CAPSTONE PROJECT
FINAL REPORT

Topic - Customer Churn Prediction.

By Vedika N. Kshatriya.

INDEX

Contents

List of Figures

List of Appendix Figures

List of Tables

List of Appendix Tables

1. INTRODUCTION

a) Brief introduction about the problem statement

Given historical dataset is of Supervised learning. Depending on the nature of the problem, can say that it involves classification tasks. The supervised classification eCommerce churn prediction problem statement involves developing a predictive model that can classify customers as either churned or non-churned based on historical data and relevant features. The goal is to build a model that can accurately predict whether a customer is likely to churn or not.

A predictive model that can identify customers who are likely to stop using an eCommerce platform. The goal is to anticipate and prevent customer churn, which refers to the loss of customers or their disengagement from a services or products.

b) Need of the project

The project is important because Churn Prediction and analysis will allow e-commerce companies to anticipate and determine which clients are susceptible to migrate. As Churn predicted in e-commerce will help to know the real value of the potential loss of those clients to take the necessary retention measures to reduce or avoid their migration. The project goals are as follows –

- **To propose commercial actions aimed at maintaining clients that are showing signs of churn and offer them customized offers.**
- **To develop prediction models for the customer churns in Ecommerce Company, analyzing different attributes related to customer churn.**
- **To follow the data mining methodology in a structured way based on the modelling, evaluation, and implementation of predicted models.**
- **To apply three different machine learning models.**

1. EDA AND BUSINESS IMPLICATION

a) Non – Visual understanding of the data

- **Total there are 11260 rows and 19 columns.**
- **There are 2 variables which have int64 datatype, 5 variables of float64 datatype and 12 variables is of object datatype.**
- **Here, it is observable that data type of target variable Churn is Integer. To which I will convert in Object data type.**
- **There are no duplicate records present in the dataset.**
- **Null values are present in the dataset.**
- **It is observable that data type of target variable Churn is Integer.**
- **Data seems normally distributed.**
- **Data type of cashback variable is Object.**

b) Visual understanding of the data – Univariate Analysis, Bivariate Analysis and Multivariate analysis.

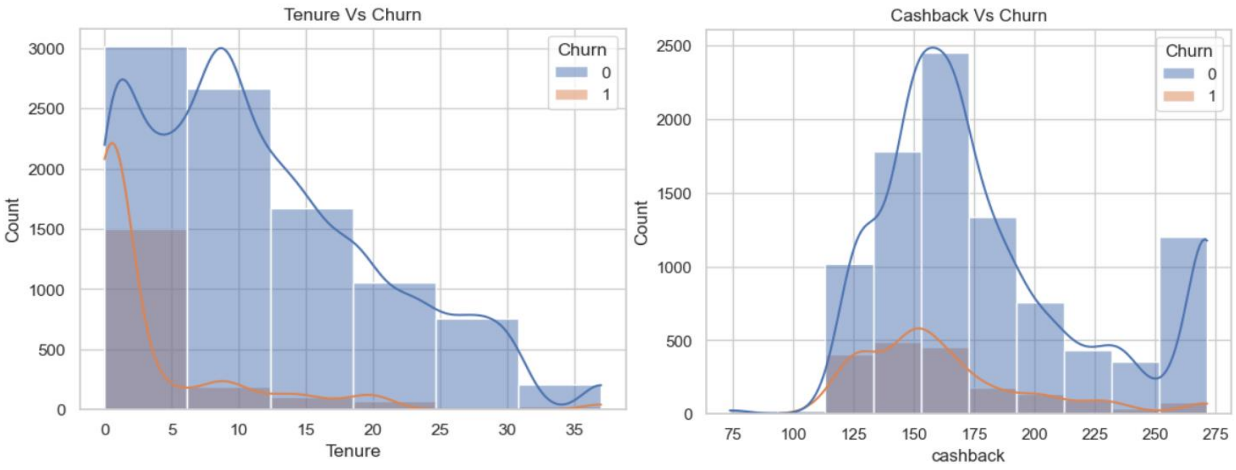


Figure 1 – Tenure vs churn and Cashback vs churn

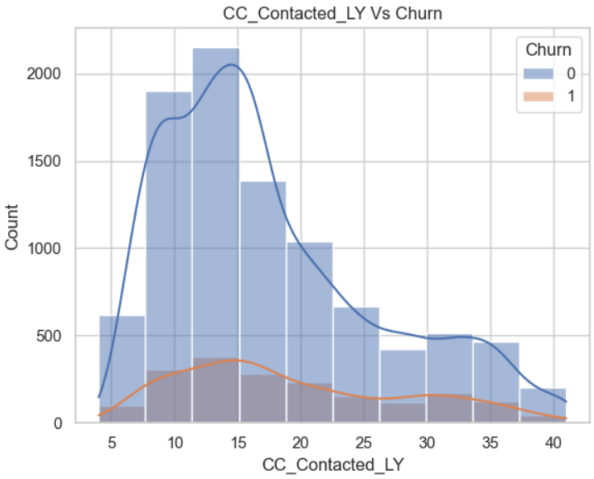


Figure 2 – CC contacted vs Churn

- **Tenure - Majority customers are those who are having tenure between 0 to 12 and Customer who churned are higher in case of customer whose tenure of account is between 0 to 6.**
- **CC Contacted LY – Customers who are churned contacted customer care 10 to 20 times.**
- **Cashback – Count of customers is highest who has earned the highest cashback which ranges between 130 to 195. Churners are those customers who earned the cashback between 125 to 175.**

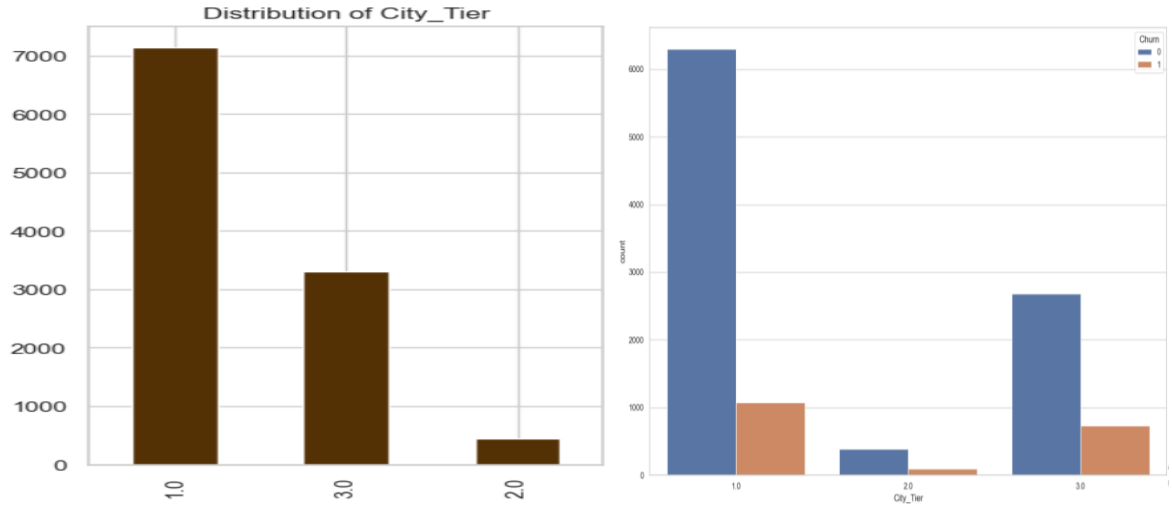


Figure 3 – City Tier vs Churn

- **City Tier – Higher number of customers who are churned and not churned are from City Tier 1 followed by City tier 3 and City Tier 2.**

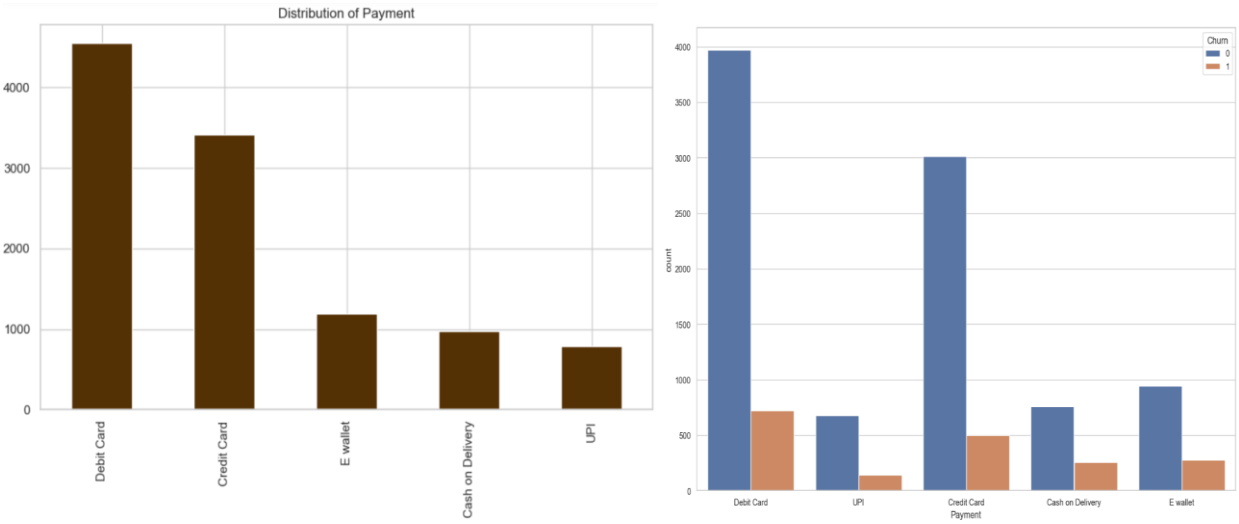


Figure 4 – Payment Vs Churn

- **Payment – Customers prefers payment by way of Debit card and Credit card payment.**

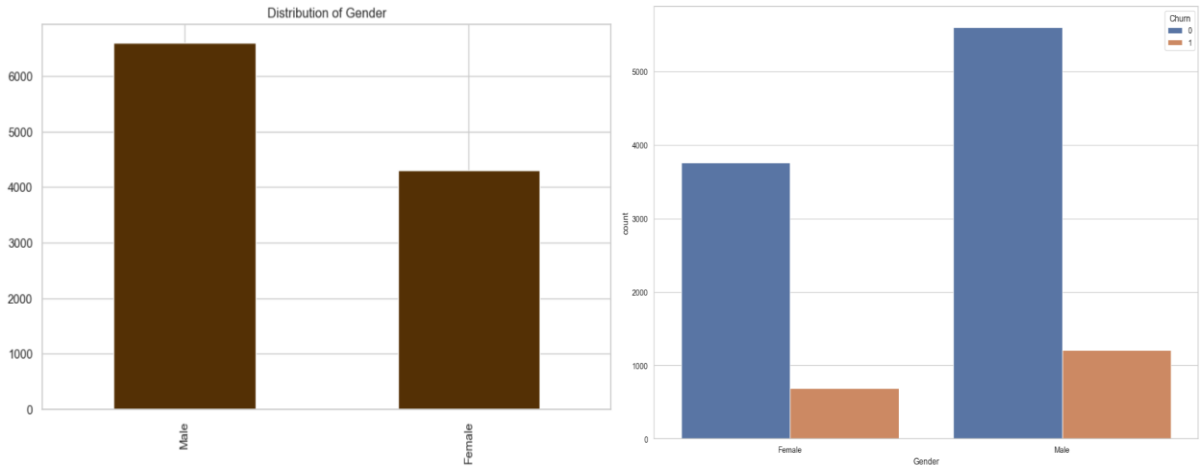


Figure 5 – Gender Vs Churn

- **Gender – More than 1000 male customers have been churned which is more in comparison with female customers.**

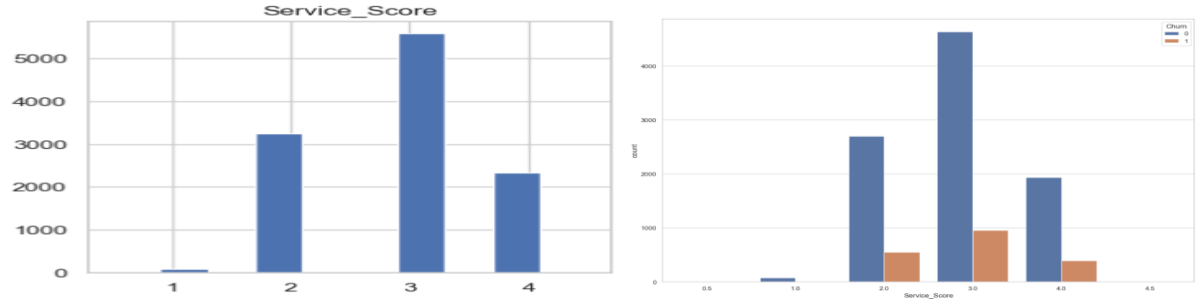


Figure 6 – Service Score Vs Churn.

- **Service Score – Satisfaction score given by customers of the account on service provided by company is ranges between 2 to 4. Customers who have given satisfaction score 3 has a greater number of customers churned.**

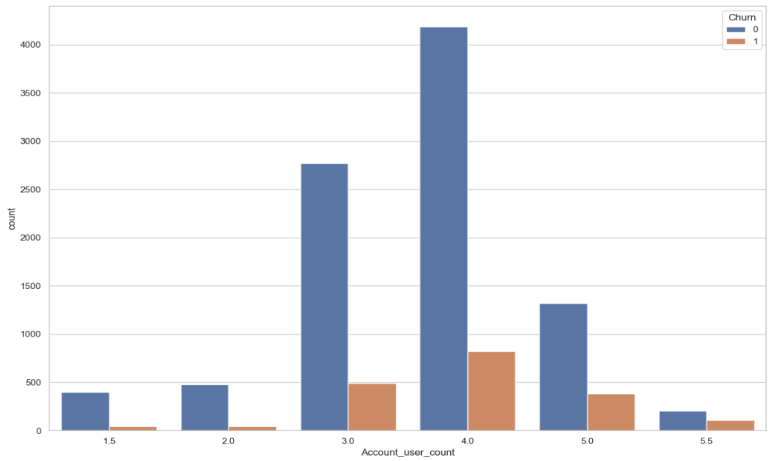


Figure 7 – Account User Count Vs Churn.

- **Account user Count - More than 4000 customers, who have account, Number of customers tagged with their account is 4.**
- **Customers who are churned is more in case of customers who have 4 accounts tagged in their a/c.**

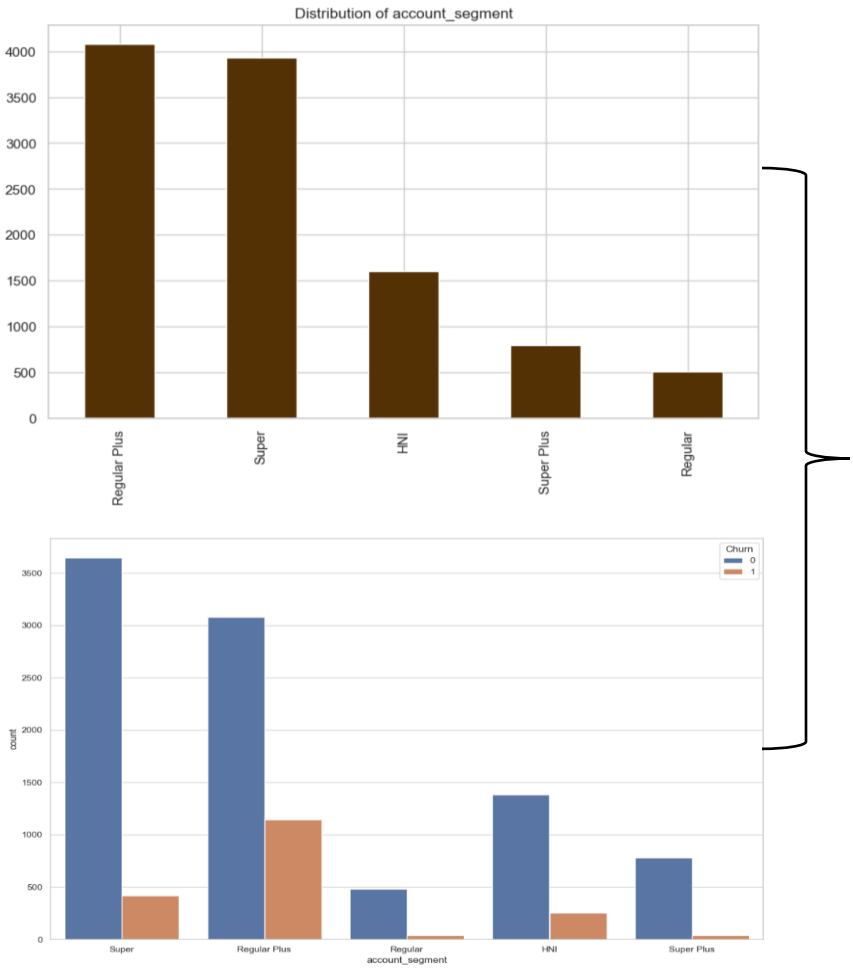


Figure 8 – Account Segment Vs Churn.

- **Account Segment – Account segmentation is done on the basis of customers spend. Number of customers who are in Super segment is more than 3500 followed by customers who segmented as Regular Plus.**
- **Higher number of customers churned is from Regular Plus segmentation and lowest number of customers churned is from Regular and Super Plus segmentation.**

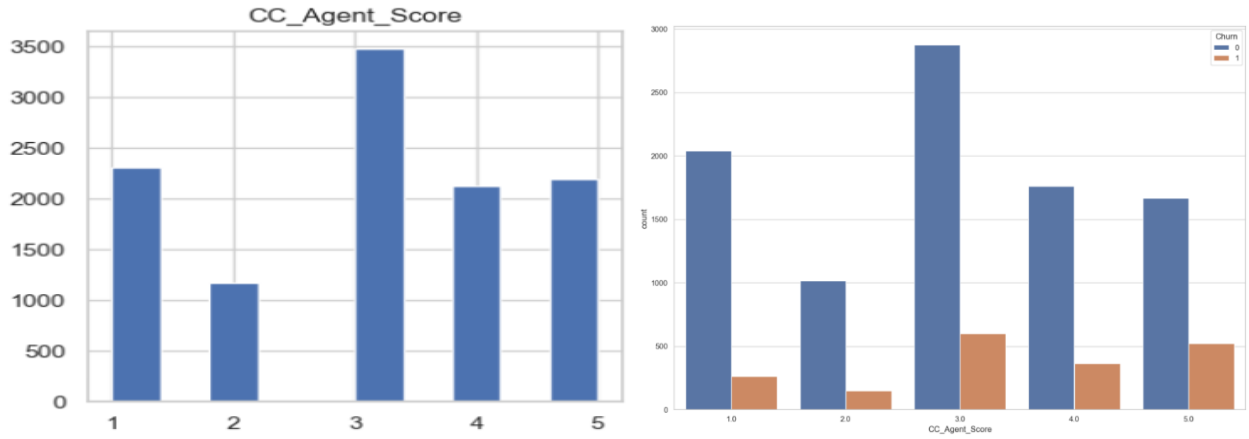


Figure 9 – CC Agent Score Vs Churn

- **CC Agent Score – More than 2500 customers of the account given Satisfaction score on customer care service provided by company is 3. Also, where we can see that it also has more than 500 churned customers.**

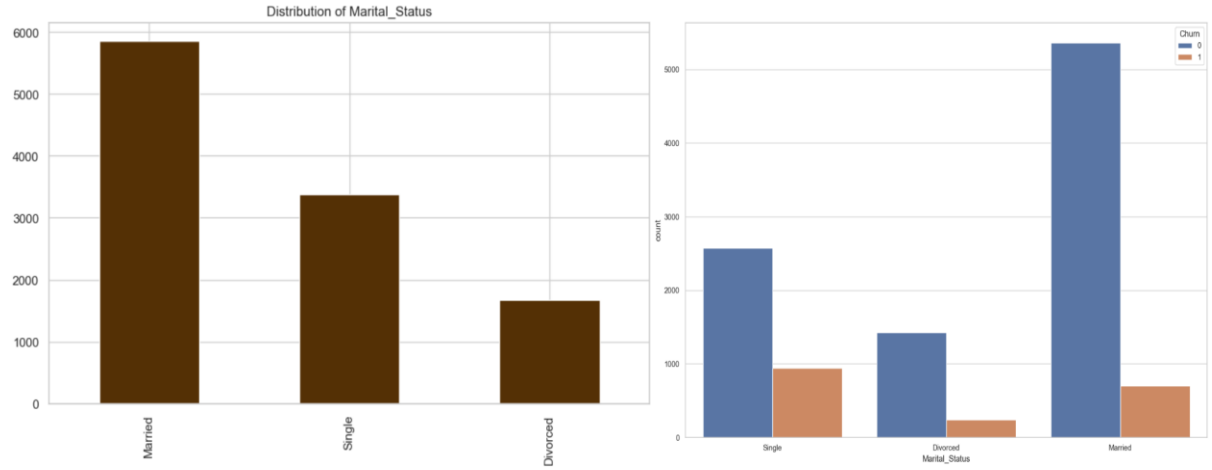


Figure 10 – Marital Status Vs Churn.

- **Marital Status – There are more than 5000 customers who are married. More than 2000 customers are Single and More than 1000 customers are Divorced. But churned customers are more in case of single customers.**

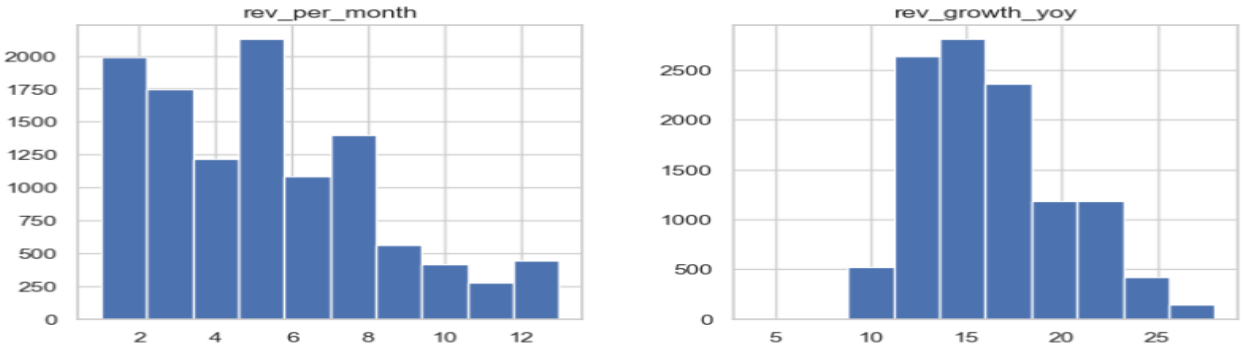


Figure 11 – Rev per Month and Rev Growth YOY Vs Churn

- **Rev per month - Churned customers are more in in case of accounts who have generated monthly average revenue ranges between 2 to 5**
- **Rev growth yoy - Number of customers churned is higher is case of customers, whose account have revenue growth percentage ranges between 12 to 16.**

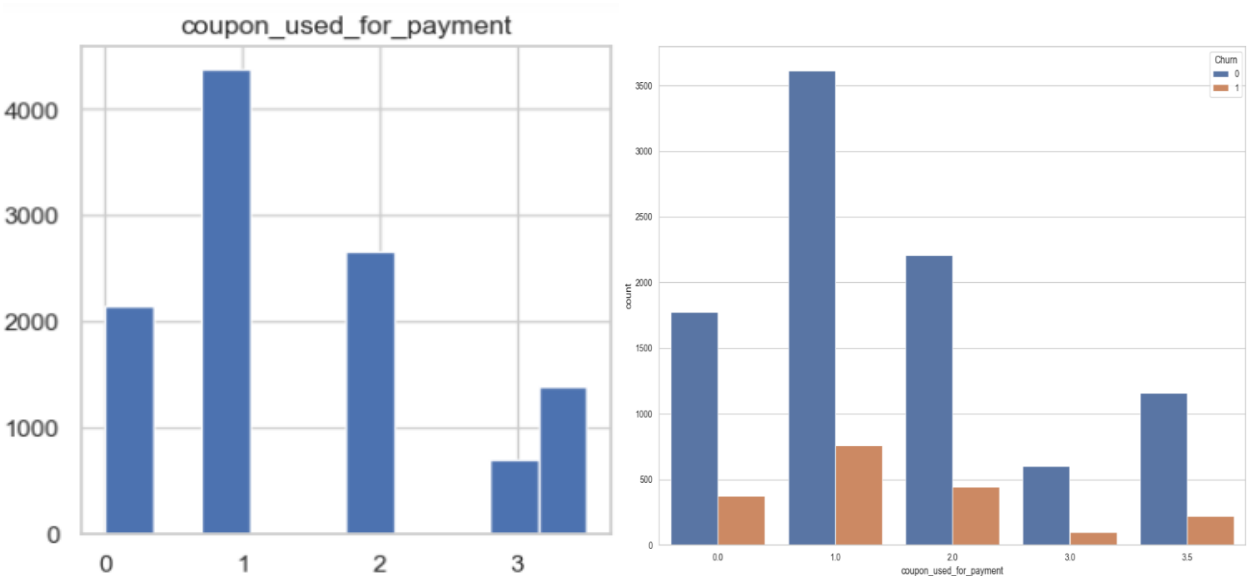


Figure 12 – Coupon Used for Payment Vs Churn

- **Coupon used for payment - Only 1 time coupon is used by more than 3500 customers for the payment. Churned customers are more in case where only 1- or 2-times coupon is used for the payment.**

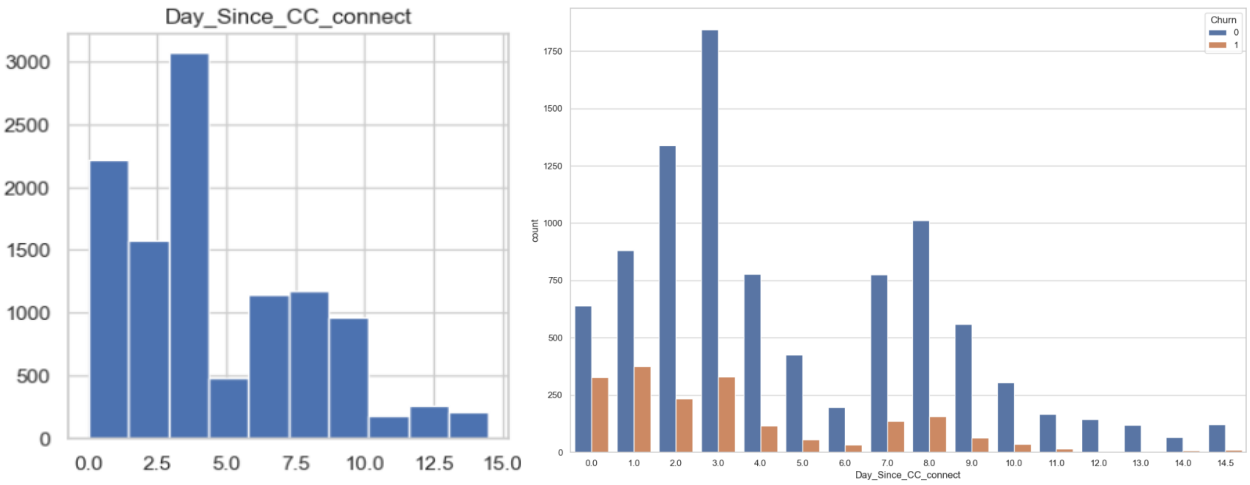
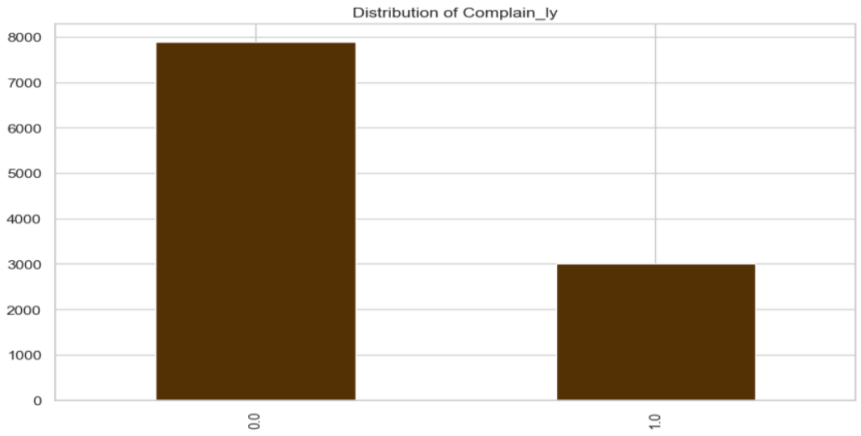


Figure 13 – Day Since CC Connect Vs Churn

- **Day since cc connect - Maximum number of customers churned where number of days since no customers in the account has contacted the customer care is between 0 to 3.**



- **Complain ly - More than 2000 customers have lodged complain in last 12 months. Churned customers are more in case where customers have raised the complaint.**

Figure 14 – Complain LY Vs Churn.

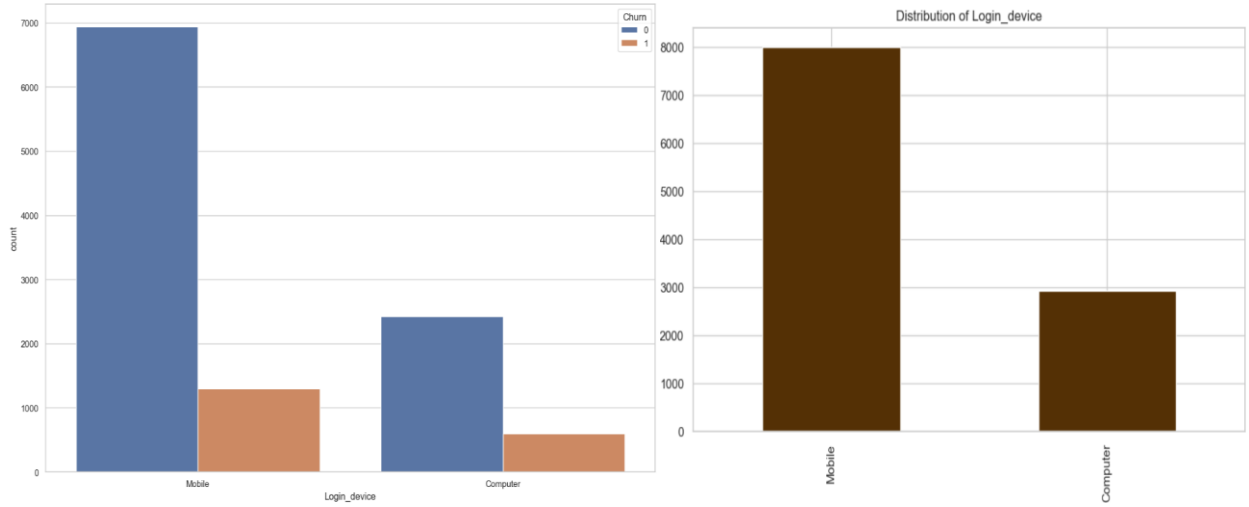


Figure 15 – Login Device Vs Churn

- **Login device - Majority number of customers use Mobile as login device and it also has higher number of churned customers.**

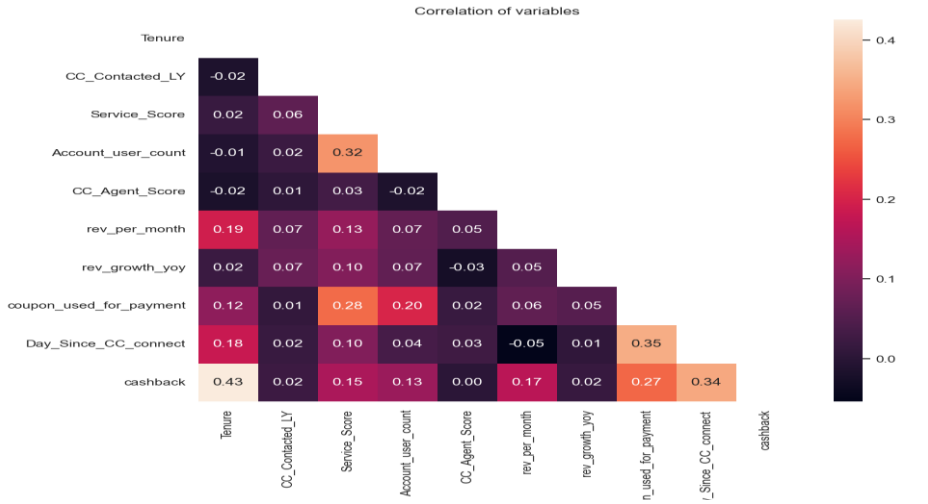
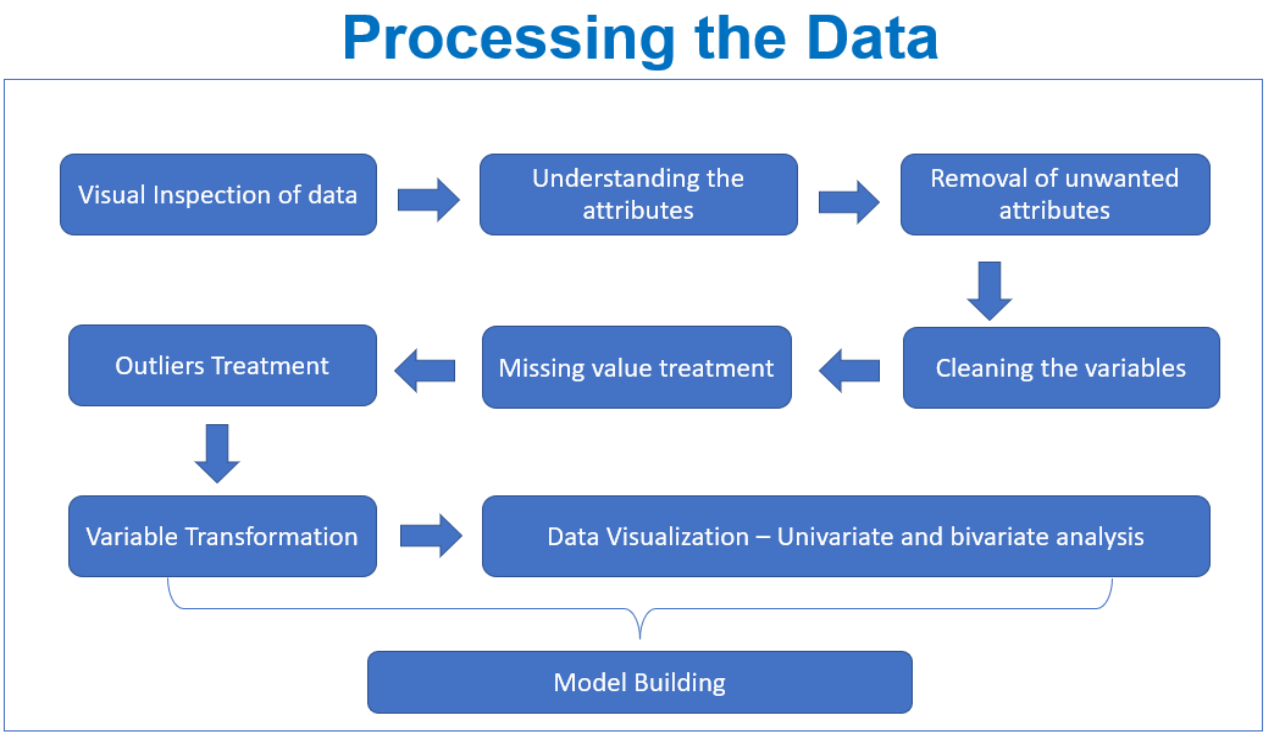


Figure 16 – Correlation Heatmap of Variables.

From the correlation plot, we can see that various attributes are not correlated to each other. Tenure and Cashback is comparatively correlated with each other. Same we can see that in case of attribute Day since CC connect and coupons used for payment.

2. DATA CLEANING AND PRE-PROCCESSING.

Approach used for identifying and treating missing values and outlier treatment



1. Treatment of Missing Value

After the visual inspection of the data, it has been observed that there are null values present in the data set. By using is_null() function in python, we can check the count of null values.

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

Table 1 – Missing values before cleaning of variables.

Also, it has been noticed that some special characters like #, @, $, &&&&, *, + are there in some variables. So, I decided to treat them by converting these special characters into the missing values.

In feature Gender, there are 4 unique values are there. So, I decided to replace 'F' with 'Female' and 'M' with 'Male'.

In feature 'account_segment' there are 7 unique values. Hence, I replace 'Regular +' to 'Regular Plus' and 'Super +' to 'Super Plus'.

By referring the below table, we can see that after replacing Special characters to missing values, increase in missing values can be seen.
Therefore, Total missing values in the data is 2.15% of the data.

Churn	0
Tenure	218
City_Tier	112
CC_Contacted_LY	102
Payment	109
Gender	108
Service_Score	98
Account_user_count	444
account_segment	97
CC_Agent_Score	116
Marital_Status	212
rev_per_month	791
Complain_ly	357
rev_growth_yoy	3
coupon_used_for_payment	3
Day_Since_CC_connect	358
cashback	473
Login_device	760
dtype: int64	

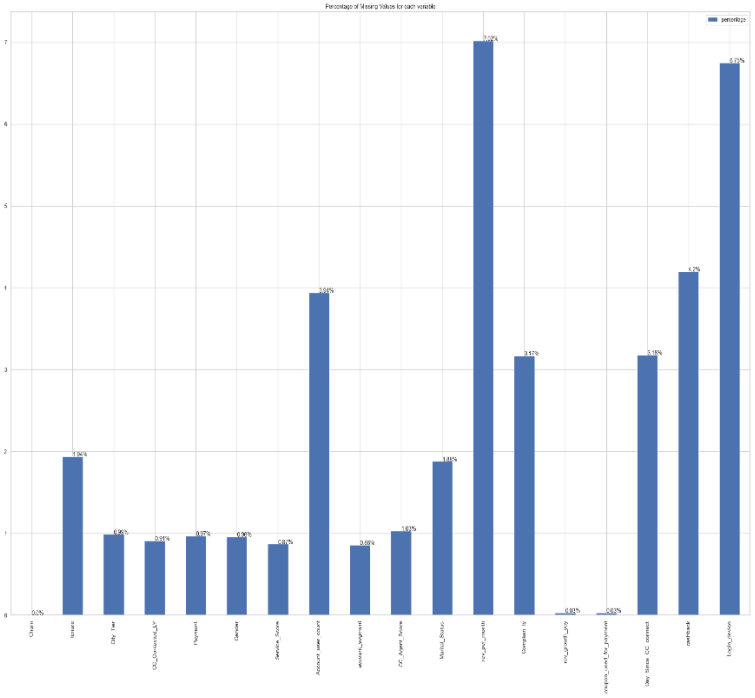


Table 2 – Increase in missing values after cleaning the variables.

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	

Table 3 – Missing values after its treatment.

Feature 'rev_per_month' has 7.02% missing values followed by 'Login_device' which has 6.75% of missing values.

I have treated missing values in categorical variable by using function 'fillna()' and replacing them with mode values.

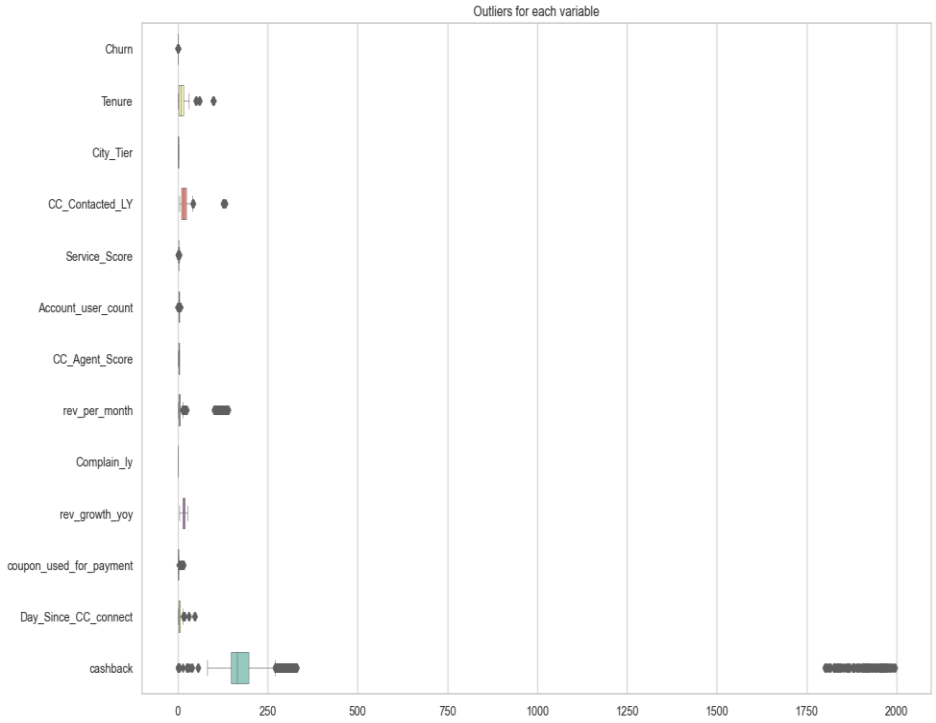
In case of Numerical variables, missing values are replaced by median using 'fillna()' function.

We can check that there are no null values present in the data.

2. Outliers Treatment

The handling of outliers is very important during the data preprocessing pipeline as the presence of outliers can prevent the model to perform best.

With the help of the following boxplots, I observed that there are some extreme values i.e., Outliers.



Outliers were treated by using Winsorization, i.e., bringing the larger outliers (Data points above the Q3 + 1.5 * IQR value) to the upper whisker value and bringing the smaller outliers (Data points below the Q1 - 1.5 * IQR value) to the lower whisker.

Figure 17 – Presence of outliers

Effects of outlier treatment on variables can be seen from the following boxplots

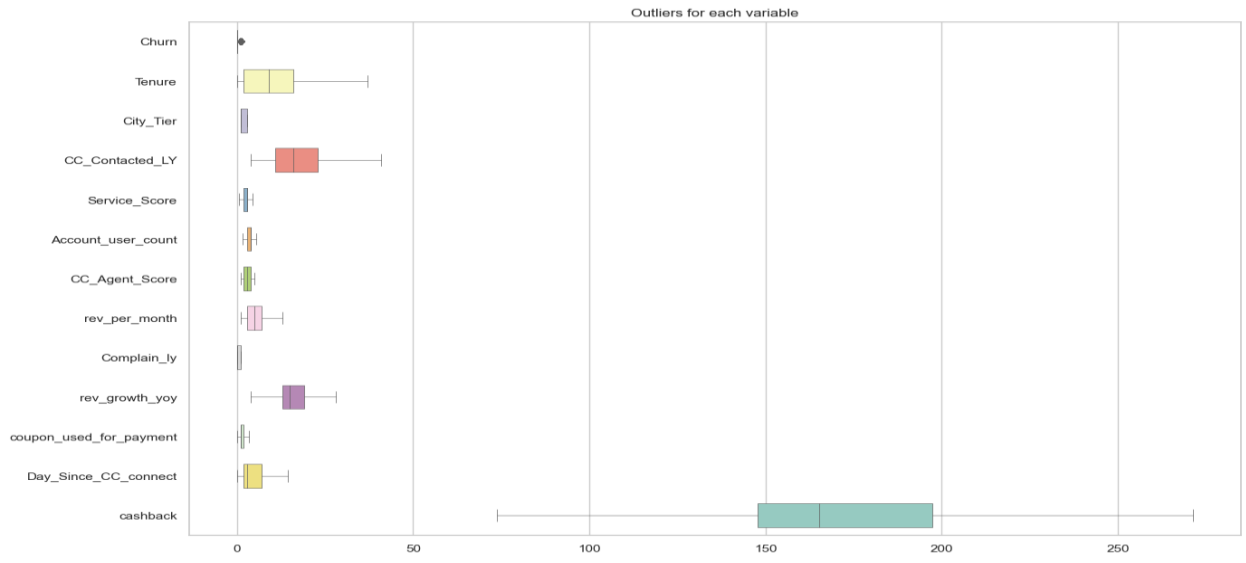


Figure 18 - Presence of outliers after treatment of outliers.

3. Need for variable transformation

Variable 'Churn', 'City_Tier' & 'Complain_ly' has data type of Integer and Float. These variables should of categorical nature. Therefore, by using 'astype()' function in python I changed the data type of variables to Object.

	count	unique	top	freq
Churn	10915	2	0	9077
City_Tier	10915.0	3.0	1.0	7148.0
Payment	10915	5	Debit Card	4549
Gender	10915	2	Male	6604
account_segment	10915	5	Regular Plus	4083
Marital_Status	10915	3	Married	5865
Complain_ly	10915.0	2.0	0.0	7901.0
Login_device	10915	2	Mobile	8001

Table 4 - Data type – Object Variables

4. Variables removed or added and why

- **Account ID is unwanted variable. Hence, I decided to drop it.**
- **In my opinion there is no need of addition of new variables.**

3. MODEL BUILDING

1. Modeling approached used and why

- **Currently the efforts to retain the customers has been very reactive. The management team is keen to take more initiatives on this front and have a targeted proactive strategy.**
- **Extensive experimentation with more than 5 different models was done to identify that could predict customer behavior so that company can take proactive steps to retain the customers wherever possible.**
- **The data was slightly imbalanced (Majority Class : Minority Class = 83:17). Hence, appropriate model evaluation metric was required to be chosen. Recall for minority class was used to finalize the best model.**

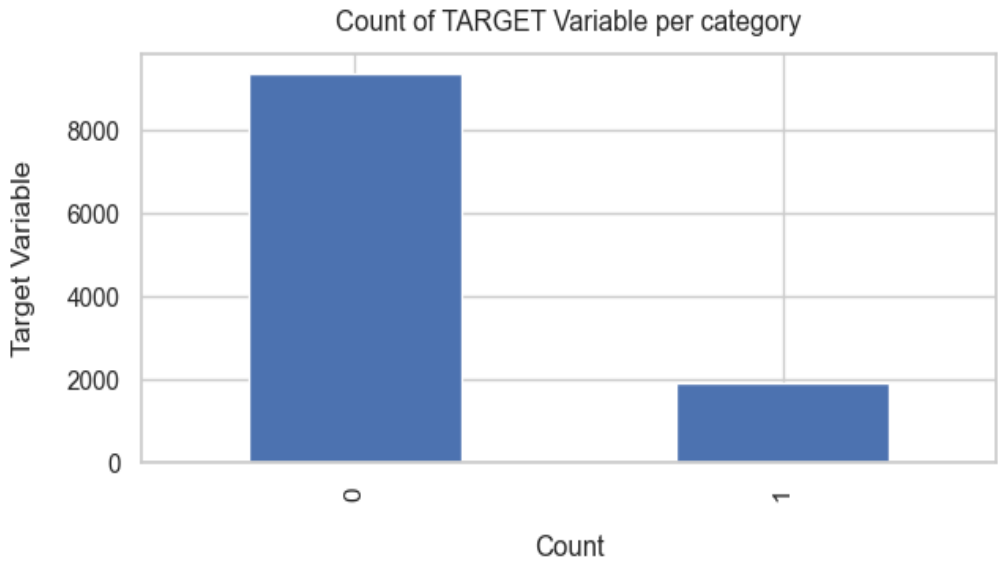


Figure 19 – Distribution of Churn

- **1896 customers are churned customers and 9364 are not churned.**

Addressing Unbalanced Data in Business:

Resampling Techniques:

- **Undersampling: This involves randomly removing samples from the majority class to balance the data. However, it may lead to the loss of important information.**
- **Oversampling: This involves duplicating or synthetically generating new samples for the minority class to balance the data. Techniques like SMOTE (Synthetic Minority Over-sampling Technique) can be employed.**
- **Hybrid Approaches: Combining undersampling and oversampling techniques can be effective in achieving a more balanced dataset.**

- **Ensemble Methods: Utilizing ensemble techniques, such as boosting algorithms (e.g., AdaBoost or XGBoost), can help improve the performance on minority classes by combining multiple weak classifiers.**
- **Collecting More Data: In some cases, collecting additional data for the minority class can help improve the balance and overall performance of the model. This approach may not always be feasible or cost-effective, but it is worth considering if possible.**

Before Building any model, we need to do some preprocessing on the data.

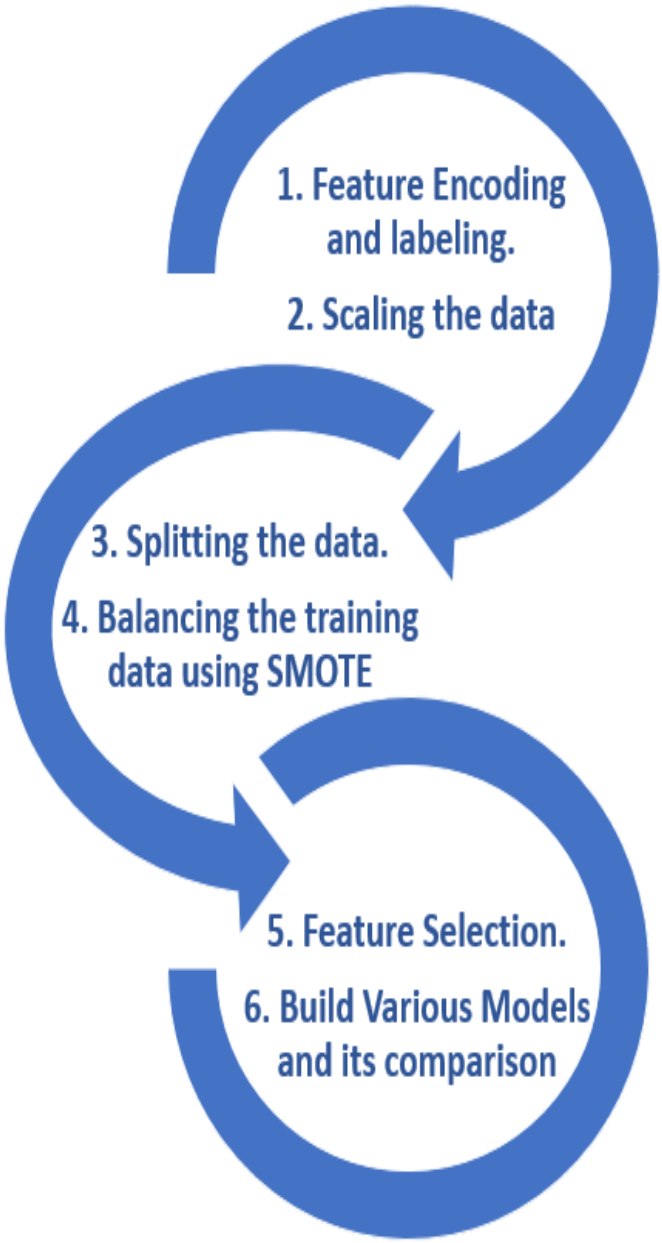


Figure 20 – Steps followed before model building.

1. Feature Encoding and labeling –

For binary class variables, need to do Label Encoding. And to encode other categorical variables I have used get_dummies() function.

2. Scaling the data –

By using StandardScaler () function, all the numerical columns are scaled.

3. Splitting the data –

Separated dependent and independent variables. Split the data into training and testing data set.

4. Balancing the training data using SMOTE –

SMOTE is specifically designed to oversample the minority class by interpolating new synthetic samples between existing minority class samples.

5. Feature Selection –

By using Recursive Feature Elimination and p-value, keeping top 14 variable which are important for the dataset.

6. Build Various Models and its comparison –

More than 5 models build and compared.

Comparisons of the models built with 14 important features –

Model Comparison								
Model Name	Train Data				Test Data			
	Accuracy	Precision	Recall	F1 Score	Accuracy	Precision	Recall	F1 Score
Logistic Regression	0.73	0.74	0.69	0.72	0.74	0.77	0.70	0.73
Decision Tree Classifier	0.75	0.79	0.69	0.74	0.75	0.79	0.70	0.74
Random Forest Classifier	0.75	0.77	0.69	0.73	0.74	0.78	0.69	0.73
Linear Discriminant Analysis	0.73	0.75	0.67	0.71	0.74	0.78	0.68	0.73
KNN Model	0.75	0.78	0.70	0.74	0.75	0.78	0.70	0.74
Guassian Naïve Bayes	0.64	0.61	0.75	0.67	0.65	0.63	0.76	0.69
Bagging Classifier	0.75	0.78	0.70	0.74	0.75	0.79	0.70	0.74
ADA Boost Classifier	0.72	0.74	0.69	0.71	0.75	0.79	0.70	0.74

Table 5 - Comparisons of the models built with 14 important features.

- **8 models tried to finalize the best amongst them. The evaluation metric comparison seen above.**
- **Decision tree classifier, Bagging classifier, ADA boost classifier and Gaussian Naïve Bayes model seems to be giving best results in case of accuracy, precision and recall.**
- **Keeping recall as a metric of comparison Gaussian Naïve Bayes could be the best model for churn prediction.**
- **Recall = True Positives / (True Positives + False negative)**

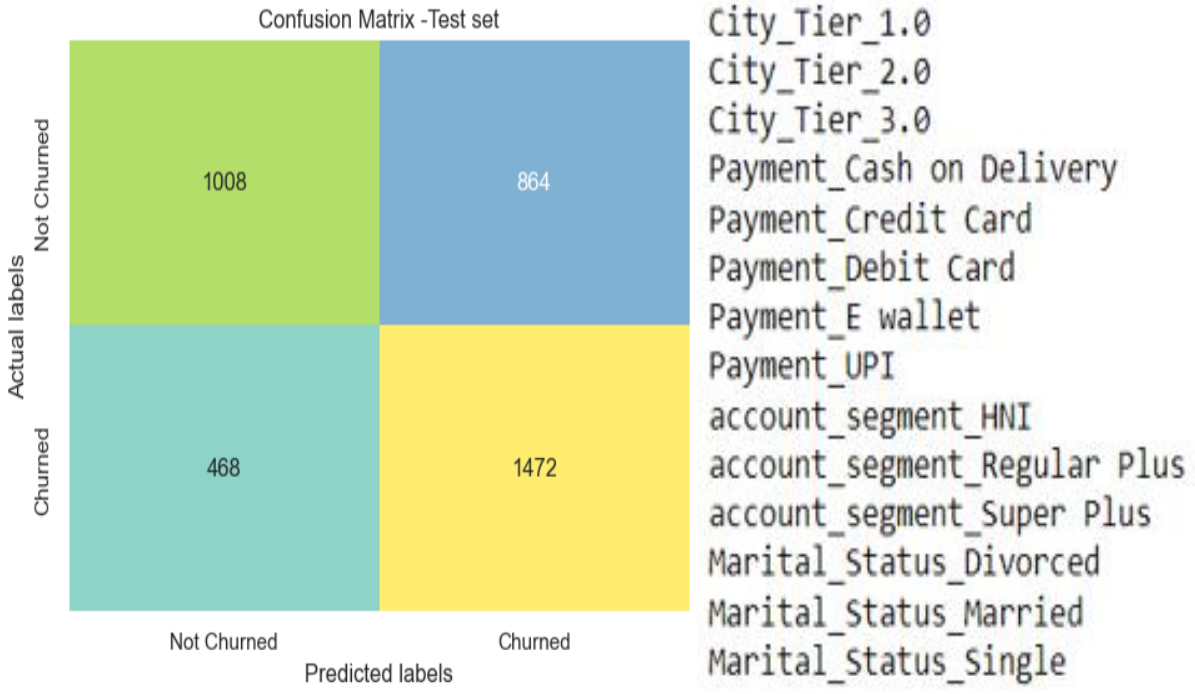


Table 6 – Confusion matrix For Gaussian Naïve Bayes and 14 Important features

In comparison with other model, confusion matrix for Gaussian naïve bayes model's shows better result.

Comparisons of the models built with 20 important features –

Model Comparison								
Model Name	Train Data				Test Data			
	Accuracy	Precision	Recall	F1 Score	Accuracy	Precision	Recall	F1 Score
Logistic Regression	0.84	0.83	0.84	0.84	0.84	0.85	0.84	0.84
Logistic Regression - After Tuning	0.84	0.84	0.84	0.84	0.84	0.85	0.83	0.84
Decision Tree Classifier - After Tuning	0.99	0.99	0.98	0.98	0.90	0.91	0.90	0.91
Random Forest Classifier - After Tuning	0.91	0.91	0.90	0.91	0.89	0.90	0.88	0.89
Linear Discriminant Analysis	0.82	0.82	0.83	0.82	0.83	0.84	0.82	0.83
KNN Model	0.89	0.86	0.93	0.89	0.88	0.85	0.92	0.88
Guassian Naïve Bayes	0.71	0.67	0.85	0.75	0.72	0.68	0.85	0.76
Bagging Classifier	0.99	0.98	0.99	0.99	0.93	0.94	0.93	0.93
ADA Boost Classifier	0.86	0.86	0.85	0.86	0.93	0.94	0.93	0.93

Table 7 – Comparisons of the models built with 20 important features.

- **KNN Model and bagging Classifier seems to be giving best results in case of accuracy, precision and recall.**
- **Keeping top 20 fetaures and build model with these features is better as all the models perform better.**
- **Hardly any model is overfitted or underfitted.**

Confusion Matrix for KNN Model and Bagging Classifier is as follows –

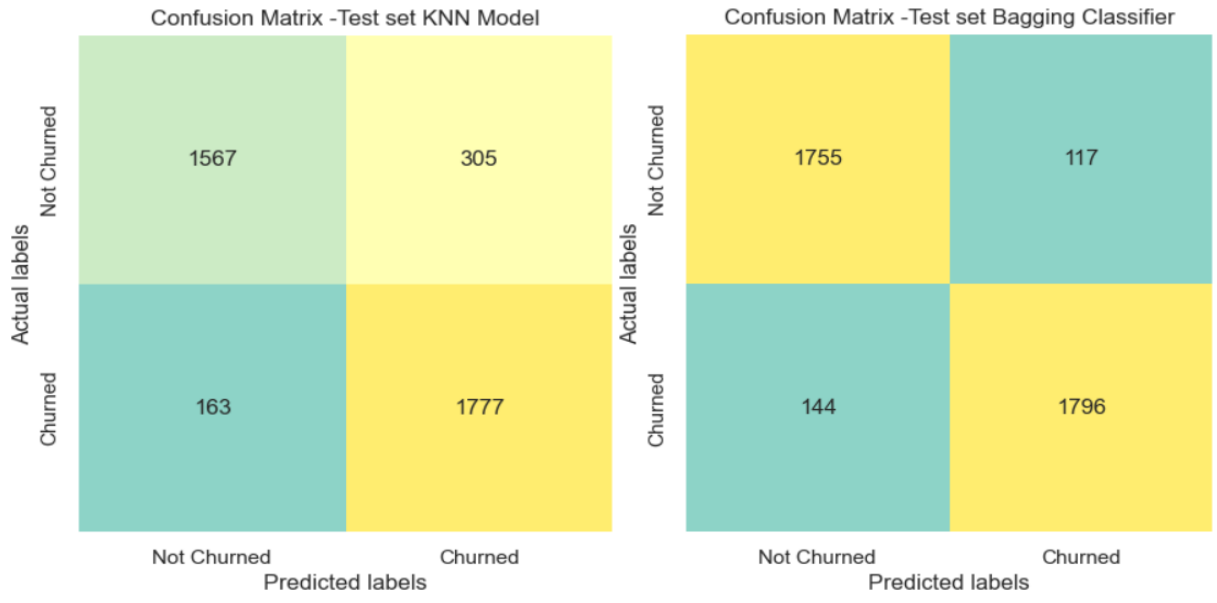


Table 8 – Confusion Matrix for KNN model and Bagging Classifier Model.

List of top 20 Features is as follows –

Tenure	Payment_E wallet
rev_per_month	Payment_UPI
Complain_ly	account_segment_HNI
Login_device	account_segment_Regular
City_Tier_1.0	account_segment_Regular Plus
City_Tier_2.0	account_segment_Super
City_Tier_3.0	account_segment_Super Plus
Payment_Cash on Delivery	Marital_Status_Divorced
Payment_Credit Card	Marital_Status_Married
Payment_Debit Card	Marital_Status_Single

Table 9 – List of top 20 Features.

```
Optimization terminated successfully.
         Current function value: 0.434363
         Iterations 7
                           Logit Regression Results
==============================================================================
Dep. Variable:                  Churn   No. Observations:                12706
Model:                          Logit   Df Residuals:                    12686
Method:                           MLE   Df Model:                           19
Date:                Fri, 16 Jun 2023   Pseudo R-squ.:                  0.3733
Time:                        23:06:45   Log-Likelihood:                -5519.0
converged:                       True   LL-Null:                       -8807.1
Covariance Type:            nonrobust   LLR p-value:                     0.000
================================================================================================
                                   coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------------------------
Tenure                          -1.6370      0.041    -40.379      0.000      -1.716      -1.558
rev_per_month                    0.5391      0.025     21.232      0.000       0.489       0.589
Complain_ly                      1.3820      0.052     26.655      0.000       1.280       1.484
Login_device                    -0.2717      0.052     -5.235      0.000      -0.373      -0.170
City_Tier_1.0                    0.6597      0.145      4.544      0.000       0.375       0.944
City_Tier_2.0                    1.1517      0.191      6.043      0.000       0.778       1.525
City_Tier_3.0                    1.3888      0.148      9.381      0.000       1.099       1.679
Payment_Cash on Delivery        -0.6117      0.143     -4.264      0.000      -0.893      -0.331
Payment_Credit Card             -1.2863      0.128    -10.081      0.000      -1.536      -1.036
Payment_Debit Card              -1.1653      0.125     -9.355      0.000      -1.409      -0.921
Payment_E wallet                -0.7673      0.144     -5.331      0.000      -1.049      -0.485
Payment_UPI                     -1.4750      0.157     -9.388      0.000      -1.783      -1.167
account_segment_HNI              1.0322      0.146      7.052      0.000       0.745       1.319
account_segment_Regular          1.9624      0.189     10.387      0.000       1.592       2.333
account_segment_Regular Plus     1.0387      0.139      7.470      0.000       0.766       1.311
account_segment_Super           -0.3220      0.141     -2.291      0.022      -0.598      -0.046
account_segment_Super Plus       0.4360      0.195      2.238      0.025       0.054       0.818
Marital_Status_Divorced         -1.9554      0.125    -15.591      0.000      -2.201      -1.710
Marital_Status_Married          -2.0045      0.112    -17.893      0.000      -2.224      -1.785
Marital_Status_Single           -0.9311      0.112     -8.308      0.000      -1.151      -0.711
================================================================================================
```

Table 10 – Statistic Model summary of features.

There is no need to drop any features because the p-value of all the features is below 0.05.

4. MODEL VALIDATION

- **KNN Model has made 1777 True Positive (TP), 1567 True Negative (TN), 163 False Negative (FN) and 305 False Positive (FP) predictions out of 3812 customers on the test data.**
- **Bagging Classifier Model has made 1796 True Positive (TP), 1755 True Negative (TN), 144 False Negative (FN) and 117 False Positive (FP) predictions out of 3812 customers on the test data.**
- **Assuming for KNN Model**
  profit of 50$ from one TP prediction
  loss of 50$ from one FN prediction
  an operational cost of 5$ spent on one FP prediction
  Overall Profit = (1777 x 50) - (163 x 50) - (305 x 5) = $79,175
  Profit/customer = 79175/3812 = $20.76

If this model were to make predictions on 20K customers in future, the Net Profit made by the company would be an estimate of $415,398.

- **Assuming for Bagging Classifier Model**
  profit of 50$ from one TP prediction
  loss of 50$ from one FN prediction
  an operational cost of 5$ spent on one FP prediction
  Overall Profit = (1796 x 50) - (144 x 50) - (117 x 5) = $82,015
  Profit/customer = 82015/3812 = $21.51

If this model were to make predictions on 20K customers in future, the Net Profit made by the company would be an estimate of $430,300.

Will go with Bagging Classifier Model.

Model can make wrong predictions such as:

1. **Predicting a customer will churn but in reality, the customer will not**
2. **Predicting a customer will not quit the service but in reality, the customer will churn**

Prediction of concern:

The second prediction is our major concern as customers renouncing the services would lead to loss and our aim is to build a prediction model to minimize the churn

Minimizing false negatives:

Recall score should be maximized. Greater the Recall score, higher the chances of predicting the customers who may churn.

5. FINAL INTERPRETATION / RECOMMENDATION.

a) Interpretation

Building profiles of churned customers w.r.t different account segment –

1. **Account Segment Regular –**

- **a total of 39 churned customers**
- **with a Tenure IQR of 4 to 20 months**
- **largely from Tier1 and Tier3 cities**
- **average of 13 customer care contacts past year**
- **mostly Female customers who preferred E-wallet payment**
- **mostly Single and used Mobile to login**
- **service score of 2 to 4**
- **average**
  - **users per account: 4**
  - **complain: 0.6**
  - **coupons used: 3 to 4 times**
  - **cashback: Rs.271**
  - **reached out to customer care within 3 to 12 days**

2. **Account Segment Regular Plus –**

- **a total of 1102 churned customers**
- **with a Tenure IQR of 1 month or less**
- **largely from Tier1 cities**
- **average of 18 customer care contacts past year, Minimum 4**
- **mostly male customers who preferred Debit card payment**
- **mostly Single and used Mobile to login**
- **service score of 2 to 4**
- **average**
  - **users per account: 4**
  - **complain: 0.6**
  - **coupons used: 1 to 2 times**
  - **cashback: Rs.143**
  - **reached out to customer care within 2 to 14 days**

3. **Account Segment Super –**

- **a total of 408 churned customers**
- **with a Tenure IQR of 4 months or less**
- **largely from Tier1 and Tier 3 cities**
- **average of 21 customers care contacts past year, Minimum 6**
- **mostly male customers who preferred Debit card payment**
- **mostly Single and used Mobile to login**

- service score of 2 to 4
- average
    users per account: 1
    complain: 0.50
    coupons used: 1 to 2 times
    cashback: Rs.173.73
    reached out to customer care within 4 to 11 days.

4. Account segment Super Plus –
- a total of 38 churned customers
- with a Tenure IQR of 2 to 11 months.
- largely from Tier1 and Tier 3 cities
- average of 19 to 20 customers care contacts past year, Minimum 8.
- mostly male customers who preferred Debit card payment
- mostly Single and used Computer to login
- service score of 2 to 4
- average
    users per account: 3 to 4
    complain: 0.60
    coupons used: 3 to 4 times
    cashback: Rs.245.82
    reached out to customer care within 8 to 15 days.

5. Account segment HNI –
- a total of 251 churned customers
- with a Tenure IQR of 0 to 11 months.
- largely from Tier1 and Tier 3 cities
- average of 22 customers care contacts past year, Minimum 6.
- mostly male customers who preferred Debit card payment
- mostly Married and used Mobile to login
- service score of 2 to 4
- average
    users per account: 4
    complain: 0.40
    coupons used: 2 to 4 times
    cashback: Rs.206
    reached out to customer care within 6 to 15 days.

Other Insights –

- **Frequency of churn is higher in –**
  - **City Tier 1.**
  - **Male Customers.**
  - **Single Customers.**
  - **New customers whose tenure of account is between 0 to 5.**
  - **Customers paying with debit cards and credit cards.**
  - **Customers who are preferring mobile phone to login.**
  - **Customers who are using 1 coupon.**
- **Number of days since no customers in the account has contacted the customer care is 3 for highest number of customers which is more than 1750 customers and 14 for lowest number of customers which is less than 250 customers.**
- **Maximum number of customers churned where number of days since no customers in the account has contacted the customer care is between 0 to 3.**
- **More than 1000 customers, Revenue growth percentage is ranges between 13 to 14.**
- **More than 2500 customers of the account given Satisfaction score on customer care service provided by company is 3. Also, where we can see that it also has more than 500 churned customers.**
- **Higher number of customers churned is from Regular Plus segmentation and lowest number of customers churned is from Regular and Super Plus segmentation.**

b) Recommendations

- **New advertising campaigns should be set up by the company to retain the top customers and to attract the new customers.**
- **To retain the customers in all 3 city tiers, company must try to understand the reason behind churning of customers.**
- **Company should offer and implements such strategies, discounts for customers who are making payments via any mode. And also, should resolve the obstacles customers are facing while making payments.**
- **Customers who are segmented other than Regular plus, HNI, Super plus should be targeted.**
- **Single and married customers are churning more than divorced customers. So, exclusive offers can be form to retain this customer base.**
- **Marketing by the company can play important role.**
- **Services provided by the company to the customers should be reviewed and if there is any lacking, that should be improved.**
- **Customer's query should get resolved on prior basis.**
- **Review system should get improved.**

- Company should conduct surveys and take feedbacks from the customers on timely basis to analyze the behavior and opinions of the customers.
- Follow-up calls are recommended.
- Provide targeted offers to Female customers who prefer E-wallet/Mobile, from the Regular account segments.
- Also provide exclusive family offers for Married customers from HNI segment as churn rate is higher among them.
- Exit interview can be conducted to get feedback from outgoing customers and work towards the betterment of the services provided.

APPENDIX

a) Non visual understanding of the data

- **First 5 rows of the data –**

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

APPENDIX TABLE 1 – 1ST FIVE ROWS

- **Last 5 rows of the data –**

	AccountID	Churn	Tenure	City_Tier	CC_Contacted_LY	Payment	Gender	Service_Score	Account_user_count	account_segment	CC_Agent_Score	Mari
11255	31255	0	10	1.0	34.0	Credit Card	Male	3.0	2	Super	1.0	
11256	31256	0	13	1.0	19.0	Credit Card	Male	3.0	5	HNI	5.0	
11257	31257	0	1	1.0	14.0	Debit Card	Male	3.0	2	Super	4.0	
11258	31258	0	23	3.0	11.0	Credit Card	Male	4.0	5	Super	4.0	
11259	31259	0	8	1.0	22.0	Credit Card	Male	3.0	2	Super	3.0	

APPENDIX TABLE 2 – LAST FIVE ROWS

Understanding of attributes –

- **Total there are 11260 rows and 19 columns.**
- **There are 2 variables which have int64 datatype, 5 variables of float64 datatype and 12 variables is of object datatype.**

We can conclude the same by referring the below image –

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

APPENDIX TABLE 3 - DATASET DESCRIPTION

- **Data Summary –**

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

	count	unique	top	freq
Tenure	11158	38	1	1351
Payment	11151	5	Debit Card	4587
Gender	11152	4	Male	6328
Account_user_count	11148	7	4	4569
account_segment	11163	7	Super	4062
Marital_Status	11048	3	Married	5860
rev_per_month	11158	59	3	1746
rev_growth_yoy	11260	20	14	1524
coupon_used_for_payment	11260	20	1	4373
Day_Since_CC_connect	10903	24	3	1816
cashback	10789.0	5693.0	155.62	10.0
Login_device	11039	3	Mobile	7482

APPENDIX TABLE 4 - DATASET SUMMARY.

Here, it is observable that data type of target variable Churn is Integer. To which I will convert in Object data type.

a) Variable Transformation

Variable 'Churn', 'City_Tier' & 'Complain_ly' has data type of Integer and Float. These variables should of categorical nature. Therefore, by using 'astype()' function in python I changed the data type of variables to Object.

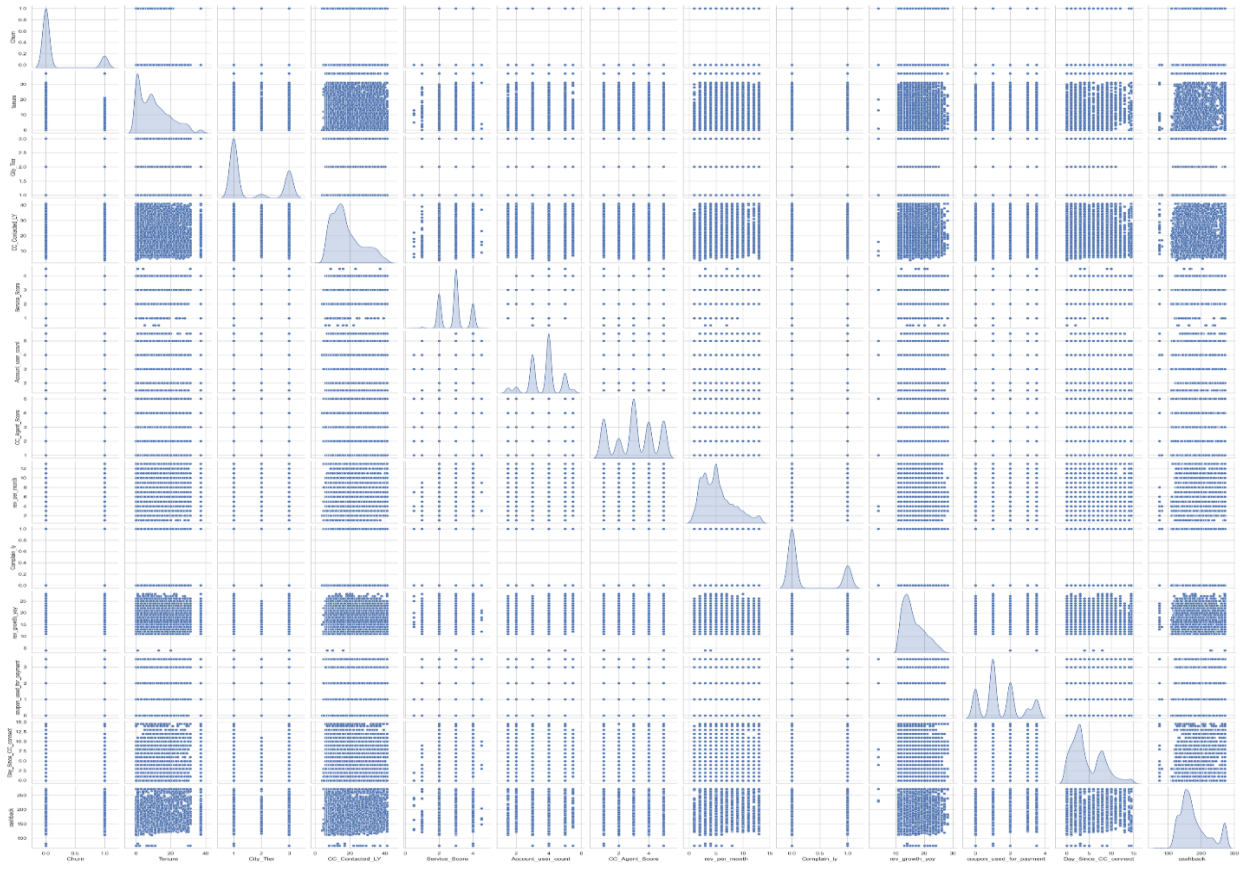
	count	unique	top	freq
Churn	10915	2	0	9077
City_Tier	10915.0	3.0	1.0	7148.0
Payment	10915	5	Debit Card	4549
Gender	10915	2	Male	6604
account_segment	10915	5	Regular Plus	4083
Marital_Status	10915	3	Married	5865
Complain_ly	10915.0	2.0	0.0	7901.0
Login_device	10915	2	Mobile	8001

APPENDIX TABLE 5 – DATA TYPE – OBJECT VARIABLES.

b) Visual Inspection of the data

Pairplot –

Pairplot shows the relationship between the variables in the form of scatterplot and the distribution of the variable in the form of histogram.



APPENDIX FIGURE 1 – PAIRPLOT OF THE VARIABLES.

From the graph, we can see that there is no linear relationship between variables.

c) Model Building and Interpretation with Top 14 Variables.

First five rows of the data after processing –

	Churn	Tenure	CC_Contacted_LY	Gender	Service_Score	Account_user_count	CC_Agent_Score	rev_per_month	Complain_ly	rev_growth_yoy	...	Paym
0	1	-0.703389	-1.381597	0	0.137106	-0.767912	-0.766978	1.292699	1	-1.385245	...	
1	1	-1.152004	-1.148541	1	0.137106	0.312490	-0.038699	0.598626	1	-0.321228	...	
2	1	-1.152004	1.415077	1	-1.249194	0.312490	-0.038699	0.251589	1	-0.587232	...	
3	1	-1.152004	-0.332844	1	-1.249194	0.312490	1.417859	0.945662	0	1.806806	...	
4	1	-1.152004	-0.682429	1	-1.249194	-0.767912	1.417859	-0.789520	0	-1.385245	...	

5 rows × 30 columns

APPENDIX TABLE 6 – SCALED DATA

By using Recursive Feature Elimination, keeping top 15 variable which are important for the dataset.
Following are the important features –

	Feature	Rank
13	City_Tier_1.0	1
14	City_Tier_2.0	1
15	City_Tier_3.0	1
16	Payment_Cash on Delivery	1
17	Payment_Credit Card	1
18	Payment_Debit Card	1
19	Payment_E wallet	1
20	Payment_UPI	1
21	account_segment_HNI	1
22	account_segment_Regular	1
24	account_segment_Super	1
25	account_segment_Super Plus	1
26	Marital_Status_Divorced	1
27	Marital_Status_Married	1
28	Marital_Status_Single	1

Appendix Table 7– List of top 15 variables.

Statistical summary of the data is as follows –

Optimization terminated successfully.
Current function value: 0.575684
Iterations 6

Logit Regression Results

Dep. Variable:	Churn	No. Observations:	12706
Model:	Logit	Df Residuals:	12690
Method:	MLE	Df Model:	15
Date:	Sat, 27 May 2023	Pseudo R-squ.:	0.1695
Time:	21:56:25	Log-Likelihood:	-7314.6
converged:	True	LL-Null:	-8807.1
Covariance Type:	nonrobust	LLR p-value:	0.000

	coef	std err	z	P>|z|	[0.025	0.975]
City_Tier_1.0	1.1499	0.134	8.604	0.000	0.888	1.412
City_Tier_2.0	1.3973	0.170	8.242	0.000	1.065	1.730
City_Tier_3.0	2.0623	0.136	15.172	0.000	1.796	2.329
Payment_Cash on Delivery	-0.6748	0.121	-5.596	0.000	-0.911	-0.438
Payment_Credit Card	-1.1242	0.109	-10.347	0.000	-1.337	-0.911
Payment_Debit Card	-1.0859	0.106	-10.217	0.000	-1.294	-0.878
Payment_E wallet	-0.8102	0.122	-6.664	0.000	-1.049	-0.572
Payment_UPI	-0.9476	0.132	-7.181	0.000	-1.206	-0.689
account_segment_HNI	0.7936	0.131	6.078	0.000	0.538	1.049
account_segment_Regular	0.2586	0.163	1.591	0.111	-0.060	0.577
account_segment_Regular Plus	1.6675	0.124	13.398	0.000	1.424	1.911
account_segment_Super	0.0736	0.126	0.582	0.561	-0.174	0.321
account_segment_Super Plus	-0.5991	0.165	-3.623	0.000	-0.923	-0.275
Marital_Status_Divorced	-2.0025	0.105	-19.015	0.000	-2.209	-1.796
Marital_Status_Married	-1.9166	0.093	-20.621	0.000	-2.099	-1.734
Marital_Status_Single	-0.8541	0.093	-9.205	0.000	-1.036	-0.672

APPENDIX TABLE 8 - REGRESSION MODEL SUMMARY 1

From the above summary report it has been clearly observed that 2 variables i.e., account_segment_regular and account_segment_super have p value more than 0.05. Hence, it is better to drop these features as well.

Summary report after dropping these variables is as follows –

Optimization terminated successfully.
Current function value: 0.575805
Iterations 6

Logit Regression Results

Dep. Variable:	Churn	No. Observations:	12706
Model:	Logit	Df Residuals:	12692
Method:	MLE	Df Model:	13
Date:	Sat, 27 May 2023	Pseudo R-squ.:	0.1693
Time:	21:56:25	Log-Likelihood:	-7316.2
converged:	True	LL-Null:	-8807.1
Covariance Type:	nonrobust	LLR p-value:	0.000

	coef	std err	z	P>|z|	[0.025	0.975]
City_Tier_1.0	1.2029	0.118	10.182	0.000	0.971	1.434
City_Tier_2.0	1.4547	0.158	9.180	0.000	1.144	1.765
City_Tier_3.0	2.1038	0.120	17.563	0.000	1.869	2.339
Payment_Cash on Delivery	-0.6448	0.115	-5.596	0.000	-0.871	-0.419
Payment_Credit Card	-1.0949	0.103	-10.665	0.000	-1.296	-0.894
Payment_Debit Card	-1.0581	0.101	-10.518	0.000	-1.255	-0.861
Payment_E wallet	-0.7742	0.116	-6.659	0.000	-1.002	-0.546
Payment_UPI	-0.9211	0.127	-7.248	0.000	-1.170	-0.672
account_segment_HNI	0.7069	0.062	11.426	0.000	0.586	0.828
account_segment_Regular Plus	1.5784	0.050	31.834	0.000	1.481	1.676
account_segment_Super Plus	-0.6869	0.116	-5.940	0.000	-0.914	-0.460
Marital_Status_Divorced	-1.9877	0.103	-19.299	0.000	-2.190	-1.786
Marital_Status_Married	-1.9052	0.090	-21.125	0.000	-2.082	-1.728
Marital_Status_Single	-0.8445	0.090	-9.370	0.000	-1.021	-0.668

APPENDIX TABLE 9 - REGRESSION SUMMARY 2.

So, for model building I will use these 14 features.

- ## Logistic Regression Model

Logistic regression is an example of supervised learning. It is used to calculate or predict the probability of a binary (yes/no) event occurring.

Logit regression, also known as logistic regression, is a statistical model used for binary classification problems. It models the relationship between a set of input variables (or features) and a binary outcome variable by applying a logistic function to the linear combination of the input variables

Accuracy score – Train Data	Accuracy score – Test Data
0.7283561951877671	0.7405561385099685

Classification report for train and test data set –
Train data –

Classification report on training data set - Logistic Regression Model

	precision	recall	f1-score	support
0	0.72	0.77	0.74	4481
1	0.74	0.69	0.72	4413
accuracy			0.73	8894
macro avg	0.73	0.73	0.73	8894
weighted avg	0.73	0.73	0.73	8894

APPENDIX TABLE 10 – CLASSIFICATION REPORT OF LOGISTIC REGRESSION ON TRAINING DATASET.

Test Data –

Classification report on test data set - Logistic Regression Model

	precision	recall	f1-score	support
0	0.72	0.78	0.75	1872
1	0.77	0.70	0.73	1940
accuracy			0.74	3812
macro avg	0.74	0.74	0.74	3812
weighted avg	0.74	0.74	0.74	3812

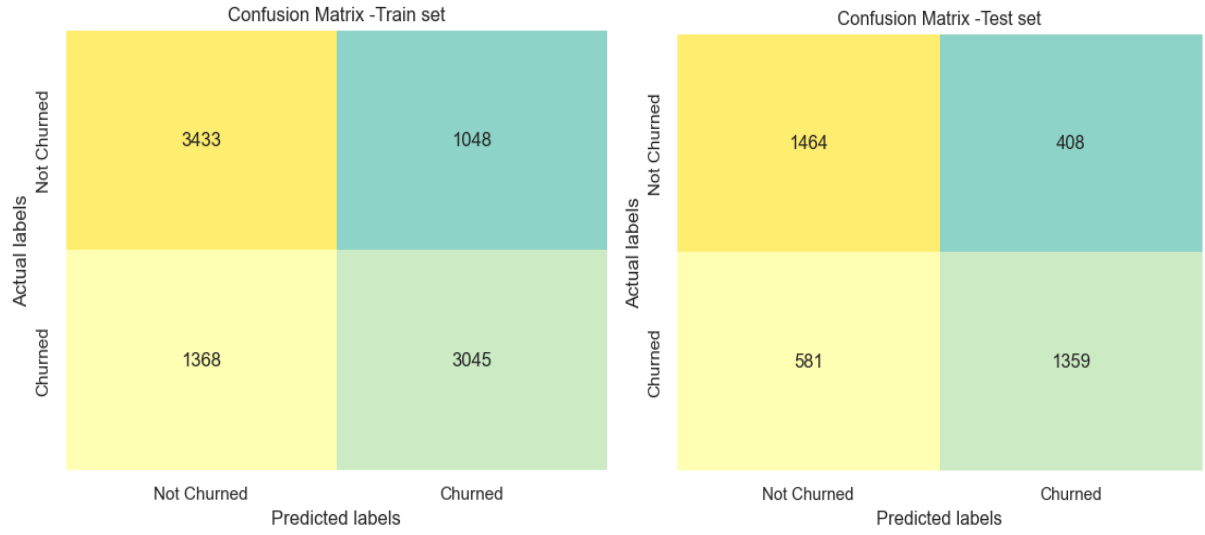
APPENDIX TABLE 11 - CLASSIFICATION REPORT OF LOGISTIC REGRESSION ON TESTING DATASET.

Validation of the model:

• The model is not over-fitted or under-fitted.

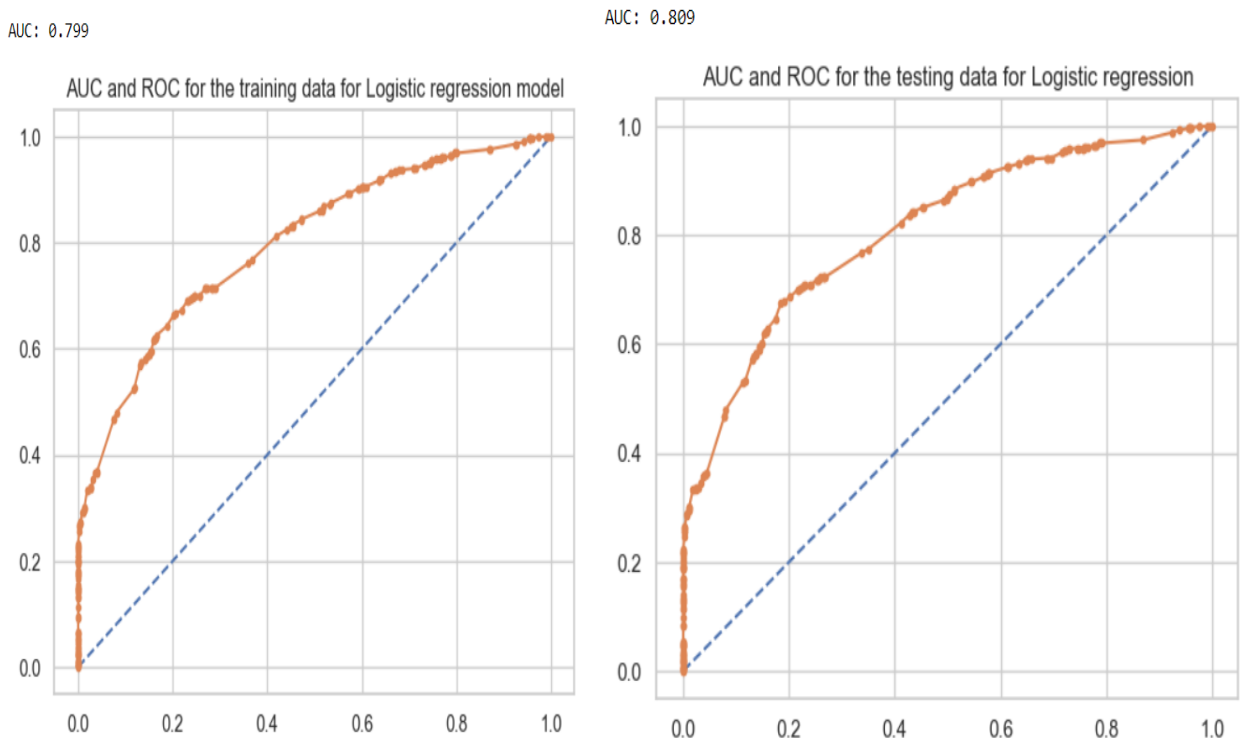
• The error in the test data is slightly higher than the train data, which is absolutely fine because the error margin is low and the error in both train and test data is not too high. Thus, the model is not over-fitted or under-fitted.

Confusion matrix for train and test dataset –



APPENDIX TABLE 12 - CONFUSION MATRIX FOR TRAINING AND TESTING DATASET – LOGISTIC REGRESSION

AUC and ROC curve for Training and testing dataset –



APPENDIX FIGURE 2- AUC AND ROC CURVE FOR TRAINING AND TESTING DATASET – LOGISTIC REGRESSION.

- ## Decision Tree Classifier

DecisionTreeClassifier is a class capable of performing multi-class classification on a dataset. In case that there are multiple classes with the same and highest probability, the classifier will predict the class with the lowest index amongst those classes.

Parameters –

```
Fitting 5 folds for each of 1 candidates, totalling 5 fits

GridSearchCV(cv=5, estimator=DecisionTreeClassifier(random_state=1), n_jobs=-1,
             param_grid={'criterion': ['gini'], 'max_depth': [10],
                         'min_samples_leaf': [10], 'min_samples_split': [50]},
             verbose=1)
```

Best parameter using gridsearch CV –

```
{'criterion': 'gini', 'max_depth': 10, 'min_samples_leaf': 10, 'min_samples_split': 50}

DecisionTreeClassifier(max_depth=10, min_samples_leaf=10, min_samples_split=50,
                       random_state=1)
```

Accuracy Score for training and testing dataset –

Accuracy score – Train Data	Accuracy score – Test Data
0.7472453339329885	0.74501573976915

Classification report for train and test data –

Train Data –

```
Classification report on train data set - Decision tree Model
              precision    recall  f1-score   support

           0       0.73      0.81      0.77      4481
           1       0.79      0.69      0.74      4413

    accuracy                           0.75      8894
   macro avg       0.76      0.75      0.75      8894
weighted avg       0.76      0.75      0.75      8894
```

APPENDIX TABLE 13 - CLASSIFICATION REPORT OF LOGISTIC REGRESSION ON TRAINING DATASET.

Test data –

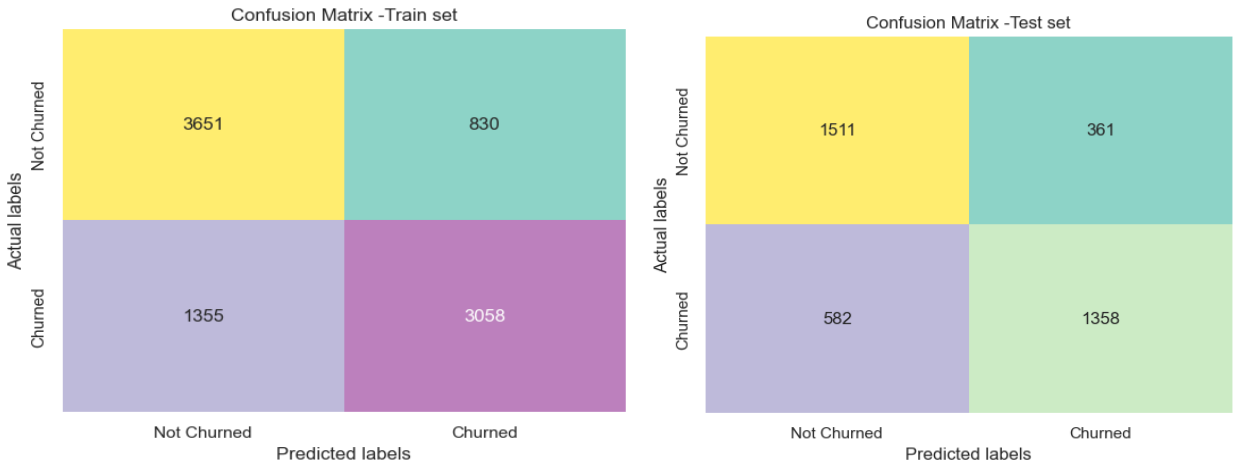
```
Classification report on test data set - Decision tree Model
              precision    recall  f1-score   support

           0       0.72      0.81      0.76      1872
           1       0.79      0.70      0.74      1940

    accuracy                           0.75      3812
   macro avg       0.76      0.75      0.75      3812
weighted avg       0.76      0.75      0.75      3812
```

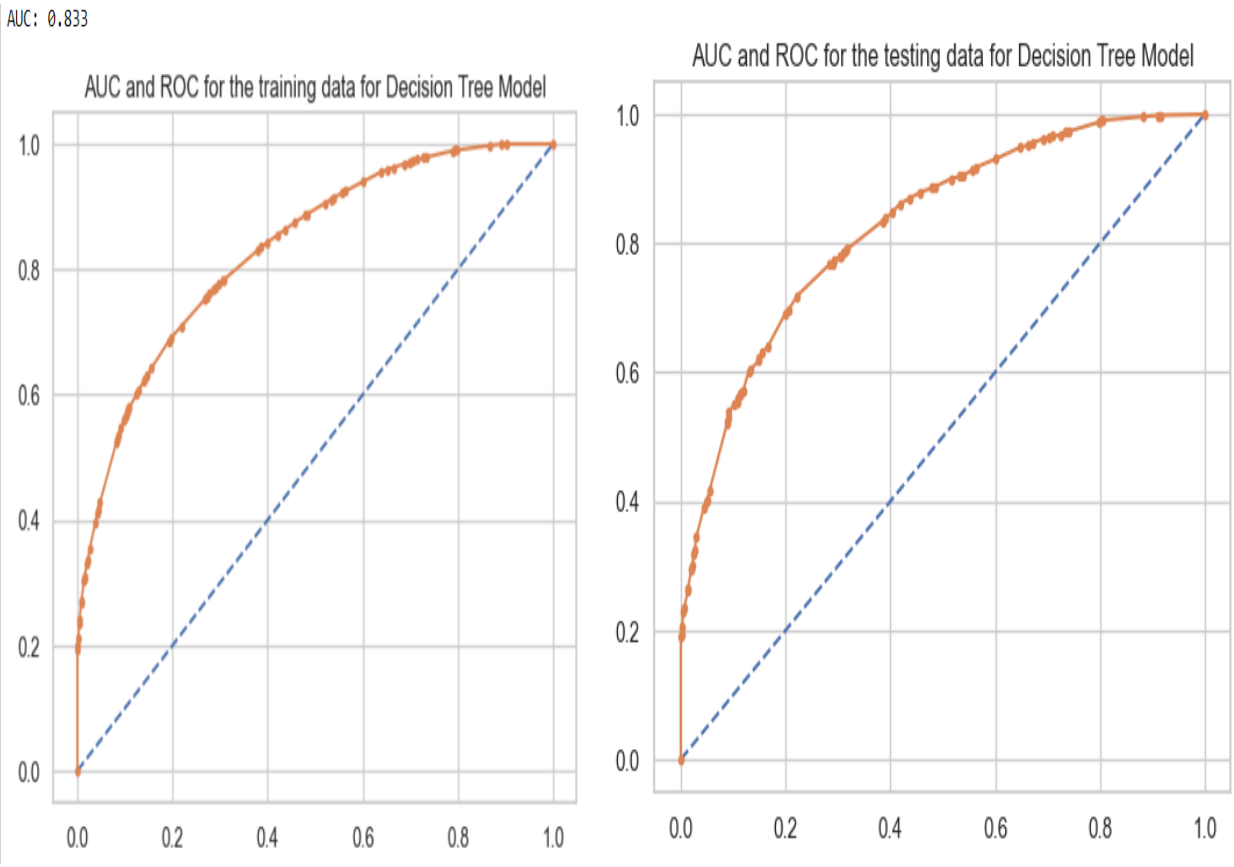
Appendix Table 14 - Classification report of logistic regression on training dataset.

Confusion Matrix for training and testing dataset –



APPENDIX TABLE 15 - CONFUSION MATRIX FOR TRAINING AND TESTING DATASET FOR DECISION TREE.

AUC and ROC curve for training and testing dataset –



APPENDIX FIGURE 3 - AUC AND ROC CURVE FOR TRAINING AND TESTING DATASET FOR DECISION TREE

- ## Random Forest Classifier

A random forest classifier is a machine learning algorithm that belongs to the ensemble learning family. It is based on the concept of decision trees and combines multiple decision trees to create a more robust and accurate model.

Parameter –

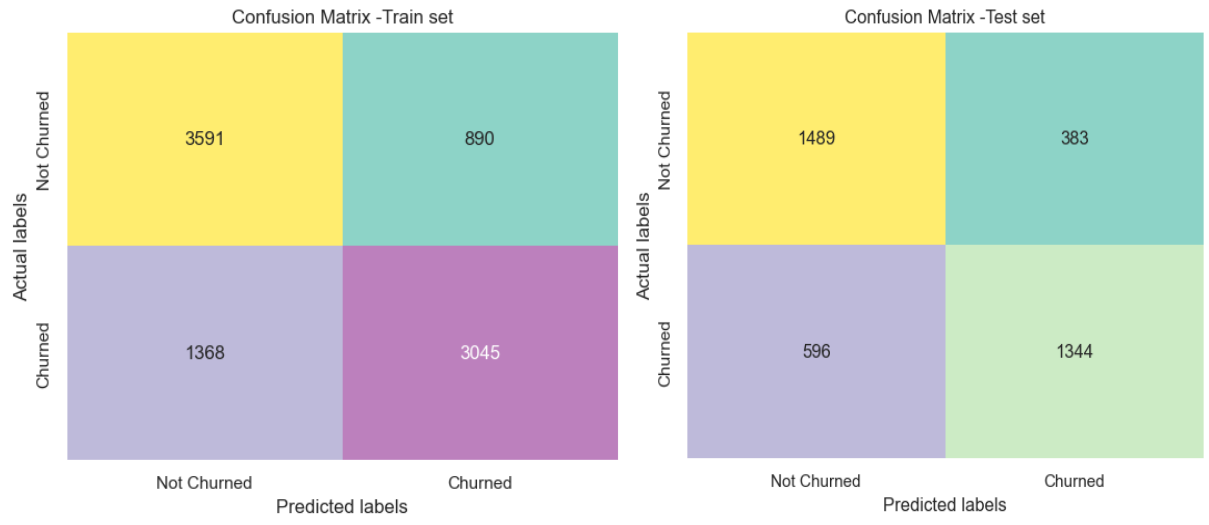
```
GridSearchCV(cv=5, estimator=RandomForestClassifier(random_state=1),
             param_grid={'max_depth': [10, 20, 30, 40, 50, 60],
                         'max_features': [11, 12, 13, 14],
                         'min_samples_leaf': [10, 50, 100],
                         'min_samples_split': [50, 60, 70],
                         'n_estimators': [100, 200]})
```

Best Parameter –

```
{'max_depth': 10,
 'max_features': 12,
 'min_samples_leaf': 10,
 'min_samples_split': 50,
 'n_estimators': 100}
```

Accuracy score – Train Data	Accuracy score – Test Data
0.7512930065212503	0.7387198321091291

Confusion Matrix of train and test data set –



APPENDIX TABLE 16 - CONFUSION MATRIX FOR TRAINING AND TESTING DATASET FOR RANDOM FOREST CLASSIFIER

Classification report for training and testing dataset –

Classification report on train data set - Random Forest

	precision	recall	f1-score	support
0	0.72	0.80	0.76	4481
1	0.77	0.69	0.73	4413
accuracy			0.75	8894
macro avg	0.75	0.75	0.75	8894
weighted avg	0.75	0.75	0.75	8894

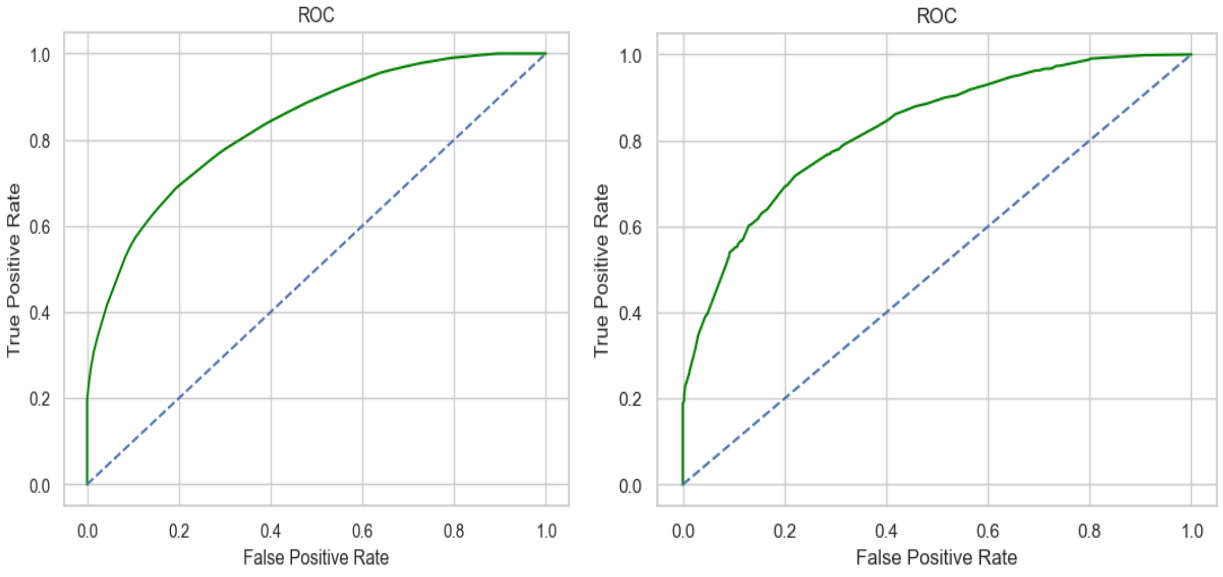
APPENDIX TABLE 17 - CLASSIFICATION REPORT FOR RANDOM FOREST CLASSIFIER OF TRAINING DATASET.

Classification report on test data set - Random Forest

	precision	recall	f1-score	support
0	0.71	0.80	0.75	1872
1	0.78	0.69	0.73	1940
accuracy			0.74	3812
macro avg	0.75	0.74	0.74	3812
weighted avg	0.75	0.74	0.74	3812

APPENDIX TABLE 18 - CLASSIFICATION REPORT FOR RANDOM FOREST CLASSIFIER OF TESTING DATASET.

AUC ROC curve for training and testing dataset –



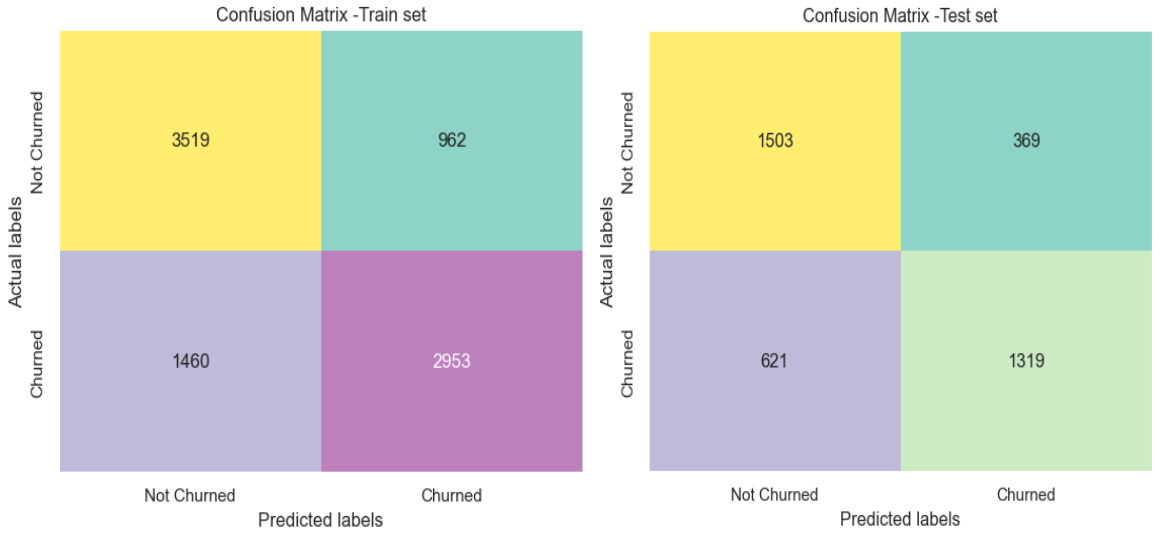
Appendix Figure 4- AUC and ROC curve for training and testing dataset for Random Forest Classifier

- ## Linear Discriminant Analysis

Linear Discriminant Analysis (LDA) is a dimensionality reduction and classification technique used in machine learning and statistics. It is a supervised learning algorithm that finds a linear combination of features to project the data into a lower-dimensional space while maximizing the separation between classes.

Accuracy score – Train Data	Accuracy score – Test Data
0.7276815830897234	0.7402938090241343

Confusion Matrix of train and test data set –



APPENDIX TABLE 19 - CONFUSION MATRIX FOR TRAINING AND TESTING DATASET FOR LDA

Classification report for training and testing dataset –

Classification report on train data set - Linear Discriminent analysis

	precision	recall	f1-score	support
0	0.71	0.79	0.74	4481
1	0.75	0.67	0.71	4413
accuracy			0.73	8894
macro avg	0.73	0.73	0.73	8894
weighted avg	0.73	0.73	0.73	8894

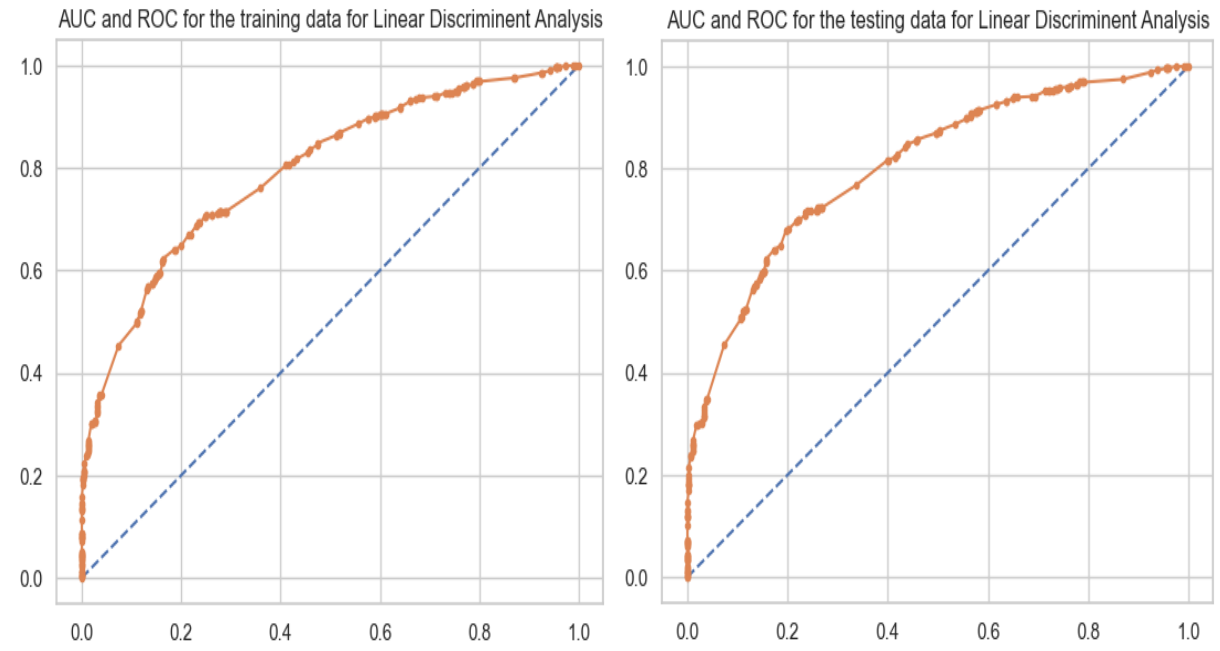
APPENDIX TABLE 20 - CLASSIFICATION REPORT FOR LDA OF TRAINING DATASET

Classification report on test data set - Linear Discriminent analysis

	precision	recall	f1-score	support
0	0.71	0.80	0.75	1872
1	0.78	0.68	0.73	1940
accuracy			0.74	3812
macro avg	0.74	0.74	0.74	3812
weighted avg	0.75	0.74	0.74	3812

Appendix Table 21 - Classification Report for LDA of testing dataset

AUC ROC curve for training and testing dataset –



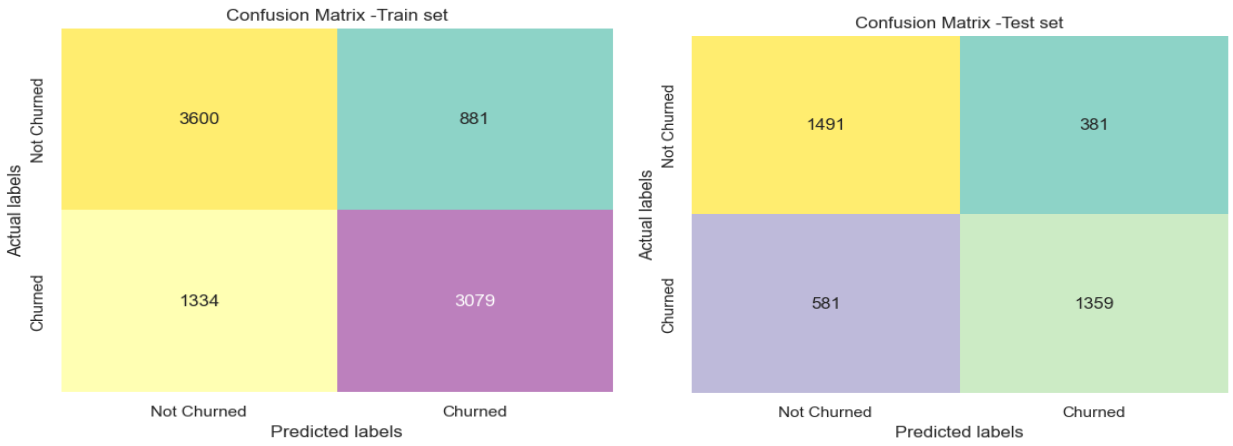
APPENDIX FIGURE 5 - AUC AND ROC CURVE FOR TRAINING AND TESTING DATASET FOR LDA

- KNN Model

K-Nearest Neighbors (KNN) is a popular supervised machine learning algorithm used for both classification and regression tasks. It is a non-parametric algorithm, meaning it doesn't make any assumptions about the underlying data distribution. Instead, it relies on the proximity of instances in the feature space.

Accuracy score – Train Data	Accuracy score – Test Data
0.7509557004722285	0.7476390346274921

Confusion Matrix of train and test data set –



Appendix Table 22 - Confusion matrix for training and testing dataset for KNN

Classification report for training and testing dataset –

Classification report on train data set - KNN Model

	precision	recall	f1-score	support
0	0.73	0.80	0.76	4481
1	0.78	0.70	0.74	4413
accuracy			0.75	8894
macro avg	0.75	0.75	0.75	8894
weighted avg	0.75	0.75	0.75	8894

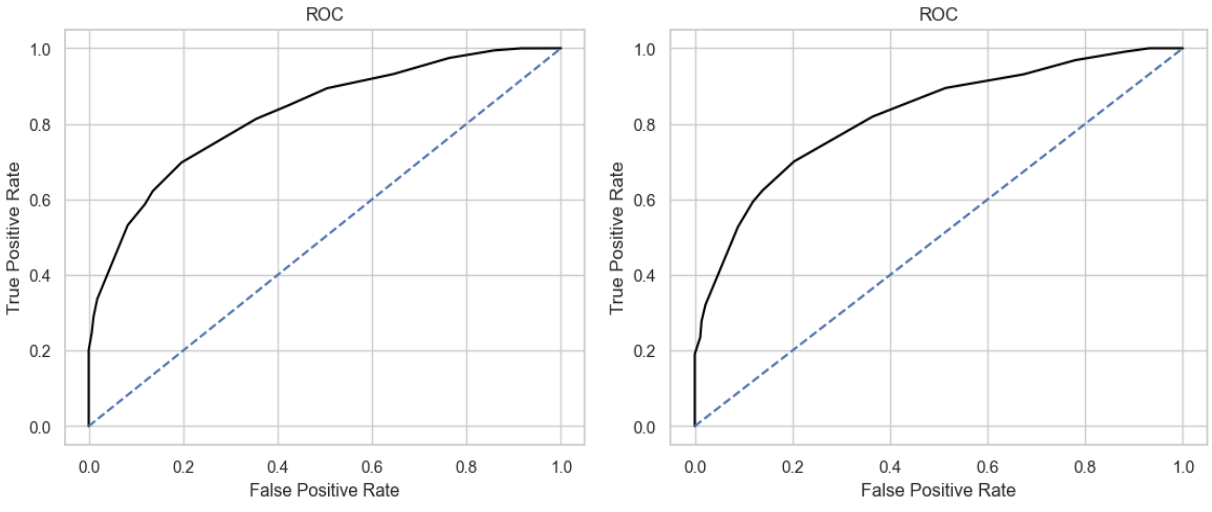
APPENDIX TABLE 23 - CLASSIFICATION REPORT FOR KNN OF TRAINING DATASET

Classification report on train data set - KNN Model

	precision	recall	f1-score	support
0	0.72	0.80	0.76	1872
1	0.78	0.70	0.74	1940
accuracy			0.75	3812
macro avg	0.75	0.75	0.75	3812
weighted avg	0.75	0.75	0.75	3812

APPENDIX TABLE 24 - CLASSIFICATION REPORT FOR KNN OF TESTING DATASET

AUC ROC curve for training and testing dataset –



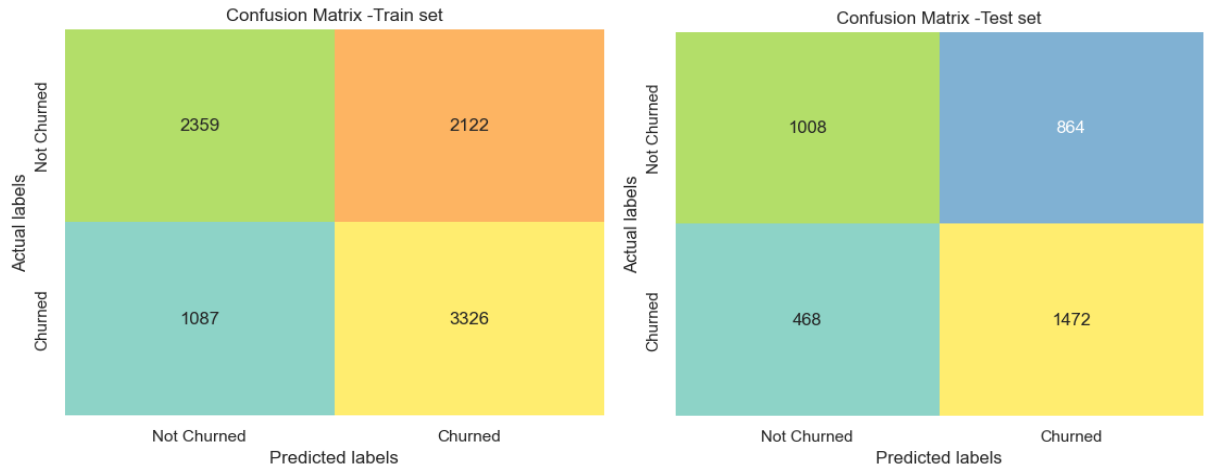
Appendix Figure 6 - AUC and ROC curve for training and testing dataset for KNN

- ## Naïve bayes Model

Naive Bayes is a classification algorithm that is based on Bayes' theorem with the assumption of independence among the features. It is called "naive" because it assumes that the presence or absence of a particular feature is unrelated to the presence or absence of any other feature, which is often an oversimplification in real-world scenarios. Despite this simplifying assumption, Naive Bayes can be effective in many practical applications and has been widely used in various fields, including text classification, spam filtering, sentiment analysis, and recommendation systems.

Accuracy score – Train Data	Accuracy score – Test Data
0.6391949628963346	0.6505771248688352

Confusion Matrix of train and test data set –



APPENDIX TABLE 25 - CONFUSION MATRIX FOR TRAINING AND TESTING DATASET FOR NB

Classification report for training and testing dataset –

Classification report on train data set - Gaussian Naive Bayes Model

	precision	recall	f1-score	support
0	0.68	0.53	0.60	4481
1	0.61	0.75	0.67	4413
accuracy			0.64	8894
macro avg	0.65	0.64	0.63	8894
weighted avg	0.65	0.64	0.63	8894

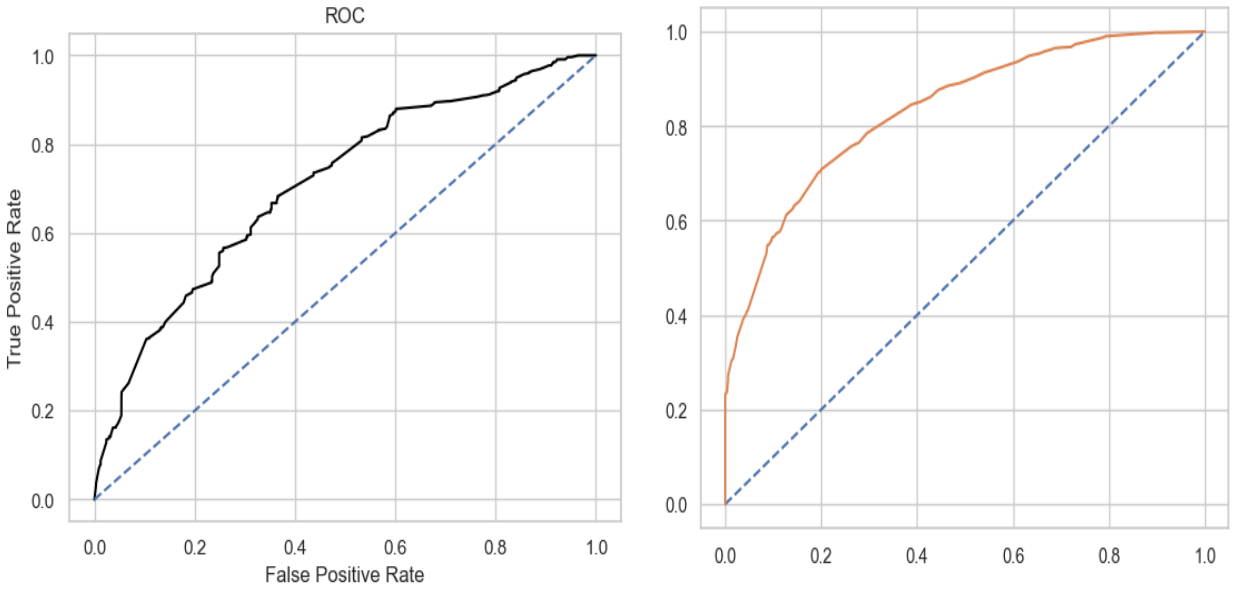
APPENDIX TABLE 26 - CLASSIFICATION REPORT FOR NB OF TRAINING DATASET

Classification report on train data set - Gaussian Naive Bayes Model

	precision	recall	f1-score	support
0	0.68	0.54	0.60	1872
1	0.63	0.76	0.69	1940
accuracy			0.65	3812
macro avg	0.66	0.65	0.65	3812
weighted avg	0.66	0.65	0.65	3812

APPENDIX TABLE 27 - CLASSIFICATION REPORT FOR NB OF TESTING DATASET

AUC ROC curve for training and testing dataset –



APPENDIX FIGURE 7 - AUC AND ROC CURVE FOR TRAINING AND TESTING DATASET FOR NB

- ## ADA Boost classifier Model

AdaBoost, short for Adaptive Boosting, is a machine learning ensemble method that combines multiple weak classifiers to create a strong classifier. Each weak classifier is trained on a subset of the training data, and the final classification decision is made by combining the predictions of all weak classifiers.

Parameters –

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

Accuracy Score, Confusion Matrix and Classification Report of training data –

```
0.723184169102766
[[3391 1090]
 [1372 3041]]
              precision    recall  f1-score   support

           0       0.71      0.76      0.73      4481
           1       0.74      0.69      0.71      4413

    accuracy                           0.72      8894
   macro avg       0.72      0.72      0.72      8894
weighted avg       0.72      0.72      0.72      8894
```

APPENDIX TABLE 28 – AS, CM, CR FOR TRAINING DATA OF ADA BOOST

Accuracy Score, Confusion Matrix and Classification Report of testing data –

```
0.7339979013641134
[[1500  372]
 [ 573 1367]]
              precision    recall  f1-score   support

           0       0.72      0.80      0.76      1872
           1       0.79      0.70      0.74      1940

    accuracy                           0.75      3812
   macro avg       0.75      0.75      0.75      3812
weighted avg       0.76      0.75      0.75      3812
```

APPENDIX TABLE 29 - AS, CM, CR FOR TESTING DATA OF ADA BOOST

d) Model Building and Interpretation with top 20 variables

- **Logistic Regression Model**

Accuracy score – Train Data	Accuracy score – Test Data
0.7283561951877671	0.7405561385099685

Classification report for train and test data set –

Train data –

Classification report on training data set - Logistic Regression Model

	precision	recall	f1-score	support
0	0.84	0.83	0.84	4481
1	0.83	0.84	0.84	4413
accuracy			0.84	8894
macro avg	0.84	0.84	0.84	8894
weighted avg	0.84	0.84	0.84	8894

Tuned - Table 1 - Classification report of logistic regression on training dataset.

Test Data –

Classification report on test data set - Logistic Regression Model

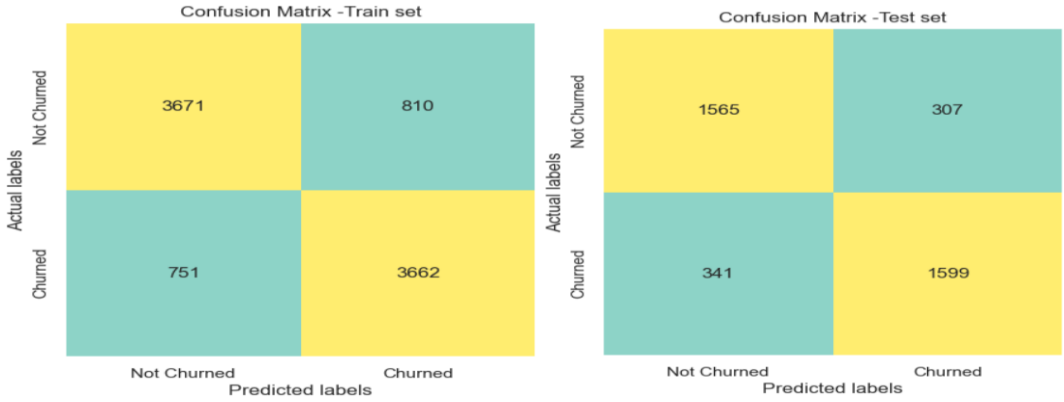
	precision	recall	f1-score	support
0	0.83	0.85	0.84	1872
1	0.85	0.84	0.84	1940
accuracy			0.84	3812
macro avg	0.84	0.84	0.84	3812
weighted avg	0.84	0.84	0.84	3812

Tuned - Table 2 - Classification report of logistic regression on testing dataset.

Validation of the model:

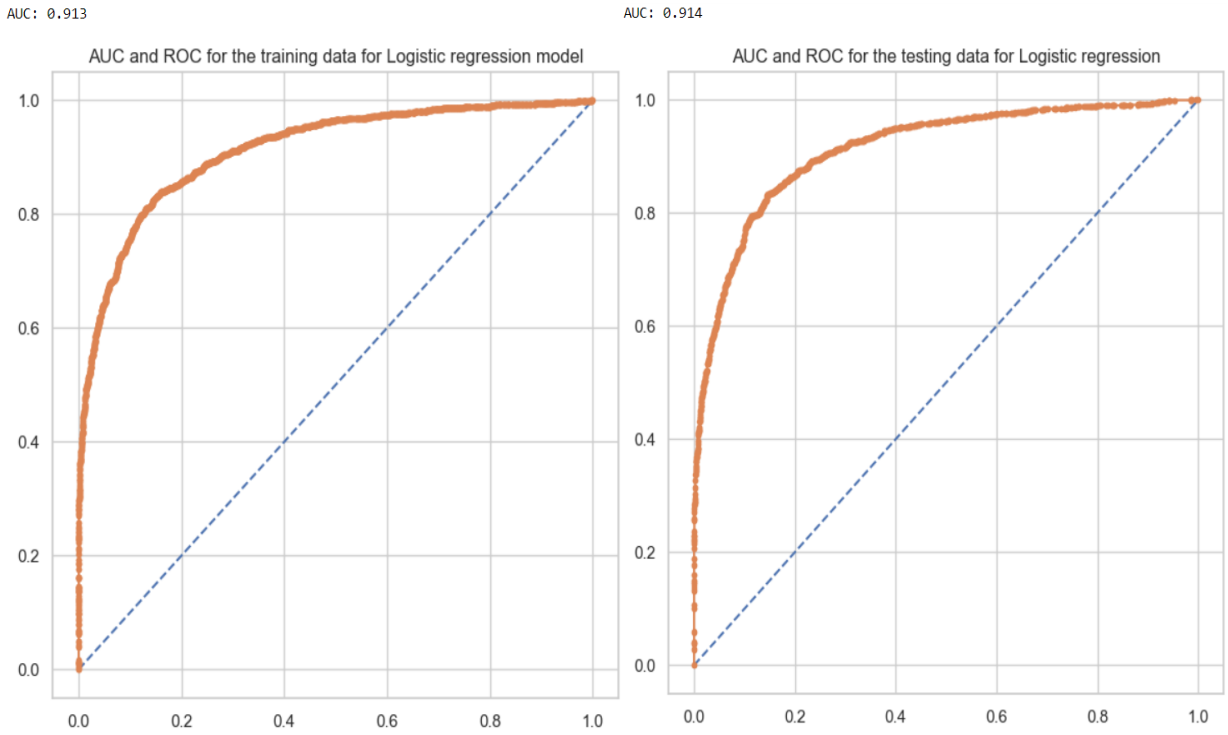
• The model is not over-fitted or under-fitted.

Confusion matrix for train and test dataset –



Tuned - Table 3 - Confusion matrix for training and testing dataset – Logistic regression

AUC and ROC curve for Training and testing dataset –



Tuned - Figure 1 - AUC and ROC curve for training and testing dataset – logistic regression.

- **Logistic Regression Model – Tuned**

Classification report for train and test data set –

Train data –

Classification report on training data set - Logistic Regression Model

	precision	recall	f1-score	support
0	0.84	0.84	0.84	4481
1	0.84	0.84	0.84	4413
accuracy			0.84	8894
macro avg	0.84	0.84	0.84	8894
weighted avg	0.84	0.84	0.84	8894

Tuned - Table 4 - Classification report of logistic regression Tuned on training dataset.

Test Data –

Classification report on test data set - Logistic Regression Model

	precision	recall	f1-score	support
0	0.83	0.85	0.84	1872
1	0.85	0.83	0.84	1940
accuracy			0.84	3812
macro avg	0.84	0.84	0.84	3812
weighted avg	0.84	0.84	0.84	3812

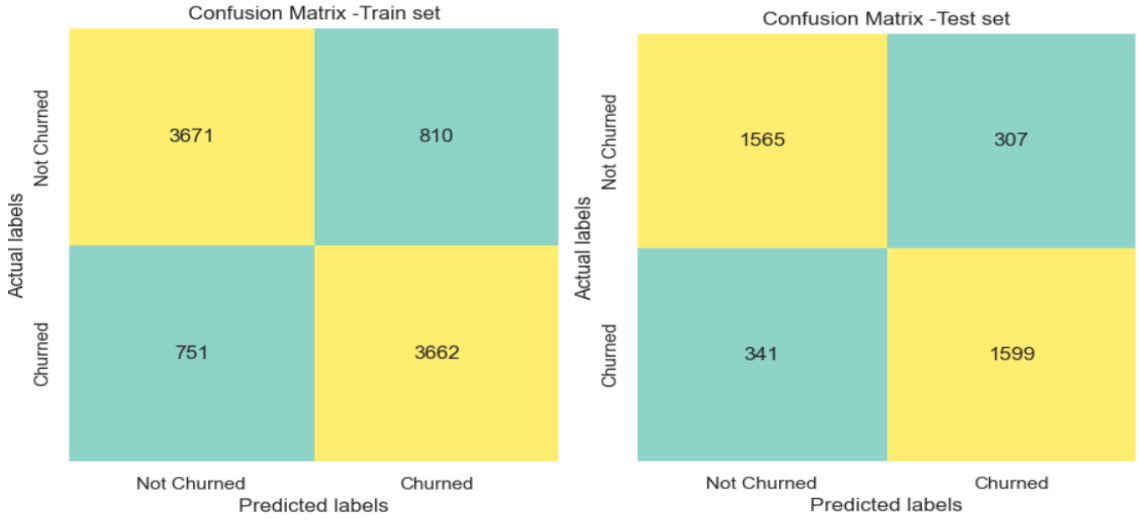
Tuned - Table 5 - Classification report of logistic regression Tuned on testing dataset.

Validation of the model:

• The model is not over-fitted or under-fitted.

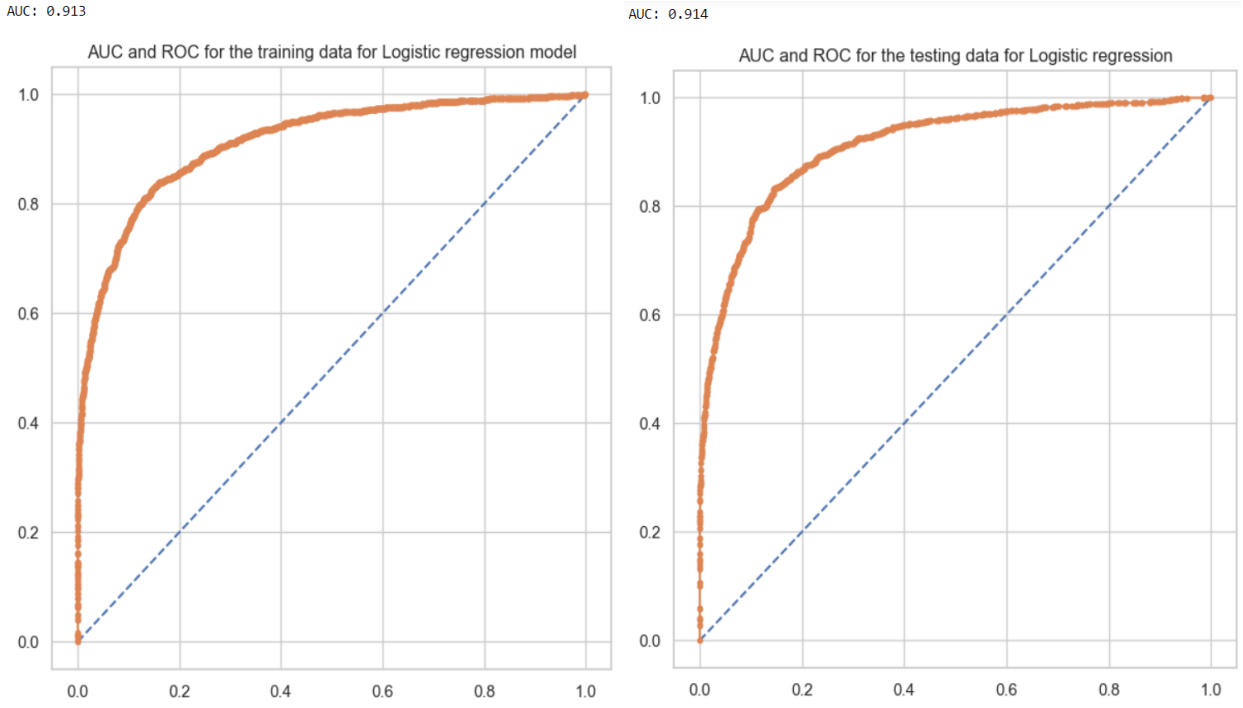
• The error in the test data is slightly higher than the train data, which is absolutely fine because the error margin is low and the error in both train and test data is not too high. Thus, the model is not over-fitted or under-fitted.

Confusion matrix for train and test dataset –



Tuned - Table 6 - Confusion matrix for training and testing dataset – Logistic regression Tuned

AUC and ROC curve for Training and testing dataset –



Tuned - Figure 2 - AUC and ROC curve for training and testing dataset – logistic regression Tuned

- ## Decision Tree Classifier

DecisionTreeClassifier is a class capable of performing multi-class classification on a dataset. In case that there are multiple classes with the same and highest probability, the classifier will predict the class with the lowest index amongst those classes.

Parameters –

```
Fitting 5 folds for each of 1 candidates, totalling 5 fits

GridSearchCV(cv=5, estimator=DecisionTreeClassifier(random_state=1), n_jobs=-1,
             param_grid={'criterion': ['gini'], 'max_depth': [10],
                         'min_samples_leaf': [10], 'min_samples_split': [50]},
             verbose=1)
```

Best parameter using gridsearch CV –

```
{'criterion': 'gini', 'max_depth': 10, 'min_samples_leaf': 10, 'min_samples_split': 50}

DecisionTreeClassifier(max_depth=10, min_samples_leaf=10, min_samples_split=50,
                       random_state=1)
```

Classification report for train and test data –
Train Data –

Classification report on train data set - Decision tree Model

	precision	recall	f1-score	support
0	0.98	0.99	0.99	4481
1	0.99	0.98	0.98	4413
accuracy			0.99	8894
macro avg	0.99	0.99	0.99	8894
weighted avg	0.99	0.99	0.99	8894

Tuned - Table 7 - Classification report of logistic regression on training dataset.

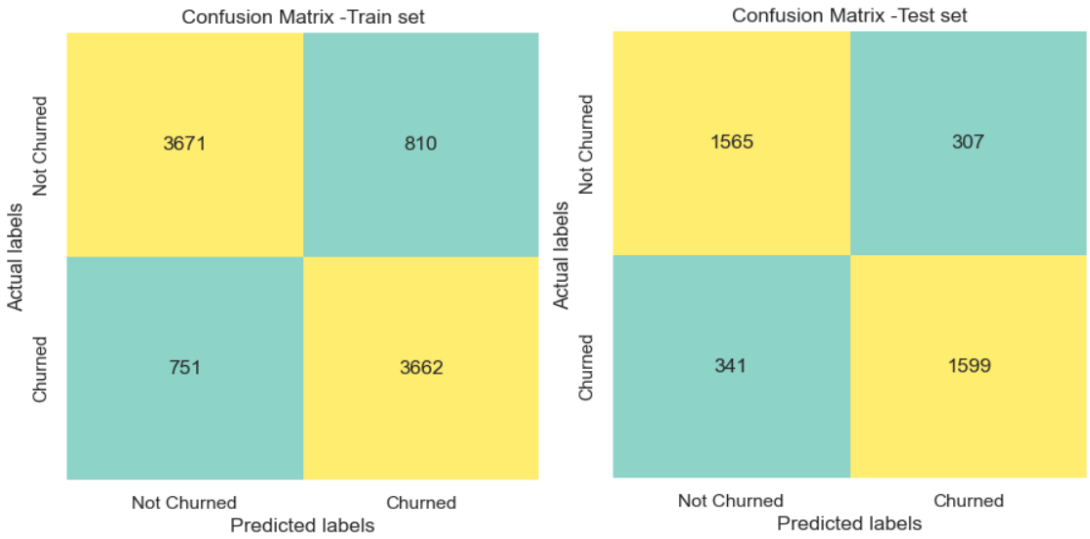
Test data –

Classification report on test data set - Decision tree Model

	precision	recall	f1-score	support
0	0.90	0.91	0.90	1872
1	0.91	0.90	0.91	1940
accuracy			0.90	3812
macro avg	0.90	0.90	0.90	3812
weighted avg	0.90	0.90	0.90	3812

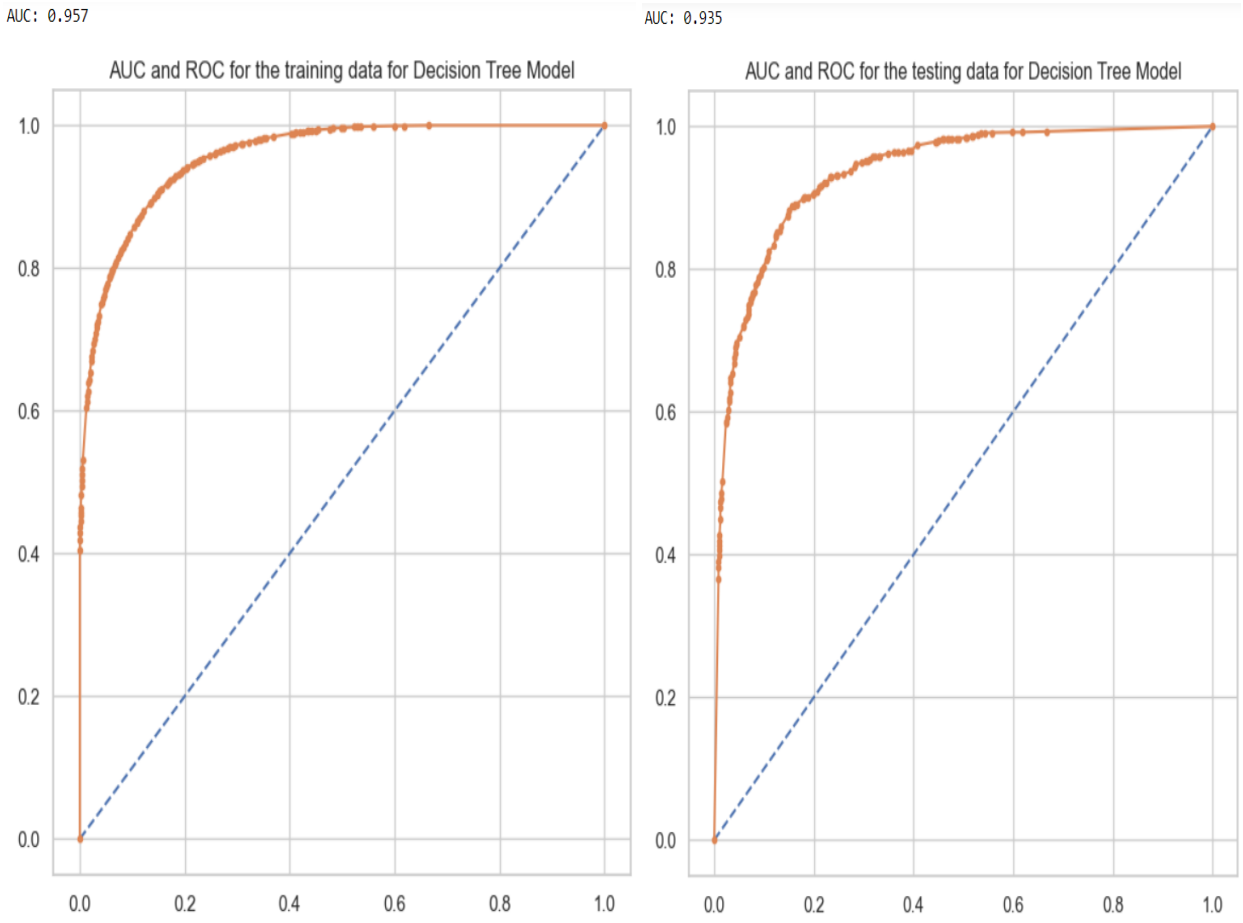
Tuned - Table 8 - Classification report of logistic regression on training dataset.

Confusion Matrix for training and testing dataset –



Tuned - Table 9 - Confusion matrix for training and testing dataset for decision tree.

AUC and ROC curve for training and testing dataset –



Tuned - Figure 3 - AUC and ROC curve for training and testing dataset for Decision Tree

- ## Random Forest Classifier

A random forest classifier is a machine learning algorithm that belongs to the ensemble learning family. It is based on the concept of decision trees and combines multiple decision trees to create a more robust and accurate model.

Parameter –

```
GridSearchCV(cv=5, estimator=RandomForestClassifier(random_state=1),
             param_grid={'max_depth': [10, 20, 30, 40, 50, 60],
                         'max_features': [11, 12, 13, 14],
                         'min_samples_leaf': [10, 50, 100],
                         'min_samples_split': [50, 60, 70],
                         'n_estimators': [100, 200]})
```

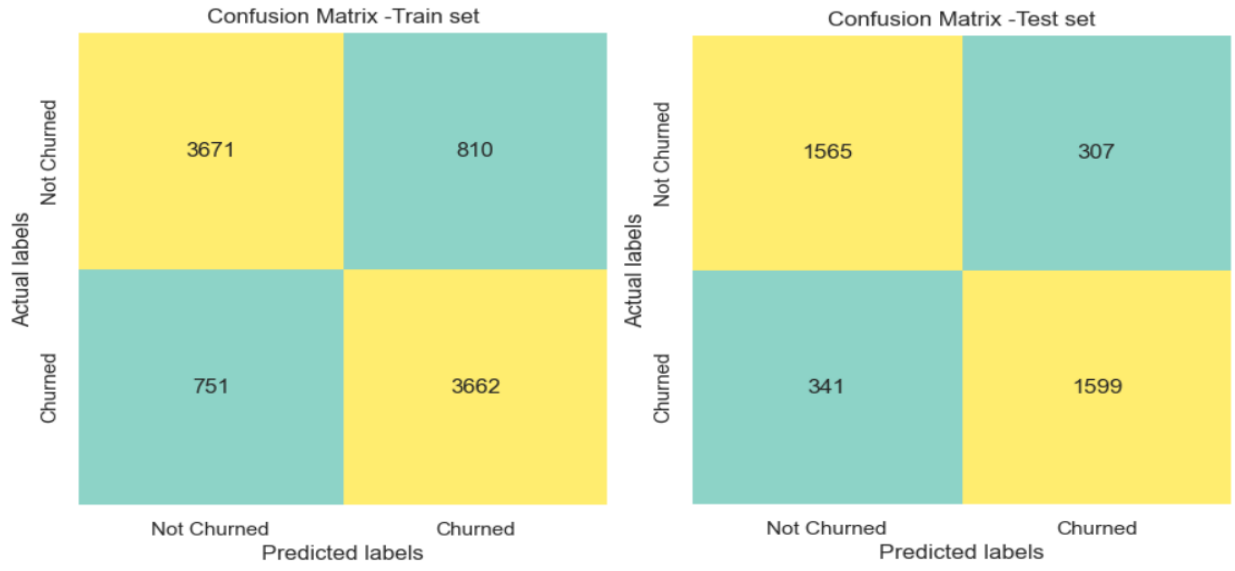
Best Parameter –

```
{'max_depth': 20,
 'max_features': 13,
 'min_samples_leaf': 10,
 'min_samples_split': 50,
 'n_estimators': 200}
```

Best Estimator –

```
RandomForestClassifier(max_depth=20, max_features=13, min_samples_leaf=10,
                       min_samples_split=50, n_estimators=200, random_state=1)
```

Confusion Matrix of train and test data set –



Tuned - Table 10 - Confusion matrix for training and testing dataset for Random Forest Classifier

Classification report for training and testing dataset –

Classification report on train data set - Random Forest

	precision	recall	f1-score	support
0	0.91	0.91	0.91	4481
1	0.91	0.90	0.91	4413
accuracy			0.91	8894
macro avg	0.91	0.91	0.91	8894
weighted avg	0.91	0.91	0.91	8894

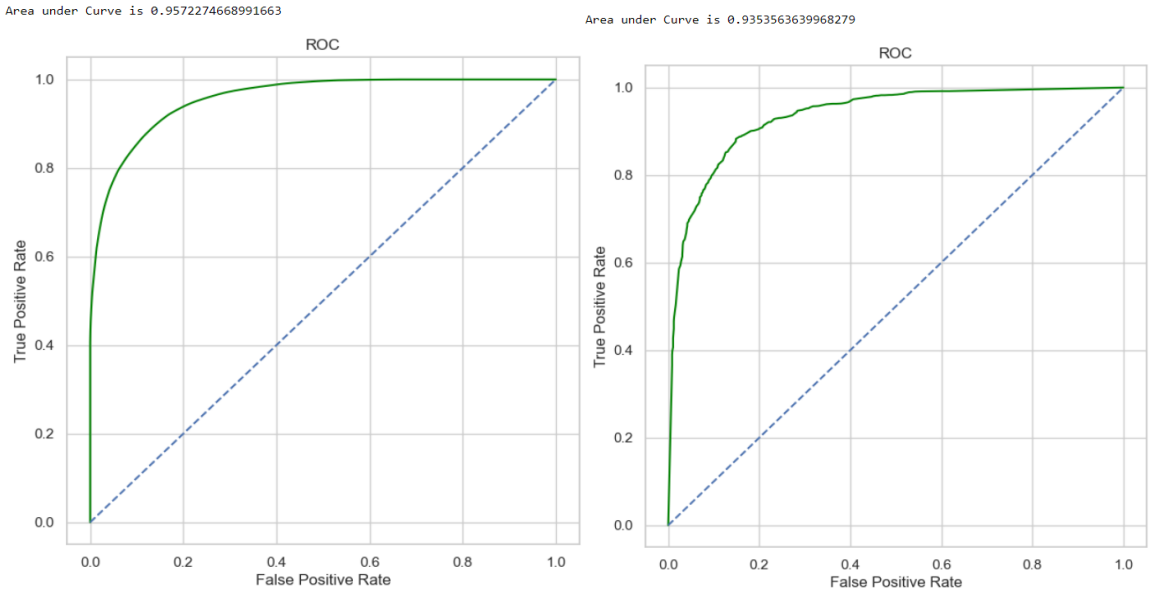
Tuned - Table 11 - Classification Report for Random Forest classifier of training dataset.

Classification report on test data set - Random Forest

	precision	recall	f1-score	support
0	0.88	0.90	0.89	1872
1	0.90	0.88	0.89	1940
accuracy			0.89	3812
macro avg	0.89	0.89	0.89	3812
weighted avg	0.89	0.89	0.89	3812

Tuned - Table 12 - Classification Report for Random Forest classifier of testing dataset.

AUC ROC curve for training and testing dataset –

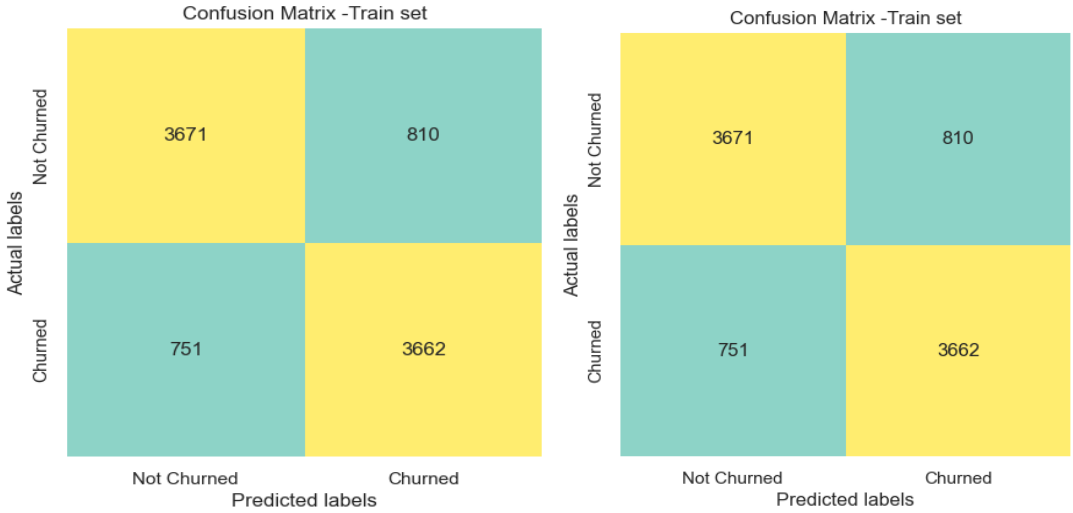


Tuned - Figure 4 - AUC and ROC curve for training and testing dataset for Random Forest Classifier

- **Linear Discriminant Analysis**

Accuracy score – Train Data	Accuracy score – Test Data
0.8244884191589836	0.8300104931794333

Confusion Matrix of train and test data set –



Tuned - Table 13 - Confusion matrix for training and testing dataset for LDA

Classification report for training and testing dataset –

Classification report on train data set - Linear Discriminent analysis

	precision	recall	f1-score	support
0	0.83	0.82	0.82	4481
1	0.82	0.83	0.82	4413
accuracy			0.82	8894
macro avg	0.82	0.82	0.82	8894
weighted avg	0.82	0.82	0.82	8894

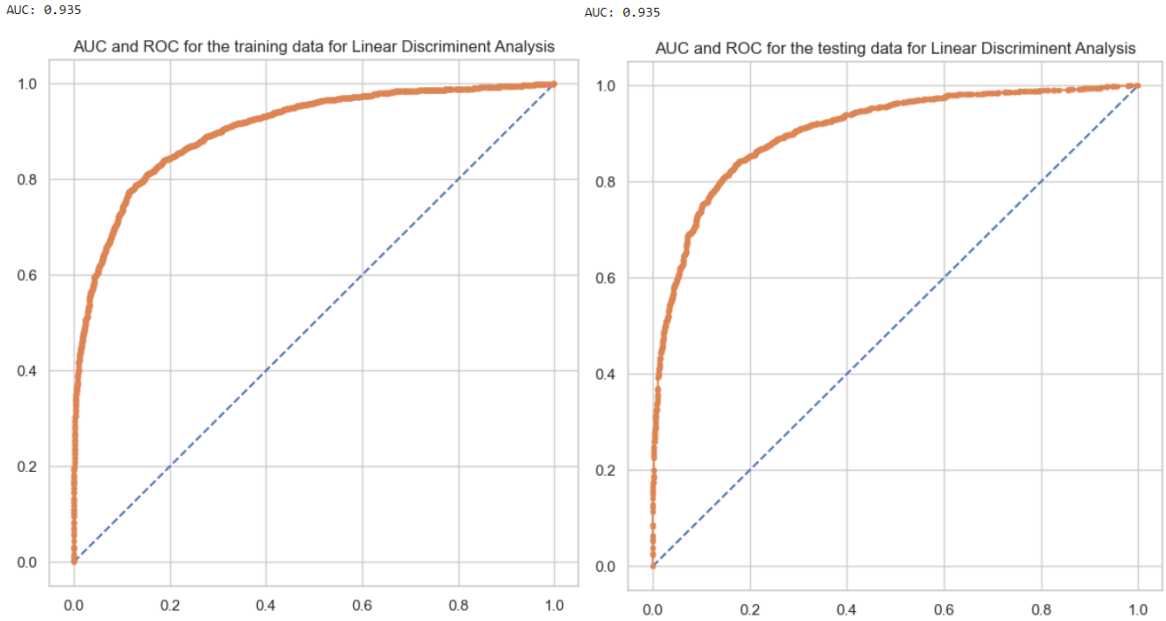
Tuned - Table 14 - Classification Report for LDA of training dataset

Classification report on test data set - Linear Discriminent analysis

	precision	recall	f1-score	support
0	0.82	0.84	0.83	1872
1	0.84	0.82	0.83	1940
accuracy			0.83	3812
macro avg	0.83	0.83	0.83	3812
weighted avg	0.83	0.83	0.83	3812

Tuned - Table 15 - Classification Report for LDA of testing dataset

AUC ROC curve for training and testing dataset –

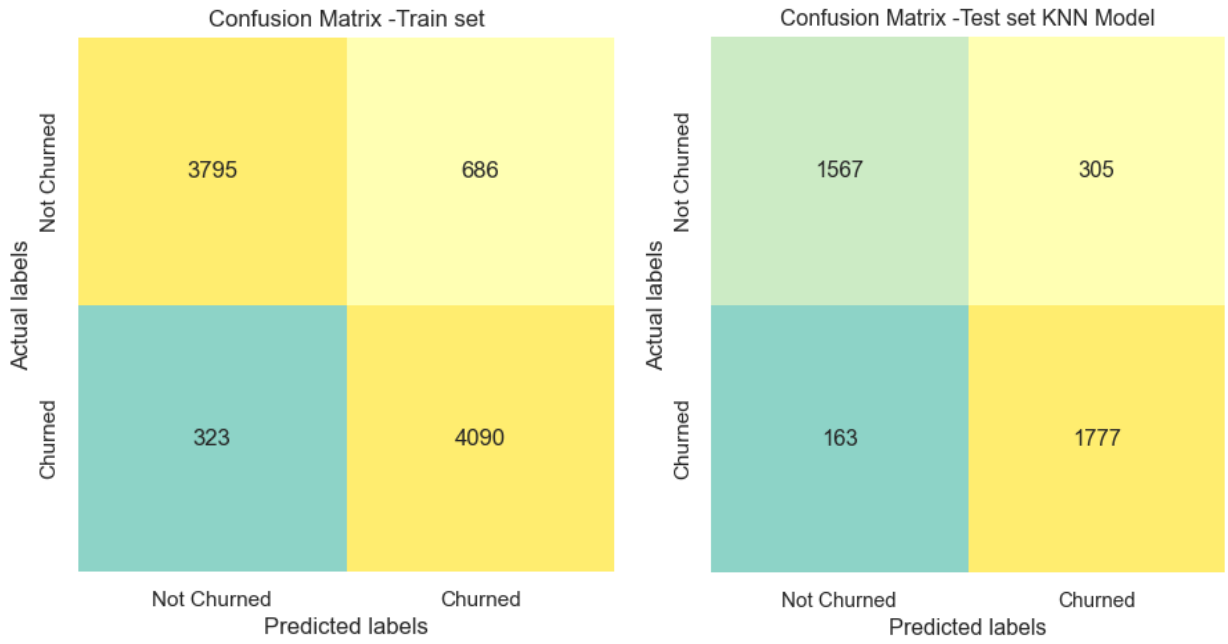


Tuned - Figure 5 - AUC and ROC curve for training and testing dataset for LDA

- **KNN Model**

Accuracy score – Train Data	Accuracy score – Test Data
0.8865527321789971	0.8772298006295908

Confusion Matrix of train and test data set –



Tuned - Table 16 - Confusion matrix for training and testing dataset for KNN

Classification report for training and testing dataset –

Classification report on train data set - KNN Model

	precision	recall	f1-score	support
0	0.92	0.85	0.88	4481
1	0.86	0.93	0.89	4413
accuracy			0.89	8894
macro avg	0.89	0.89	0.89	8894
weighted avg	0.89	0.89	0.89	8894

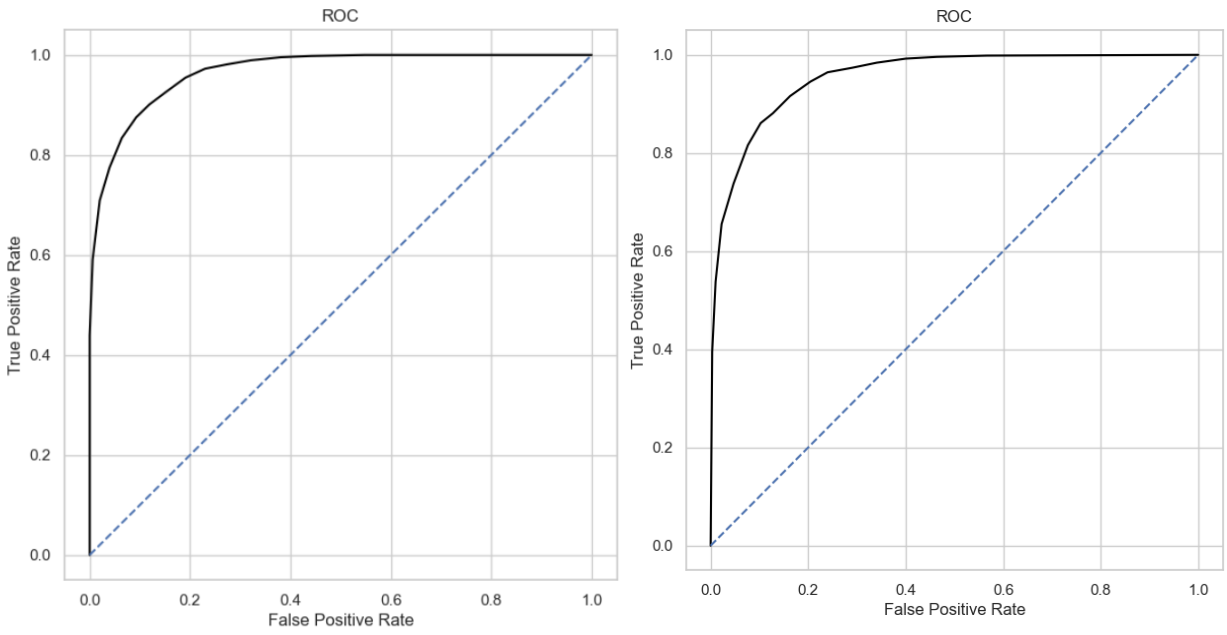
Tuned - Table 17 - Classification Report for KNN of training dataset

Classification report on train data set - KNN Model

	precision	recall	f1-score	support
0	0.91	0.84	0.87	1872
1	0.85	0.92	0.88	1940
accuracy			0.88	3812
macro avg	0.88	0.88	0.88	3812
weighted avg	0.88	0.88	0.88	3812

Tuned - Table 18 - Classification Report for KNN of testing dataset

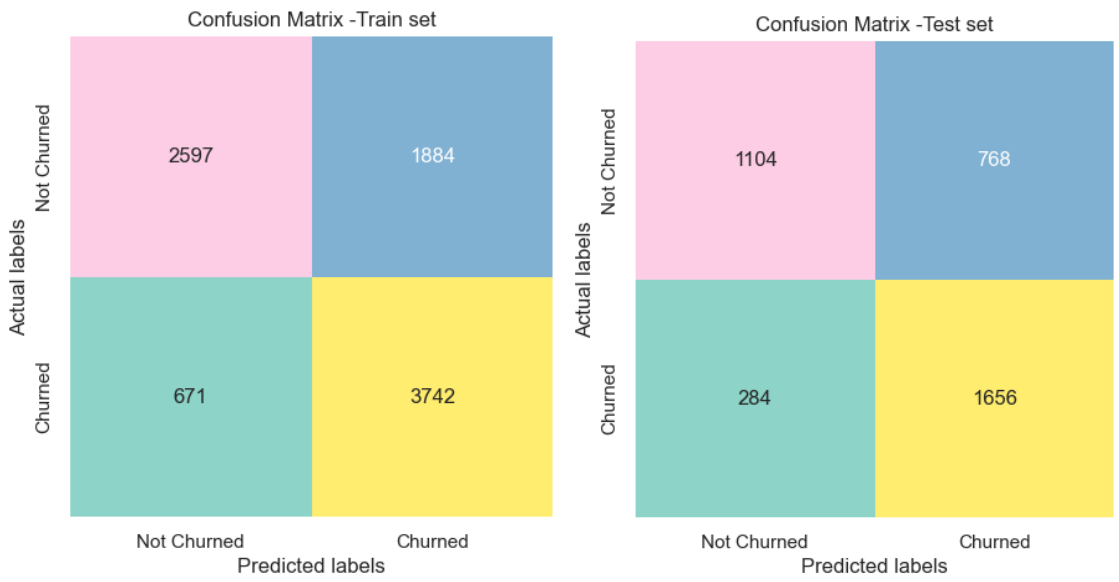
AUC ROC curve for training and testing dataset –



Tuned - Figure 6 - AUC and ROC curve for training and testing dataset for KNN

- **Naïve bayes Model**

Confusion Matrix of train and test data set –



Tuned - Table 19 - Confusion matrix for training and testing dataset for NB

Classification report for training and testing dataset –

Classification report on train data set - Gaussian Naive Bayes Model

	precision	recall	f1-score	support
0	0.79	0.58	0.67	4481
1	0.67	0.85	0.75	4413
accuracy			0.71	8894
macro avg	0.73	0.71	0.71	8894
weighted avg	0.73	0.71	0.71	8894

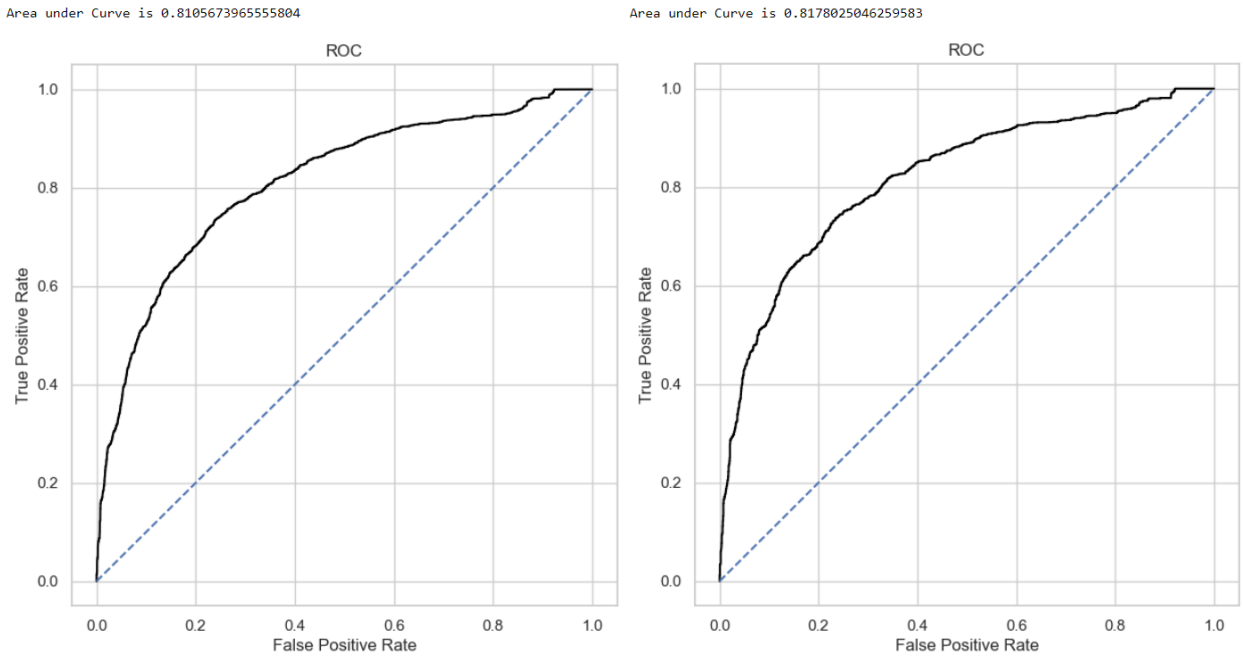
Tuned - Table 20 - Classification Report for NB of training dataset

Classification report on train data set - Gaussian Naive Bayes Model

	precision	recall	f1-score	support
0	0.80	0.59	0.68	1872
1	0.68	0.85	0.76	1940
accuracy			0.72	3812
macro avg	0.74	0.72	0.72	3812
weighted avg	0.74	0.72	0.72	3812

Tuned - Table 21 - Classification Report for NB of testing dataset

AUC ROC curve for training and testing dataset –



Tuned - Figure 7 - AUC and ROC curve for training and testing dataset for NB

- ## ADA Boost classifier Model

Parameters –

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

Accuracy Score, Confusion Matrix and Classification Report of training data –

```
0.8587812008095346
[[3891  590]
 [ 666 3747]]
              precision    recall  f1-score   support

           0       0.85      0.87      0.86      4481
           1       0.86      0.85      0.86      4413

    accuracy                           0.86      8894
   macro avg       0.86      0.86      0.86      8894
weighted avg       0.86      0.86      0.86      8894
```

Tuned - Table 22 - AS, CM, CR for training data of ADA boost

Accuracy Score, Confusion Matrix and Classification Report of testing data –

```
0.8587812008095346
[[3891  590]
 [ 666 3747]]
              precision    recall  f1-score   support

           0       0.85      0.87      0.86      4481
           1       0.86      0.85      0.86      4413

    accuracy                           0.86      8894
   macro avg       0.86      0.86      0.86      8894
weighted avg       0.86      0.86      0.86      8894
```

Tuned - Table 23 - AS, CM, CR for testing data of ADA boost

- ## Bagging Classifier

Parameters –

```
BaggingClassifier(base_estimator=DecisionTreeClassifier(), n_estimators=100,
                  random_state=1)
```

Accuracy Score, Confusion Matrix and Classification Report of training data –

```
0.9851585338430403
[[4401   80]
 [  52 4361]]
              precision    recall  f1-score   support

           0       0.99      0.98      0.99      4481
           1       0.98      0.99      0.99      4413

    accuracy                           0.99      8894
   macro avg       0.99      0.99      0.99      8894
weighted avg       0.99      0.99      0.99      8894
```

Tuned - Table 24 - AS, CM, CR for testing data of bagging classifier

Accuracy Score, Confusion Matrix and Classification Report of testing data –

```
0.9315320041972718
[[1755  117]
 [ 144 1796]]
              precision    recall  f1-score   support

           0       0.92      0.94      0.93      1872
           1       0.94      0.93      0.93      1940

    accuracy                           0.93      3812
   macro avg       0.93      0.93      0.93      3812
weighted avg       0.93      0.93      0.93      3812
```

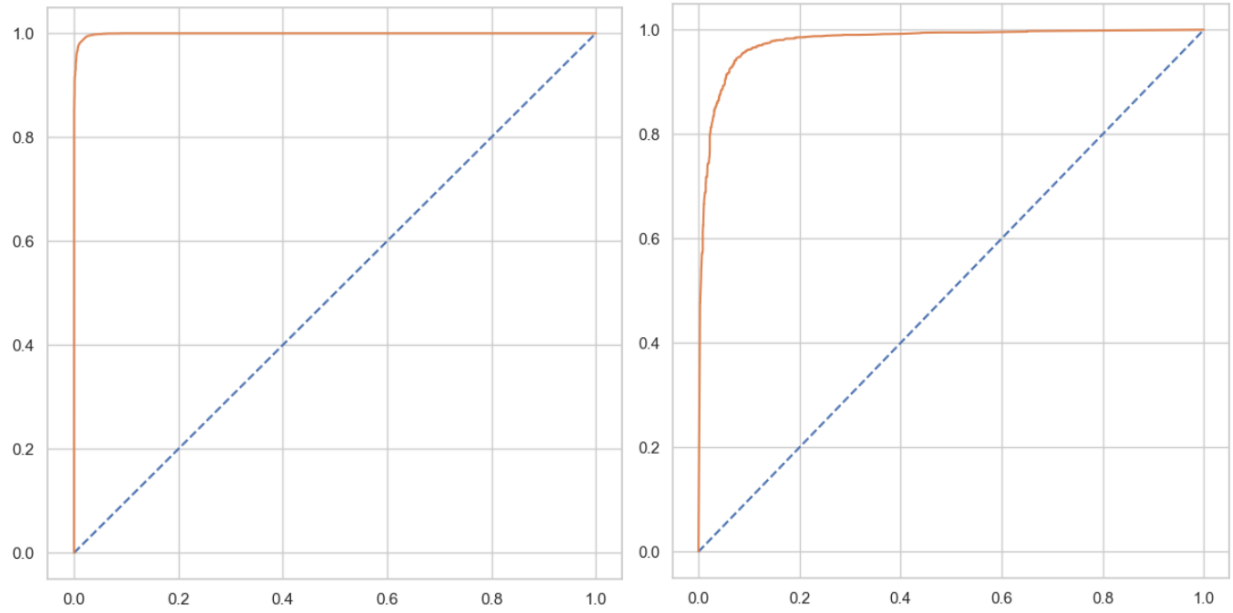
Tuned - Table 25 - AS, CM, CR for testing data of bagging classifier

AUC ROC curve for training and testing dataset –

AUC: 0.999

[<matplotlib.lines.Line2D at 0x22197ab4130>]

AUC: 0.978



Tuned - Figure 8 – Auc roc curve for training and testing dataset of bagging classifier