Capstone Project : Customer Churn Prediction (Part - 2)

Haresh Tayade
PGP – DSBA
Online sept'2021
Date – 14/08/2022

Table of Contents

DTH Churn Prediction

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 the potential churners. In this company, account churn is a major thing because a single account can have multiple customers. Hence, by losing one account the company might be losing more than one customer.

Need of the study:

The study of the customer churn is essential to as companies usually have a greater focus on customer acquisition and keep retention as a secondary priority and DTH industry is the most competitive industry with high acquisition costs and changing tariff plans. It can cost five times more to attract a new customer than it does to retain an existing one. Additionally, the existing customers have the potential to get the company new customers via word of referral helping the business in indirect selling. The study would be to focus on discovering the impact of the independent variables on the dependent variables and also understand the Root Cause Analysis of the factors leading the customer churning on a granular level.

Understanding Business Opportunities:

The Root Cause Analysis of the bleeding factors will indicate the severity of the existing issue and the creation of a churn prediction model will define the direction of the business in the upcoming days, which in turn will help the company to restructure its business model for customer retention and increasing goodwill. The company will be able to assess the touch points –

- *Product offers and packs.*
- *Financial process.*
- *Customer Care Services.*

Data Report

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

The data has been collected via –

- *KYC details of the customers.*
- *Customer care call assistance details.*
- *Aggregated revenue collection details on different periods.*
- *Demographic details captured from login devices.*

Summary from project 1 – Exploratory Datasets:

Dataset overview:

- *Dataset consisted of information about customers falling into various categories such as Personal, Business values derived from them, report of service offered to them and their level of satisfaction in response to company services.*
- *From the initial analysis it was observed that there were lot of null and unwanted values under various features. Presence of so many null values can be a hinder in good performance of models.*
- *Accordingly, unwanted variables were replaced by null values and later imputed, some of the values were renamed as they were actually a reflection of other values of the same features.*
- *And for this project I chose KNN imputation to impute null values based on similar data available in the Dataset.*

Univariate – Bivariate Analysis:

It was observed from the plots that following feature had outliers

'Tenure',

- ***'CC_Contacted_LY', 'rev_growth_yoy',***
- ***'Service_Score', coupon_used_for_payment',***
- ***'Account_user_count', 'Day_Since_CC_connect',***
- ***'rev_per_month', cashback'***

- *Also, the data in case of all variables were skewed, in either direction*
- *Independent features had very less or second to none correlation among them, which is ideal for model building as otherwise model may not often identify features which are truly important.*

Initial Observations:

From initial observation following were the insights identified:

- *The consumers belonging to Tier 1 cities churns less compared to Tier 2 and Tier 3 cities. The Tier 3 cities show the maximum impact in churn ratio, possibly, due to the lack of infrastructure development and customer service delays and may be preferring the traditional local owned cable operators to DTH connections.*
- *The consumers who have contacted Customer care in around 35 or more time seems to be churning more. Above case could have been due to lack of satisfaction among these customers in regards to company services.*
- *The consumers with payment modes 'Cash On Delivery' and 'E wallet' have churned inflated more than the ones using 'Credit/Debit Cards' and 'UPI'.*
- *There was no such significant difference in percentage of churning customers among both the genders.*
- *The consumers who have rated the service score from 2.0 – 4.0 are the ones who are more likely to churn compared to the ones who have given the lowest rating. Practically people with the lowest rating should have churned but in our case, it wasn't so, which may be possible because the Service Score recorded by the customer may not be an actual reflection of the level of satisfaction they have in accordance to the services.*
- *Company needs to develop better means to gain knowledge of actual customer satisfaction.*
- *The churn proportion was seen higher in case of 6 users for a single account which indicates that the account might be shared by friends/relatives dividing the monetary expense in return and common scenarios have been observed that shared accounts get dissolved easily due to personal clashes, financial factors and others.*

- *The customers using Regular Plus accounts seem to be churning more among others, possibly due to inadequate feature when compared to premiums*
- *The consumers belonging to the 'Single' and 'Divorced' category of marital status have churned more compared to the married consumers*
- *The revenue generated per month does not indicate significant pattern among churners and non-churners*

Splitting the data into training and test

The dataset was split into Train and Test data, further outlier treatment and null value Imputation was performed as part of pre-processing the data to build suitable model.

Checking the dimensions of the training and test data

```
X_train (7882, 17)
X_test (3378, 17)
y_train (7882,)
y_test (3378,)
```

Model building and interpretation:

Build various models (You can choose to build models for either or all of descriptive, predictive or prescriptive purposes)

Test your predictive model against the test set using various appropriate performance metrics

Interpretation of the model(s):

Various models were tried in order to predict potential churners among the customers depending on various attributes of the dataset.

Since the dataset for both the target class was highly imbalanced, Class balancing technique SMOTE was also used in order to improve model performances.

Below report comparison of various model built along with their performance on original and SMOTE data.

CART Model:

- *The algorithm of decision tree models works by repeatedly partitioning the data into multiple sub-spaces, so that the outcomes in each final sub-space is as homogeneous as possible.*
- *This approach is technically called recursive partitioning.*
- *The produced result consists of a set of rules used for predicting the outcome variable, which can be either: o a continuous variable, for regression trees o a categorical variable, for classification trees*
    - *The decision rules generated by the CART predictive model are generally visualized as a binary tree.*

Cart Model:

Train Data (Original):

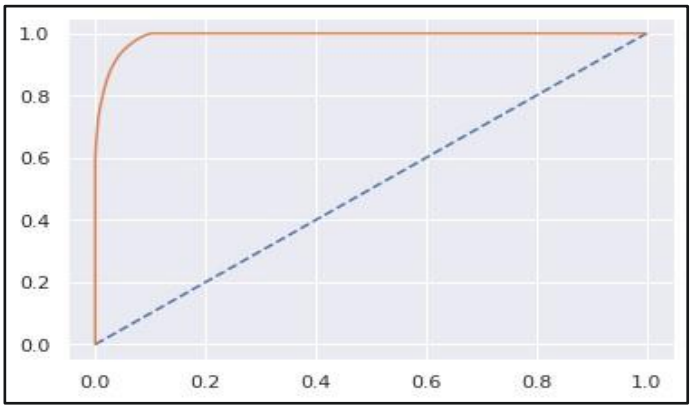
	precision	recall	f1-score	support
0	0.97	0.98	0.97	6556
1	0.89	0.85	0.87	1326
accuracy			0.96	7882
macro avg	0.93	0.91	0.92	7882
weighted avg	0.96	0.96	0.96	7882

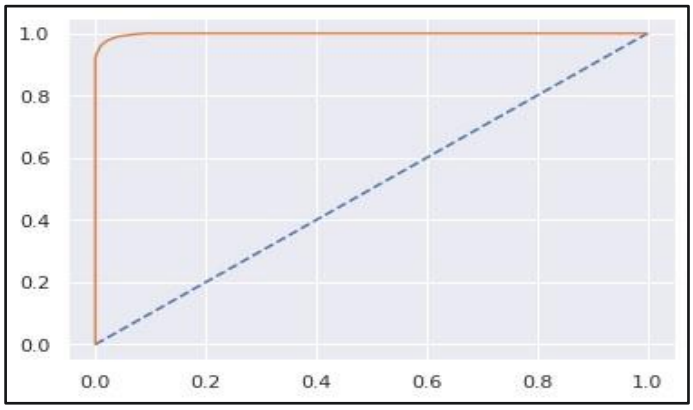
Train Data (SMOTE):

	precision	recall	f1-score	support
0	0.97	0.98	0.98	6556
1	0.98	0.97	0.98	6556
accuracy			0.98	13112
macro avg	0.98	0.98	0.98	13112
weighted avg	0.98	0.98	0.98	13112

AUC/ROC Curve:
AUC: 0.991

AUC/ROC Curve:
AUC: 0.999





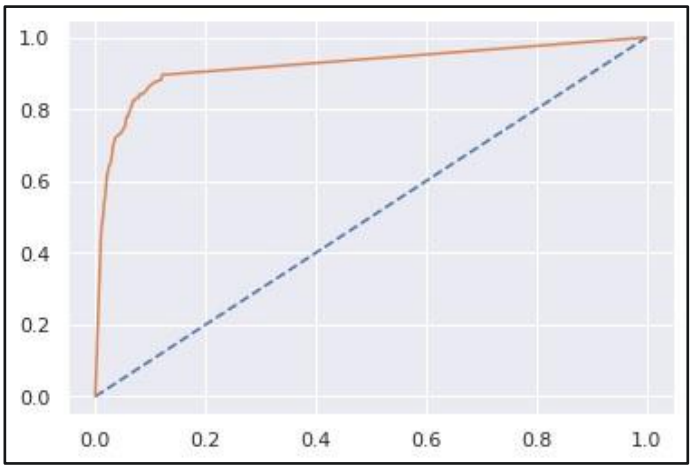
Test Data (Original):

	precision	recall	f1-score	support
0	0.94	0.96	0.95	2808
1	0.80	0.72	0.76	570
accuracy			0.92	3378
macro avg	0.87	0.84	0.86	3378
weighted avg	0.92	0.92	0.92	3378

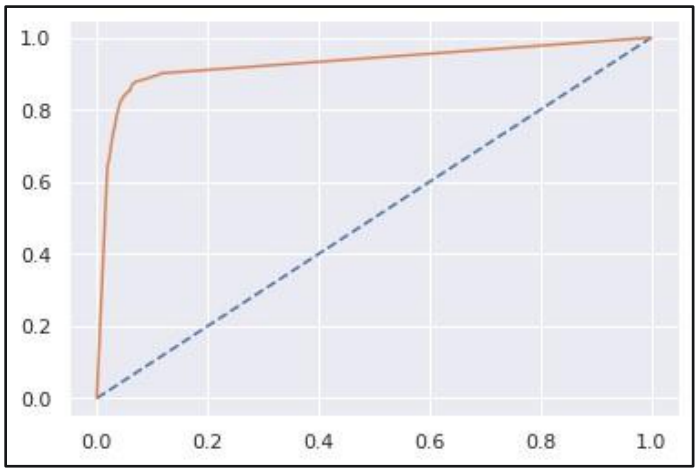
Test Data (SMOTE):

	precision	recall	f1-score	support
0	0.96	0.96	0.96	2808
1	0.80	0.82	0.81	570
accuracy			0.93	3378
macro avg	0.88	0.89	0.89	3378
weighted avg	0.94	0.93	0.94	3378

AUC/ROC Curve:
AUC: 0.921



AUC/ROC Curve:
AUC: 0.928



Variable Importance:

As per the cart model following features are not so important and hence can be dropped for model building

account_segment
Gender
coupon_used_for_payment
Service_Score

	Imp
Tenure	0.358386
Day_Since_CC_connect	0.076044
Complain_ly	0.075760
CC_Agent_Score	0.072811
cashback	0.063756
rev_per_month	0.053801
rev_growth_yoy	0.047238
CC_Contacted_LY	0.044788
Payment	0.036806
City_Tier	0.035157
Marital_Status	0.033750
Account_user_count	0.025596
Login_device	0.023237
account_segment	0.020636
Gender	0.016057
coupon_used_for_payment	0.009926
Service_Score	0.006251

Performance of CART Model after dropping unimportant variable on Test Set of original and SMOTE data:

Test Data (Original):

	precision	recall	f1-score	support
0	0.95	0.96	0.96	2808
1	0.81	0.75	0.78	570
accuracy			0.93	3378
macro avg	0.88	0.86	0.87	3378
weighted avg	0.93	0.93	0.93	3378

Test Data (SMOTE):

	precision	recall	f1-score	support
0	0.95	0.95	0.95	2808
1	0.75	0.77	0.76	570
accuracy			0.92	3378
macro avg	0.85	0.86	0.85	3378
weighted avg	0.92	0.92	0.92	3378

It can be observed model performance has improved by dropping unimportant factors, which implies that if CART model is used for Churn Prediction, then we no longer require these unimportant features in the dataset, thus saving valuable storage.

Artificial neural network:

- *An artificial neuron network (ANN) is a computational model based on the structure and functions of biological neural networks.*
- *Information that flows through the network affects the structure of the ANN because a neural network changes - or learns, in a sense - based on that input and output.*
- *ANNs are considered nonlinear statistical data modeling tools where the complex relationships between inputs and outputs are modeled or patterns are found.*
- *ANN is also known as a neural network.*
- *Artificial Neural Network primarily consists of three layers:*
- ***Input Layer:*** *○ As the name suggests, it accepts inputs in several different formats provided by the programmer.*
- ***Hidden Layer:***
  - *The hidden layer presents in-between input and output layers. It performs all the calculations to find hidden features and patterns.*
- ***Output Layer:***
  - *The input goes through a series of transformations using the hidden layer, which finally results in output that is conveyed using this layer.*
  - *The artificial neural network takes input and computes the weighted sum of the inputs and includes a bias. This computation is represented in the form of a transfer function.*

$$\sum_{i=1}^{n} Wi * Xi + b$$

  - *It determines weighted total is passed as an input to an activation function to produce the output. Activation functions choose whether a node should fire or not. Only those who are fired make it to the output layer. There are distinctive activation functions available that can be applied upon the sort of task we are performing.*

Advantages of Artificial Neural Network (ANN)

- ***Parallel processing capability:***
  Artificial neural networks have a numerical value that can perform more than one task simultaneously.
- ***Storing data on the entire network:***
  Data that is used in traditional programming is stored on the whole network, not on a database. The disappearance of a couple of pieces of data in one place doesn't prevent the network from working.
- ***Capability to work with incomplete knowledge:***
  After ANN training, the information may produce output even with inadequate data. The loss of performance here relies upon the significance of missing data.

- ***Having a memory distribution:***

  For ANN is to be able to adapt, it is important to determine the examples and to encourage the network according to the desired output by demonstrating these examples to the network. The succession of the network is directly proportional to the chosen instances, and if the event can't appear to the network in all its aspects, it can produce false output.
- ***Having fault tolerance:***

  Extortion of one or more cells of ANN does not prohibit it from generating output, and this feature makes the network fault-tolerance.

Disadvantages of Artificial Neural Network:

- ***Assurance of proper network structure:***

  There is no particular guideline for determining the structure of artificial neural networks. The appropriate network structure is accomplished through experience, trial, and error.
- ***Unrecognized behavior of the network:***

  It is the most significant issue of ANN. When ANN produces a testing solution, it does not provide insight concerning why and how. It decreases trust in the network.
- ***Hardware dependence:***

  Artificial neural networks need processors with parallel processing power, as per their structure. Therefore, the realization of the equipment is dependent.

- ***Difficulty of showing the issue to the network:***
- *ANNs can work with numerical data. Problems must be converted into numerical values before being introduced to ANN. The presentation mechanism to be resolved here will directly impact the performance of the network. It relies on the user's abilities.*
- ***The duration of the network is unknown:***
- *The network is reduced to a specific value of the error, and this value does not give us optimum results.*

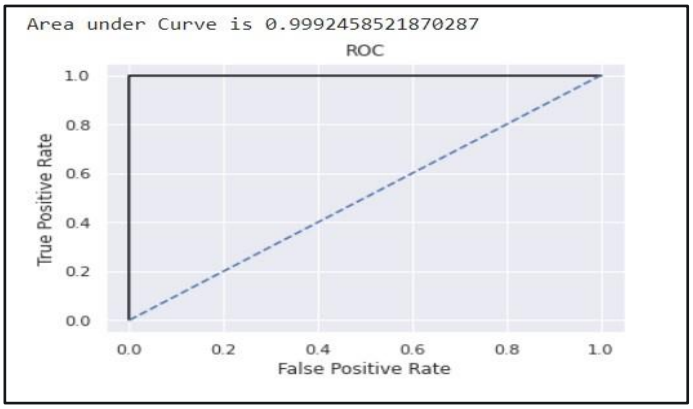
ANN Model:

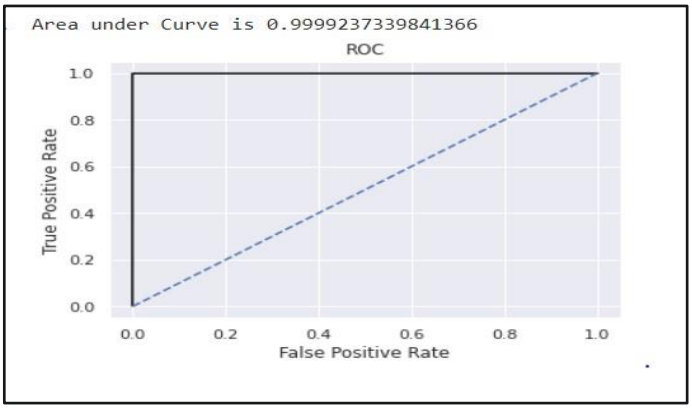
Train Data (Original) Train Data (SMOTE)

	precision	recall	f1-score	support
0	1.00	1.00	1.00	6556
1	1.00	1.00	1.00	1326
accuracy			1.00	7882
macro avg	1.00	1.00	1.00	7882
weighted avg	1.00	1.00	1.00	7882

	precision	recall	f1-score	support
0	1.00	1.00	1.00	6556
1	1.00	1.00	1.00	6556
accuracy			1.00	13112
macro avg	1.00	1.00	1.00	13112
weighted avg	1.00	1.00	1.00	13112

AUC/ROC Curve:





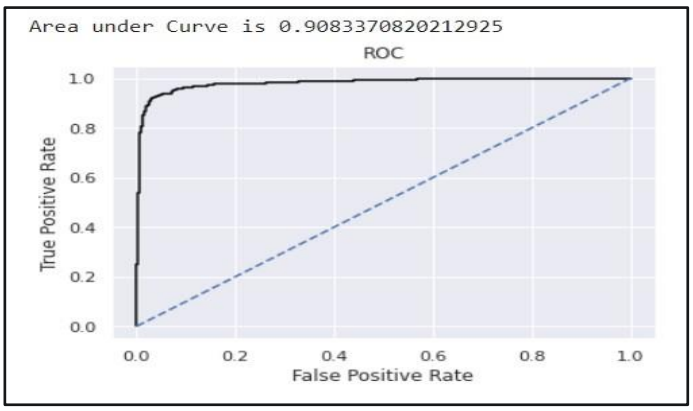
Test Data (Original):

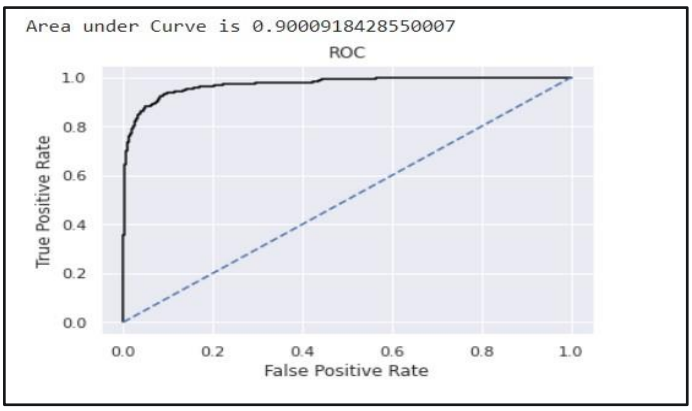
	precision	recall	f1-score	support
0	0.97	0.99	0.98	2808
1	0.94	0.83	0.88	570
accuracy			0.96	3378
macro avg	0.95	0.91	0.93	3378
weighted avg	0.96	0.96	0.96	3378

Test Data (SMOTE):

	precision	recall	f1-score	support
0	0.99	0.85	0.91	2808
1	0.56	0.95	0.70	570
accuracy			0.86	3378
macro avg	0.77	0.90	0.81	3378
weighted avg	0.92	0.86	0.88	3378

AUC/ROC Curve:





Inferences from ANN:

Performance of the ANN model after SMOTE has improved on test, although the most important factor Recall value has improved considerably, but Precision has taken a hit, implying that there are lot of false churning prediction among customers who are non-churners

Logistic Regression Model:

- *Logistic Regression is a Machine Learning classification algorithm that is used to predict the probability of a categorical dependent variable. In logistic regression, the dependent variable is a binary variable that contains data coded as 1 (yes, success, etc.) or 0 (no, failure, etc.). In other words, the logistic regression model predicts P(Y=1) as a function of X.*

Logistic Regression Assumptions

- *Binary logistic regression requires the dependent variable to be binary.*
- *For a binary regression, the factor level 1 of the dependent variable should represent the desired outcome.*
- *Only the meaningful variables should be included.*
- *The independent variables should be independent of each other. That is, the model should have little or no multicollinearity.*
- *The independent variables are linearly related to the log odds.*
- *Logistic regression requires quite large sample sizes ie. we don't want curse of dimensionality to effect our data*

Log. Regression Model:

Train Data (Original)

Classification Report of the training data of Logisitic Regression Model:

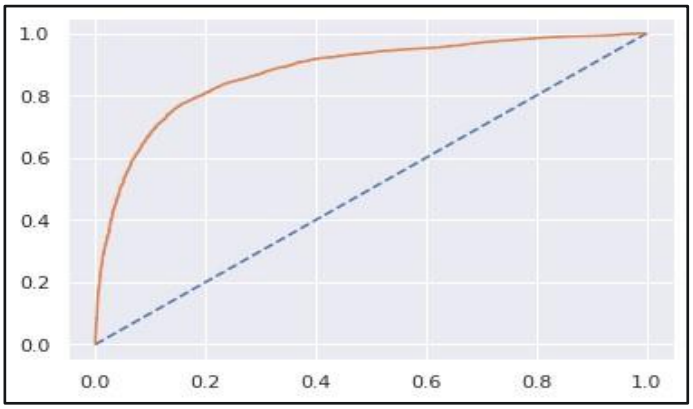
	precision	recall	f1-score	support
0	0.90	0.97	0.93	6556
1	0.75	0.45	0.56	1326
accuracy			0.88	7882
macro avg	0.82	0.71	0.75	7882
weighted avg	0.87	0.88	0.87	7882

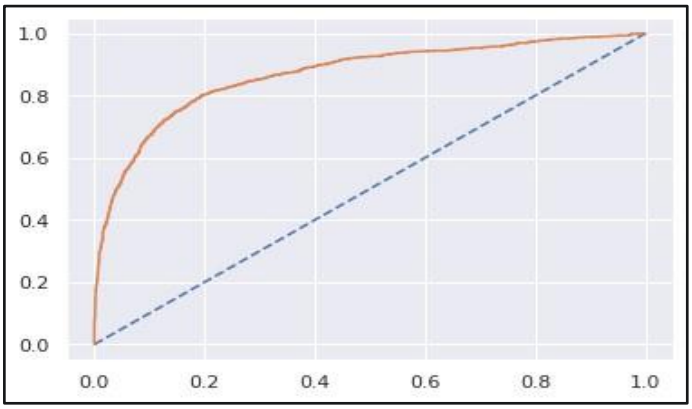
Train Data (SMOTE)

Classification Report of the training data of Logisitic Regression Model:

	precision	recall	f1-score	support
0	0.82	0.77	0.80	6556
1	0.79	0.83	0.81	6556
accuracy			0.80	13112
macro avg	0.80	0.80	0.80	13112
weighted avg	0.80	0.80	0.80	13112

AUC/ROC Curve:





Test Data (Original):

Classification Report of the test data of Logisitic Regression Model:

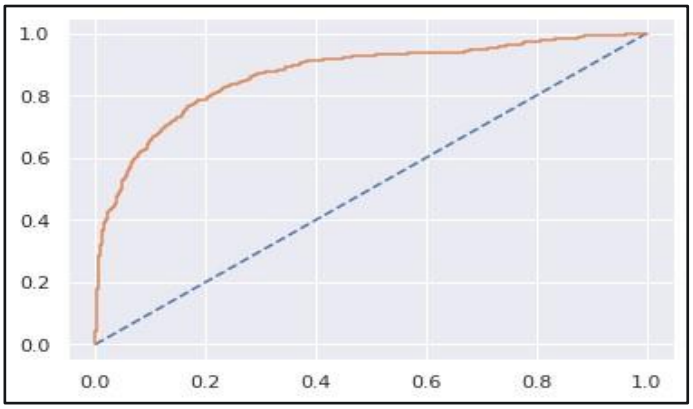
	precision	recall	f1-score	support
0	0.96	0.76	0.85	2808
1	0.42	0.83	0.55	570
accuracy			0.78	3378
macro avg	0.69	0.80	0.70	3378
weighted avg	0.86	0.78	0.80	3378

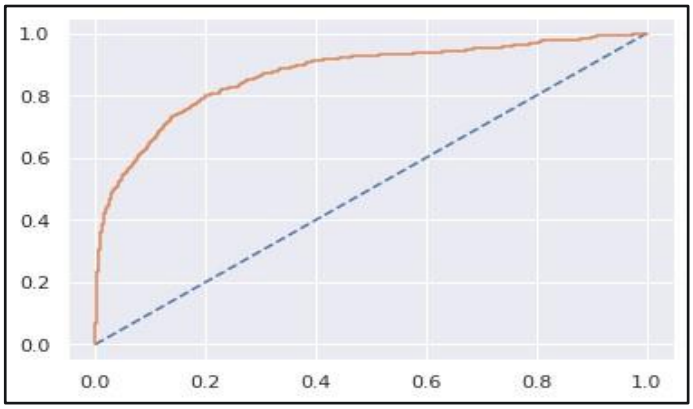
Test Data (SMOTE):

Classification Report of the test data of Logisitic Regression Model:

	precision	recall	f1-score	support
0	0.90	0.97	0.93	2808
1	0.79	0.46	0.58	570
accuracy			0.89	3378
macro avg	0.84	0.72	0.76	3378
weighted avg	0.88	0.89	0.87	3378

AUC/ROC Curve:





Inferences from Logistic Regression:

Although the model performance has improved after SMOTE, but still it has not performed to the sattisfaction to be put into production.

K-Nearest Neighbour:

- *K-Nearest Neighbour is one of the simplest Machine Learning algorithms based on Supervised Learning technique.*
- *K-NN algorithm assumes the similarity between the new case/data and available cases(nearest neighbour points) and put the new case into the category that is most similar to the available categories.*
- *K-NN algorithm stores all the available data and classifies a new data point based on the similarity. This means when new data appears then it can be easily classified into a well suite category by using K- NN algorithm.*
- *The approach to find nearest neighbors using distance between the query point and all other points is called the brute force.*
- *Becomes time costly and inefficient with increase in number of points*
- *Determining the optimal K is the challenge in K Nearest Neighbor classifiers.*
- *Larger value of K suppresses impact of noise but prone to majority class dominating.*

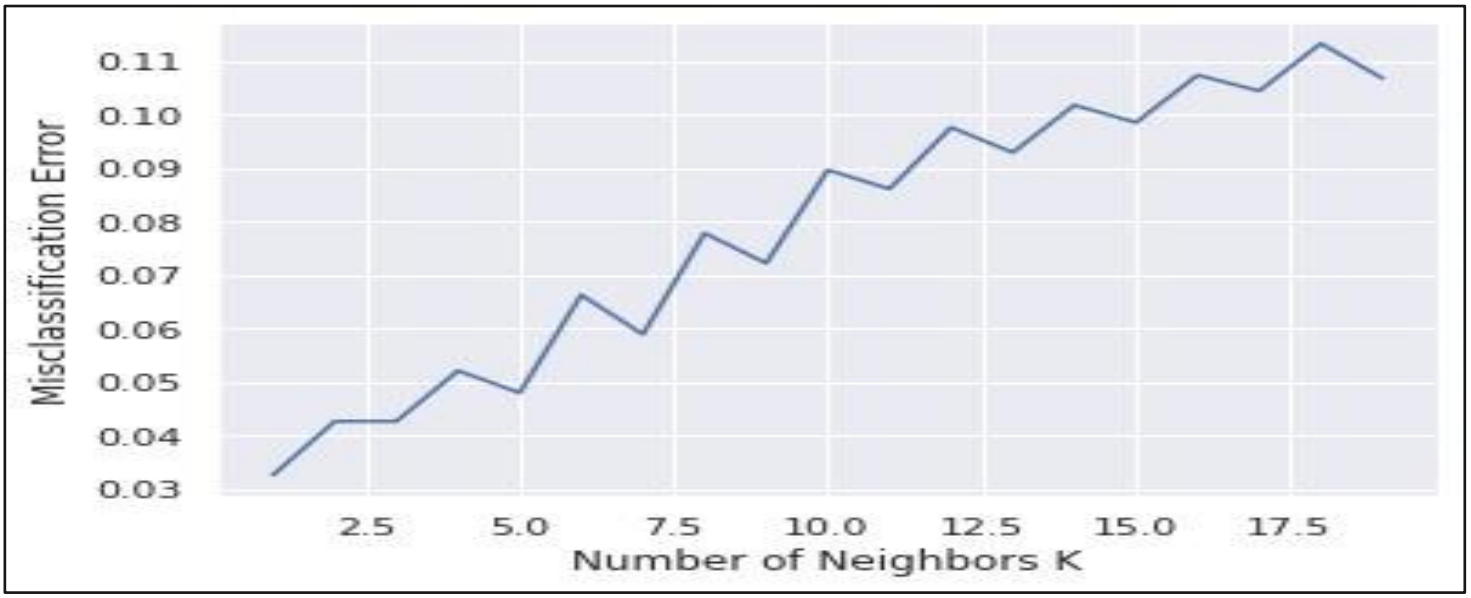
Radius Neighbour Classifier:

- *Implements learning based on number of neighbors within a fixed radius r of each training point, where r is a floating-point value specified by the user.*
- *May be a better choice when the sampling is not uniform. However, when there are many attributes and data is sparse, this method becomes ineffective due to curse of dimensionality.*

KD Tree nearest neighbour:

- *Approach helps reduce the computation time.*
- *Very effective when we have large data points but still not too many dimensions*
- *Classification is computed from a simple majority vote of the nearest neighbors of each point.*
- *Suited for classification where relationship between features and target classes is numerous, complex and difficult to understand and yet items in a class tend to be fairly homogenous on the values of attributes*
- *Not suitable if the data is too noisy and the target classes do not have clear demarcation in terms of attribute values*

Values for various parameter for the KNN model were decided plotting MCEs for various K values and the K values with the least MCE was used to build KNN model



Model Performances on Original and SMOTE Data:

Train Data (Original) *Train Data (SMOTE)*

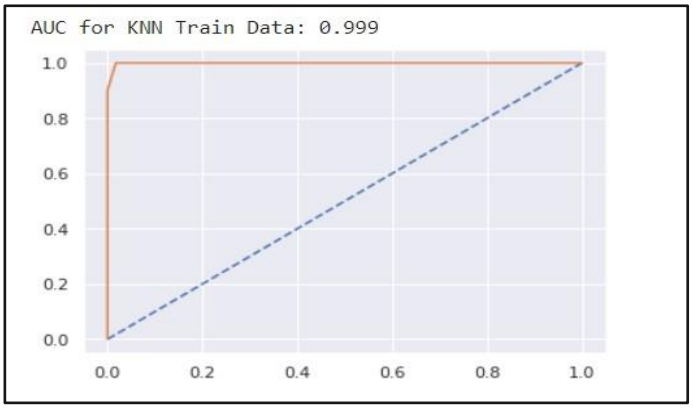
Classification Report of the train data of KNN Model:

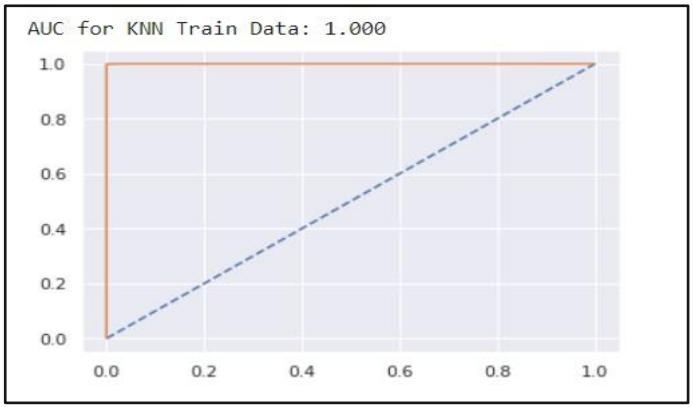
	precision	recall	f1-score	support
0	0.98	1.00	0.99	6556
1	1.00	0.90	0.95	1326
accuracy			0.98	7882
macro avg	0.99	0.95	0.97	7882
weighted avg	0.98	0.98	0.98	7882

Classification Report of the train data of KNN Model:

	precision	recall	f1-score	support
0	1.00	1.00	1.00	6556
1	1.00	1.00	1.00	6556
accuracy			1.00	13112
macro avg	1.00	1.00	1.00	13112
weighted avg	1.00	1.00	1.00	13112

AUC/ROC Curve:





Test Data (Original):

Classification Report of the test data of KNN Model:

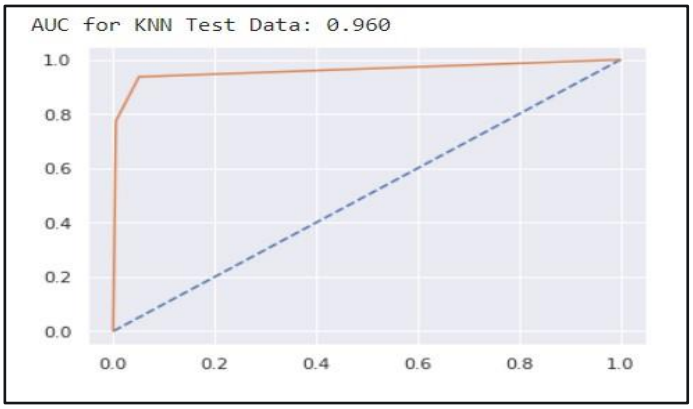
	precision	recall	f1-score	support
0	0.96	0.99	0.97	2808
1	0.97	0.77	0.86	570
accuracy			0.96	3378
macro avg	0.96	0.88	0.92	3378
weighted avg	0.96	0.96	0.96	3378

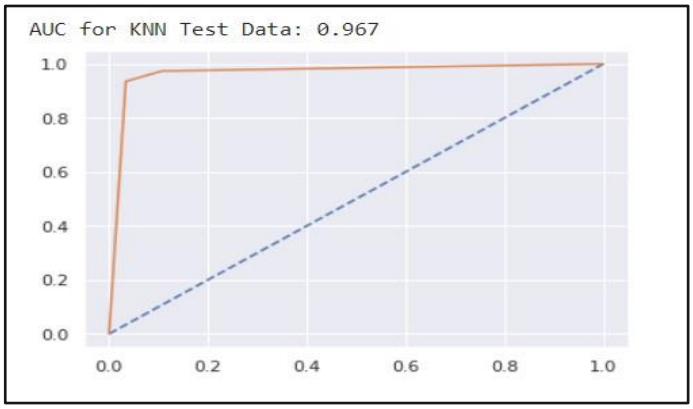
Test Data (SMOTE):

Classification Report of the test data of KNN Model:

	precision	recall	f1-score	support
0	0.99	0.97	0.98	2808
1	0.85	0.94	0.89	570
accuracy			0.96	3378
macro avg	0.92	0.95	0.93	3378
weighted avg	0.96	0.96	0.96	3378

AUC/ROC Curve:





Inferences from KNN Model:

The model performance has improved after SMOTE considerably well on test data. Model has a very high Recall value, suggesting that it is able to predict potential churners with very high accuracy.

2. Model Tuning

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

Applying Ensemble Methods on the original and SMOTE Data:

Random Forest Classification Model:

- *The random forest is a classification algorithm consisting of many decisions trees.*
- *It uses bagging and feature randomness when building each individual tree to try to create an uncorrelated forest of trees whose prediction by committee is more accurate than that of any individual tree.*
- *A forest is comprised of trees. It is said that the more trees it has, the more robust a forest is.*
- *Random forests create decision trees on randomly selected data samples, gets prediction from each tree and selects the best solution by means of voting.*
- *It also provides a pretty good indicator of the feature importance.*
- *How does the algorithm work?*

It works in four steps:

o Select random samples from a given dataset. o Construct a decision tree for each sample and get a prediction result from each decision tree. o Perform a vote for each predicted result. o Select the prediction result with the most votes as the final prediction.

Model Performances on Original and SMOTE Data:

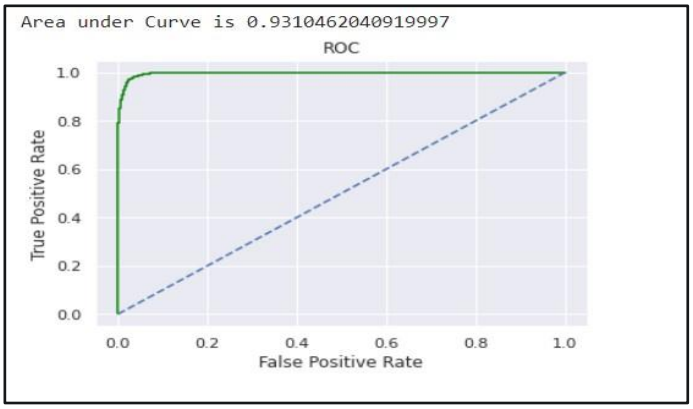
Train Data (Original)

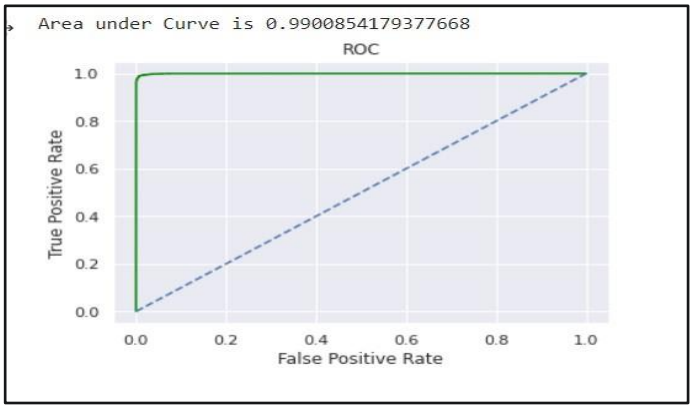
	precision	recall	f1-score	support
0	0.97	1.00	0.98	6556
1	0.98	0.87	0.92	1326
accuracy			0.97	7882
macro avg	0.97	0.93	0.95	7882
weighted avg	0.97	0.97	0.97	7882

Train Data (SMOTE)

	precision	recall	f1-score	support
0	0.99	0.99	0.99	6556
1	0.99	0.99	0.99	6556
accuracy			0.99	13112
macro avg	0.99	0.99	0.99	13112
weighted avg	0.99	0.99	0.99	13112

AUC/ROC Curve:





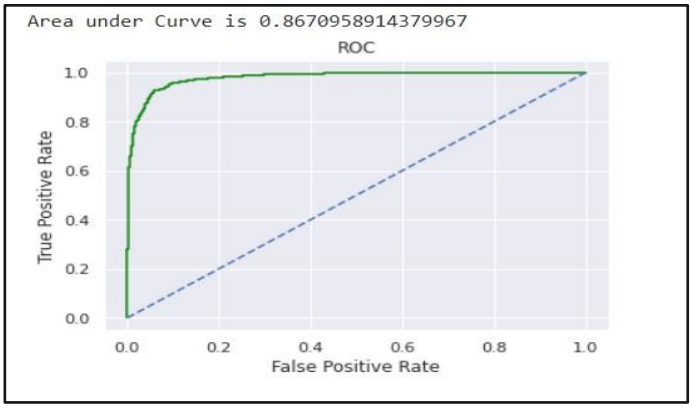
Test Data (Original):

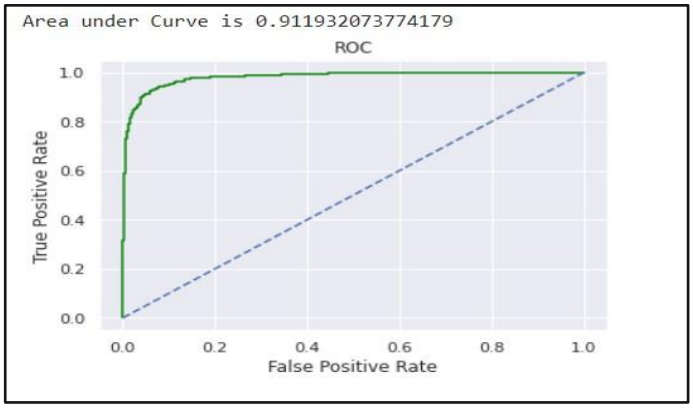
	precision	recall	f1-score	support
0	0.95	0.99	0.97	2808
1	0.92	0.75	0.82	570
accuracy			0.95	3378
macro avg	0.94	0.87	0.90	3378
weighted avg	0.95	0.95	0.94	3378

Test Data (SMOTE):

	precision	recall	f1-score	support
0	0.97	0.98	0.97	2808
1	0.88	0.85	0.86	570
accuracy			0.95	3378
macro avg	0.92	0.91	0.92	3378
weighted avg	0.95	0.95	0.95	3378

AUC/ROC Curve:





Variable Importance:

	Imp
Tenure	0.344193
Complain_ly	0.083314
Day_Since_CC_connect	0.069849
cashback	0.059649
CC_Agent_Score	0.056994
CC_Contacted_LY	0.054504
rev_per_month	0.053757
rev_growth_yoy	0.048181
Payment	0.041549
Marital_Status	0.040298
account_segment	0.033574
City_Tier	0.027549
Account_user_count	0.026827
coupon_used_for_payment	0.018560
Login_device	0.016725
Gender	0.014948
Service_Score	0.009530

As per the random forest model following features are not so important and hence can be dropped for model building

- *Login_devices*
- *Gender*
- *coupon_used_for_payment*
- *Service_Score*

Performance of Random Forest Model after dropping unimportant variable on Test Set of original and SMOTE data:

	precision	recall	f1-score	support
0	0.96	0.99	0.97	2808
1	0.92	0.80	0.86	570
accuracy			0.95	3378
macro avg	0.94	0.89	0.91	3378
weighted avg	0.95	0.95	0.95	3378

	precision	recall	f1-score	support
0	0.97	0.97	0.97	2808
1	0.85	0.85	0.85	570
accuracy			0.95	3378
macro avg	0.91	0.91	0.91	3378
weighted avg	0.95	0.95	0.95	3378

Inferences from RF Model:

- *The model performance has improved after SMOTE considerably well on test data. Model has a very high Recall value, suggesting that it is able to predict potential churners with very high accuracy.*
- *It can be observed model performance has improved by dropping unimportant factors, which implies that if Random Forrest model is used for Churn Prediction, then we no longer require these unimportant features in the dataset, thus saving valuable storage*

Boosting:

- *It is a machine learning ensemble meta-algorithm for principally reducing bias, and furthermore variance in supervised learning, and a group of machine learning algorithms that convert weak learner to string ones.*
- *Sequential ensemble methods where the base learners are generated sequentially. • Example: Adaboost, Stochastic Gradient Boosting*

AdaBoosting (Adaptive Boosting):

In AdaBoost, the successive learners are created with a focus on the ill fitted data of the previous learner
Each successive learner focuses more and more on the harder to fit data i.e. their residuals in the previous tree

Model Performance on Train and test Data:

Train Data (Original)

Classification Report of the train data of AdaBoosting Model:

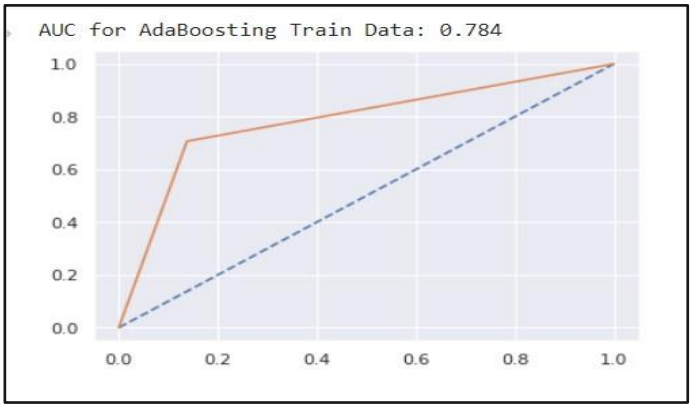
	precision	recall	f1-score	support
0	0.94	0.86	0.90	6556
1	0.51	0.71	0.59	1326
accuracy			0.84	7882
macro avg	0.72	0.78	0.74	7882
weighted avg	0.86	0.84	0.85	7882

Train Data (SMOTE)

Classification Report of the train data of AdaBoosting Model:

	precision	recall	f1-score	support
0	0.76	0.86	0.81	6556
1	0.84	0.73	0.78	6556
accuracy			0.80	13112
macro avg	0.80	0.80	0.80	13112
weighted avg	0.80	0.80	0.80	13112

AUC/ROC Curve:



Test Data (Original):

Test Data (SMOTE):

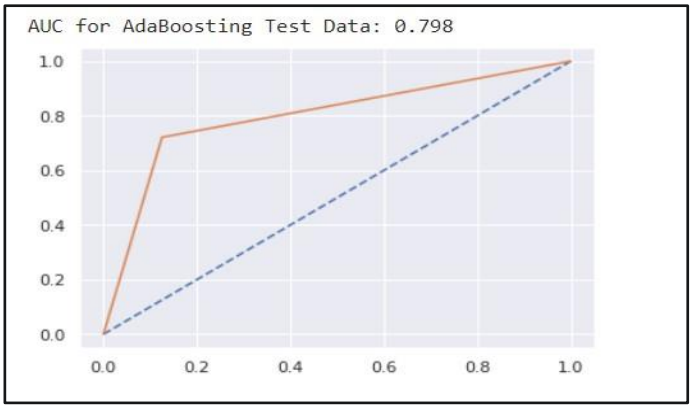
Classification Report of the test data of AdaBoosting Model:

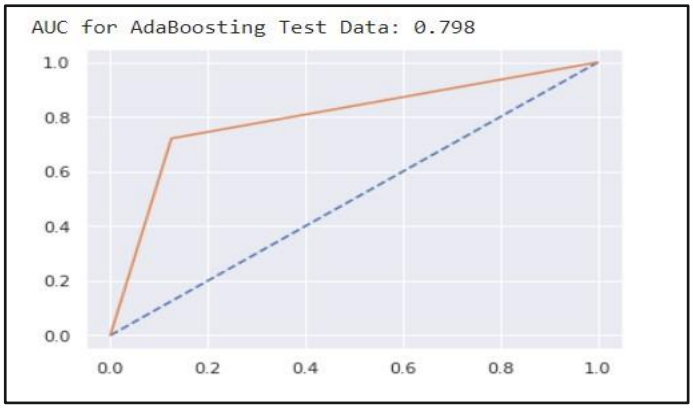
	precision	recall	f1-score	support
0	0.94	0.87	0.91	2808
1	0.54	0.72	0.62	570
accuracy			0.85	3378
macro avg	0.74	0.80	0.76	3378
weighted avg	0.87	0.85	0.86	3378

Classification Report of the test data of AdaBoosting Model:

	precision	recall	f1-score	support
0	0.94	0.87	0.91	2808
1	0.54	0.72	0.62	570
accuracy			0.85	3378
macro avg	0.74	0.80	0.76	3378
weighted avg	0.87	0.85	0.86	3378

AUC/ROC Curve:





Variable Importance:

	Imp
Tenure	1.0
Marital_Status	0.0
cashback	0.0
Day_Since_CC_connect	0.0
coupon_used_for_payment	0.0
rev_growth_yoy	0.0
Complain_ly	0.0
rev_per_month	0.0
CC_Agent_Score	0.0
City_Tier	0.0
account_segment	0.0
Account_user_count	0.0
Service_Score	0.0
Gender	0.0
Payment	0.0
CC_Contacted_LY	0.0
Login_device	0.0

AdaBoosting model seems to be considering only one features as important for Churn prediction and seems to be misjudging while considering importance of other independent features.

Hence for this dataset it cannot be used for Churn Prediction

Gradient Boosting:

Gradient boosting classifiers are the AdaBoosting method combined with weighted minimization, after which the classifiers and weighted inputs are recalculated.

The objective of Gradient Boosting classifiers is to minimize the loss, or the difference between the actual class value of the training example and the predicted class value.

It isn't required to understand the process for reducing the classifier's loss, but it operates similarly to gradient descent in a neural network.

Model Performances on Original and SMOTE Data:

Train Data (Original)

Classification Report of the train data of Gradient Boosting Model:

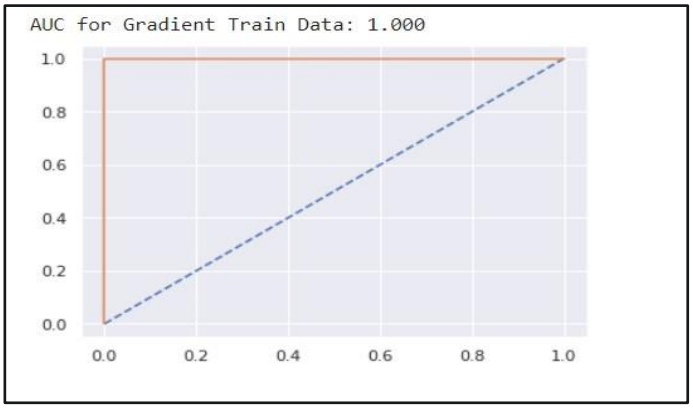
	precision	recall	f1-score	support
0	1.00	1.00	1.00	6556
1	1.00	1.00	1.00	1326
accuracy			1.00	7882
macro avg	1.00	1.00	1.00	7882
weighted avg	1.00	1.00	1.00	7882

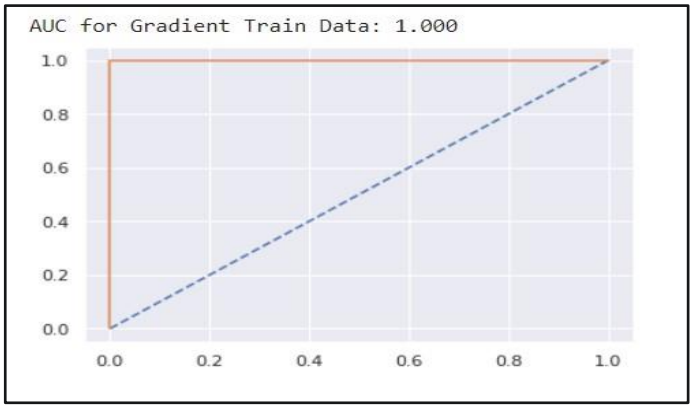
Train Data (SMOTE)

Classification Report of the train data of Gradient Boosting Model:

	precision	recall	f1-score	support
0	1.00	1.00	1.00	6556
1	1.00	1.00	1.00	6556
accuracy			1.00	13112
macro avg	1.00	1.00	1.00	13112
weighted avg	1.00	1.00	1.00	13112

AUC/ROC Curve:





Test Data (Original):

Test Data (SMOTE):

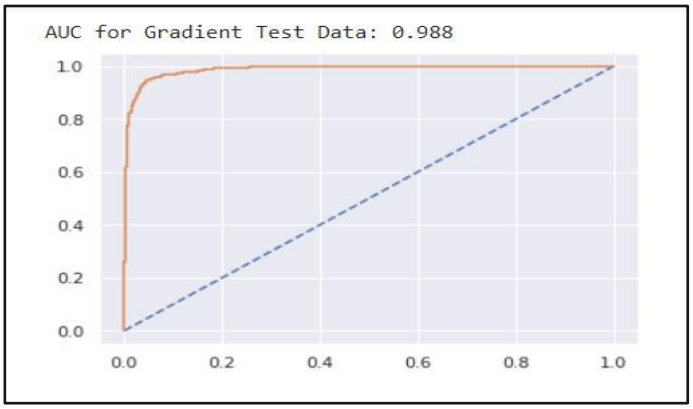
Classification Report of the test data of Gradient Boosting Model:

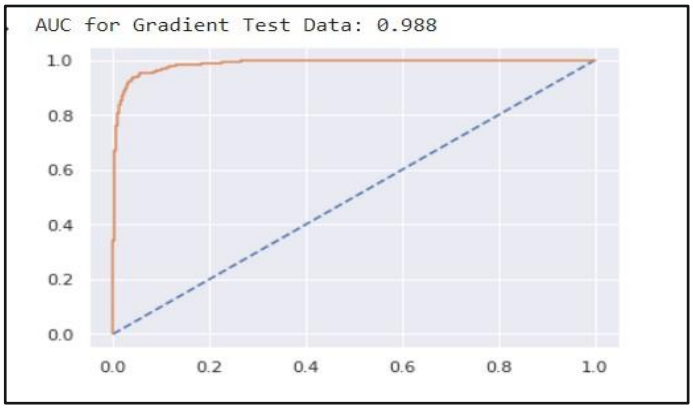
	precision	recall	f1-score	support
0	0.97	0.98	0.98	2808
1	0.90	0.87	0.88	570
accuracy			0.96	3378
macro avg	0.94	0.92	0.93	3378
weighted avg	0.96	0.96	0.96	3378

Classification Report of the test data of Gradient Boosting Model:

	precision	recall	f1-score	support
0	0.97	0.98	0.98	2808
1	0.91	0.86	0.88	570
accuracy			0.96	3378
macro avg	0.94	0.92	0.93	3378
weighted avg	0.96	0.96	0.96	3378

AUC/ROC Curve:





Variable Importance:

	Imp
Tenure	0.434508
Complain_ly	0.101191
Day_Since_CC_connect	0.063264
rev_per_month	0.055385
CC_Agent_Score	0.049371
cashback	0.045625
Payment	0.037376
CC_Contacted_LY	0.034400
Marital_Status	0.033376
rev_growth_yoy	0.030310
account_segment	0.030119
City_Tier	0.022449
Account_user_count	0.021701
coupon_used_for_payment	0.015059
Service_Score	0.010495
Gender	0.008877
Login_device	0.006493

As per the random forest model following features are not so important and hence can be dropped for model building

- *CC_Contacted_LY*
- *Gender*
- *Service_Score*

Performance of Gradient Boosting Model after dropping unimportant variable on Test Set of SMOTE data:

Classification Report of the test data of Gradient Boosting Model:

	precision	recall	f1-score	support
0	0.97	0.98	0.98	2808
1	0.90	0.87	0.88	570
accuracy			0.96	3378
macro avg	0.93	0.92	0.93	3378
weighted avg	0.96	0.96	0.96	3378

Inferences from Gradient Boosting Model:

- *The model performance has improved after SMOTE considerably well on test data. Model has a very high Recall value, suggesting that it is able to predict potential churners with very high accuracy.*
- *It can be observed model performance has improved by dropping unimportant factors, which implies that if Gradient Boosting model is used for Churn Prediction, then we no longer require these unimportant features in the dataset, thus saving valuable storage.*
- *We can select Gradient Boosting Model for churn prediction based on various factors like model simplification and upper-hand in model performance over other models.*

Bagging (Bootstrap Aggregation):

- *Reduced chances of over fitting by training each model only with a randomly chosen subset of the training data. Training can be done in parallel.*
- *Essentially trains a large number of "strong" learners in parallel (each model is an over fit for that subset of the data)*
- *Combines (averaging or voting) these learners together to "smooth out" predictions.*
- *For this case we are using Random Forest Classifier as our base estimator.*

Model Performance on Train and test Data:

Train Data (Original)

Classification Report of the train data of Bagging Model:

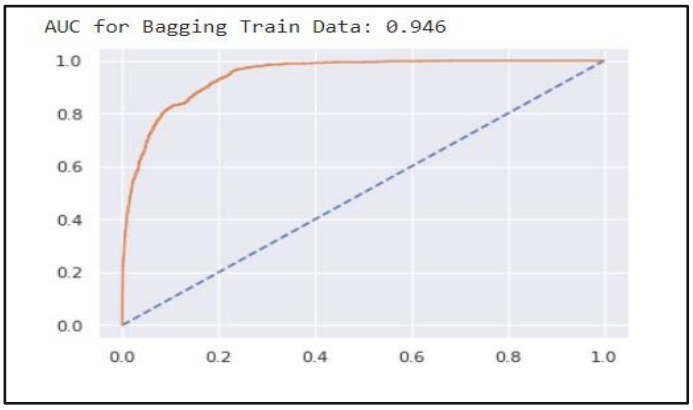
	precision	recall	f1-score	support
0	0.91	0.98	0.94	6556
1	0.84	0.53	0.65	1326
accuracy			0.90	7882
macro avg	0.88	0.75	0.80	7882
weighted avg	0.90	0.90	0.89	7882

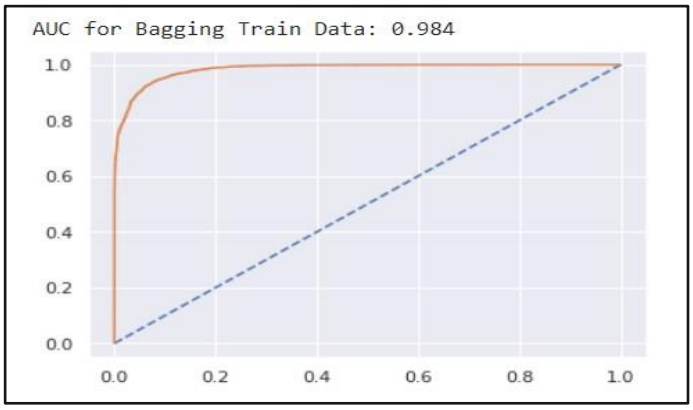
Train Data (SMOTE)

Classification Report of the train data of Bagging Model:

	precision	recall	f1-score	support
0	0.93	0.93	0.93	6556
1	0.93	0.93	0.93	6556
accuracy			0.93	13112
macro avg	0.93	0.93	0.93	13112
weighted avg	0.93	0.93	0.93	13112

AUC/ROC Curve:





Test Data (Original):

Classification Report of the test data of Bagging Model:

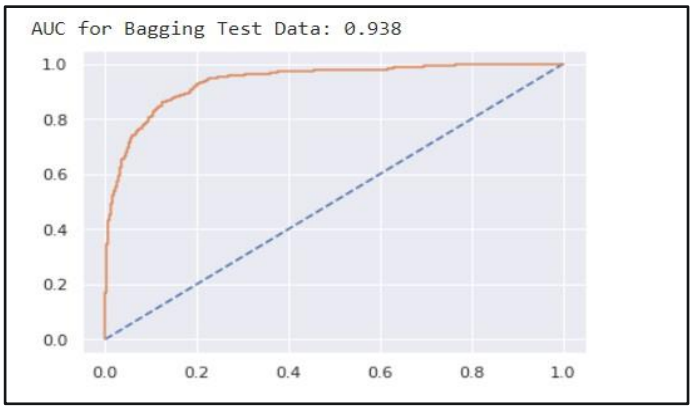
	precision	recall	f1-score	support
0	0.91	0.98	0.94	2808
1	0.83	0.51	0.63	570
accuracy			0.90	3378
macro avg	0.87	0.75	0.79	3378
weighted avg	0.89	0.90	0.89	3378

Test Data (SMOTE):

Classification Report of the test data of Bagging Model:

	precision	recall	f1-score	support
0	0.95	0.93	0.94	2808
1	0.69	0.76	0.72	570
accuracy			0.90	3378
macro avg	0.82	0.84	0.83	3378
weighted avg	0.91	0.90	0.90	3378

AUC/ROC Curve:



Inference from Bagging Model:

It can be observed that Although the model performance has improved after SMOTE, but still it has not performed to the sattisfaction to be put into production.

Overview of Model Performance on Original and SMOTE Data:

Original Data:

	CART Train	CART Test	Neural Network Train	Neural Network Test	Log_Reg Train	Log_Reg Test	KNN Train	KNN Test
Accuracy	0.97	0.93	1.0	0.96	0.88	0.89	0.98	0.96
AUC	1.00	0.91	1.0	0.91	0.87	0.87	1.00	0.96
Recall	0.89	0.75	1.0	0.83	0.45	0.46	0.90	0.77
Precision	0.92	0.81	1.0	0.94	0.75	0.79	1.00	0.97
F1 Score	0.90	0.78	1.0	0.88	0.56	0.58	0.95	0.86

SMOTE Data:

	CART Train	CART Test	Neural Network Train	Neural Network Test	Log_Reg Train	Log_Reg Test	KNN Train	KNN Test
Accuracy	0.98	0.92	1.0	0.86	0.80	0.78	1.0	0.96
AUC	1.00	0.91	1.0	0.90	0.88	0.87	1.0	0.97
Recall	0.98	0.77	1.0	0.95	0.83	0.83	1.0	0.94
Precision	0.99	0.75	1.0	0.56	0.79	0.42	1.0	0.85
F1 Score	0.98	0.76	1.0	0.70	0.81	0.55	1.0	0.89

Inferences:

By Using SMOTE model performance of different model have been sattisfactory.

In most cases, models have been able to identify more than 80% of true Churners.

Ensemble Methods:

Original Data:

	Random Forest Train	Random Forest Test	Bagging Train	Bagging Test	AdaBoosting Train	AdaBoosting Test	Gradient Boosting Train	Gradient Boosting Test
Accuracy	0.99	0.95	0.90	0.90	0.84	0.85	1.0	0.96
AUC	0.97	0.89	0.95	0.93	0.78	0.80	1.0	0.99
Recall	0.94	0.80	0.53	0.51	0.71	0.72	1.0	0.87
Precision	0.99	0.92	0.84	0.83	0.51	0.54	1.0	0.90
F1 Score	0.96	0.86	0.65	0.63	0.59	0.62	1.0	0.88

SMOTE Data:

	Random Forest Train	Random Forest Test	Bagging Train	Bagging Test	AdaBoosting Train	AdaBoosting Test	Gradient Boosting Train	Gradient Boosting Test
Accuracy	0.99	0.95	0.93	0.90	0.80	0.85	1.0	0.96
AUC	0.99	0.91	0.98	0.94	0.80	0.80	1.0	0.98
Recall	0.99	0.85	0.93	0.76	0.73	0.72	1.0	0.87
Precision	0.99	0.85	0.93	0.69	0.84	0.54	1.0	0.90
F1 Score	0.99	0.85	0.93	0.72	0.78	0.62	1.0	0.88

Inferences:

Performance of Ensemble methods are comapratively better than general models.
Additionally by Using SMOTE, there has been a vast improvement in model performance of different model
In most cases, models have been able to identify more than 85% of true Churners.

Final Model Selection:

Among the various models built for DTH Dataset, following models have performed exceeding well when compared to other.

- *Artificial Neural Network*
- *KNN*
- *Gradient Boosting*

For DTH Dataset the final model that I chose to use for Churn Predicition will be ***Gradient Boosting,*** *due to following advantages over the other 2 models*

- *Speed: Constructing weak models is computationally cheap ⭘ Often provides predictive accuracy that cannot be beat.*
- *Lots of flexibility - can optimize on different loss functions and provides several hyperparameter tuning options that make the function fit very flexible.*
- *No data pre-processing required - often works great with categorical and numerical values as is.*

- *For ANN, there is no particular guideline for determining the structure of artificial neural networks. The appropriate network structure is accomplished through experience, trial, and error*
  - *Artificial neural networks need processors with parallel processing power, as per their structure.*

- *For KNN, Accuracy depends on the quality of the data. With large data, the prediction stage might be slow. Sensitive to the scale of the data and irrelevant features. Require high memory – need to store all of the training data. Given that it stores all of the training, it can be computationally expensive*

Final Insights and Suggestion:

With using Gradient Boosting Model, following insights were derived which was also reflected while initial Exploratory Data Analysis:
Following features seems to be insignificant while using Gradient Boosting Model:

- *CC_Contacted_LY*
- *Gender*
- *Service_Score*

Recommendations:

- *Since, there is no significance in the Gender feature, we can opt for disregarding this feature for the study.*
- *However, customer satisfaction rate plays a crucial role in Customer Churn. Therefore, we can change the grading system of numerical to a string based system viz Highly Satisfied, Satisfied and Not Satisfied. This will enable the customers to understand the rating system better and will help in recording the correct data.*
- *Additionally, it has been noted that the feature CC_contacted_LY holds no significance in building our model. It indicates that the calling history of the customer cannot truly determine the churn ratio as even customers who never contacted the helpline can also churn and vice versa.*