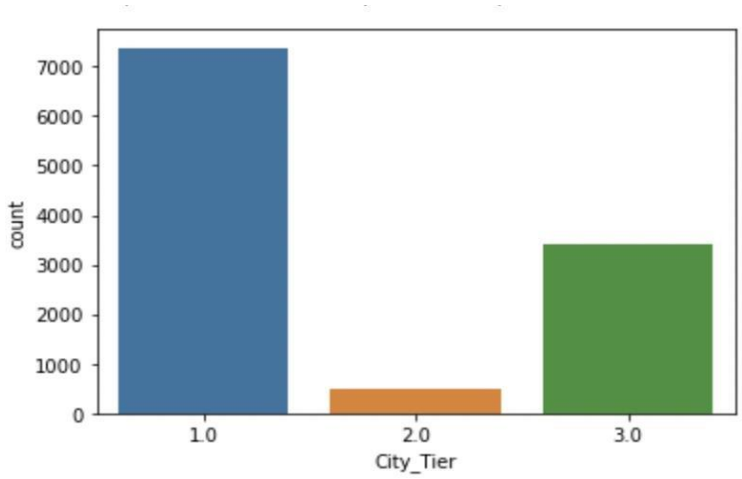
• All numeric variables with the exception of rev_growth_yoy have outliers. Some outliers for certain variables are closer to the whisker, whereas there are a group of outliers that are far beyond the whisker with no in between values. For instance, rev_per_month has a huge space between 30 and 100 indicating absence of values in that range. Those outliers in the extreme values do not correlate with corresponding outliers in cashback field. We cannot rule out these outliers as incorrect values, they may belong to hotels with many rooms. But models like logistic regression are sensitive to outliers and may not give good performance if outliers are left untreated.

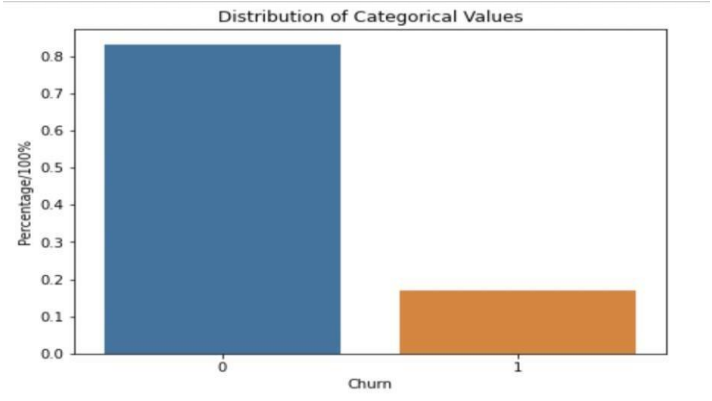
• Hence, we can follow two approaches to modelling – one set of data with outliers treated for outlier sensitive models and other set of data with outliers not treated (left as-is) for outlier resistant models such as Random forest.

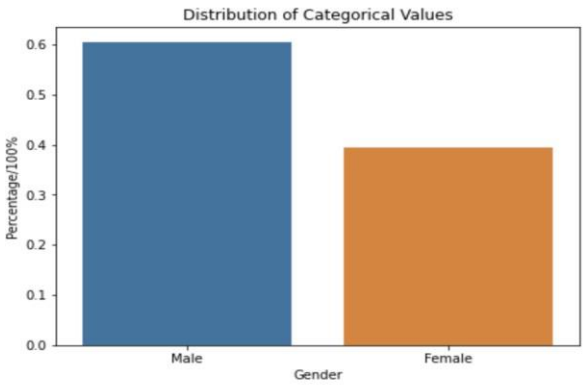
• Coupon_used_for_payment has a very limited range 0 to 16. Hence, for the purpose of this analysis, the outliers will not be treated (similar to categorical variables).

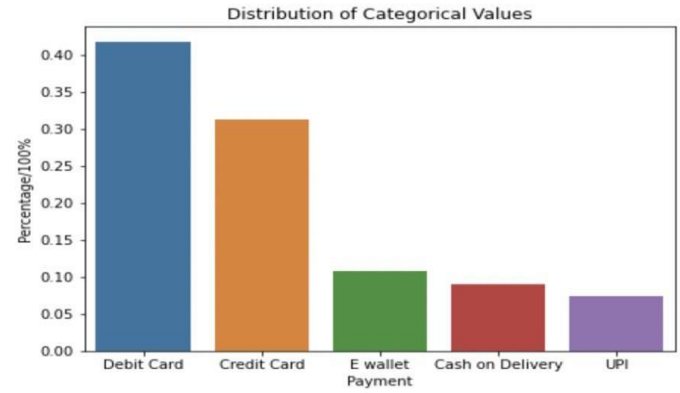
• All numeric variables with the exception of rev_growth_yoy have a high positive skew.

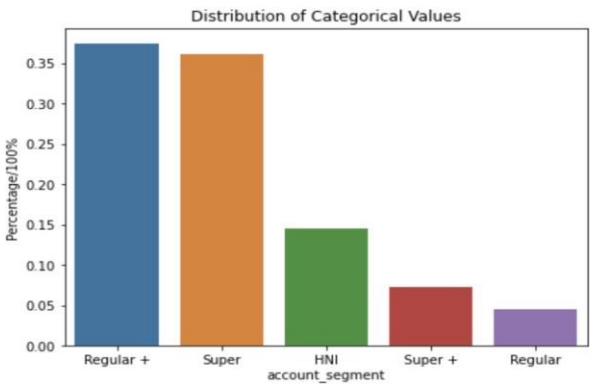
Categorical fields – Count plot

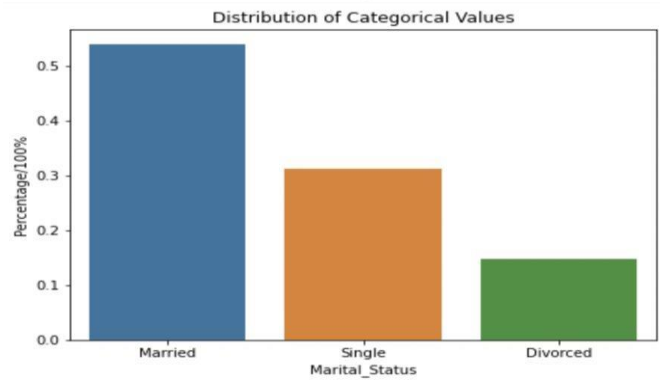


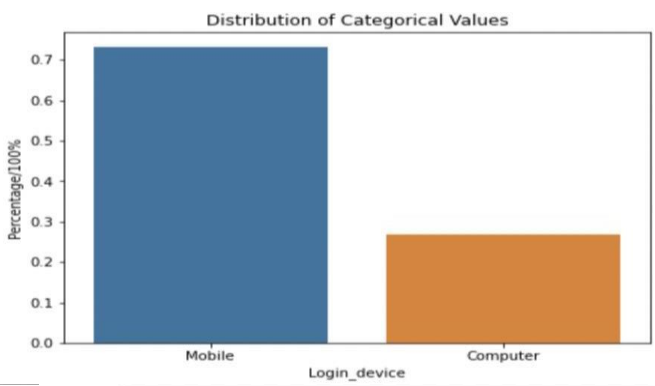

Distribution of Categorical Values
Percentage/100%
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0
1
Churn

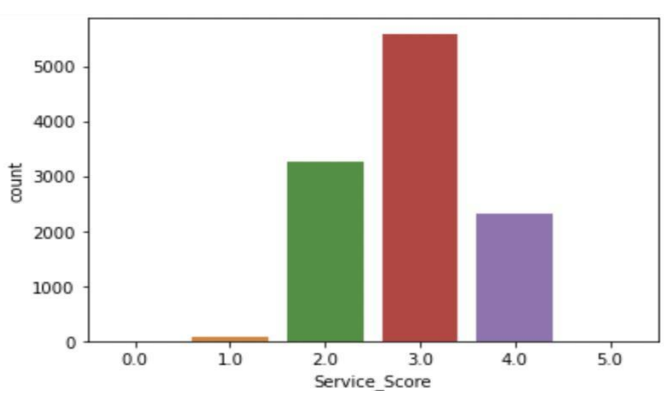

Distribution of Categorical Values
Percentage/100%
0.0
0.1
0.2
0.3
0.4
0.5
0.6
Male
Female
Gender

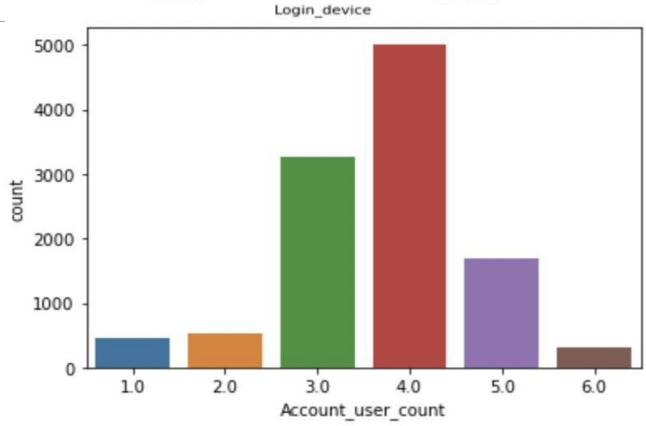

Distribution of Categorical Values
Percentage/100%
0.00
0.05
0.10
0.15
0.20
0.25
0.30
0.35
0.40
Debit Card
Credit Card
E wallet
Cash on Delivery
UPI
Payment

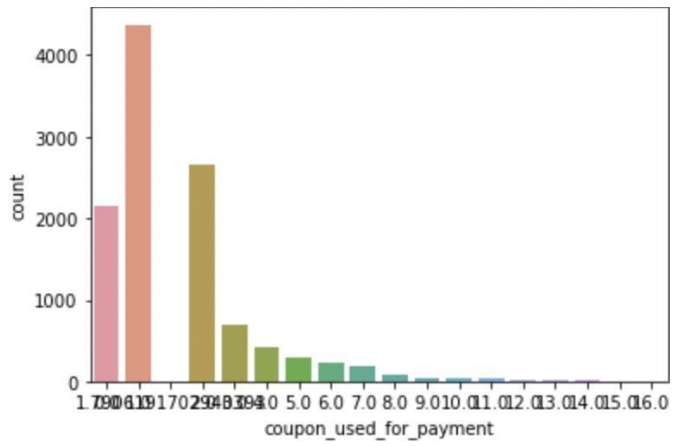

Distribution of Categorical Values
Percentage/100%
0.00
0.05
0.10
0.15
0.20
0.25
0.30
0.35
Regular +
Super
HNI
Super +
Regular
account_segment

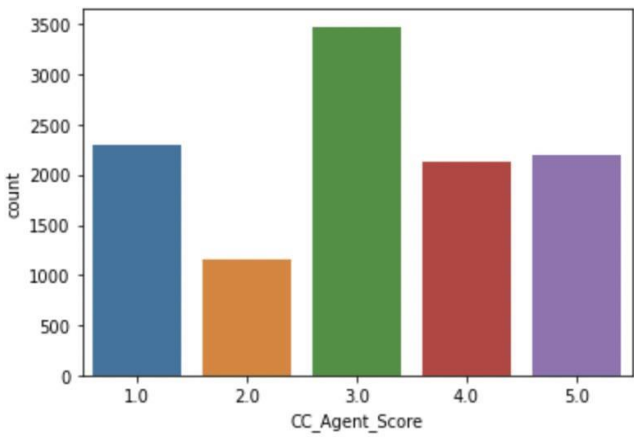

Distribution of Categorical Values
Percentage/100%
0.0
0.1
0.2
0.3
0.4
0.5
Married
Single
Divorced
Marital_Status

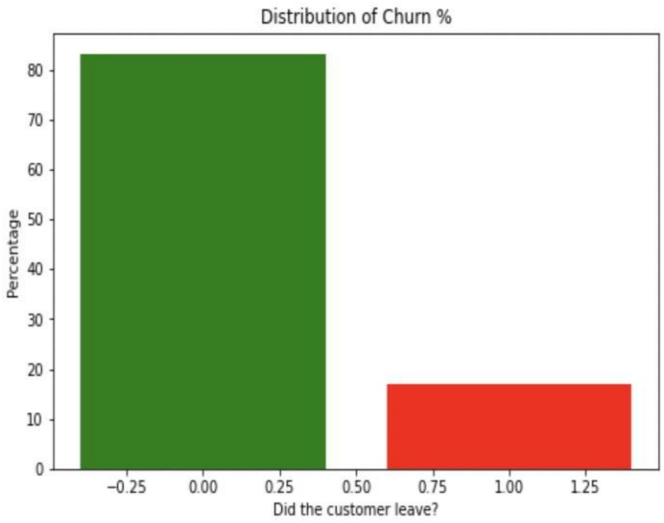

Distribution of Categorical Values
Percentage/100%
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
Mobile
Computer
Login_device


count
0
1000
2000
3000
4000
5000
0.0
1.0
2.0
3.0
4.0
5.0
Service_Score


count
0
1000
2000
3000
4000
5000
1.0
2.0
3.0
4.0
5.0
6.0
Account_user_count







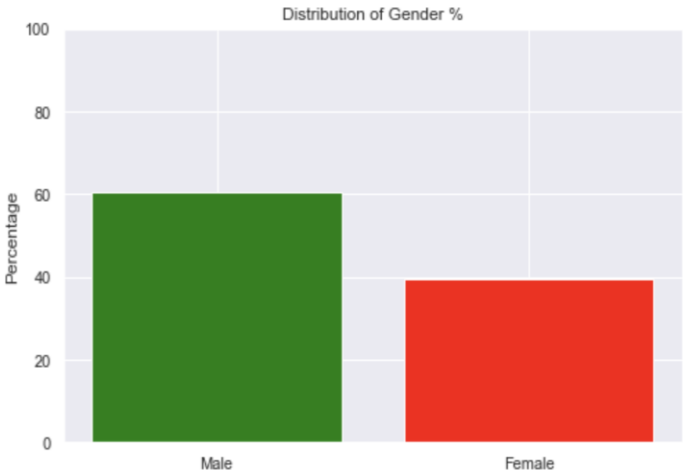
Observation: This is an imbalanced dataset with target variable containing 16.8% churn. The prediction will be for minority class.

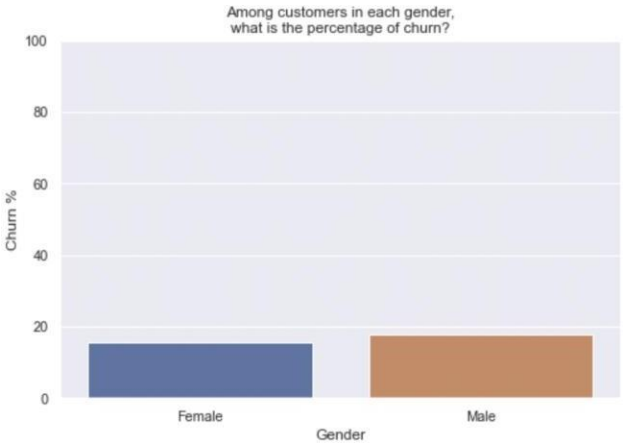
Observations:

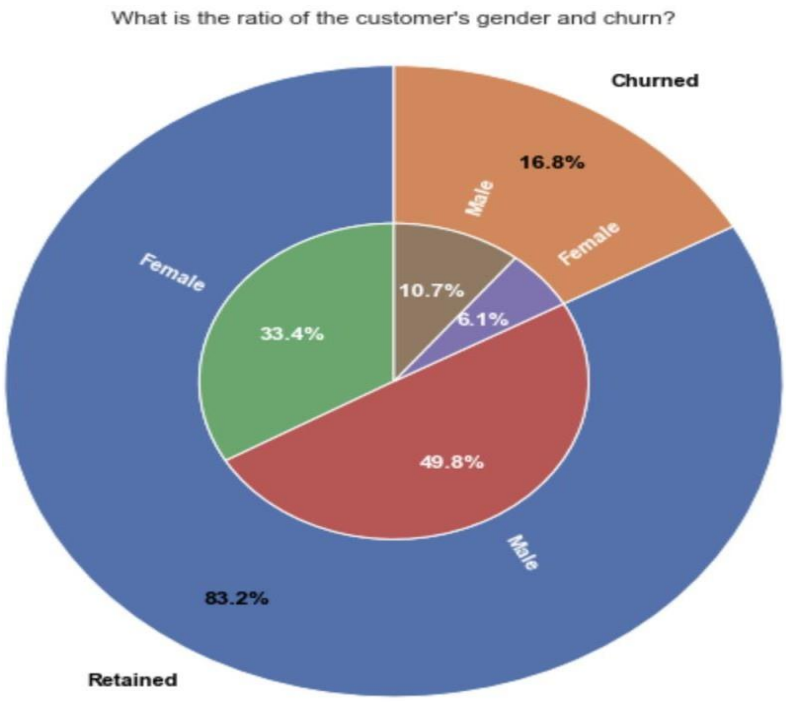
- This is an imbalanced dataset with target variable containing 16.8% churn
- Tier 1 cities have more accounts followed by Tier 3 cities
- Most of the accounts pay through debit card followed by credit card. UPI ranks last amongst payment methods
- Number of male account holders outnumber females
- Regular plus and Super are top two account segment types by number
- Top score for both Customer service agent and Service score is 3
- Married customers have the most accounts followed by single
- Most accounts do not have a customer complaint filed last year
- Most account holders use Mobile for logging and using services

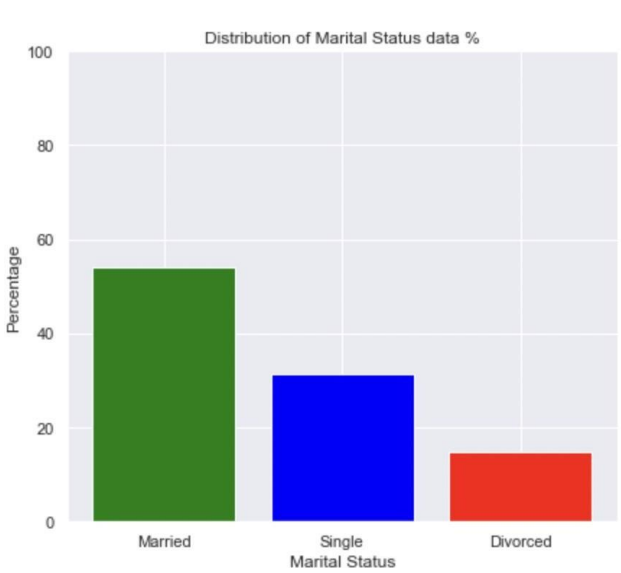
3.2. Bivariate analysis & Multivariate analysis

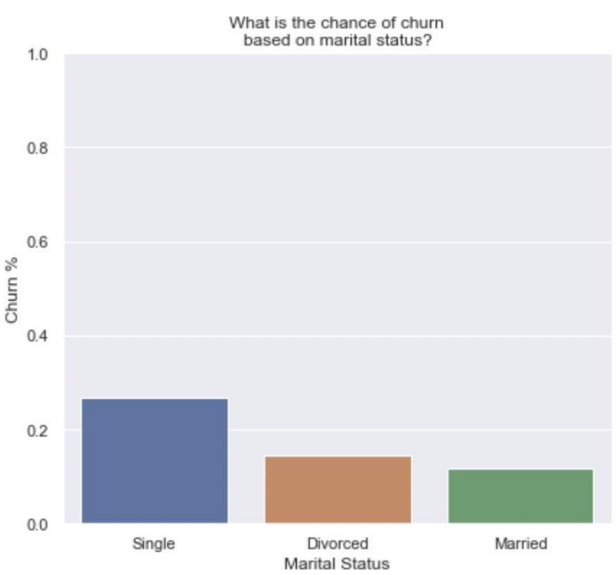
Bivariate analysis shows the relationship between two variables. Here, the predictor variables have been taken and their relationship with the target variable has been plotted. The influence of the predictor variables on the target variable can be observed in these bivariate plots.

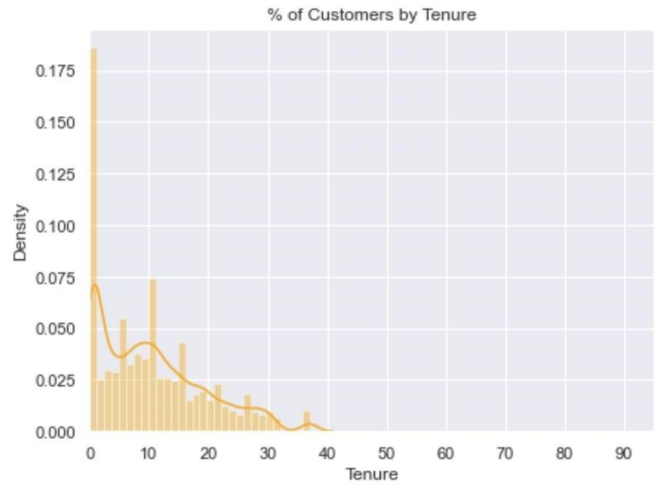

Distribution of Gender %
Percentage
100
80
60
40
20
0
Male
Female

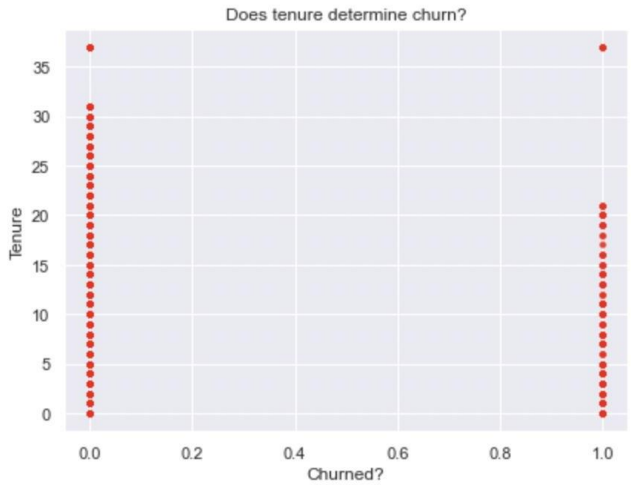

Among customers in each gender,
what is the percentage of churn?
Churn %
100
80
60
40
20
0
Female
Male
Gender

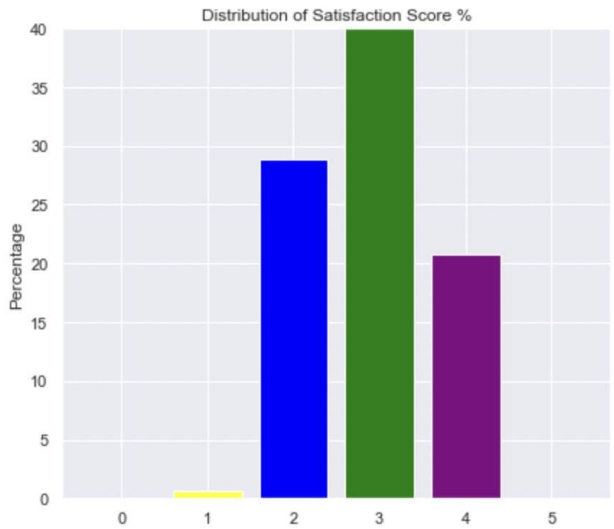

What is the ratio of the customer's gender and churn?
Churned
16.8%
Male
Female
10.7%
6.1%
33.4%
49.8%
Female
Male
83.2%
Retained

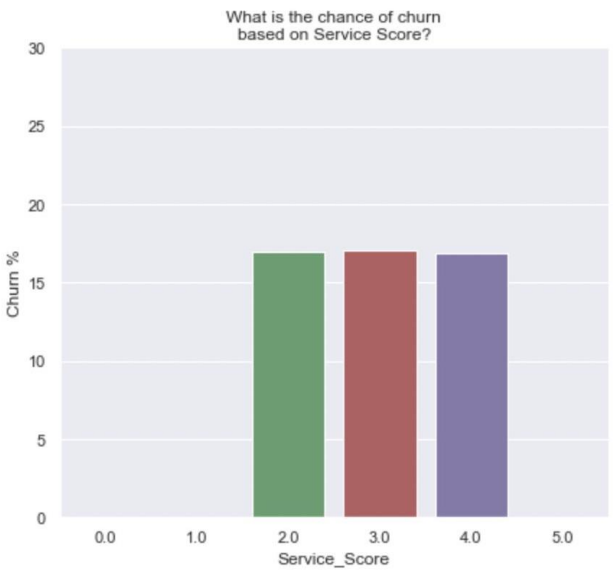

Distribution of Marital Status data %
Percentage
100
80
60
40
20
0
Married
Single
Divorced
Marital Status

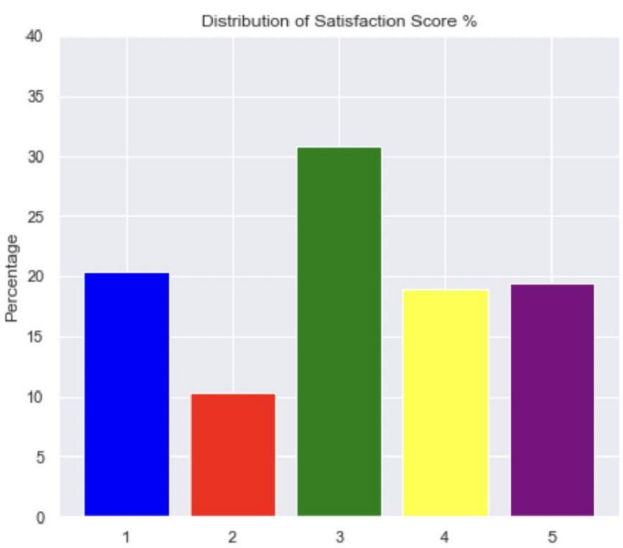

What is the chance of churn
based on marital status?
Churn %
1.0
0.8
0.6
0.4
0.2
0.0
Single
Divorced
Married
Marital Status

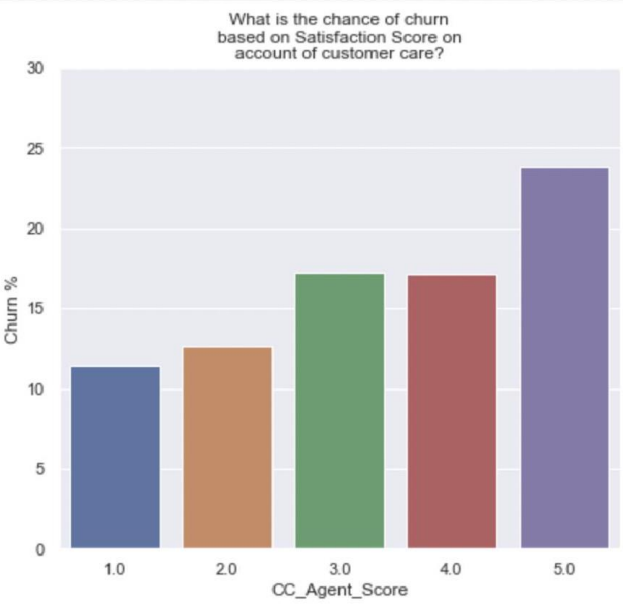

% of Customers by Tenure
Density
Tenure

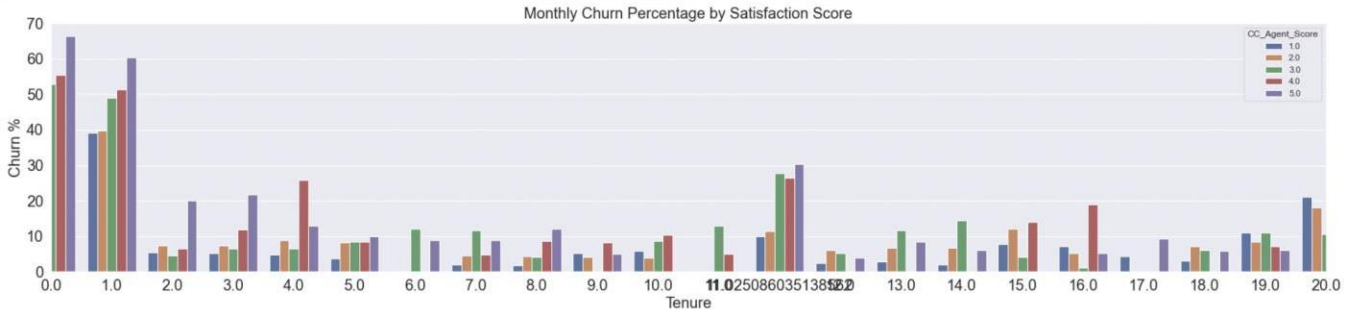

Does tenure determine churn?
Tenure
Churned?

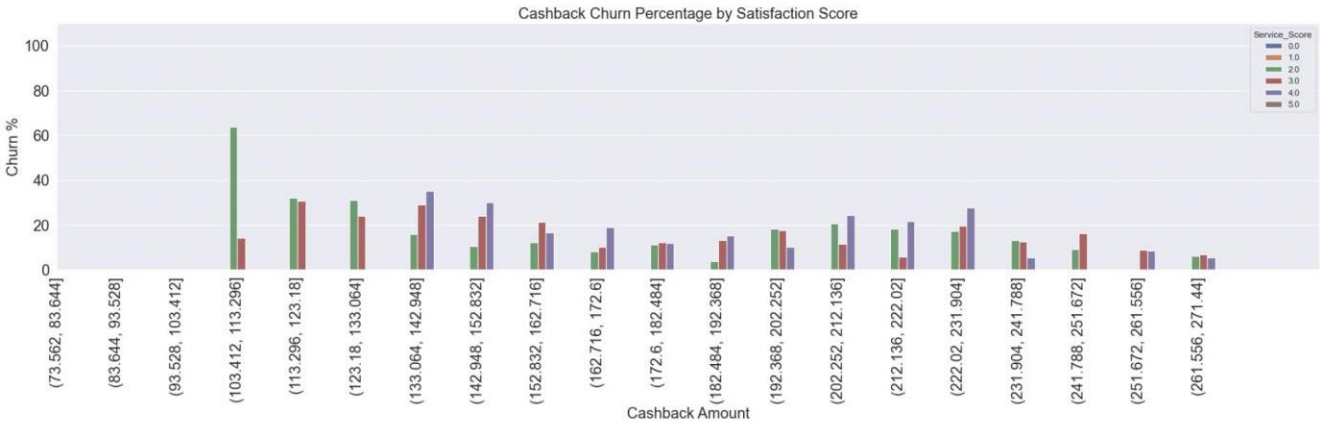

Distribution of Satisfaction Score %
Percentage


What is the chance of churn based on Service Score?
Churn %
Service_Score

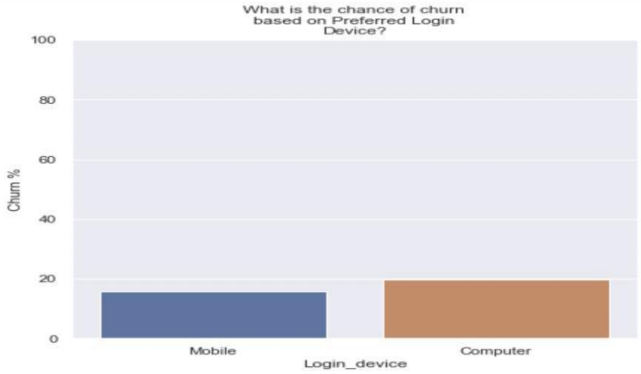

Distribution of Satisfaction Score %
Percentage

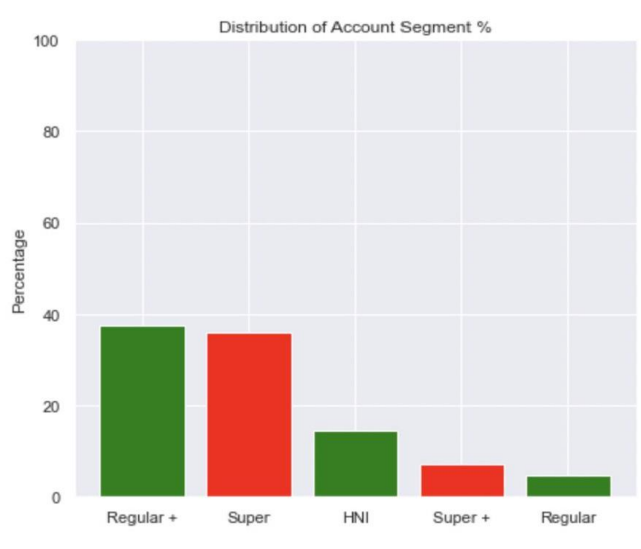

What is the chance of churn based on Satisfaction Score on account of customer care?
Churn %
CC_Agent_Score

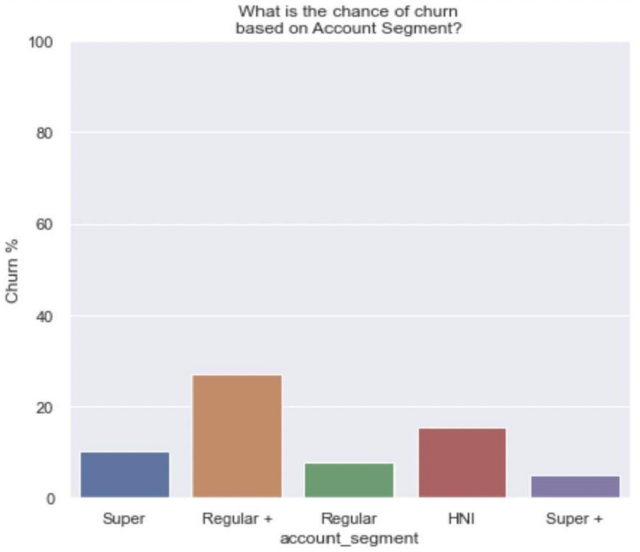

Monthly Churn Percentage by Satisfaction Score
Churn %
Tenure
CC_Agent_Score
1.0
2.0
3.0
4.0
5.0


Cashback Churn Percentage by Satisfaction Score
Churn %
Cashback Amount
Service_Score
0.0
1.0
2.0
3.0
4.0
5.0
(73.562, 83.644]
(83.644, 93.528]
(93.528, 103.412]
(103.412, 113.296]
(113.296, 123.18]
(123.18, 133.064]
(133.064, 142.948]
(142.948, 152.832]
(152.832, 162.716]
(162.716, 172.6]
(172.6, 182.484]
(182.484, 192.368]
(192.368, 202.252]
(202.252, 212.136]
(212.136, 222.02]
(222.02, 231.904]
(231.904, 241.788]
(241.788, 251.672]
(251.672, 261.556]
(261.556, 271.44]

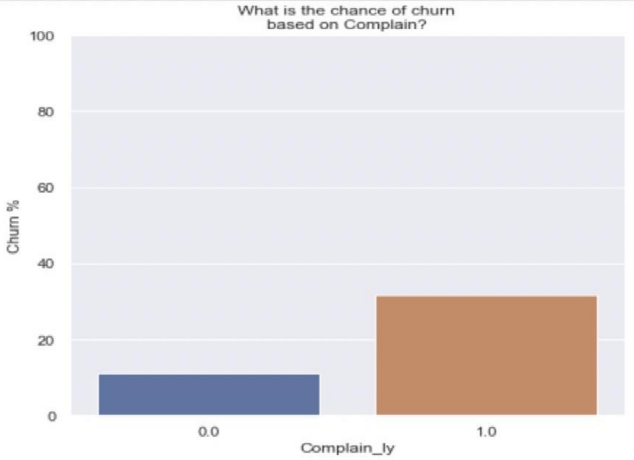

Distribution of Preferred Login Device %
Percentage
Mobile
Computer

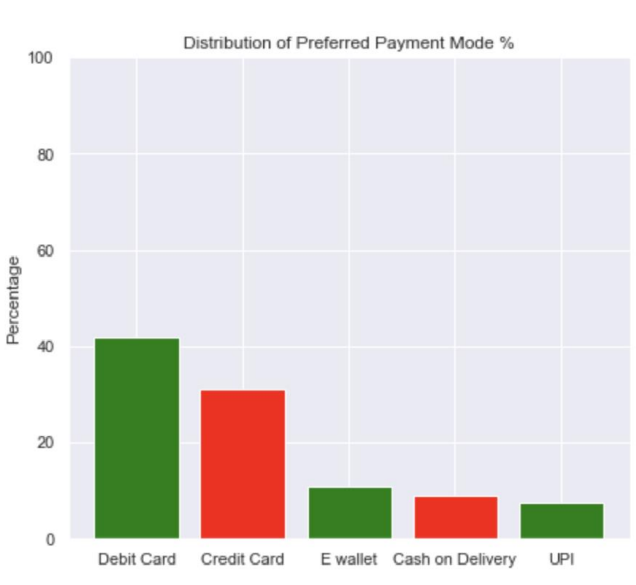

What is the chance of churn based on Preferred Login Device?
Churn %
Mobile
Computer
Login_device

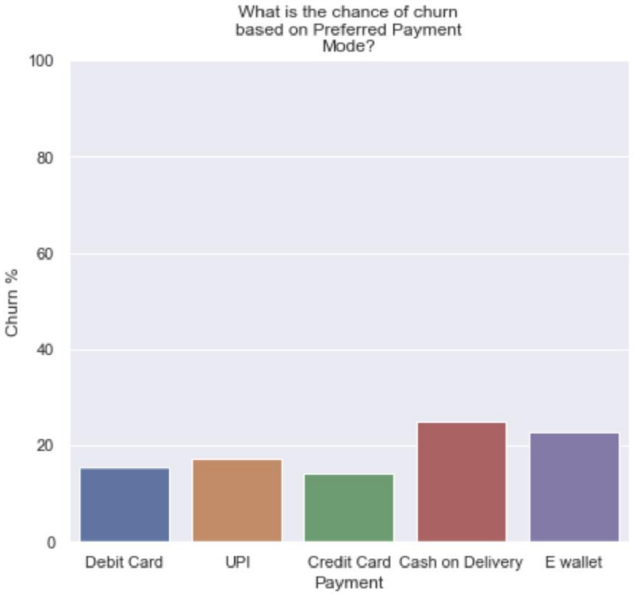

Distribution of Account Segment %
Percentage
100
80
60
40
20
0
Regular +
Super
HNI
Super +
Regular

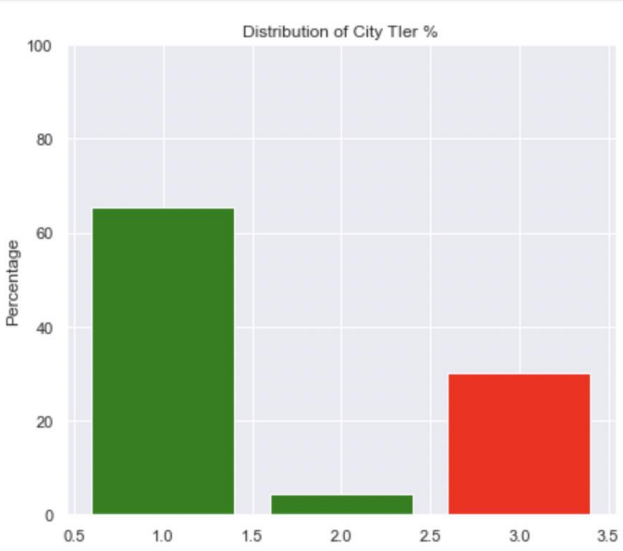

What is the chance of churn based on Account Segment?
Churn %
100
80
60
40
20
0
Super
Regular +
Regular
HNI
Super +
account_segment

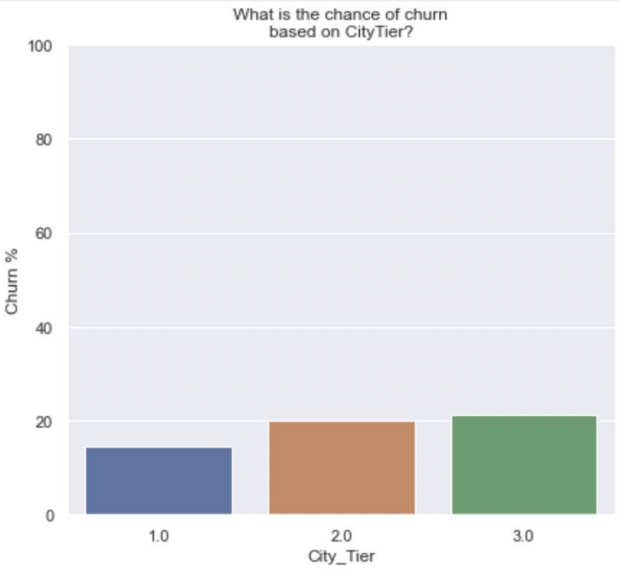

Distribution of Complain %
Percentage
100
80
60
40
20
0
−0.25
0.00
0.25
0.50
0.75
1.00
1.25


What is the chance of churn based on Complain?
Churn %
100
80
60
40
20
0
0.0
1.0
Complain_ly

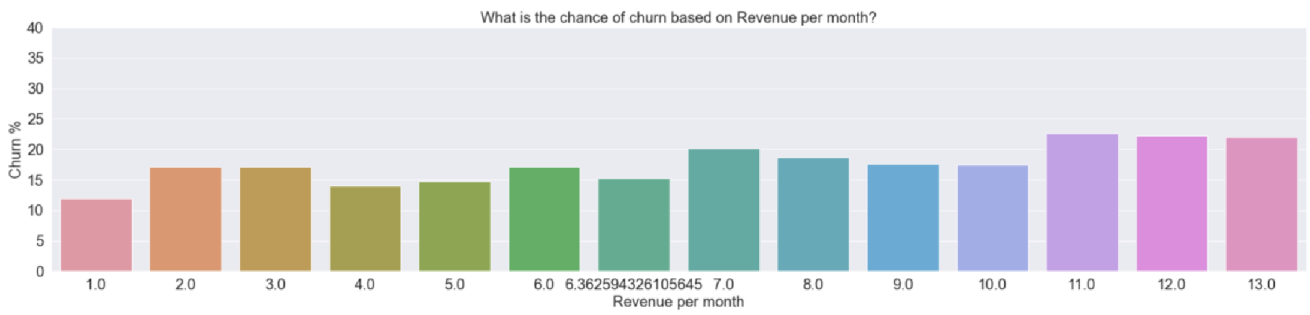

Distribution of Preferred Payment Mode %
Percentage
100
80
60
40
20
0
Debit Card
Credit Card
E wallet
Cash on Delivery
UPI

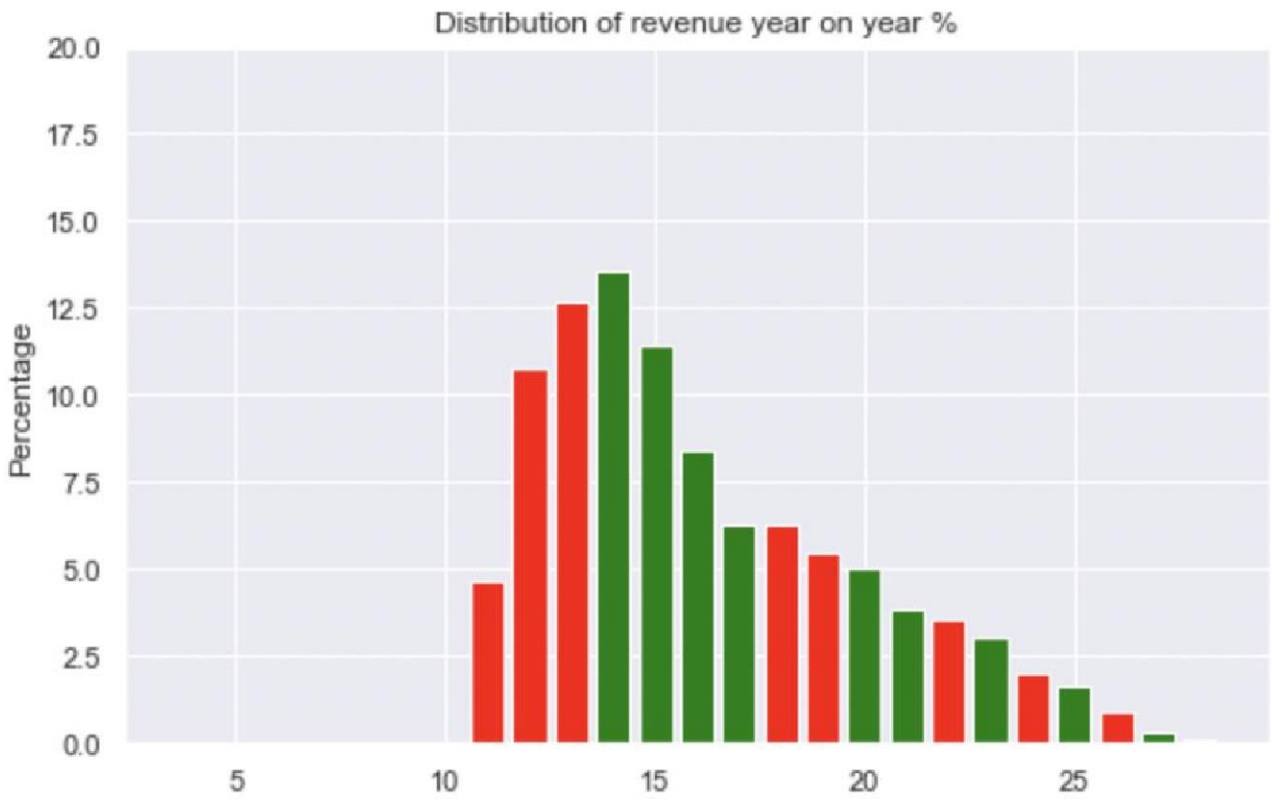

What is the chance of churn based on Preferred Payment Mode?
Churn %
100
80
60
40
20
0
Debit Card
UPI
Credit Card
Cash on Delivery
E wallet
Payment

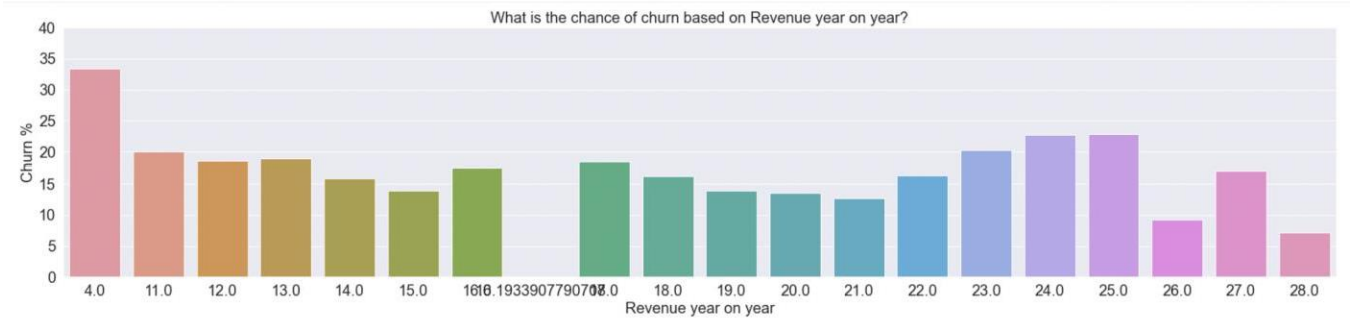

Distribution of City TIer %
Percentage
100
80
60
40
20
0
0.5
1.0
1.5
2.0
2.5
3.0
3.5

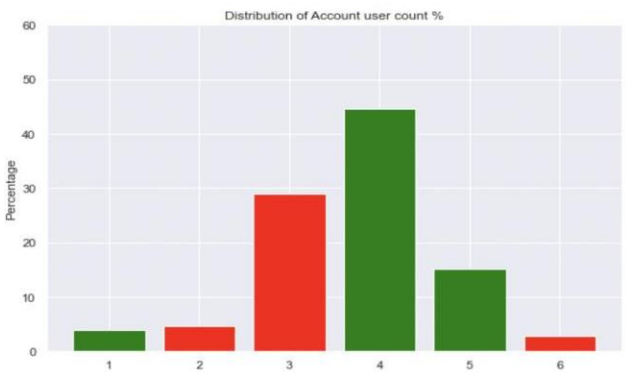

What is the chance of churn based on CityTier?
Churn %
100
80
60
40
20
0
1.0
2.0
3.0
City_Tier

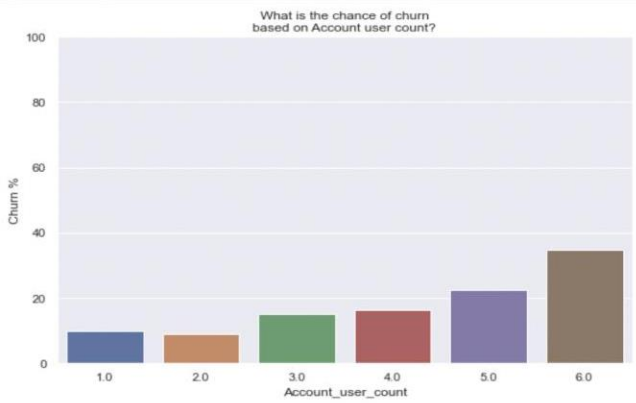

Distribution of revenue per month %
Percentage
20.0
17.5
15.0
12.5
10.0
7.5
5.0
2.5
0.0
0
2
4
6
8
10
12
14

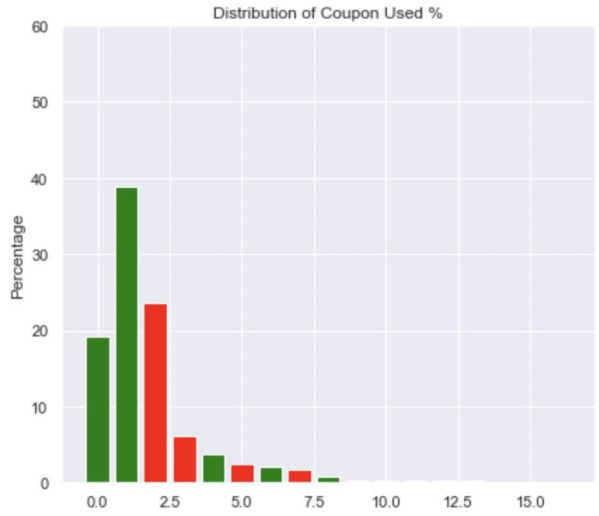

What is the chance of churn based on Revenue per month?
Churn %
40
35
30
25
20
15
10
5
0
1.0
2.0
3.0
4.0
5.0
6.0
6.362594326105645
7.0
8.0
9.0
10.0
11.0
12.0
13.0
Revenue per month

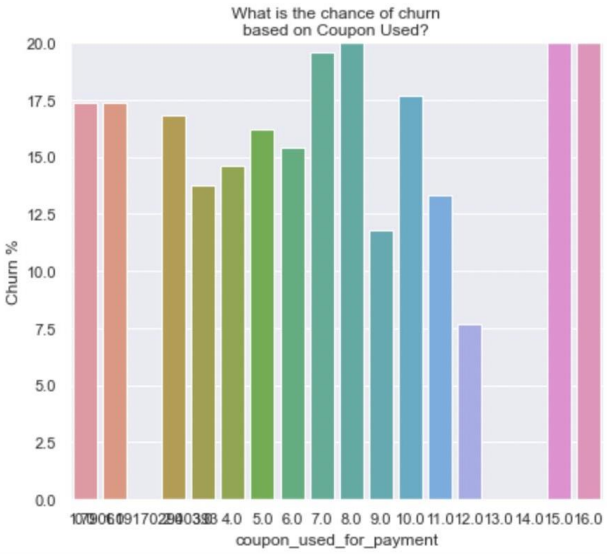

Distribution of revenue year on year %
Percentage
20.0
17.5
15.0
12.5
10.0
7.5
5.0
2.5
0.0
5
10
15
20
25

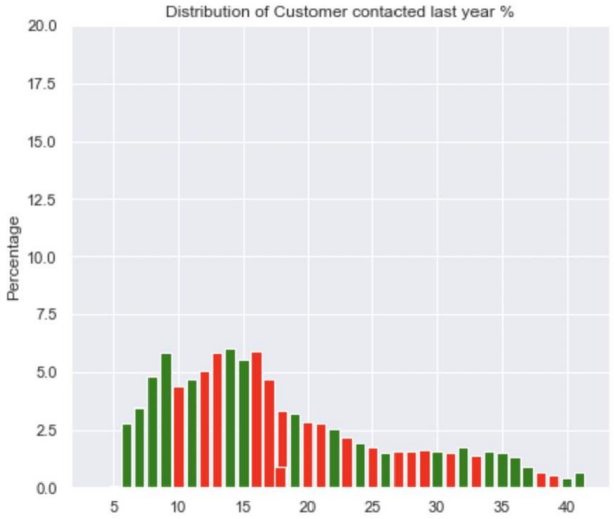

What is the chance of churn based on Revenue year on year?
Churn %
Revenue year on year
4.0
11.0
12.0
13.0
14.0
15.0
18.0
19.0
20.0
21.0
22.0
23.0
24.0
25.0
26.0
27.0
28.0

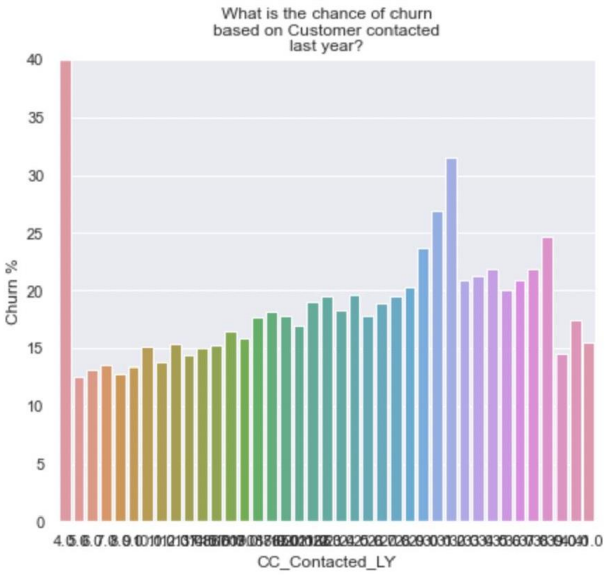

Distribution of Account user count %
Percentage
1
2
3
4
5
6

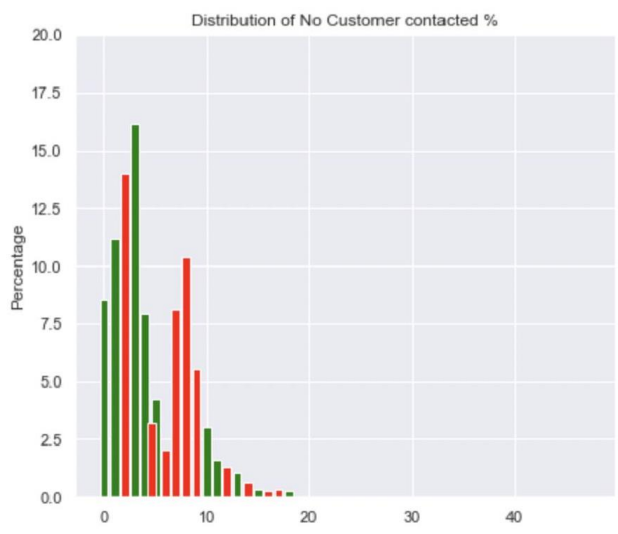

What is the chance of churn based on Account user count?
Churn %
Account_user_count
1.0
2.0
3.0
4.0
5.0
6.0

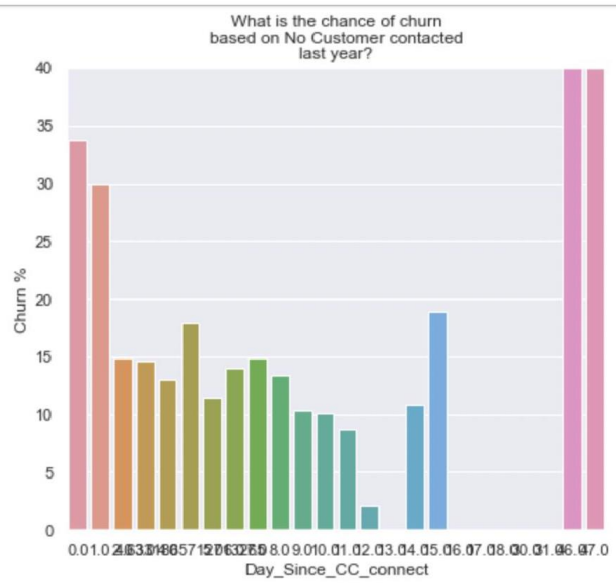

Distribution of Coupon Used %
Percentage
0
10
20
30
40
50
60
0.0
2.5
5.0
7.5
10.0
12.5
15.0


What is the chance of churn based on Coupon Used?
Churn %
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0
3.0 4.0 5.0 6.0 7.0 8.0 9.0 10.0 11.0 12.0 13.0 14.0 15.0 16.0
coupon_used_for_payment

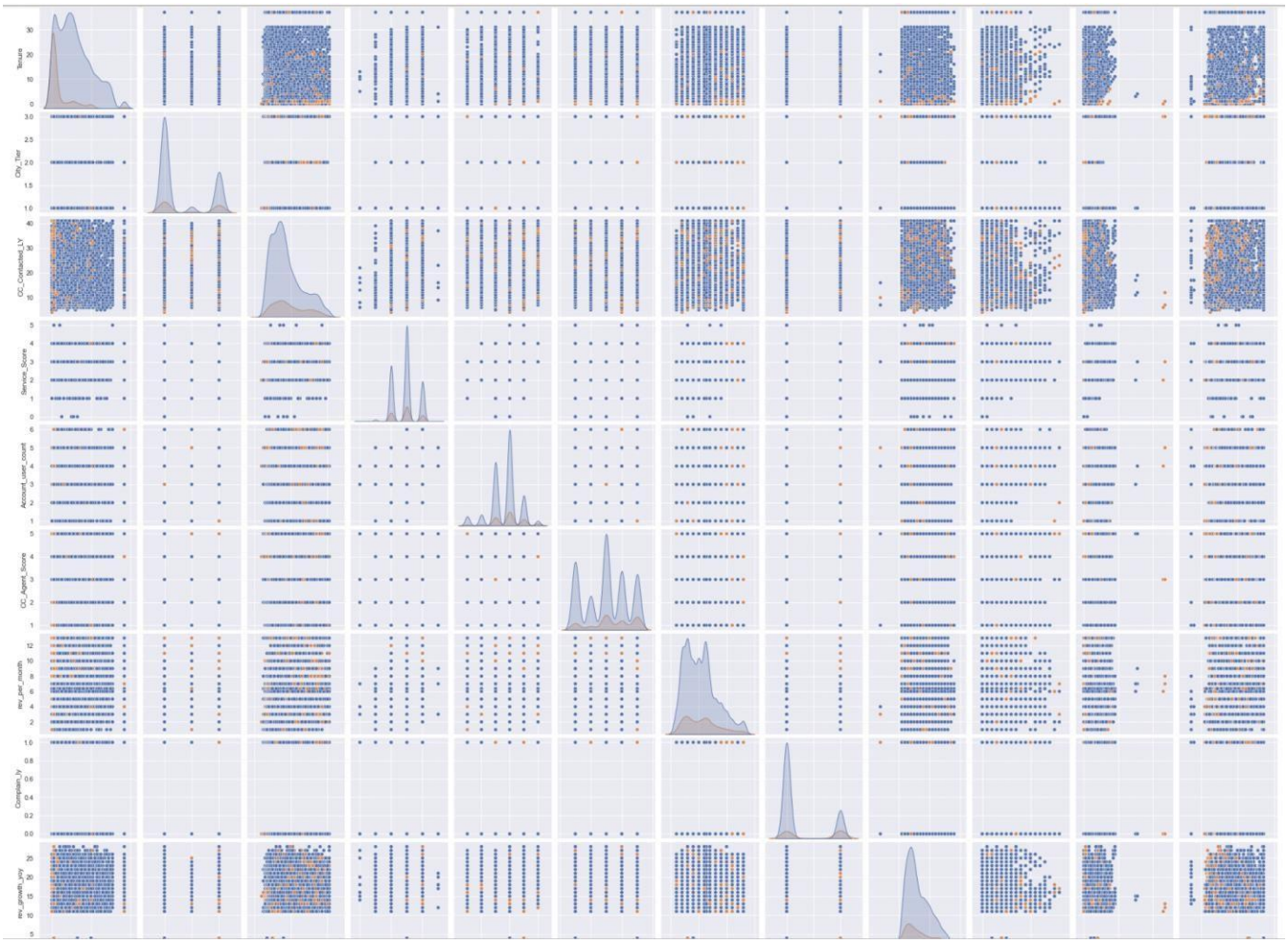

Distribution of Customer contacted last year %
Percentage
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0
5
10
15
20
25
30
35
40


What is the chance of churn based on Customer contacted last year?
Churn %
0
5
10
15
20
25
30
35
40
CC_Contacted_LY


Distribution of No Customer contacted %
Percentage
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0
0
10
20
30
40


What is the chance of churn based on No Customer contacted last year?
Churn %
0
5
10
15
20
25
30
35
40
Day_Since_CC_connect

Pair plot with hue set as target variable



Observations:

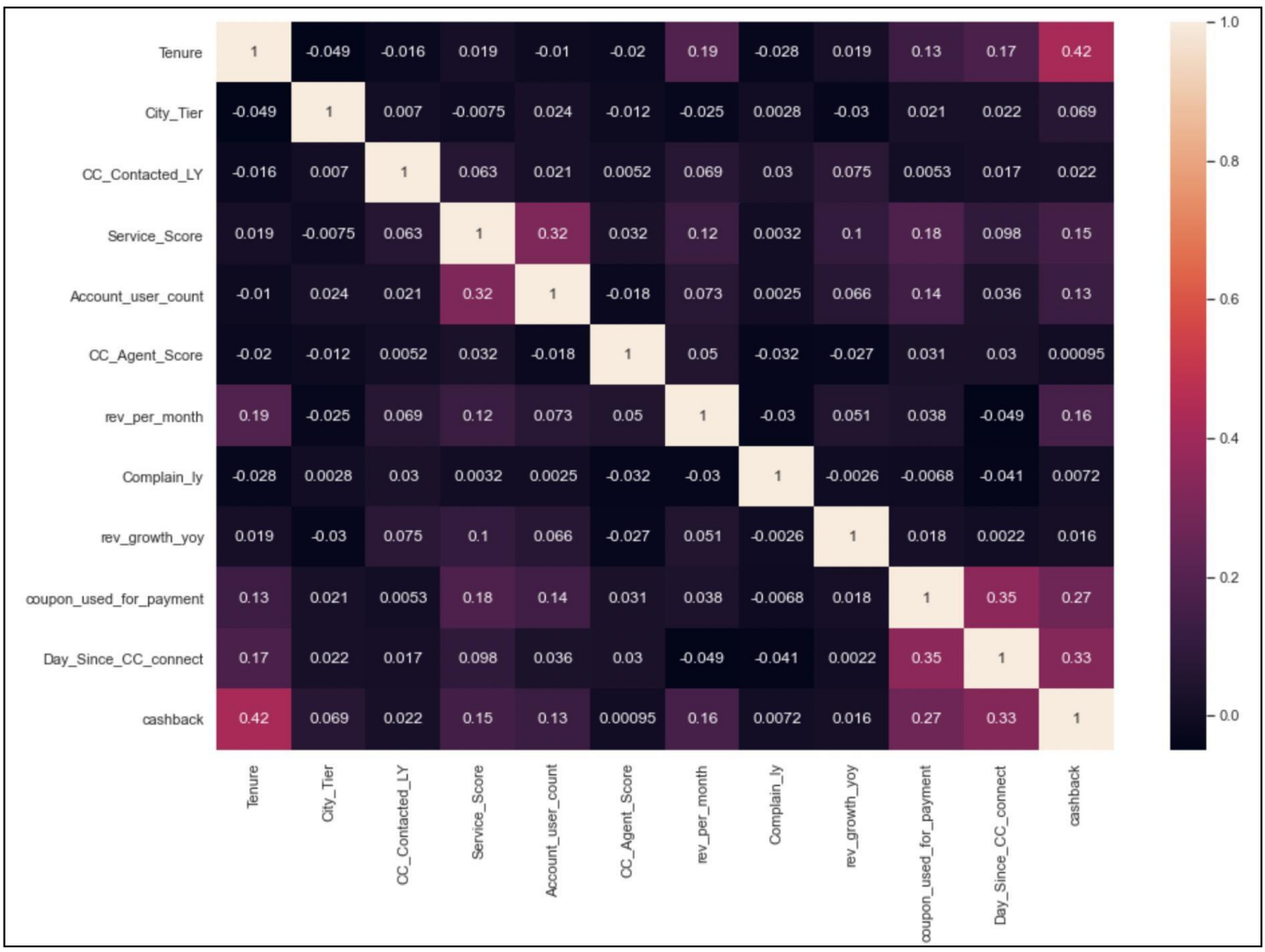
• Some of the variables clearly show presence of some clusters for e.g., rev_per_month and Day_Since_CC_connect

• The diagonal kde plot for Tenure shows a slight separation with customers who churned falling on the lower side of Tenure

• There is no linear relationship between any two continuous variables

Correlation heat map

The following heat map shows the correlation (pearson's) between various numeric predictors in the dataset. The purpose of doing a correlation matrix is to check if any of the variables have a strong correlation that can contribute to multi-collinearity. If there is a strong correlation > 0.80 between two variables, one of them can be dropped as it would not add much to the model's performance.



		correlation
cashback	Tenure	0.422706
Day_Since_CC_connect	coupon_used_for_payment	0.354672
cashback	Day_Since_CC_connect	0.331911
Service_Score	Account_user_count	0.315423

Observation: None of the numeric variables show a strong correlation. Hence there is no need to drop any variable.

3.3 Missing value treatment

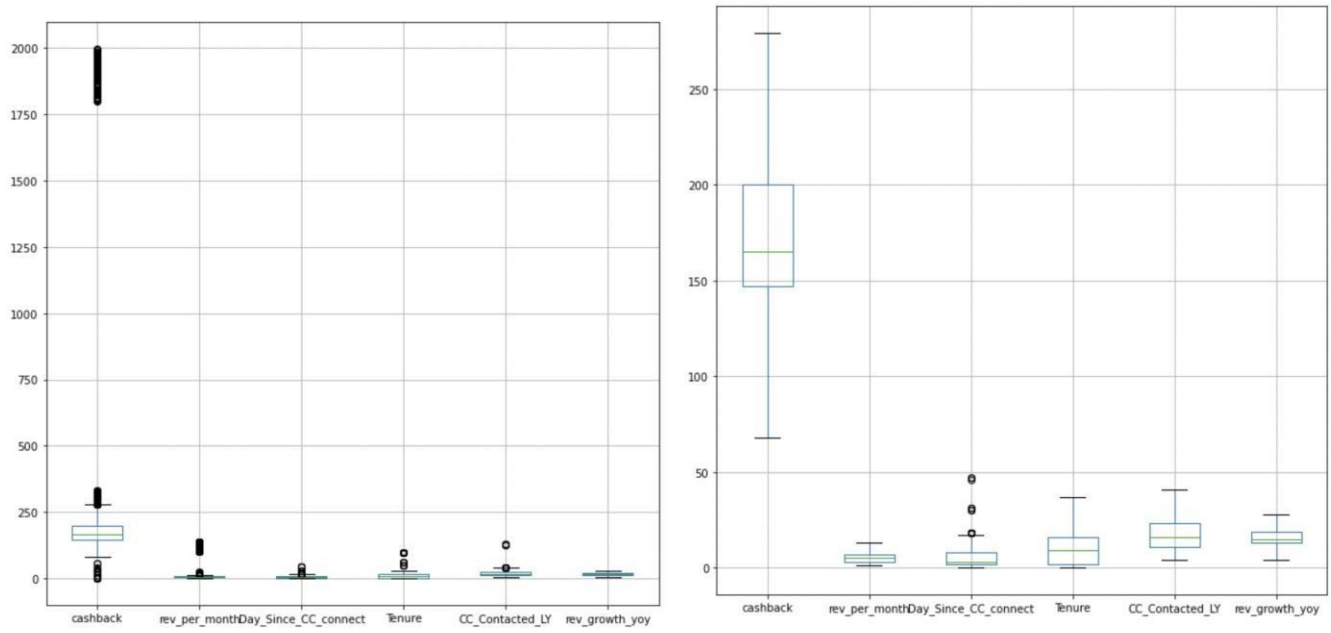
Percentage of nulls
Percentage nulls or missing values present in the predictor variables of the dataset are as follows:

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

Missing value treatment has been done using mean and mode.
For all the categorical variable missing value treatment has been done using mode. And for numeric missing value treatment has been done using mean.

```
Churn                          0
Tenure                         0
City_Tier                      0
CC_Contacted_LY                0
Payment                        0
Gender                         0
Service_Score                  0
Account_user_count             0
account_segment                0
CC_Agent_Score                 0
Marital_Status                 0
rev_per_month                  0
Complain_ly                    0
rev_growth_yoy                 0
coupon_used_for_payment        0
Day_Since_CC_connect           0
cashback                       0
Login_device                   0
dtype: int64
```

3.4 Outlier Treatment:



o Some outliers for certain variables are closer to the whisker, whereas there are a group of outliers that are far beyond the whisker with no in between values. For instance, rev_per_month has a huge space between 30 and 100 indicating absence of values in that range. Those outliers in the extreme values do not correlate with corresponding outliers in cashback field. We cannot rule out these outliers as incorrect values, they may belong to hotels with many rooms. But models like logistic regression are sensitive to outliers and may not give good performance if outliers are left untreated.
o Hence, two approaches to modelling were performed – one set of data with outliers treated for outlier sensitive models and other set of data with outliers not treated (left as-is) for outlier resistant algorithms such as Random-forest and during tuning/trials of other algorithms.
o Coupon_used_for_payment has a very limited range 0 to 16. Hence, for the purpose of this analysis, the outliers will not be treated (similar to categorical variables).
For the outlier treated dataset, outliers beyond upper and lower whiskers were treated by capping to the lower and upper range where
o lower_range= 1st quartile - (1.5 * IQR) and
o upper_range= 3rd quartile + (1.5 * IQR)
Where, IQR = 3rd quartile – 1st quartile value

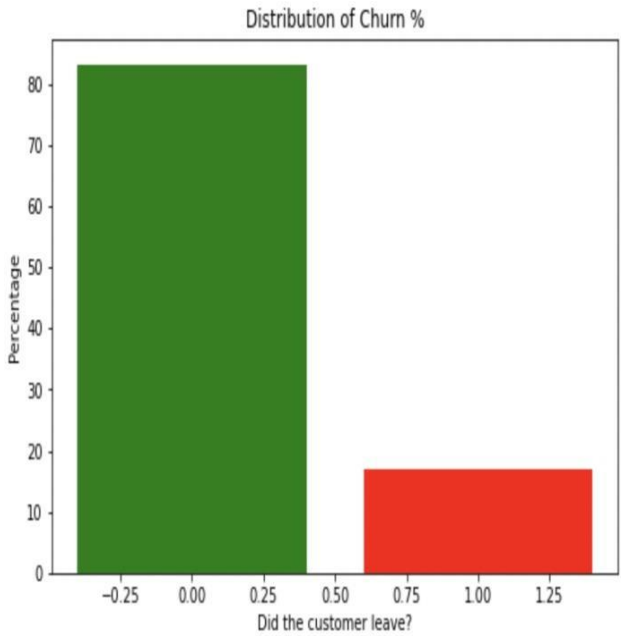
3.5 Addition of new variables

Cluster code has been added to the dataset (details about clustering is provided in section 3.5). It may be used when experimenting with model building.

4. Business Insights

4.1 Data imbalance

As can be seen from the plot below, the data is imbalanced with respect to the target variable which is the indicator whether customer has churned or not. The distribution given below shows that for every 100 customers acquired by the business, 17 have churned and 83 customers are active. This distribution is skewed towards active/current customers. The objective for this exercise is to be able to predict the customers who would churn i.e., the minority class '1'.



If a dataset has equal distribution amongst the categorical values taken by the target variable (Churn), it would be called a balanced dataset. In a balanced dataset, the model learns to predict both classes with equal efficiency as there are equal number of observations.
In case of customer churn, any business would have less of churned customers and more of active customers. Similarly, this dataset also has close to 17% churned customers. Although this mimics real-life churn data, from a modelling perspective, this may pose a challenge.

Challenges posed by an imbalanced dataset:

• If there are no sufficient observations in the minority class, the model would be unable to learn the patterns in minority class (class of interest) well and may predict majority class better. Using resampling techniques such as oversampling minority class or under-sampling majority class may be beneficial. Compared to under-sampling majority class which would result in loss of data, oversampling using techniques like SMOTE (Synthetic Minority Oversampling) may help. This technique would help generate synthetic data for churned customers based on the existing churned customer observations. However, this technique may not always guarantee better model performance. Depending on the algorithm used and the type of SMOTE used, there could be overfitting in the train dataset.
• Accuracy as an evaluation metric would not be appropriate in imbalanced classification problems. Even if the algorithm predicts all customers as belonging to majority class, it would still result in an

accuracy of 83.2%. Using Precision, Recall or F1-score for the minority class may be a better approach for evaluation.
In this problem, SMOTE would be one of the data treatments given. Models built on imbalanced data would be compared against models built on SMOTE balanced dataset and the best performance measure yielding data would be chosen. It may be possible that SMOTE does not give a good performance, hence a comparison would be required to make that decision.

4.2 Business Insights from EDA

• **Customer feedback:** 78% of customers have rated service as 3 or less than 3 (out of a scale of 5). Likewise, 61% of customers have rated customer care agents a score of 3 or less than 3 (out of a scale of 5). This indicates that the customer feedback is pointing to dissatisfaction or bare satisfaction in service and also in customer care engagement.
• **Relationship between Tenure and Churn:** In the bivariate histogram for Tenure vs Churn, it can be seen that the churn is very high for low tenures. For tenure between 0 and 1, around 51.85% customers have churned. The reason for this churn needs to be drilled down and addressed. Once the tenure increases, the histogram shows that proportion of churn decreases.
• **Relationship between Account segment and Churn:**
More customers in Regular Plus plan seem to churn. A comparison of this plan and competitors plan with same features and for same pricing range could be done to ascertain if the plan or pricing needs to be changed. Also, customer feedback for customers on this plan could be obtained and analysed to find out why there is more churn in this account segment.
• **Relationship between Monthly revenue and Churn:**
The % churn in high revenue customers is slightly more than the churn in lower revenue customers. This is would a cause for concern for the DTH provider that not only is there more churn but more high revenue customers are churning.
• **Relationship between Days since customer care connect and Churn**
Days since customer care connect for churned customers is lesser than for active customers. This shows that churn has happened shortly after the customers have contacted customer care.
• **Relationship between Customer care contacted last year and Churn**
The number of times customer care was contacted previous year was more in churned customers compared to active customers.
• **Relationship between Complaints made last year and Churn**
Proportion of customers who complained is significantly more in churned customers compared to active customers.
• **Relationship between User count and Churn**
Proportion of accounts with user counts 5 and 6 is more in churned customers compared to active customers.
• **Relationship between Payment type and Churn**

Proportion of customers who have paid through E-wallet and Cash on delivery is more within churned customers compared to active customers.

• **Relationship between City tier and Churn**

Proportion of customers who reside in Tier 3 cities is more within churned customers compared to active customers. Whether there is more competition in those cities compared to Tier-1 cities needs to be explored by the DTH provider.

• **Relationship between Marital status and Churn**

Single customers have churned more compared to married or divorced customers.

• **Relationship between Gender and Churn:**

Above is a count plot shows the overall distribution of how many males and females are present in the dataset.

- Also from the above visualization we can conclude than there is overall churn of male and female is 16.8% and 83.2% customers have not churned.
- In that 10.7% of male and 6.1% of female has been churned.
- And in not churned around 49.8% male has not churned and 33.4% female has not churned.
- Also from the above plot the percentage of likely to churn in female is 17% and in male is 19%

4.3. Clustering

• Cluster analysis or clustering is the task of grouping a set of objects in such a way that objects in the same group (called a cluster) are more similar to each other than to those in other groups (clusters).

• Clustering is an unsupervised task and given all the features in the dataset, the clustering algorithm is allowed to group customers such that each group has similar customers and customers of different groups are dissimilar.

• For this purpose, the processed dataset (cleaned, scaled, nulls imputed, outlier treated, categorical features encoded) without the target variable was used.

• As the dataset contained both continuous and categorical predictor variables, kprototypes function from Kprototypes library was used.

• The algorithm was run for 2,3 and 4 clusters and 3 clusters seemed to have the best separation. The cluster profile was formed by grouping the observations by clusters and finding the mean for all the features.

• Although churn was not part of the features for clustering, it was added part of the cluster profile so that it is possible to appreciate how churn varies for each cluster.

Churn	Tenure	City_Tier	Service_Score	CC_Agent_Score	cashback	rev_per_month	Day_Since_CC_connect	CC_Contacted_LY	rev_growth_yoy
-0.407944	0.896023	-0.048452	0.29073	0.084537	0.954519	0.416538	0.542456	0.072997	0.115922
2.222345	-0.800134	0.186111	0.007296	0.232539	-0.38328	0.071427	-0.379061	0.184486	-0.021804
-0.449975	-0.306321	-0.02875	-0.184065	-0.126617	-0.475129	-0.283058	-0.218866	-0.104159	-0.065552

4.4 Business Insights from cluster profiling

• Even though this is an unsupervised algorithm without the use of target variable in clustering, the profiling came up with clear separation of groups only for those features that also showed a high F-statistic and high Chi square value in the bivariate analysis.
• Higher the tenure, lesser the churn. But the cluster profile also shows that for low tenures (cluster 1 and 0), this is not holding good. To further explore this, it would be good to bin Tenure and check the relationship with Churn through a stacked bar.
• High churn clusters had contacted customer care almost twice as the least churn customers
• The cashback was lower for high churn cluster compared to low churn cluster
• High churn cluster contacted customer care less times compared to low churn customers
• The cluster with maximum complaints last year also has the maximum churn. The cluster with minimum complaints last year has the minimum churn. It could be suggested that if any customer files a complaint, the complaint be followed up and resolved until the customer is satisfied. This process needs to be looked at if it can be strengthened.

5. Modelling Approach

5.1. Algorithms applicable for the given problem

• 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' – will not churn and '1' – will churn
• Since there is a target variable 'Churn' to be predicted, this is a supervised learning problem
• There are several algorithms that can be used for classification problems such as these
o Linear classification: Logistic Regression, Linear Discriminant Analysis, Naïve Bayes
o Non-linear classification algorithms: SVMs non-linear adaptations, Decision tree, K-nearest neighbor, Artificial Neural Network
o Ensemble models: Random Forest, Adaboost, Gradient Boost
• These algorithms have certain assumptions about the data on which they are fit. Depending on the nature of the data, algorithms would give good or poor performance.

5.2. Methodology

• Data was split into train and test set in the ratio 70:30. 7882 records were assigned to train dataset and 3378 records were assigned to test dataset. The selection was done in such a manner that both sets had similar distribution of target variable as in the original dataset (i.e., 16.8% churn=1s).
• 8 algorithms were chosen and for each algorithm
▪ Base model (with default hyperparameters) was constructed and evaluation on train and test datasets performed
▪ Different data was used and performance measured and recorded
▪ Hyperparameters for the algorithm were tuned using SKlearn library's GridSearchCV and also manually if required.
▪ Performance was again measured for the tuned algorithm
▪ Feature importance was extracted from the model through in-built attributes for certain models. Amongst black box models, Sklearn's Permutation feature importance was used as a wrapper function on the model to obtain feature importance.

5.3. Evaluation metrics for model comparison

• The final best model was decided based on comparison of evaluation metrics for train and test data of all the models. The following criteria was used:
o The train and test performances should be comparable – i.e., no overfit or underfit. The model should be robust and reliable for use with unknown data.
o Maximum Precision score for 1s. The problem statement states that Revenue assurance team do not want to give unnecessary freebies. If precision for 1s (churn customers) is high, then the actual churns out of the model's predicted churns would be high and hence the revenue assurance team's criteria would be satisfied.
Precision = True positive/ (True positive + False positive)
o High F1-score (to ensure Recall is also good) for Churns. 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 churns 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.
o Model should be interpretable.
o Model should not be computationally expensive (like KNN).

6. Model Building and Tuning

8 algorithms were tried for this dataset – Logistic Regression, Linear Discriminant Analysis, Support Vector Machine, Artificial Neural Network, K-Nearest Neighbours, Random Forest, XGboost, Gradient Boost. Algorithm implementations from Sklearn package was used for all algorithms. In addition to SKlearn, Logistic regression algorithm was also executed using Statsmodel package. The Appendix contains detailed information for all the 8 models including details on the base and the tuned model performance along with their interpretations.
For sake of brevity, this section contains a tabulated form of individual models (for each of the 8 algorithms) and the results from tuning exercise. The feature importances for the top 3 best performing models are also shown. Only for Gradient Boost, the selected model, all details are provided. For complete details on all the algorithms, base model and best model performance metrics, confusion matrix, classification report for train and test data, feature importance and model interpretation.

6.1. Effort for model tuning

Model performance improvement was accomplished using the following methods:
- Around 8 different algorithms were tried across linear, non-linear and ensemble methods.
- The first model for each algorithm was the base model with the default hyperparameters. Model tuning to achieve better performance was done by tuning the hyperparameters using Grid Search CV, a function offered by Sklearn to automatically try out multiple parameter options for each hyperparameter.
- The underlying data was changed (Smote resampled/ non-resampled, outlier treated/not treated) and the effect of model performance observed for some of the models.
- Ensemble methods were used (Random Forest, XGboost, Gradient Boost) and their hyperparameters were also tuned.

The following tables show all the algorithms tried, various model tuning trials done and the performance for each one of them in train and test dataset. The data treatment done is also shown as part of the table. Hyperparameters are part of the code and hence not included here. The best model for each algorithm has been highlighted in green. The model reference number can be used to locate the model in the python code.

6.1.1. Logistic Regression

Logistic regression model performs well when the data is linearly separable. It assumes that there is linearity between target and predictor variables. It is a parametric model hence it can be fast compared to KNN. It also provides coefficients that helps with model interpretability. Hence the first model tried was Logistic regression. Both SKLearn and Statsmodel (python libraries) implementations were tried for Logistic regression.

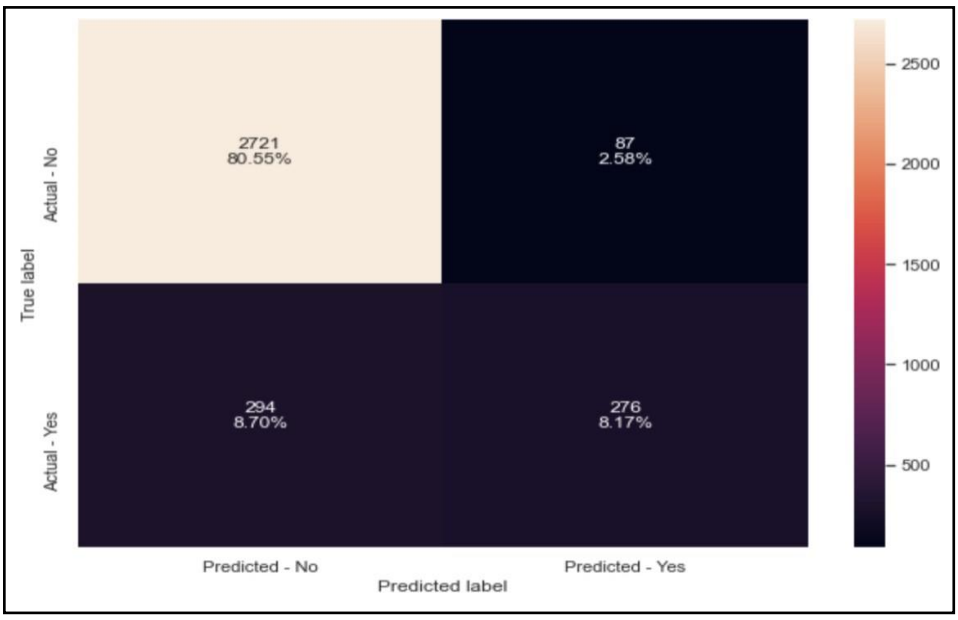
Model Refrence	Data Treatment		Hyper Paramete	Train Data				Test Data			
	Smote	Scaling		Accuracy	Precision	Recall	F1 - Score	Accuracy	Precision	Recall	F1 - Score
Logistic Regression with Default parameter	NO	Yes	Base Model	0.89	0.78	0.51	0.61	0.89	0.76	0.48	0.59
Logistic Regression with Grid Search	NO	Yes	GridSearch CV	0.89	0.77	0.51	0.61	0.89	0.75	0.48	0.59
Logistic Regression with SMOTE	YES	YES	SMOTE	0.81	0.46	0.79	0.58	0.81	0.45	0.76	0.57

Table 1 ***Model tuning and Performances - Logistic regression***

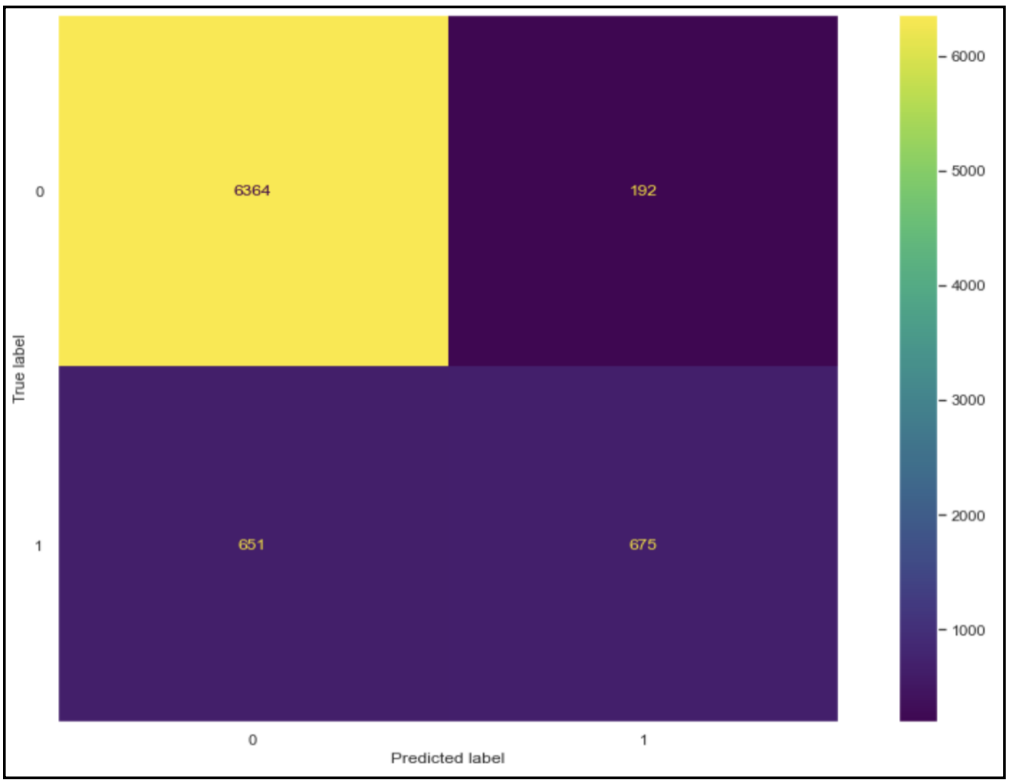
6.1.1.1. SKLearn Base model with default hyperparameters (Also the best model)

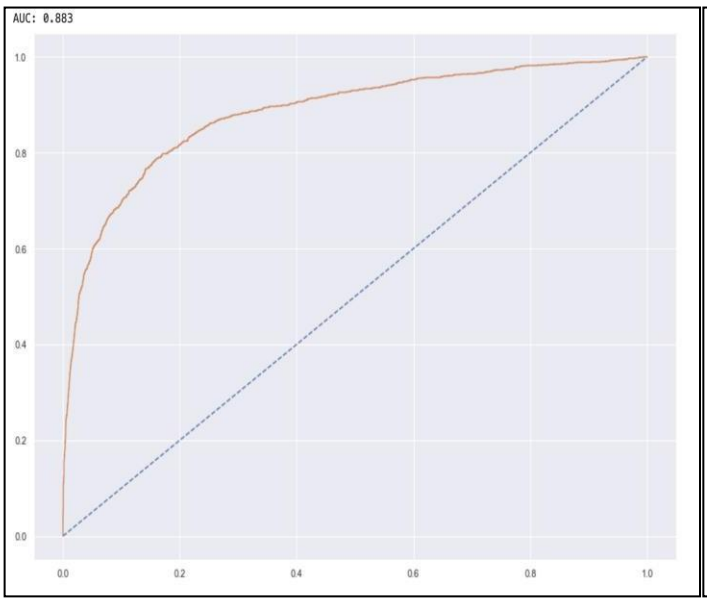
```
Accuracy on training set :  0.8930474498858157
Accuracy on test set :  0.8872113676731794
Recall on training set :  0.5090497737556561
Recall on test set :  0.4842105263157895
Precision on training set :  0.7785467128027682
Precision on test set :  0.7603305785123967
F1 on training set :  0.615595075239398
F1 on test set :  0.5916398713826366
```

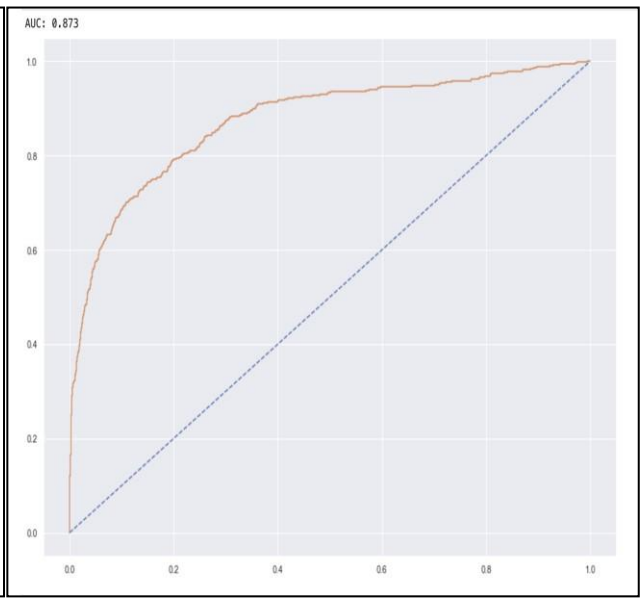
Confusion Matrix of Test data:



Confusion Matrix of Train data:







AUC of training and test set is: 0.883 and 0.873

SKLearn Logistic regression - model tuning

• GridSearchCV function from Sklearn library was used to tune the hyperparameters. It was found that tuned hyperparameters did not perform better than the base model.

• Further, Scaled data and Smote resampled data was used with the base model. It was found that Smote resulted in overfitting precision of class

• Scaling also did not improve performance compared to non-scaled data.

6.1.2. Linear Discriminant Analysis

Linear Discriminant Analysis (LDA) is also another machine learning classifier. It works well when there are linearly separable classes in data. It assumes that the underlying data has a gaussian distribution but can perform well even if assumptions are violated.

Model Refrence	Data Treatment		Hyper Parameter	Train Data				Test Data			
	Smote	Scaling		Accuracy	Precision	Recall	F1 - Score	Accuracy	Precision	Recall	F1 - Score
LDA with Default parameter	NO	Yes	Base Model	0.88	0.76	0.47	0.58	0.88	0.75	0.45	0.56
LDA with Grid Search	NO	Yes	GridSearch CV	0.89	0.76	0.46	0.58	0.88	0.75	0.46	0.56

Table 6- 2 Model tuning and Performances - LDA

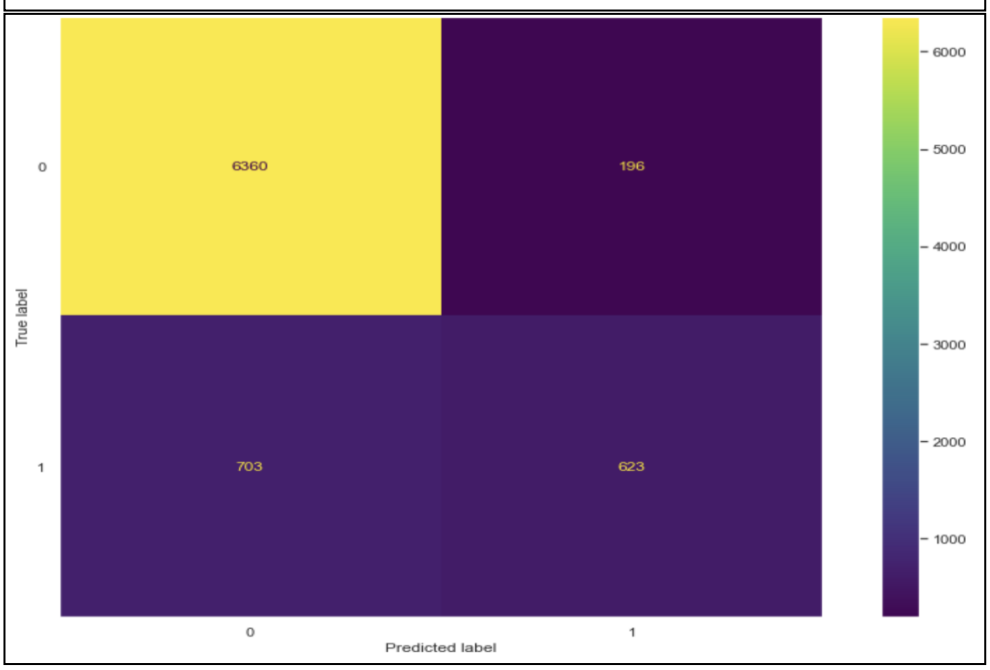
Linear Discriminant Analysis (LDA) is also another machine learning classifier. It works well when there are linearly separable classes in data. It assumes that the underlying data has a gaussian distribution but can perform well even if assumptions are violated.

- SKlearn's Linear Discriminant Analysis function was used for modelling
- The base model was run with default hyperparameters and the performance metrics noted
- The model's hyperparameters were tuned using GridSearchCV function and model constructed using the best parameters selected by GridSearchCV. This did not provide much improvement over the base model except that f1-score improved by 0.01
- Data used was outlier treated dataset
- Model performance metrics for base model and tuned model have been provided below

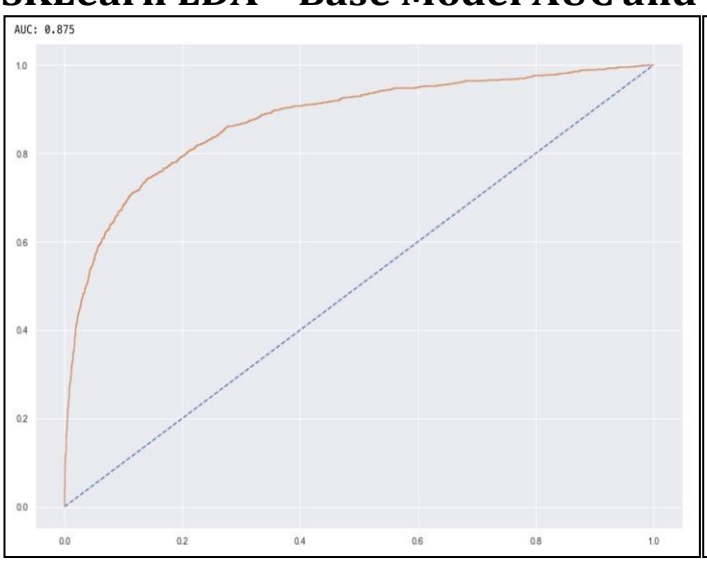
6.1.1.2. Base model of LDA with default hyperparameters (Also the best model)

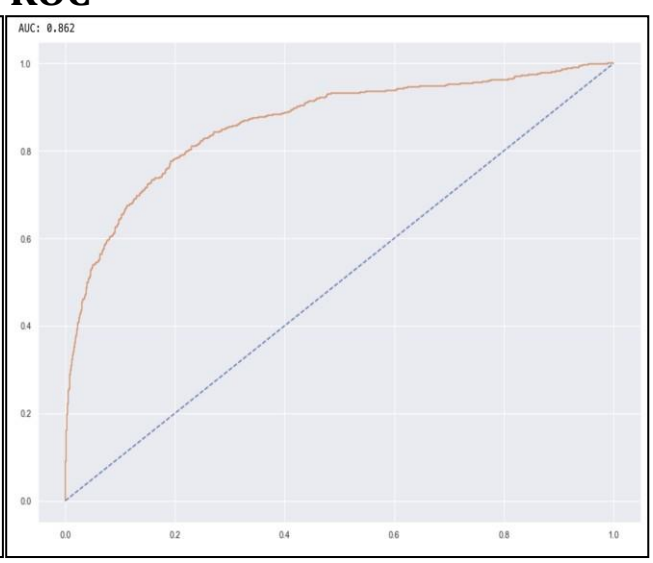
```
Accuracy on training set :  0.8859426541486932
Accuracy on test set :  0.8821788040260509
Recall on training set :  0.4698340874811463
Recall on test set :  0.45614035087719296
Precision on training set :  0.7606837606837606
Precision on test set :  0.7471264367816092
F1 on training set :  0.5808857808857808
F1 on test set :  0.5664488017429193
```

AUC training and Testing is: 0.875 and 0.862



SKLearn LDA – Base Model AUC and ROC





LDA - Model interpretation

This model's performance with respect to precision and recall of minority class is not a good one. A recall of 0.45 means out of 100 churning customers, model can only identify 45 correctly. The precision is not very high either. So other models may have to be looked at for this exercise.

```
{'Tenure': -0.12460958788987196,
 'City_Tier': 0.3474620264170837,
 'CC_Contacted_LY': 0.24385975318670486,
 'Service_Score': -0.07398000129431243,
 'Account_user_count': 0.32830155755099827,
 'CC_Agent_Score': 0.28026967557471594,
 'rev_per_month': 0.40036599994455696,
 'Complain_ly': 1.9931864237070127,
 'rev_growth_yoy': -0.018911474638469674,
 'coupon_used_for_payment': 0.08213185731521248,
 'Day_Since_CC_connect': -0.2476548627512002,
 'cashback': -0.2985642547017972,
 'Churn_1': 0.9790306779504511,
 'Payment_2': 0.6811009546525216,
 'Payment_3': -0.0985448676680153,
 'Payment_4': 0.11841062619318715,
 'Payment_5': -0.24917406397825007,
 'Gender_1': -0.6886432492863055,
 'account_segment_2': -1.127701127419583,
 'account_segment_3': -2.1660056362292965,
 'account_segment_4': -0.9330659098042144,
 'account_segment_5': 0.9408759010606401,
 'Marital_Status_2': -0.21280577328088415,
 'Marital_Status_3': -0.44039296343514106}
```

Observations:
The best model out of all LDA models has given the following coefficients. Compared to overall best model provided by Gradient boost, we can see that the features are not ranked in the same way. In fact, tenure seems to have a low-ranking coefficient although the model has established the inverse relationship it has with churn.
This is probably why LDA's performance is not comparable to other model's performance

Complaints made by customers the previous year and Single customers are the top two positive coefficients. If there was a complaint the previous year and if the customer is single, that increases the possibility of churn. Similarly, Credit card payment and Super segment customers have the highest negative coefficients. This implies that customers paying using credit card and customers falling under super segment accounts churn lesser. This model has not performed well compared to other models. Hence the interpretation may not be very useful

6.1.3 Ensemble method – Adaboost

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.

Model Refrence	Data Treatment		Hyper	Train Data				Test Data			
	Smote	Scaling	Parameter	Accuracy	Precision	Recall	F1 - Score	Accuracy	Precision	Recall	F1 - Score
ADA with Default parameter	NO	Yes	Base Model	0.89	0.73	0.59	0.65	0.9	0.75	0.58	0.65
ADA tuned	NO	Yes	GridSearch CV	1	1	1	1	0.98	0.99	0.88	0.93

Table 6- 3 Model tuning and Performances – ADA Boost

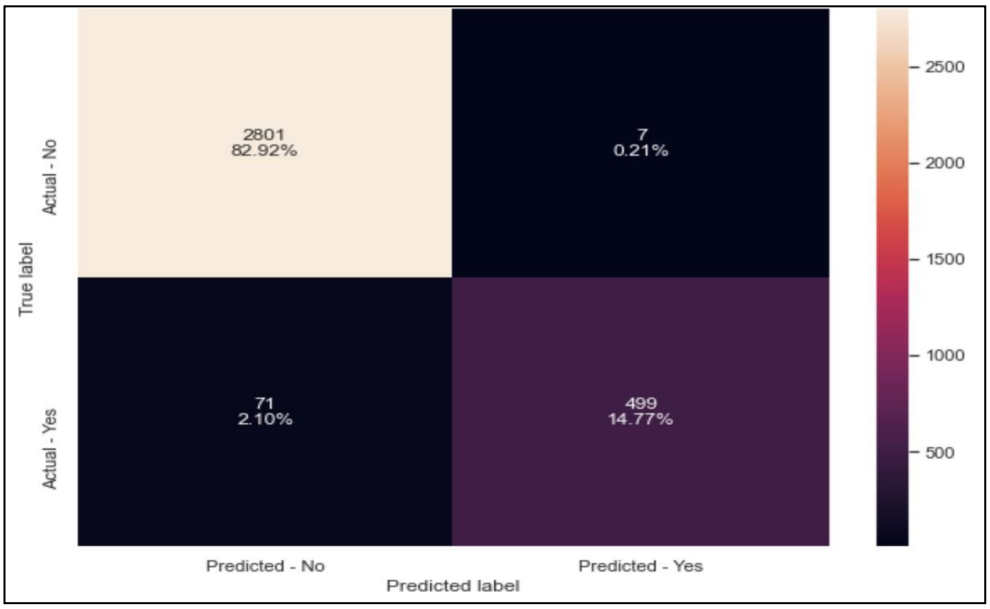
Adaboost

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.
In this modelling exercise, SKlearn's Adaboost 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 tuned model have been provided in the below sections **AUC Training : 1.00 and Testing 0.99**

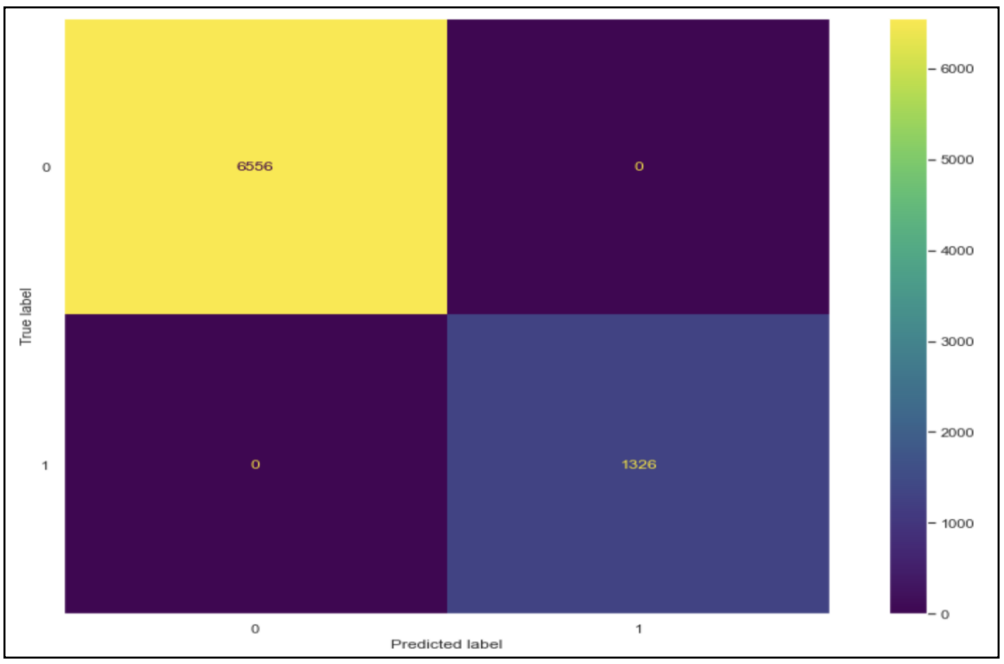
6.1.1.3. Gridsearch CV model of ADA boost is the best model (Also the best model)

```
Accuracy on training set :  1.0
Accuracy on test set :  0.9769094138543517
Recall on training set :  1.0
Recall on test set :  0.875438596491228
Precision on training set :  1.0
Precision on test set :  0.9861660079051383
F1 on training set :  1.0
F1 on test set :  0.9275092936802973
```

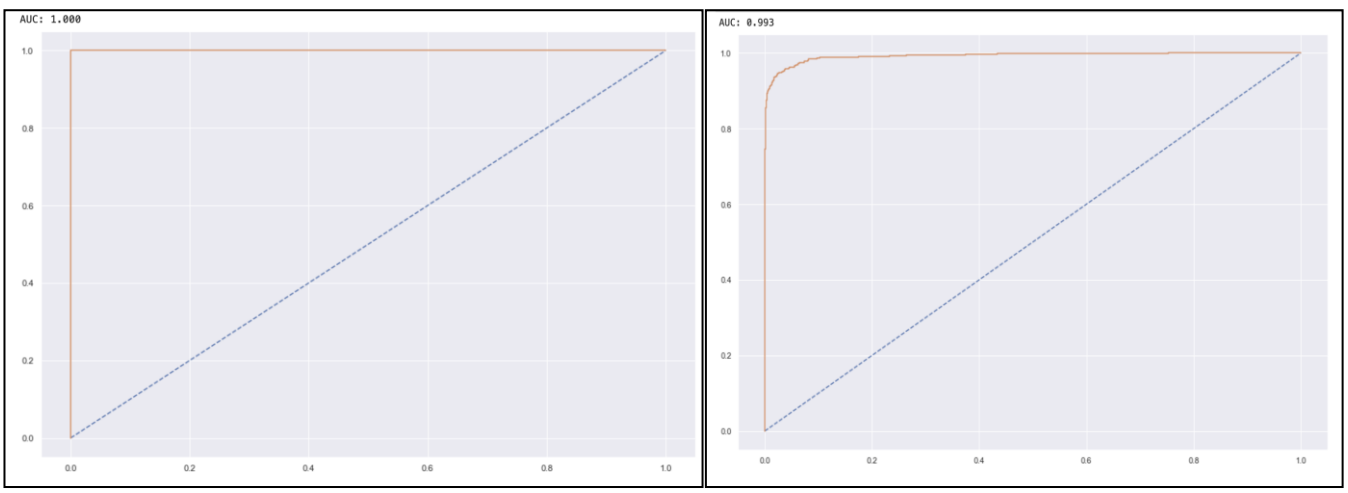
Confusion Matrix on Test Set:



Confusion Matrix on Train Set:



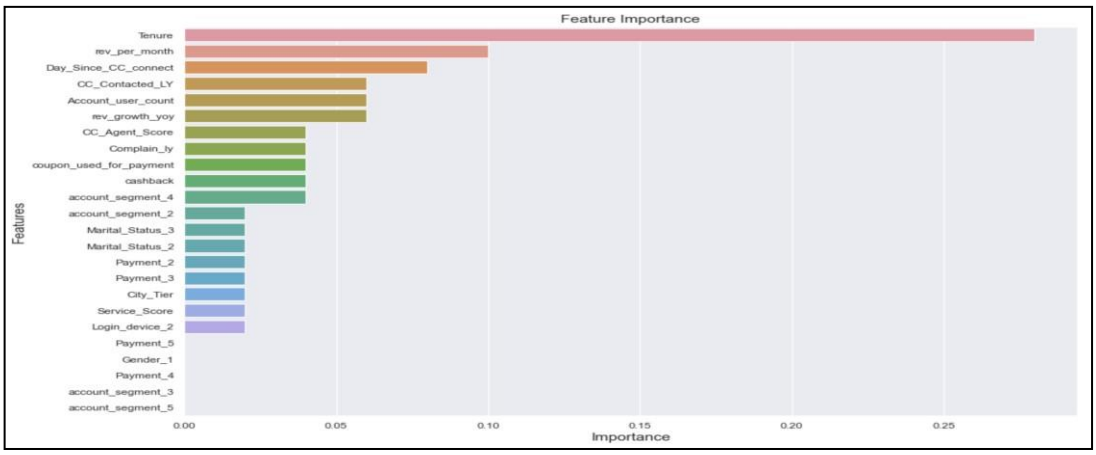
AUC of Train is: 1.0 and Test is 0.993



Model interpretation

The performance of this model when compared with other models is low. But it has not shown any overfitting or underfitting.

Feature importance from Adaboost



Since the performance for this model is not good the feature importance may not be reflective of actual data.

This model has picked Tenure to be the most important feature followed by Days since customer care last contact and Customer care contacted previous year.

- From EDA, we can observe that for lower tenures especially within the first year, the churn is higher. Hence, once a customer has been acquired, the first year is very important to keep the customer satisfied.
- Churned customers had contacted customer care more recently before churning than active customers. Per EDA, median days since last customer connect is higher for active customers. Churned customers had contacted customer care recently before churning.
- The next important parameter to predict customer churn per this model is number of times customer care was contacted by the customer. Per EDA, the median and third quantile of number of times customer care was contacted previous year is higher for churned customers compared to active/current customers.

6.1.4. Ensemble method - Gradient Boost

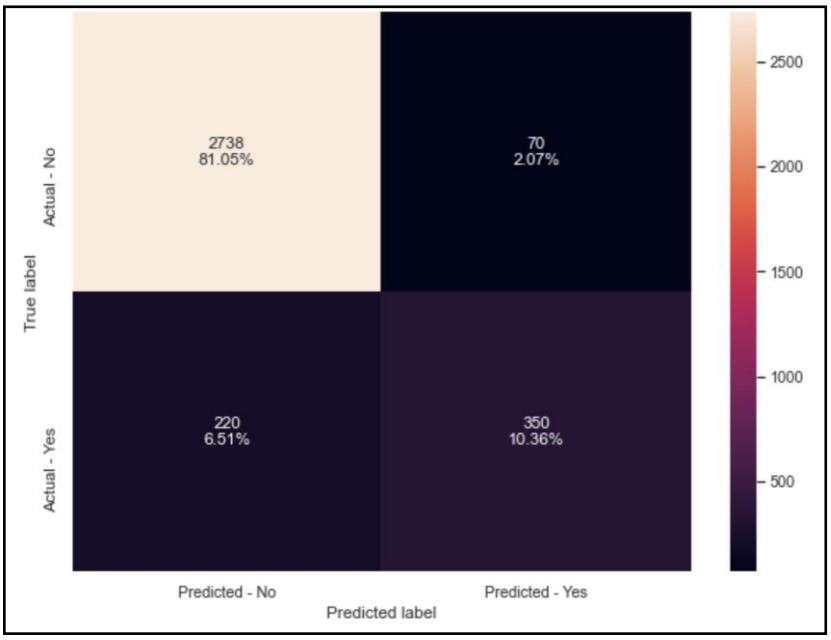
Gradient boost is an ensemble machine learning algorithm that trains underlying models in a gradual, additive and sequential manner.
In this modelling exercise, SKlearn's Gradient Boost classifier function was used for modelling:

• 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 are provided below

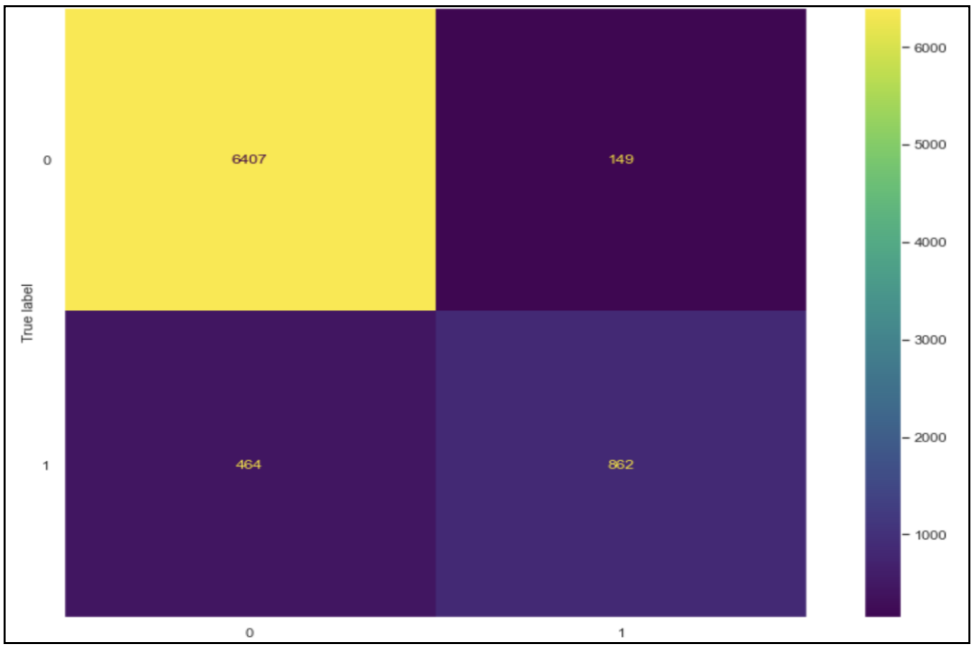
6.1.4.1. Gradient boost base model with default hyperparameters

```
Accuracy on training set :  0.9222278609489977
Accuracy on test set :  0.9141503848431024
Recall on training set :  0.6500754147812972
Recall on test set :  0.6140350877192983
Precision on training set :  0.8526211671612265
Precision on test set :  0.8333333333333334
F1 on training set :  0.7376979032948224
F1 on test set :  0.7070707070707071
```

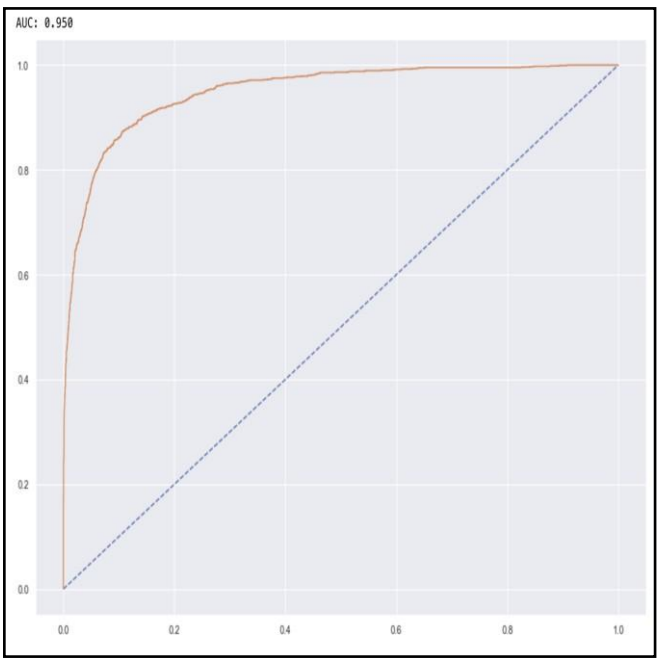
Confusion Matrix on Test Data:

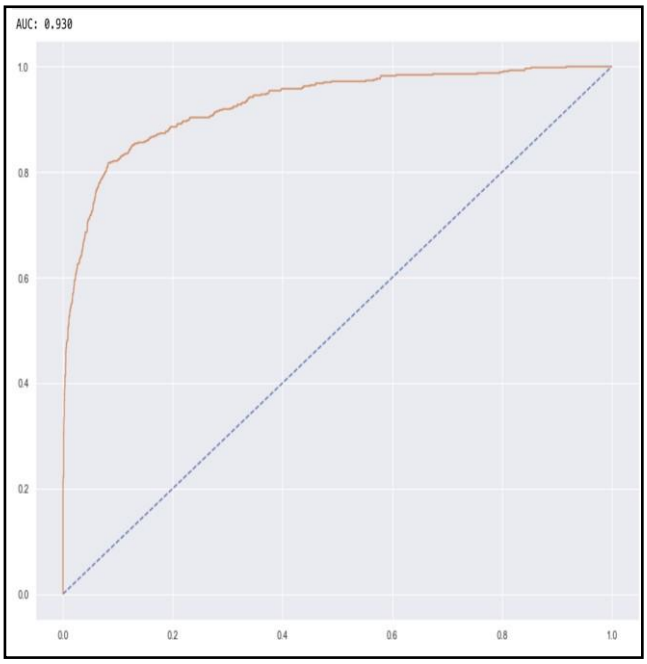


Confusion Matrix on Train Data:



AUC of Train is: 0.95 and AUC of Test is 0.93





The model is robust (no overfit or underfit) but the recall is quite poor in both train and test data. Hyperparameter tuning was done to see if performance can be improved on the model.

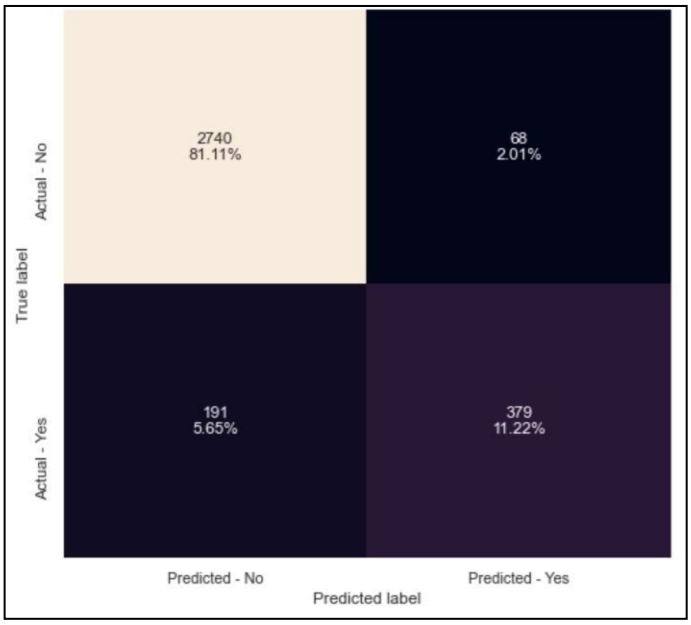
6.1.4.2. Gradient Boost - Model tuning

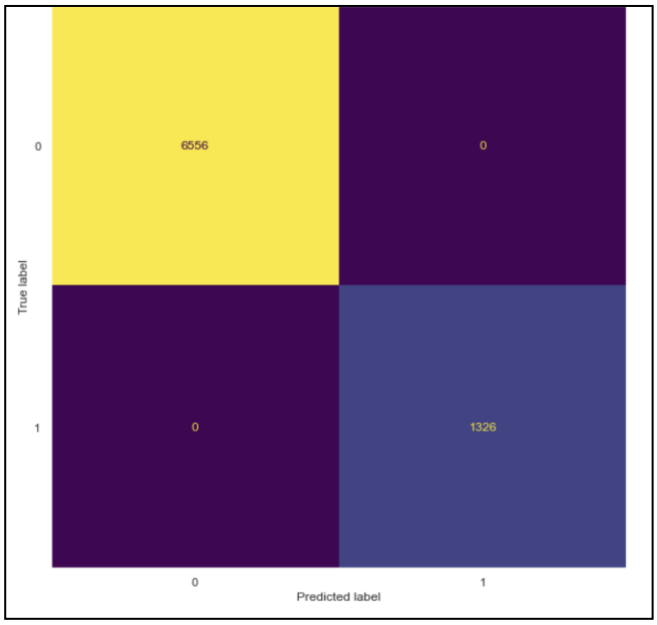
• The model's hyperparameters were tuned using GridSearchCV function from Sklearn library and model constructed using the best parameters selected. The tuning parameters for Gridsearch and individual scoring metrics for each parameter grid have been provided in Appendix

• Tuning improved the model performance considerably

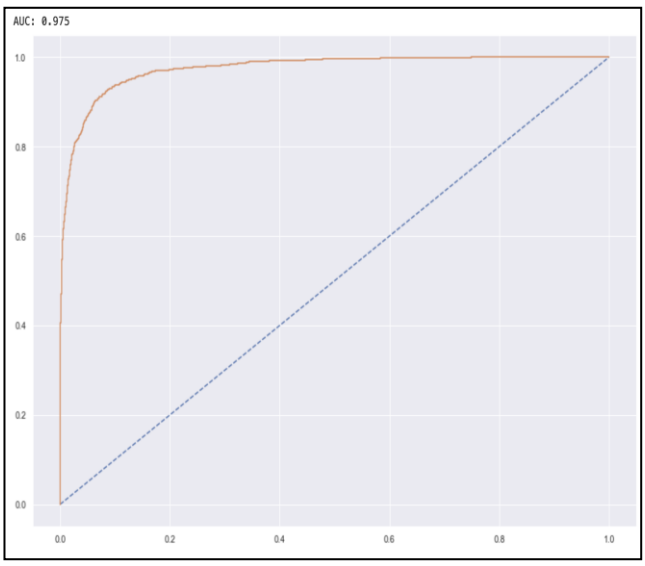
Model Refrence	Data Treatment		Hyper	Train Data				Test Data			
	Smote	Scaling	Parameter	Accuracy	Precision	Recall	F1 - Score	Accuracy	Precision	Recall	F1 - Score
GD with Default parameter	NO	Yes	Base Model	0.92	0.85	0.65	0.74	0.91	0.83	0.61	0.71
GD tuned	NO	Yes	GridSearch CV	1	1	1	1	0.98	0.99	0.92	0.95

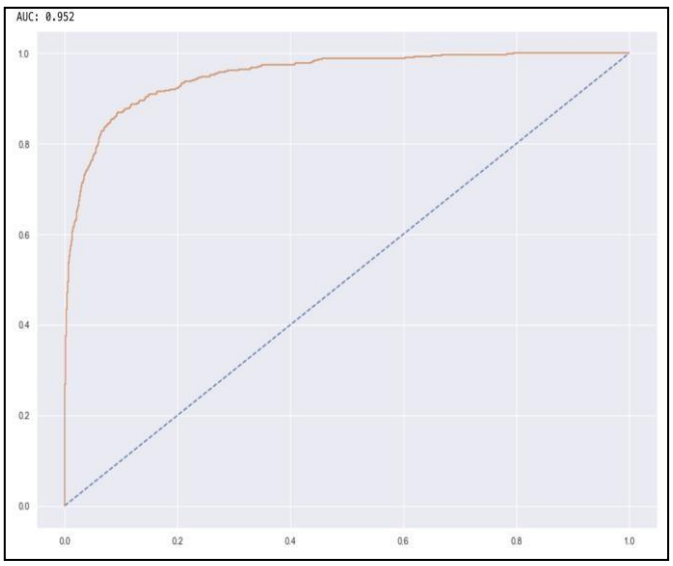
Table 6- 4 Model tuning and Performances – Gradient Boosting





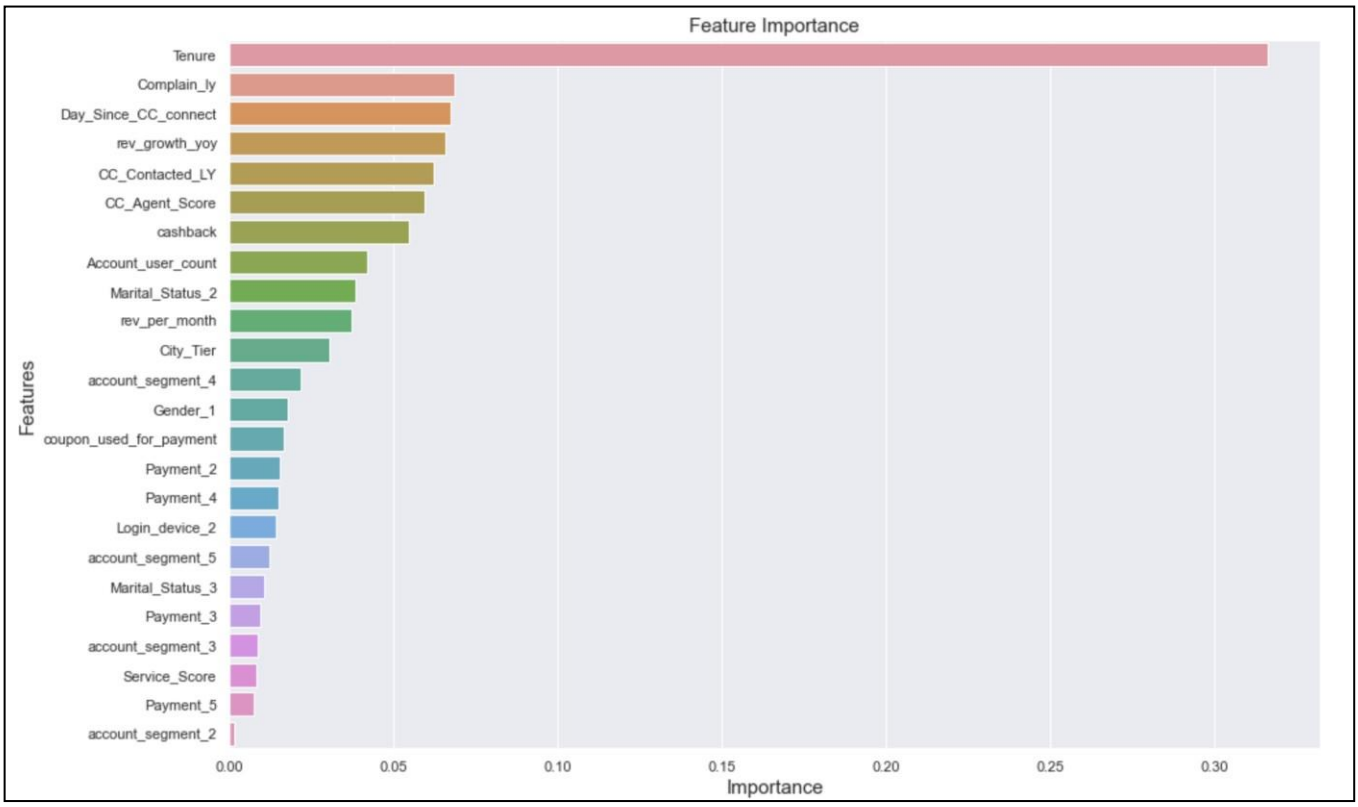
AUC of Train is: 1.0 and AUC of Test is: 0.997





The model is robust (no overfit or underfit) but the recall is quite poor in both train and test data. After Hyperparameter tuning the recall has been improved well.

Feature Importance Gradient Boost:



This model has also picked Tenure to be the most important feature followed by Complaints last year. Out of the top 5 important features, RF and XG have 2 features in common with Tenure and Complain_LY being the same top 2 important parameters. This model has picked Tenure to be the most important feature followed by Complaints last year and Customer care score. From EDA, we can observe that for lower tenures especially within the first year, the churn is higher. Hence, once a customer has been acquired, the first year is very important to keep the customer satisfied. The next risk indicator of customer churn per this model is if a complaint has been registered in the previous year. From EDA, if a complaint was registered, the risk of churn is high. Also, single customers have more propensity to churn compared to married or divorced customers.

6.1.5. Ensemble method – Random Forest

Random forest is an ensemble machine learning algorithm that uses bootstrapping to reduce variance in the underlying decision trees. It also selects only a subset of features for each node split decision. Since it is an ensemble of trees with varying features for each node, each tree is different from another. Random forest is resistant to outliers and does not require the data to be scaled.

Model Refrence	Data Treatment		Hyper	Train Data					Test Data				
	Smote	Scaling	Parameter	Accuracy	Precision	Recall	F1 - Score	AUC	Accuracy	Precision	Recall	F1 - Score	AUC
RF with Default parameter	NO	Yes	Base Model	1	1	1	1	1	0.97	0.99	0.86	0.92	0.99
RF tuned	NO	Yes	GridSearch CV	0.95	0.98	0.74	0.85	0.901	0.93	0.96	0.64	0.77	0.89

Table 6-5 Model tuning and Performances - Random Forest

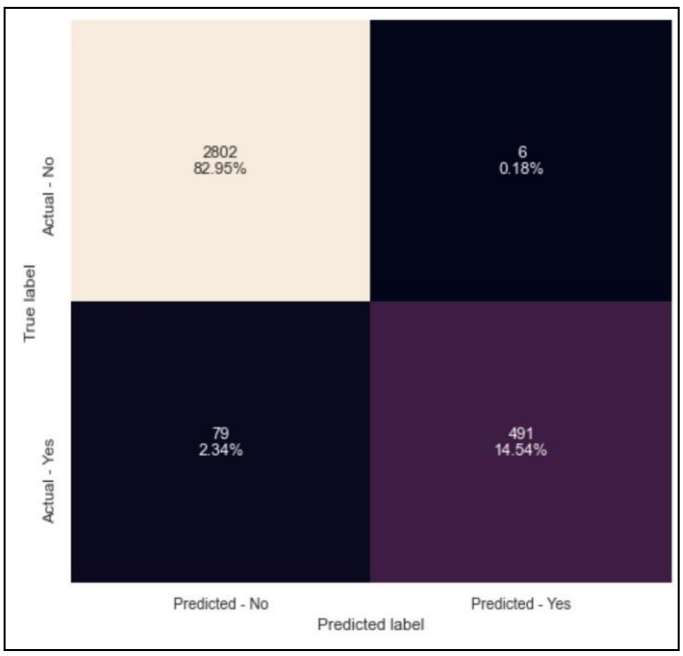
The data with outliers gave a slightly better performance compared to outlier treated data. But the difference between Recall in train and test is more than 10%. The model has overfit on train data and hence this was not considered for selection as final model.

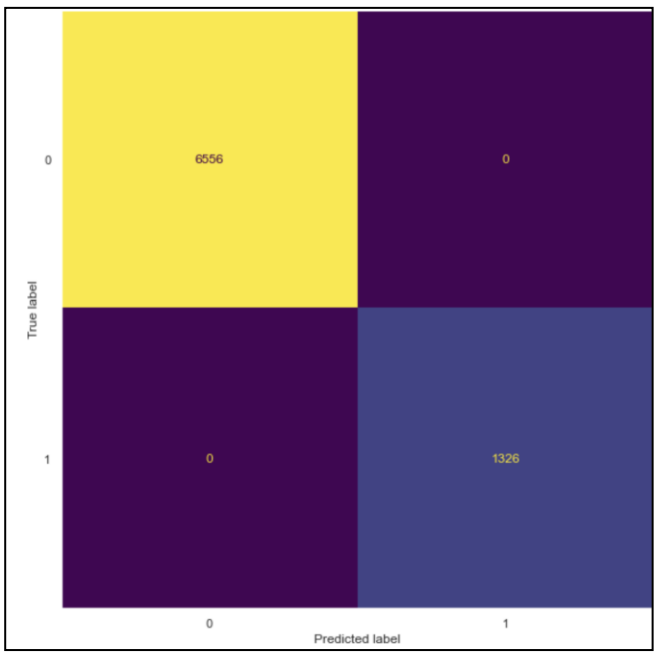
Random forest is an ensemble machine learning algorithm that uses bootstrapping to reduce variance in the underlying decision trees. It also selects only a subset of features for each node split decision. Since it is an ensemble of trees with varying features for each node, each tree is different from another. Random forest is resistant to outliers and does not require the data to be scaled.
In this modelling exercise, SKlearn's Random Forest 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

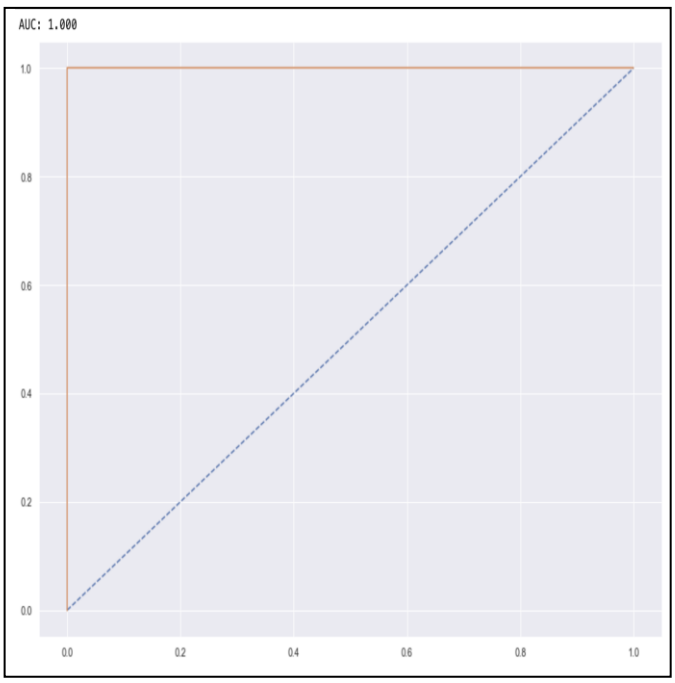
6.1.5.1. Random Forest base model with default hyperparameters

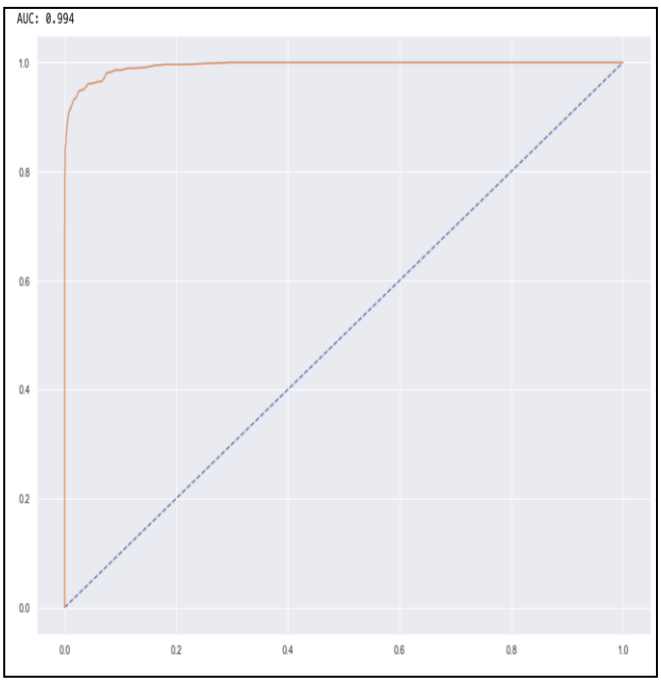
```
Accuracy on training set :  1.0
Accuracy on test set :  0.9748371817643576
Recall on training set :  1.0
Recall on test set :  0.8614035087719298
Precision on training set :  1.0
Precision on test set :  0.9879275653923542
F1 on training set :  1.0
F1 on test set :  0.9203373945641987
```





AUC and ROC of Train and Test Data: 1.00 and 0.994





Clearly, the model has overfit on the train dataset as the test dataset shows a recall of only 0.86. The model has to be tuned to reduce variance.

6.1.5.2. SKLearn Random Forest - model tuning

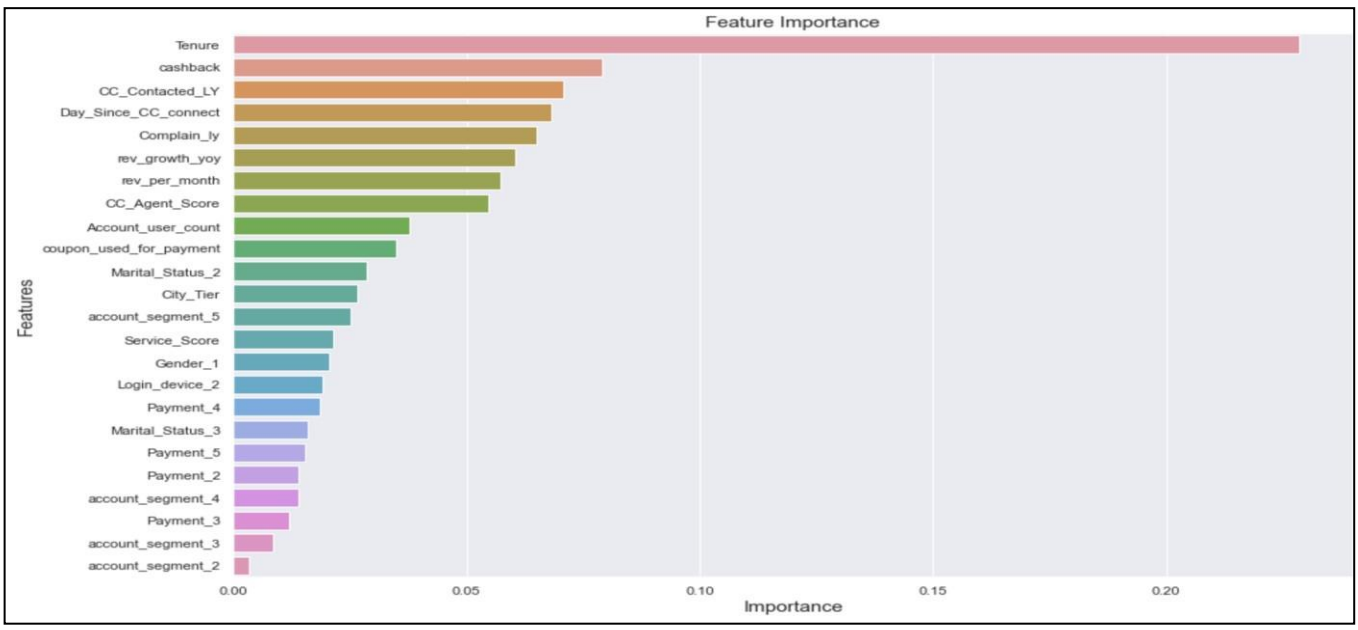
• The model's hyperparameters were tuned using GridSearchCV function and model constructed using the best parameters selected by GridSearchCV.
• As can be seen from the base model, the model had overfit on train dataset (all 1s). During tuning, the tree depth was contained, the minimum samples in each leaf increased to reduce overfit.

Model Refrence	Data Treatment		Hyper	Train Data				Test Data			
	Smote	Scaling	Parameter	Accuracy	Precision	Recall	F1 - Score	Accuracy	Precision	Recall	F1 - Score
RF with Default parameter	NO	Yes	Base Model	1	1	1	1	0.97	0.99	0.86	0.92
RF tuned	NO	Yes	GridSearch CV	0.95	0.98	0.74	0.85	0.93	0.96	0.64	0.77

6.1.5.3. Model interpretation

This model has overfitted both in the default base model as well as the hyper parameter tuned model. The tuned model in fact performed worser than the base model. Hence in the final comparison, this cannot get selected as best model.
SKlearn's permutation importance function was used to determine the features that are important to the model.

Feature Importance for Random Forest:



This model has picked Tenure to be the most important feature. However, the next two parameters in terms of importance are not the same amongst XGBoost and Random Forest. Since the performance for this model is not good the feature importance may not be reflective of actual data.
This model has picked Tenure to be the most important feature followed by Customer care contacted previous year and Days since customer care last contact.

- From EDA, we can observe that for lower tenures especially within the first year, the churn is higher. Hence, once a customer has been acquired, the first year is very important to keep the customer satisfied.
- The next important parameter to predict customer churn per this model is number of times customer care was contacted by the customer. Per EDA, the median and third quantile of number of times customer care was contacted previous year is higher for churned customers compared to active/current customers.
- Churned customers had contacted customer care more recently before churning than active customers. Per EDA, median days since last customer connect is higher for active customers. Churned customers had contacted customer care recently before churning.

6.1.6. Artificial Neural Network

Artificial Neural Network (ANN) is a powerful machine learning algorithm that can be used for classification as well as regression. This algorithm can learn the complex patterns in underlying data. There is a tendency to overfit, but that can be controlled in the tuning exercise using the hyperparameters.

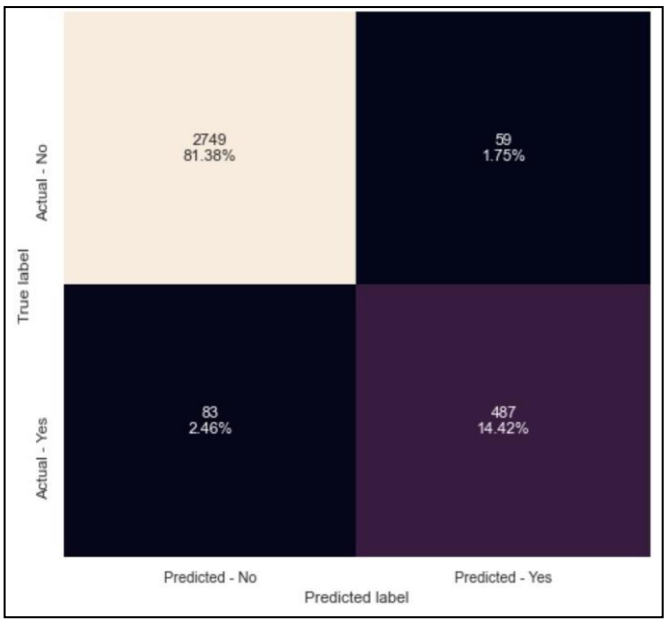
Model Refrence	Data Treatment		Hyper	Train Data					Test Data				
	Smote	Scaling	Parameter	Accuracy	Precision	Recall	F1 - Score	AUC	Accuracy	Precision	Recall	F1 - Score	AUC
ANN Tuned	NO	Yes	GridSearch CV	0.88	0.71	0.45	0.55	0.84	0.88	0.74	0.47	0.57	0.84
ANN Base model	NO	Yes	Base Model	0.98	0.96	0.94	0.95	0.996	0.96	0.89	0.85	0.87	0.98

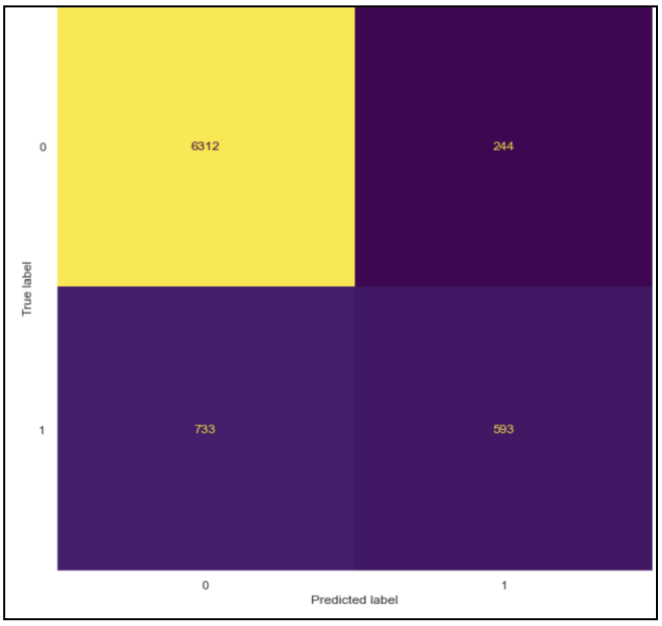
Table 6-6 Model tuning and Performances - Artificial Neural Network

- 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 scaled 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

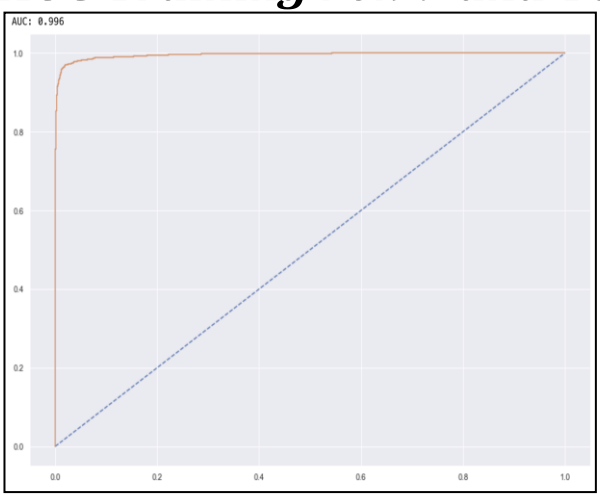
6.1.6.1. ANN base model with default hyperparameters

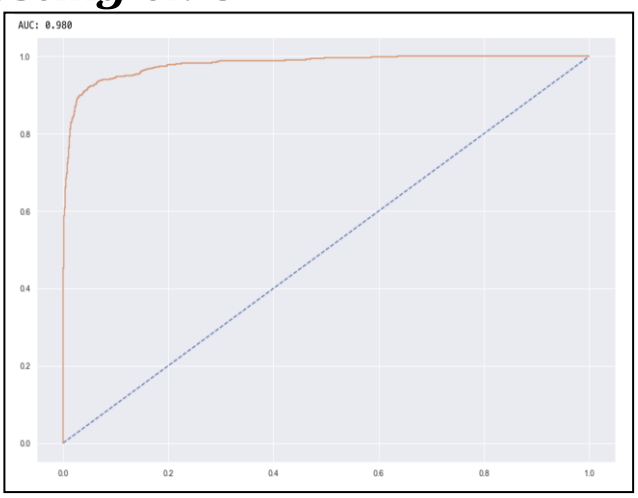
```
Accuracy on training set :  0.9824917533620908
Accuracy on test set :  0.9579632918886916
Recall on training set :  0.9381598793363499
Recall on test set :  0.8543859649122807
Precision on training set :  0.9569230769230769
Precision on test set :  0.891941391941392
F1 on training set :  0.9474485910129475
F1 on test set :  0.8727598566308243
```





AUC Training : 0.99 and Testing 0.98





6.1.7. K-Nearest Neighbour

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.

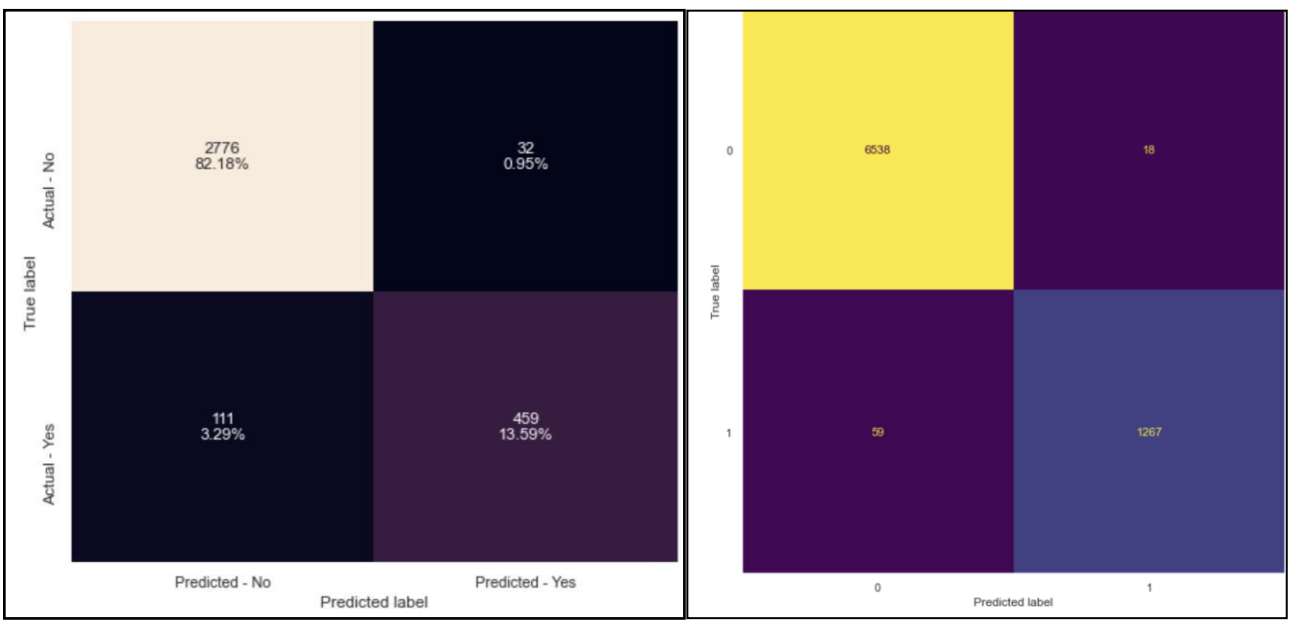
Model Refrence	Data Treatment		Hyper	Train Data				Test Data			
	Smote	Scaling	Parameter	Accuracy	Precision	Recall	F1 - Score	Accuracy	Precision	Recall	F1 - Score
KNN	NO	Yes	Base Model	0.98	0.97	0.9	0.94	0.96	0.93	0.81	0.87
KNN - Tuned	NO	Yes	GridSearch CV	0.99	0.99	0.96	0.97	0.97	0.95	0.89	0.92

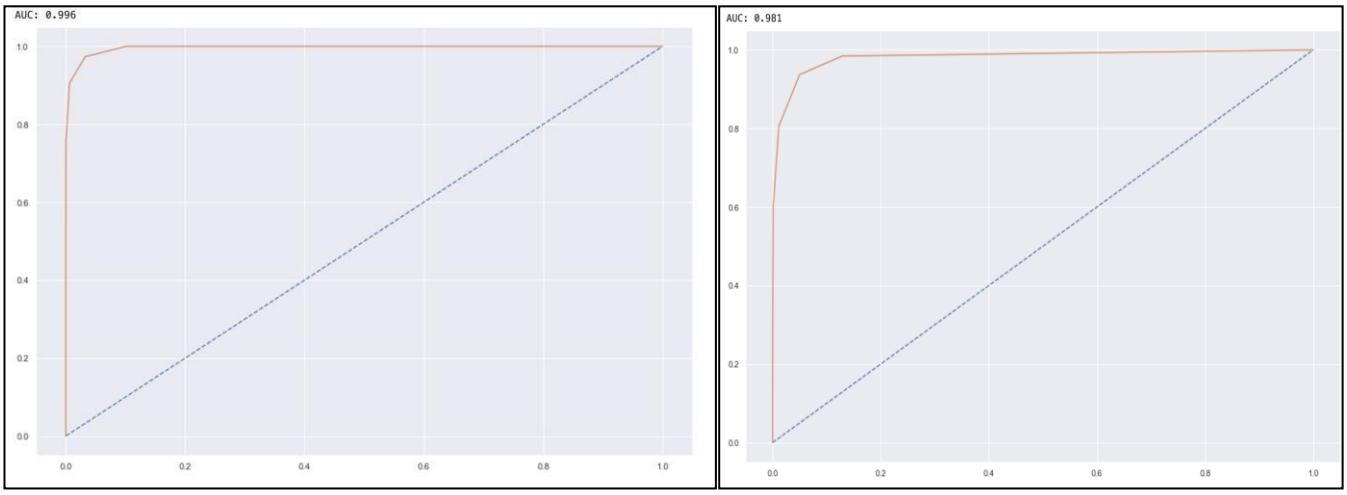
Table 6-7 Model tuning – _KNN models performance

KNN has some nice properties: it is automatically non-linear, it can detect linear or non-linear distributed data, it tends to perform very well with a lot of data points.

6.1.7.1. KNN base model with default hyperparameters

```
Accuracy on training set :  0.9790662268459782
Accuracy on test set :  0.9576672587329781
Recall on training set :  0.9049773755656109
Recall on test set :  0.8052631578947368
Precision on training set :  0.9685230024213075
Precision on test set :  0.9348268839103869
F1 on training set :  0.935672514619883
F1 on test set :  0.8652214891611687
```





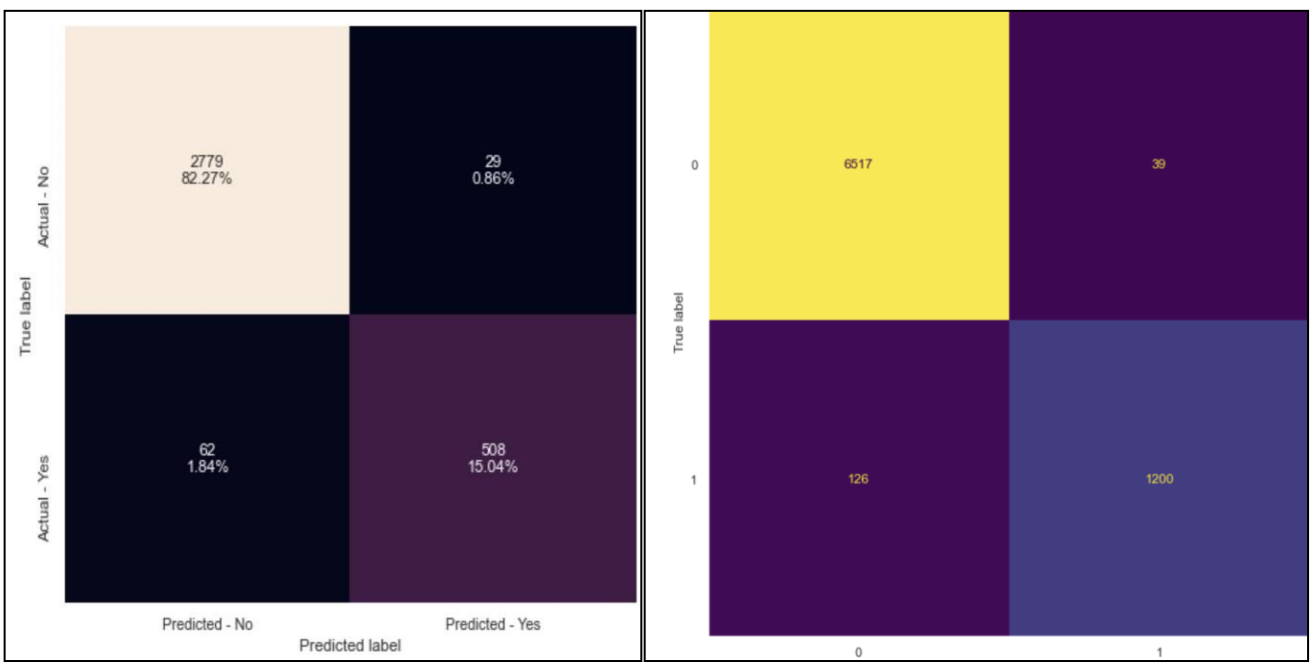
AUC of Training data is : 0.996 and AUC of test data set is: 0.981

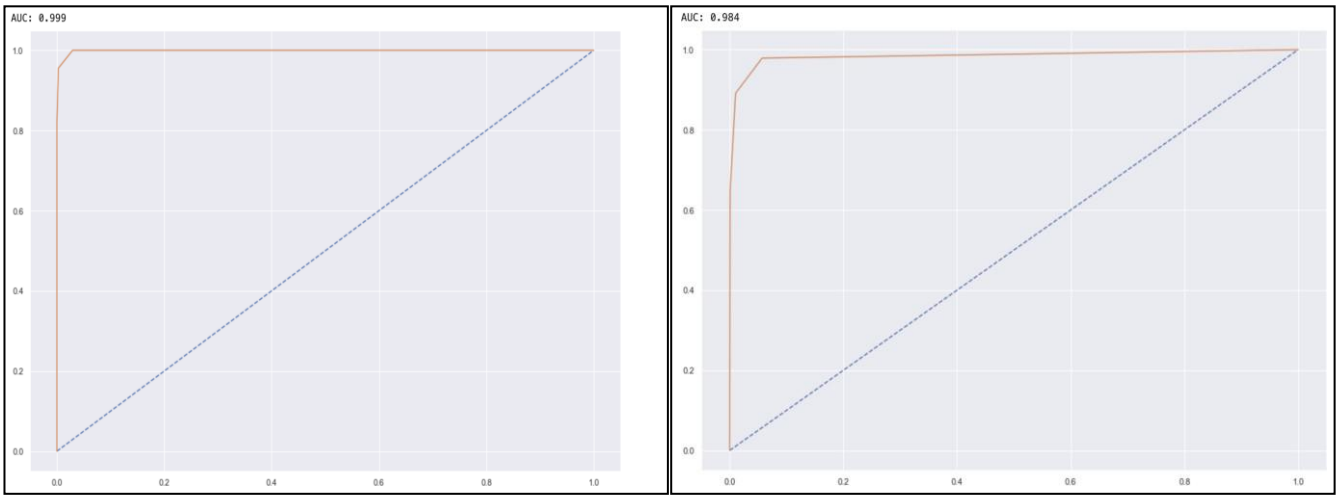
6.1.7.2. SKLearn KNN - model tuning

We first create a KNN classifier instance and then prepare a range of values of hyperparameter K from 1 to 11 that will be used by GridSearchCV to find the best value of K.

```
Accuracy on training set :  0.9902309058614565
Accuracy on test set :  0.9730609828300769
Recall on training set :  0.9555052790346908
Recall on test set :  0.8912280701754386
Precision on training set :  0.9859922178988327
Precision on test set :  0.9459962756052142
F1 on training set :  0.9705093833780161
F1 on test set :  0.9177958446251129
```

Confusion Matrix of Test and Train Data





AUC of Training data is : 0.999 and AUC of test data set is: 0.984

6.1.7.3. Model interpretation

The KNN tuned model has performed well compared to base model. However the base model is overfitted both in the train and test set. In base model the recall is 10% lesser in test compared to train. Hence we can't take KNN base model as best model. Where as the KNN Tuned model with Gridsearch cv has performed better compared to base model. The Recall of Train is 0.95 and Test is 0.89. Hence in the final comparison, we can add KNN-Tuned model for best model comparison.

6.1.8 XG Boost

XGBoost (XGB) 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.

Model Refrence	Data Treatment		Hyper	Train Data				Test Data			
	Smote	Scaling	Parameter	Accuracy	Precision	Recall	F1 - Score	Accuracy	Precision	Recall	F1 - Score
XGBoost Base Model	NO	Yes	Base Model with Random state = 1	1	1	1	1	0.97	0.96	0.83	0.89
XGBoost - Tuned	NO	Yes	GridSearch CV	1	1	1	1	0.97	0.96	0.88	0.92

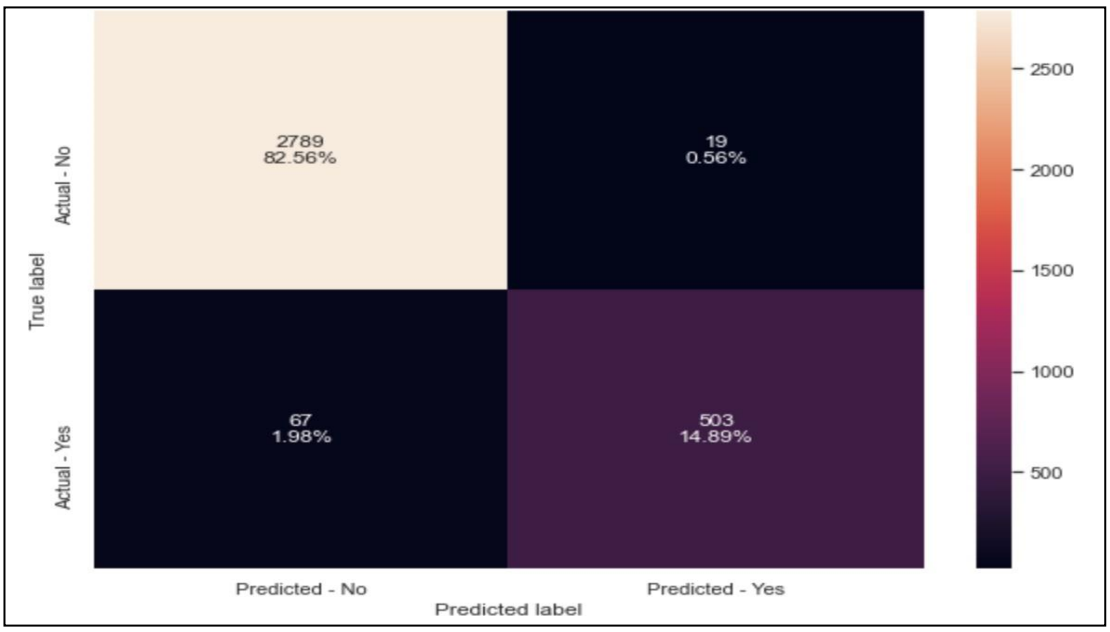
XGBoost builds upon the idea of gradient boosting algorithm with some modifications. Gradient boosted trees are built in sequence because each estimator predicts residuals of the previous estimator, which makes it slow at the time of model training as compared to build estimators in parallel.

Thus, the main concentration in XGBoost is speed enhancement and model performance.

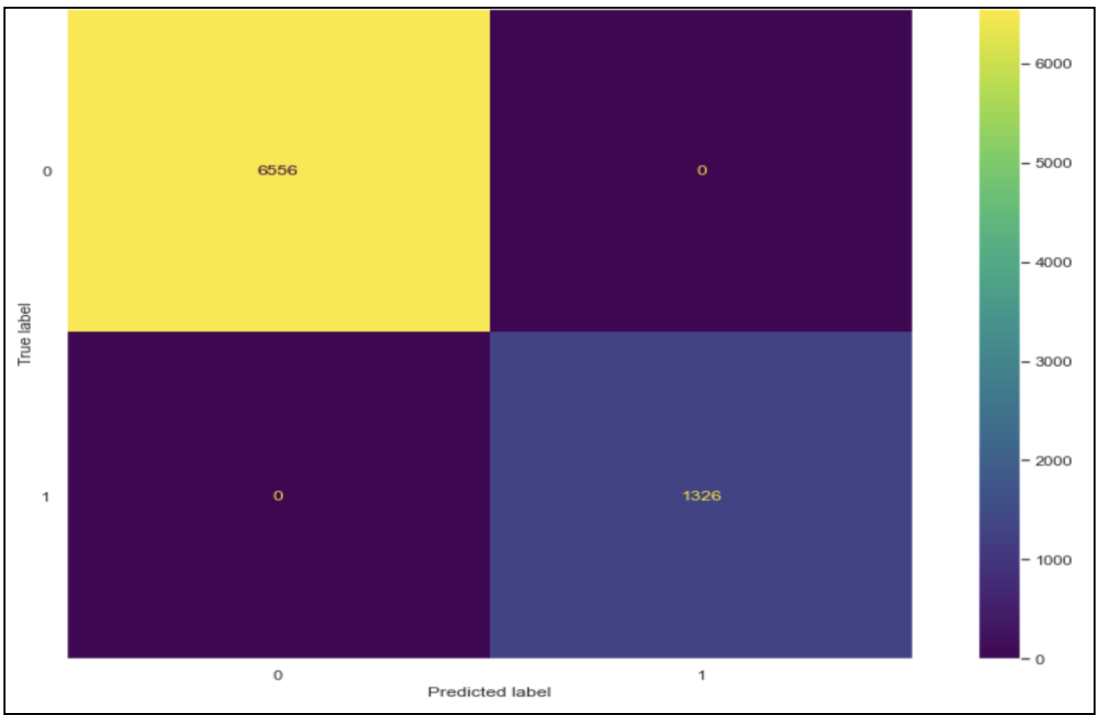
6.1.8.1. XGBoost - Tuned (best Model)

```
Accuracy on training set :  1.0
Accuracy on test set :  0.9745411486086442
Recall on training set :  1.0
Recall on test set :  0.8824561403508772
Precision on training set :  1.0
Precision on test set :  0.9636015325670498
F1 on training set :  1.0
F1 on test set :  0.9212454212454213
```

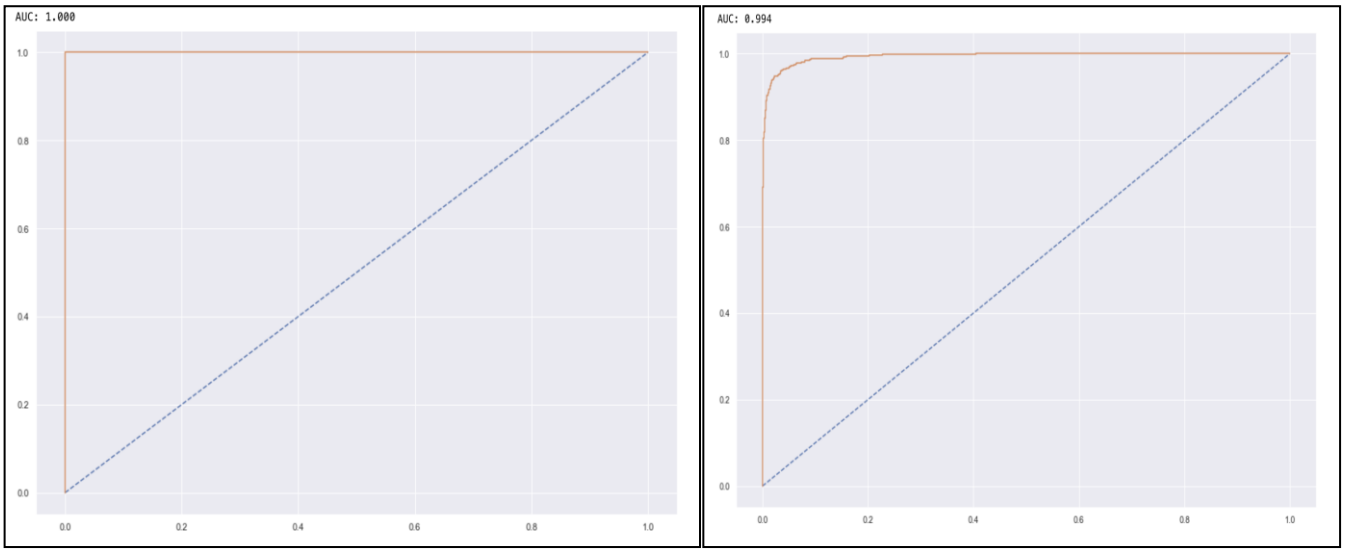
Confusion Matrix on Test Data



Confusion Matrix on Train set:

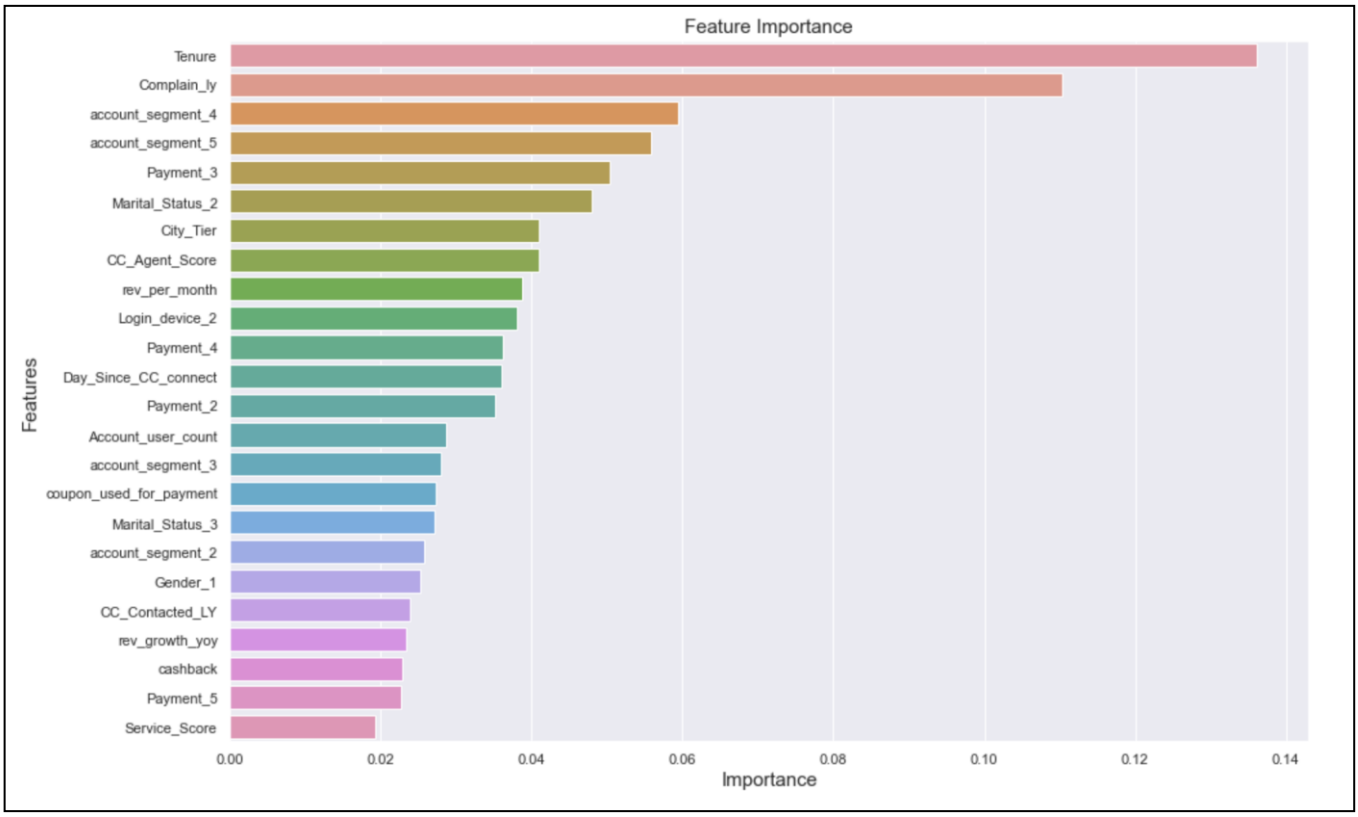


ROC and AUC of Train and Test:



AUC train: 1.00 and AUC Test: 0.994

Feature Importance for XGBoost:



This model has picked Tenure to be the most important feature followed by Customer care contacted previous year and Days since customer care last contact.

• From EDA, we can observe that for lower tenures especially within the first year, the churn is higher. Hence, once a customer has been acquired, the first year is very important to keep the customer satisfied.

• The next important parameter to predict customer churn per this model is number of times customer care was contacted by the customer. Per EDA, the median and third quantile of number of times customer care was contacted previous year is higher for churned customers compared to active/current customers.
• Churned customers had contacted customer care more recently before churning than active customers. Per EDA, median days since last customer connect is higher for active customers. Churned customers had contacted customer care recently before churning.

7. Model validation

• The given data was split into train and test dataset in the ratio 70:30. The test dataset was held out and kept for validation purpose only.
• 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.
• If train data seemed to be giving all 1's and the difference between train and test performances are not off by more than 10%, a validation of test scores was done by doing 5-fold and 10-fold cross validation on entire dataset. The scores on this cross validation were compared to test dataset scores to ensure that model had not overfitted on train dataset.

7.1. Criteria for the best performing model

Primary criteria

• **Precision, Recall & F1-score for 1s**: This is the case of class imbalance as the dataset has 16.8% churns. In this case study 'Precision' and 'Recall' of class 1 or the minority class is most important. Combining these 2 metrics, F1-score for class 1 is also used in the comparison.

• **Precision** is defined as True positive/ (True Positive + False Positive). It answers the question - Out of all customers that the model identifies as churning customers, how many are actual churners? This is the most important factor in this case as budget for offers to retain customers would be limited and more false positives would mean spending that budget on customers who would not churn anyway. 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.

• **Recall** is defined as True Positive / (True Positive + False Negative). It answers the question - 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.

• F1-score is harmonic mean of Precision and Recall. Where both Precision and Recall are important, this metric can be used as a single metric that needs to be optimized for.

Secondary 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. The following table shows performance of the best model from each algorithm built so far. The metrics given in the below table are all for minority class/churners (1s) which is the class of interest. The best performer has been highlighted in green. The figure below that shows the ROC curve for all models and the best performer is the blue line.

	Model	Train_Accuracy	Test_Accuracy	Train_Recall	Test_Recall	Train_Precision	Test_Precision	Train_f1	Test_f1
0	Logistic Regression with Smote	0.81	0.81	0.79	0.76	0.47	0.46	0.59	0.57
1	Logistic Regression	0.89	0.89	0.51	0.48	0.78	0.76	0.62	0.59
2	Logistic Regression-Tuned	0.89	0.89	0.51	0.49	0.78	0.76	0.62	0.59
3	LDA	0.89	0.88	0.47	0.46	0.76	0.75	0.58	0.57
4	LDA-Tuned	0.89	0.88	0.47	0.46	0.76	0.75	0.58	0.57
5	AdaBoost with default paramters	0.90	0.90	0.59	0.58	0.73	0.75	0.65	0.66
6	AdaBoost Tuned	1.00	0.98	1.00	0.88	1.00	0.99	1.00	0.93
7	Gradient Boosting with default parameters	0.92	0.91	0.65	0.61	0.85	0.83	0.74	0.71
8	XGBoost	1.00	0.97	1.00	0.83	1.00	0.96	1.00	0.89
9	XGBoost-Tuned	1.00	0.97	1.00	0.88	1.00	0.96	1.00	0.92
10	Gradient Boosting with init=AdaBoost	0.92	0.91	0.63	0.59	0.85	0.84	0.72	0.70
11	Gradient Boosting Tuned	1.00	0.98	1.00	0.92	1.00	0.99	1.00	0.95
12	Random Forest	1.00	0.97	1.00	0.86	1.00	0.99	1.00	0.92
13	Random Forest - Tuned	0.95	0.94	0.74	0.64	0.99	0.97	0.85	0.77
14	ANN	0.98	0.95	0.89	0.78	0.97	0.93	0.93	0.85
15	ANN-Tuned	0.88	0.88	0.45	0.47	0.71	0.74	0.55	0.57
16	KNN	0.98	0.96	0.90	0.81	0.97	0.93	0.94	0.87
17	KNN-Tuned	0.99	0.97	0.96	0.89	0.99	0.95	0.97	0.92

Table 7-1 A comparison of best model from all algorithms

7.2. Why Gradient boost is the best model?

• The hyperparameter tuned Gradient Boost model (GB_Tuned) has given the best performance in terms of test data precision, recall and f1-score which are the primary evaluation metrics. The Accuracy and AUC are also highest amongst all the models. As the model looked like it overfit on train dataset, a further 5-fold and 10-fold cross-validation was done on complete dataset in which the F1-score on all folds was either comparable or greater than test data f1-score (the test data performance held true or was better for all folds)

• The difference between train data metrics and test data metrics is within 10%

• The model is interpretable. Sklearn provided feature importance for the model

7.3. How can business use these metrics?

• A precision of 0.99 implies that out of 100 customers that the model has identified as churned, 99 would actually churn and 1 would not. Any marketing budget allocated for a targeted campaign for retention of these customers would be most optimally utilized as only 1/100 customers would be incorrectly identified as churn.

• A recall of 0.92 implies that for 100 customers who actually churn, the model would have identified 92 as churn and 8 as not-churn. This would mean that the campaign would target these 92 customers and there is scope for retaining these customers and missing out on 8 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 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 high value customer segment.

7.4. Model interpretation from best models

Gradient boost model has given very good performance after tuning. The model is robust (no overfit or underfit).
The figures below show the feature importance given by the Gradient Boost model and two other top scoring models (F1-score) – _Random Forest and XGBoost. A comparison of top 6 features that contributed to these models is provided in a table.

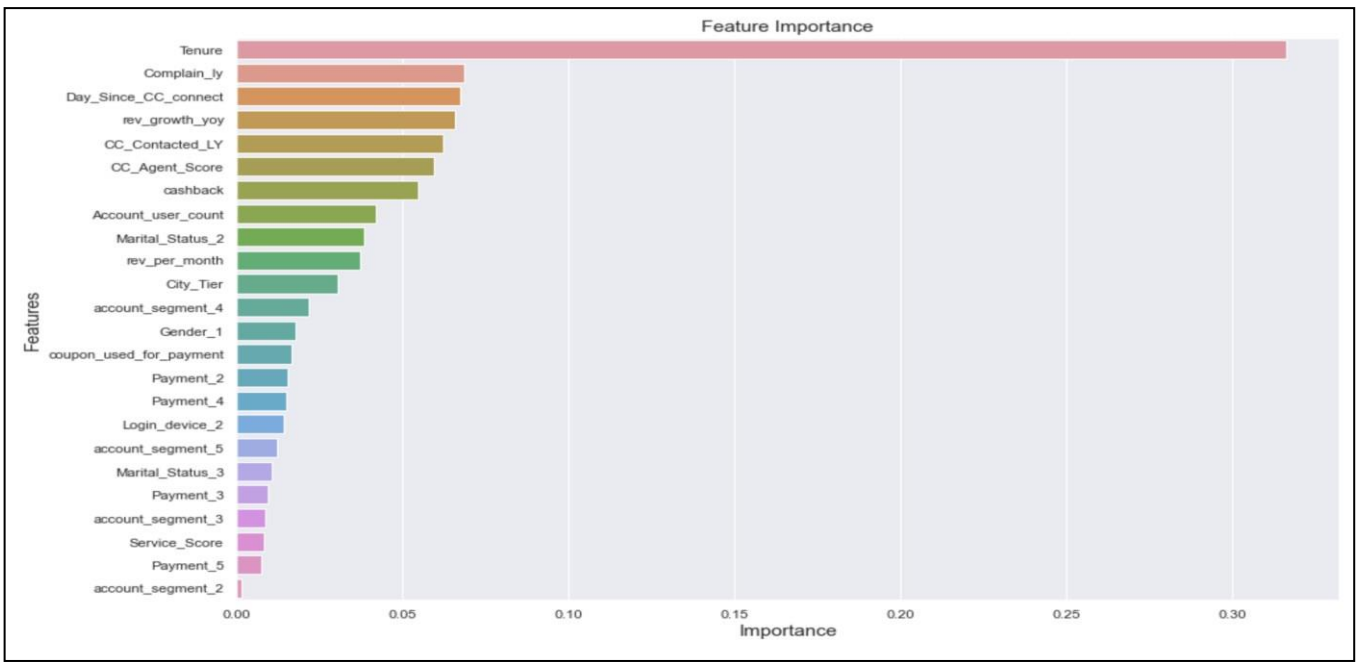


Figure 7-2 Feature importance for Gradient Boost model

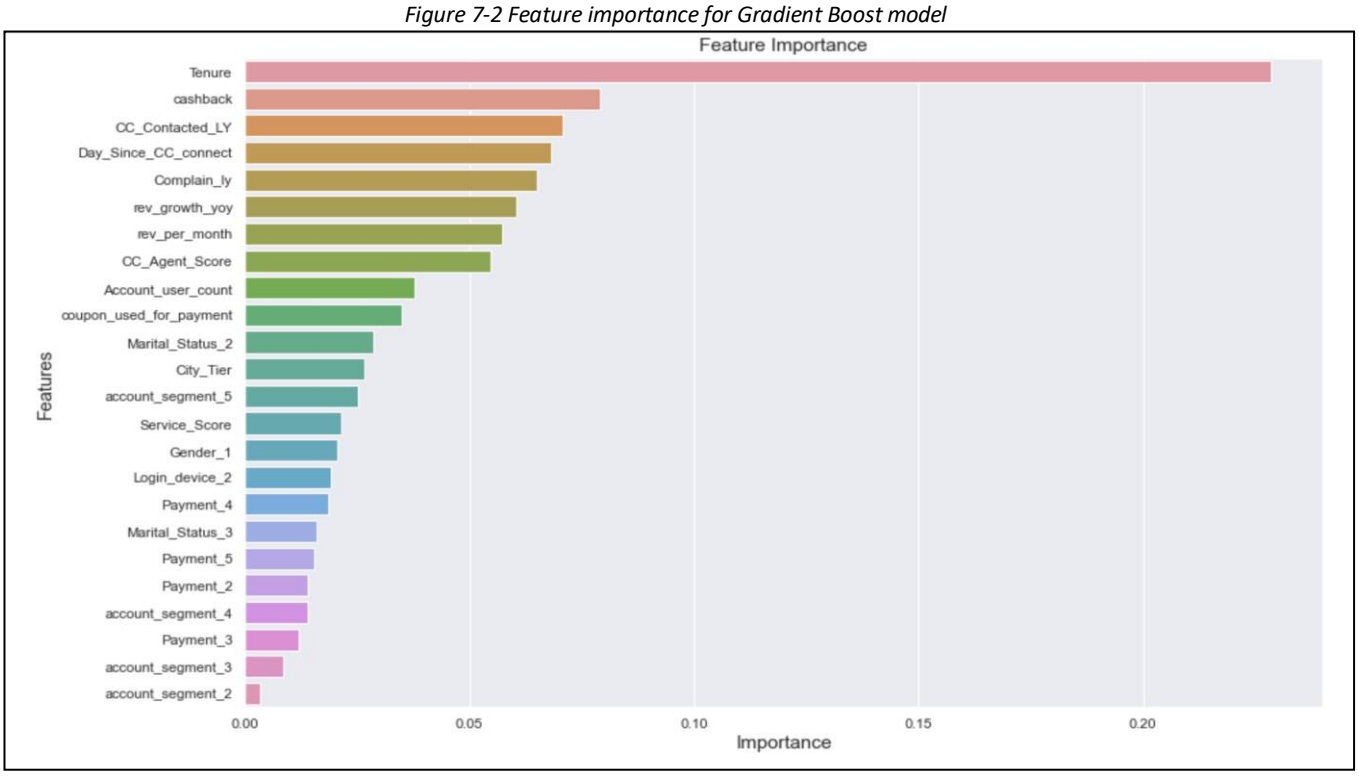


Figure 7-3 Feature importance for Random Forest model

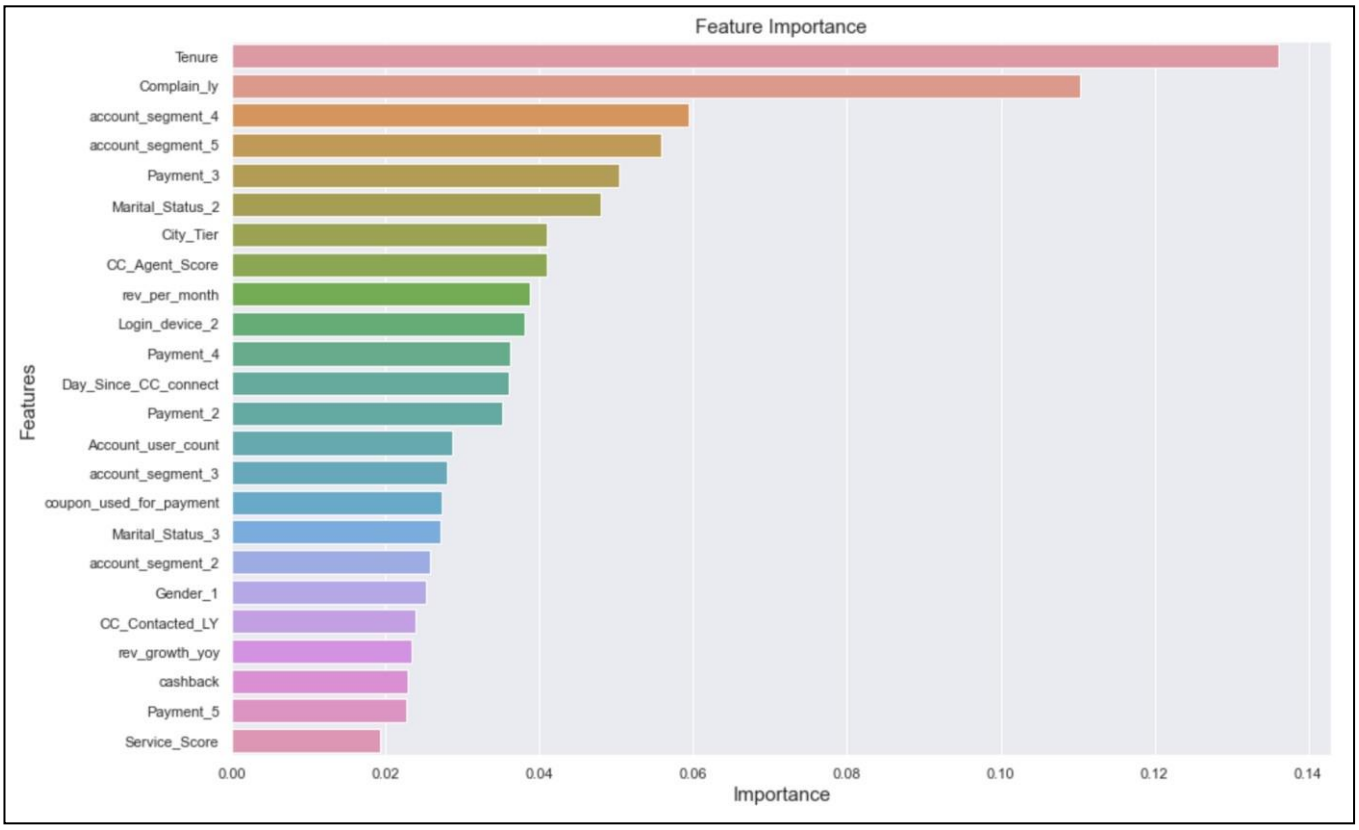


*Figure 7-4 Feature importance for XGBoost model**

	Gradient Boost	**Random Forest**	**XGBoost**
Feature1	Tenure	Tenure	Tenure
Feature2	Complain	cashback	Complain
Feature3	Days since Customer Contacted Last year	Number of times CC Last year	Account Segment_4
Feature4	Rev_growth_yoy	Days since Customer Contacted Last year	Account Segment_5
Feature5	Number of times CC Last year	Complain	Payment_3
Feature6	CC_Agent_score	Rev_growth_yoy	Marital_status_2

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.
If we look at the top3 models, the top most contributor remains the same. Was complaint made last year and customer care agent score also figure in Top6 features across these 3 models.

Top 5 features from Gradient Boost

• From EDA, we can observe that for lower tenures especially within the first month, the churn is higher. Hence, once a customer has been acquired, the first two months is very important to keep the customer satisfied.

• Churned customers had contacted customer care more recently before churning than active customers. Per EDA, median days since last customer connect is higher for active customers. Churned customers had contacted customer care recently before churning.

• The next important parameter to predict customer churn per this model is number of times customer care was contacted by the customer. Per EDA, the median and third quantile of number of times customer care was contacted previous year is higher for churned customers compared to active/current customers.

• According to ranking of important features, except tenure, the other features in top 5 are related to customer care or complaints. This may be indicative of need to monitor and improve the customer care processes.

8. Business recommendations

8.1. High churn rate in low tenure customers

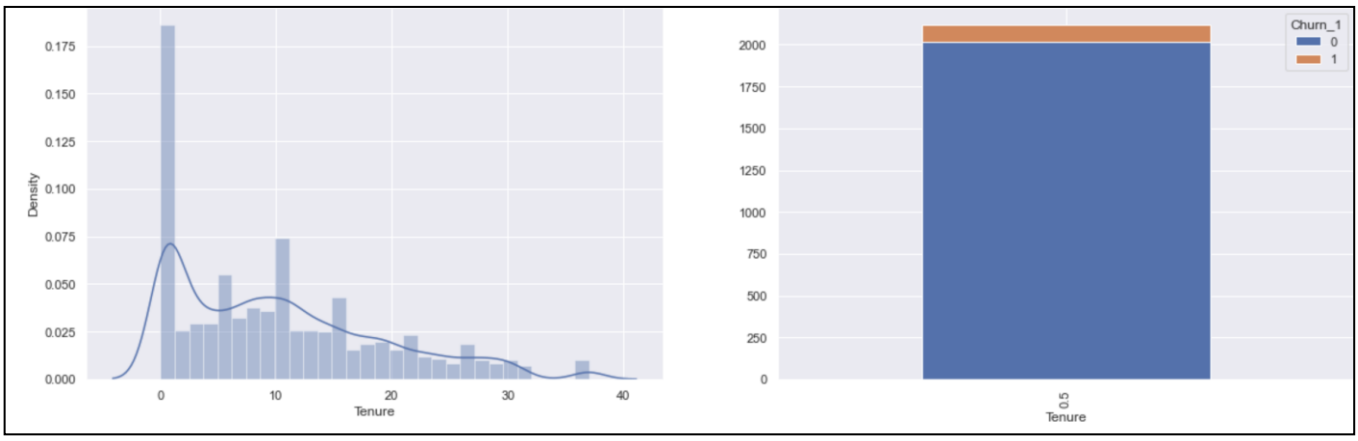


Figure 8-1 Tenure and Churn

Insight: Churn is highest between the tenure period 0 to 2

Possible Reasons: Bad first experience or Trial periods/prepaid accounts that expire automatically if no top-up is done within a predefined period. Important to determine between the above two reasons. Based on high customer care calls, complaints registered and low cashback and coupons for low tenure customers, it points to the first reason.

Business recommendations:

- Activation/Onboarding team could extend support beyond the initial setup until customers settle down with the service
- Activation team proactively engages customers for the first month or two
- Customer care take a feedback survey about the process so that any hiccups can be understood and sorted out
- To increase response rates for feedback, gift cards/coupons can be given

8.2. Existing retention programs – Cashback and Coupons

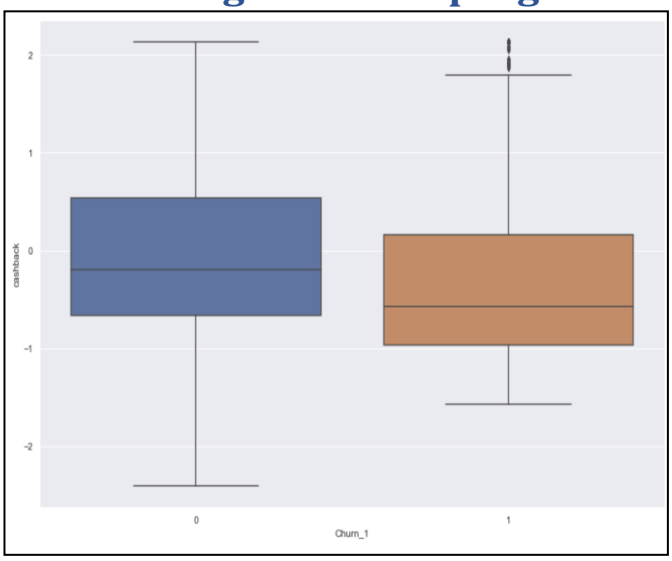


Figure 8-2 Churn and Cashback.

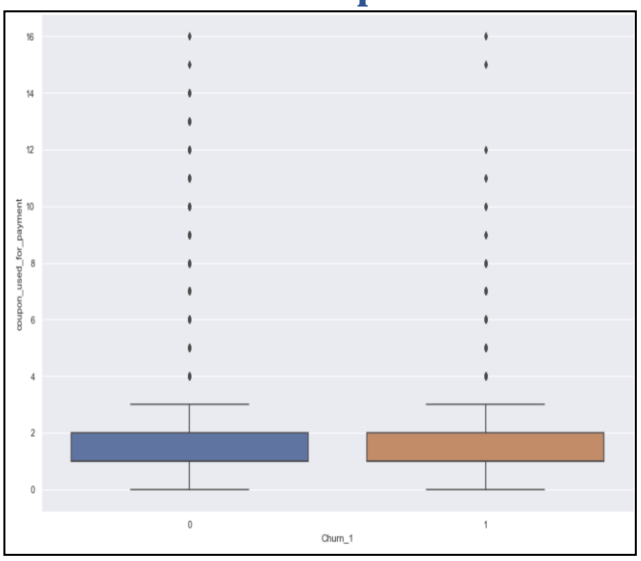


Figure 8-2 Churn and coupons used

- The churned customers as shown in first boxplot have lesser cashback
- The churned and active customers have almost used the same number of coupons for payment

Insight: The current retention programs do not seem to be focusing on the customers with higher risk of churn

Recommendation: Review whether existing cashback and coupon programs are still relevant given the current churn model. If they are not relevant, design new retention programs to address current high risk customer group.

8.3. Churn and Customer care service

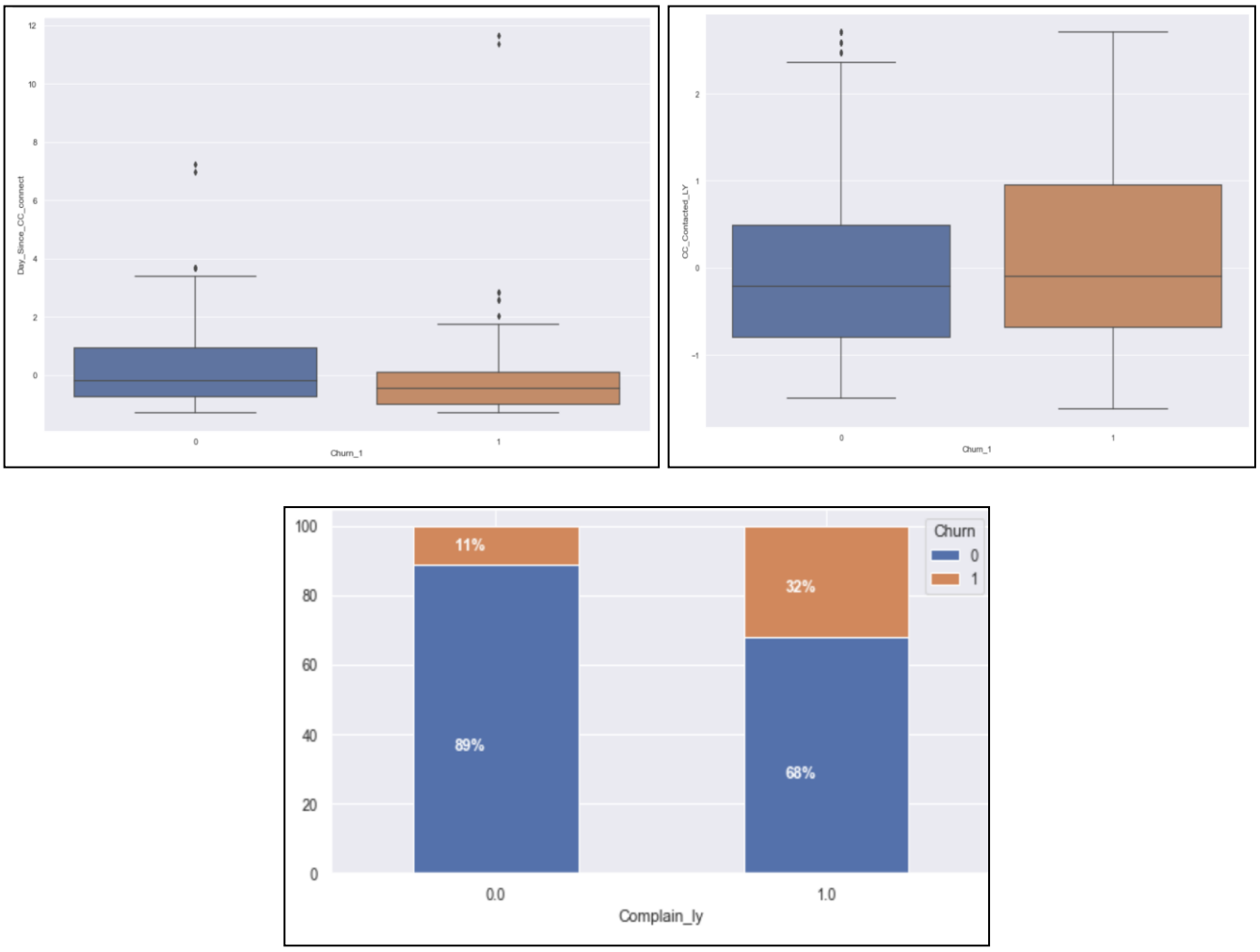


Figure 8-3 Churn and Customer care service

Churned customers seem to have contacted customer care more recently before churning

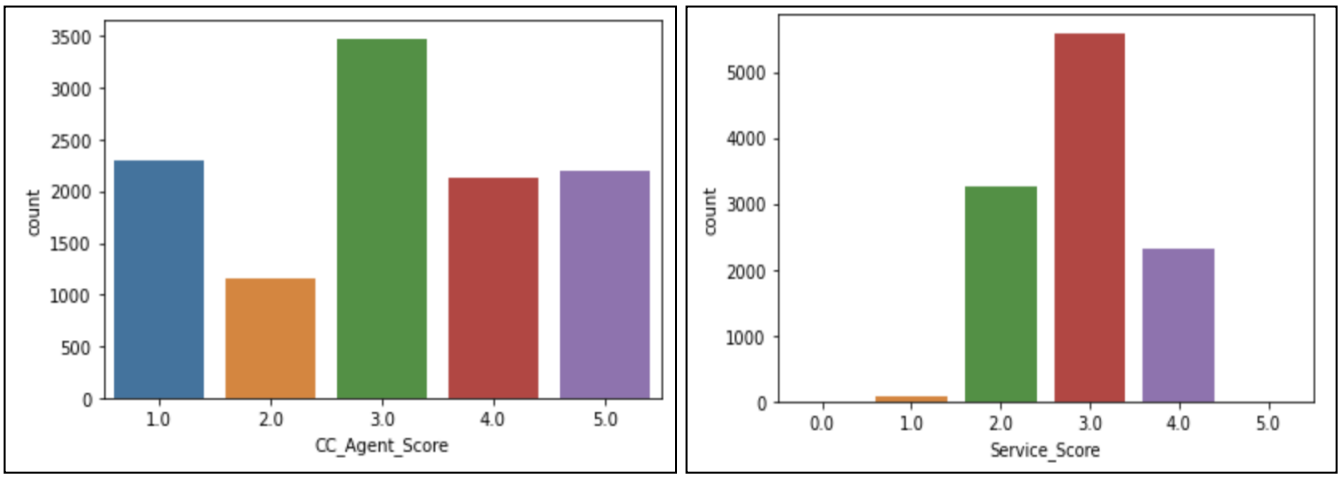
- The number of times churned customers contacted customer care in the year is higher than number of times active customers contacted customer care
- 32% of customers who registered complaint churned Vs 11% of customers who have not registered complaint in last year

Insight: These indicate behavioural changes in customer before churn happens

Recommendation: Analyze Complaints & Customer care contact reasons

- Perform Root cause analysis, identify and fix top reasons
- Establish Service level agreements (if not already present)

8.4. Customer care & Service – Customer perspective



Insight: 78% of customers have rated service as 3 or less than 3
Insight: 61% of customers have rated customer care agents a score of 3 or less than 3

Recommendation: 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

8.5. Revenue per month and Churn

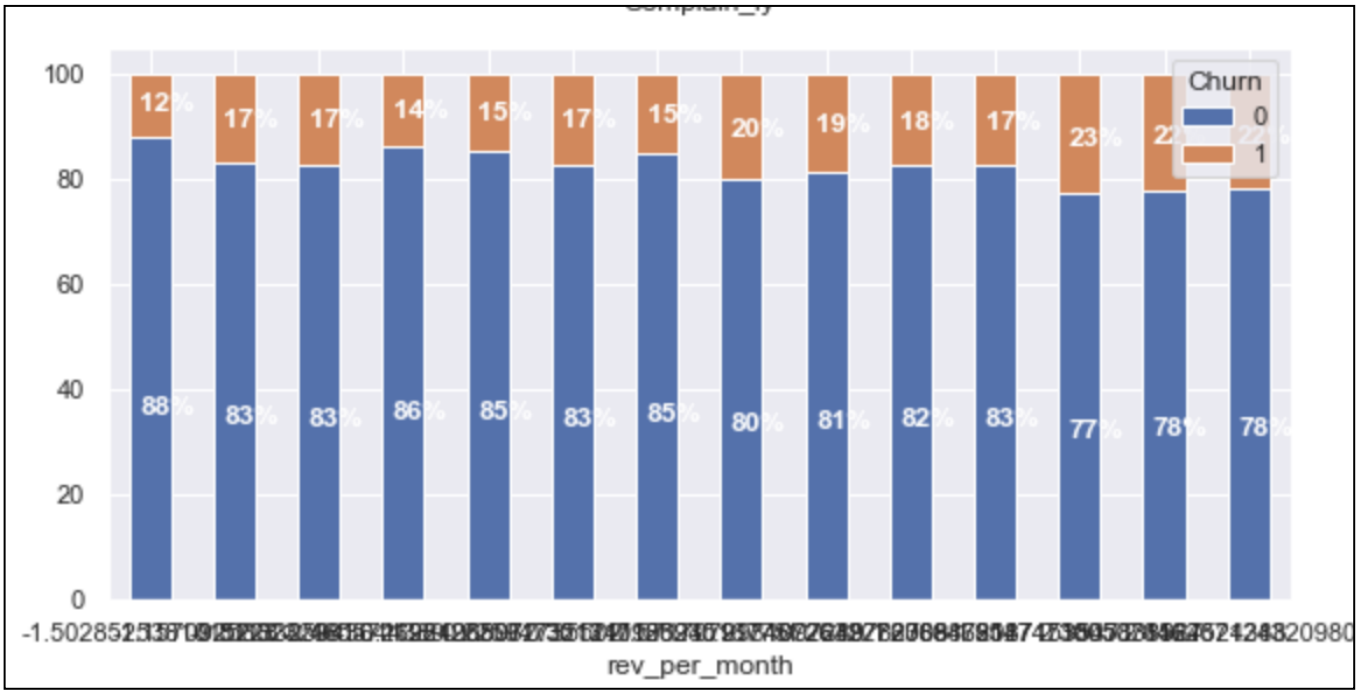


Figure 8-4 High churn in high revenue customers

The % churn of customers in higher revenue group, for revenue >=7 per month is higher than low revenue group customers
Insight: More proportion of high revenue customers are leaving compared to less revenue customers

Recommendation: Create segmented offers for high revenue customers. An illustrative segmentation based on RFM (Recency – Frequency – Monetary) has been shown below along with sample targeted recommendations for each segment. This illustration is based on the test data. A similar segmentation can be done in discussion with client based on what metrics they feel are important to segment based on.

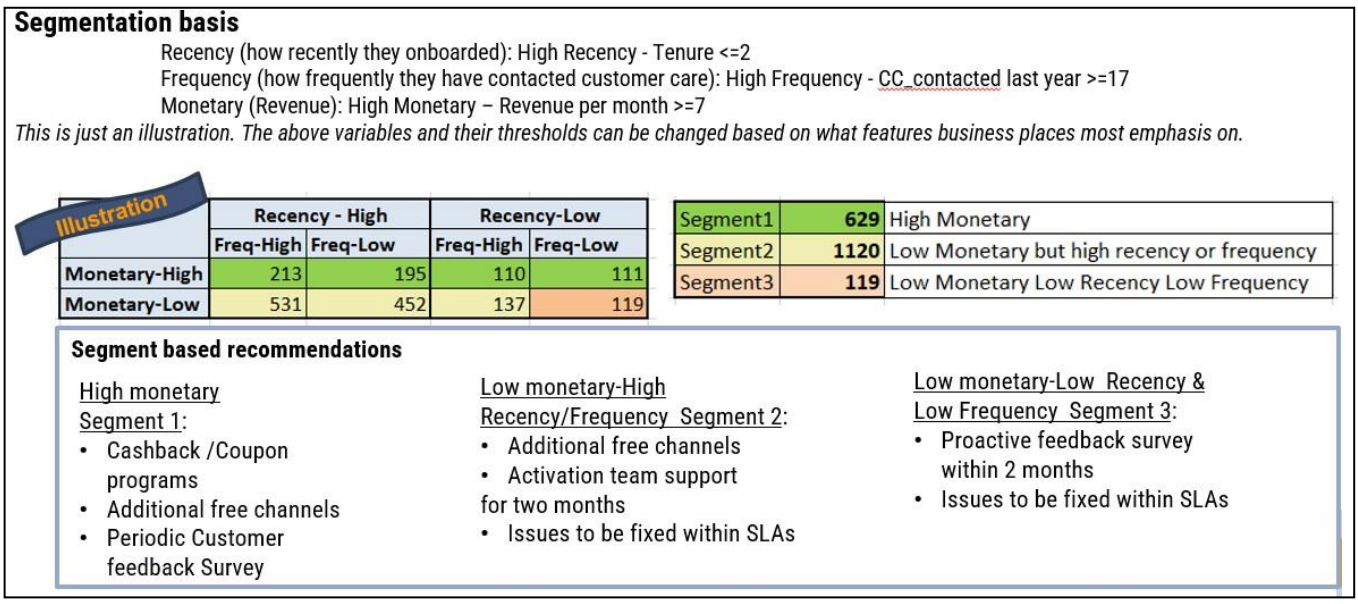


Figure 8-5 Segmented offers illustration

9. Appendix

9.1. Annexure A: Tuning done for Gradient Boost algorithm

```
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]
}
## Best parameters for above grid is learning_rate=0.5, max_depth=9,
max_features=10,min_samples_leaf=6, min_samples_split=20
## loss = deviance. Above is for 100 estimators and friedman_mse which
are default. Best f1_score = 0.917
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]
} # Trying different criterion ('mse') for same best grid settings as grid1
## For mse, score has slightly improved to 0.9124. Now let us try for mae
as well.
param_grid3 = {
'loss': ['deviance'],
'learning_rate': [0.5],
'n_estimators': [101],
'criterion': ['mae'], #
'min_samples_split': [20],
'min_samples_leaf': [6],
'max_depth':[9],
'max_features':[10]
} # Trying different criterion ('mse') for same best grid settings as grid1
## For mse, score has slightly improved to 0.9124. Now let us try for mae
as well.
## mae keeps running forever. So, we'll stick to mse and vary other
parameters of the grid to improve f1-score
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]
} # increasing just estimators for same best grid settings as grid1 with mse ## f1_score = 0.92. Increases with increase in estimators. Let's try for higher number of estimators in next round.
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]
} # estimators settled at 201 with almost same f1_score of 0.9205. Fixing it at 201 and varying other features in next round 43

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]
} # estimators settled at 201 with almost same f1_score of 0.92. Fixing it at 201 and varying other parameters in next round
 # Result: GradientBoostingClassifier(criterion='mse', learning_rate=0.5, max_depth=9,max_features=10, min_samples_leaf=6,
 # min_samples_split=20, n_estimators=201,random_state=0)
 # f1_Score: Not improved - same at 0.9205
 # max_depth and max_features have settled at 9 and 10 as in previous parameters
 # We'll slightly reduce min_samples_split and min_samples_leaf in next round
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]
}
 # Result: GradientBoostingClassifier(criterion='mse', learning_rate=0.5, max_depth=9,max_features=11, min_samples_leaf=6,
 # min_samples_split=15, n_estimators=201,random_state=0)
 # f1_Score: 0.9215
 # In the next iteration, we will fix all values as per the best parameters above and only tune learning rate
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]
}
 # Result: GradientBoostingClassifier(criterion='mse', learning_rate=0.5, max_depth=9,max_features=11, min_samples_leaf=6,
 # min_samples_split=15, n_estimators=201,random_state=0)
 # f1_Score: 0.9215
 # Learning rate settled at middle value of 0.5. We'll finalize this parameter set and evaluate on train and test

9.2. Annexure B: References

1. DTH industry: A glimpse of profits at last! | Business Standard News (business-standard.com)
2. Forbes India - Direct To My Pocket: The DTH Tug Of War
3. Customer Retention Marketing vs. Customer Acquisition Marketing | OutboundEngine
4. The Precision-Recall Plot Is More Informative than the ROC Plot When Evaluating Binary Classifiers on Imbalanced Datasets (plos.org)