Churning Analysis Capstone

Business Report

Note 2

DSBA

Vikash Kumar

Aug'22 Batch

Table of Contents

Content of Tables

Content of Figure and Plots

In continuation of Capstone Customer Churning PN1……

Train Test Split (70:30)

For the given business problem, 'Churn' is the target variable since the problem is to come up with a model to predict whether a particular customer will Churn or not.

```
X – Independent variable (Removing 'Churn' variable)

y – Dependent/ Target variable (Having only 'Churn' variable)
```

1.The train-test split procedure is used to estimate the performance of machine learning algorithms when they are used to make predictions on data not used to train the model.
2.The train_test_split() method is used to split our data into train and test sets. Samples from the original training dataset are split into the two subsets using random selection. This is to ensure that the train and test datasets are representative of the original dataset.
Procedure: First, we need to divide our data into features (X) and labels (y). The dataFrame gets divided into X_train,X_test , y_train and y_test. X_train and y_train sets are used for training and fitting the model. The X_test and y_test sets are used for testing the model if it's predicting the right outputs/labels. we can explicitly test the size of the train and test sets. 3.The procedure has one main configuration parameter, which is the size of the train and test sets. This is most commonly expressed as a percentage between 0 and 1 for either the train or test datasets, we keep our train sets larger than the test sets.
4.There is no optimal split percentage on train and test 70:30 are for our analysis.
5.The objective is to estimate the performance of the machine learning model on new data: data not used to train the model
6.This is done to ensure that datasets are a representative sample (e.g. random sample) of the original dataset, which in turn, should be a representative sample of observations from the problem domain.
7.Stratified Train-Test Splits: This classification problems do not have a balanced number of examples for each class label. As such, it is desirable to split the dataset into train and test sets in a way that preserves the same proportions of examples in each class as observed in the original dataset.

Next step is to Split the data into training and testing test. Splitting the data as 70% training and 30% testing with a random state = 42 and stratify = y. Because the y is in minority. Output of this step will be: Training independent variable (X_train), Testing independent variable (X-test), Training dependent variable (y_train) and testing dependent variable (y_test).
The Shape and size of Train and Test data

```
Number of rows and columns of the training set for the independent v
ariables: (7882, 23)
Number of rows and columns of the training set for the dependent var
iable: (7882,)
Number of rows and columns of the test set for the independent varia
bles: (3378, 23)
Number of rows and columns of the test set for the dependent variabl
e: (3378,)
```

TABLE 1 TRAIN TEST SHAPE

Resampling using SMOTE

SMOTE is a popular oversampling technique that generates synthetic samples of the minority class to balance the dataset. It creates new synthetic samples by interpolating between existing minority class samples.
This helps in increasing the representation of the minority class and improving the performance of machine learning models.

Using SMOTE can be beneficial in this scenario as it can potentially improve the model's ability to capture patterns and make accurate predictions for the minority class.

Therefore, in this case, since the target variable is imbalanced with a ratio of 82.2 to 16.8, applying SMOTE could be a suitable approach to balance the classes and enhance the model's performance.

The Size and shape of Data after Resampling

```
After OverSampling the shape of X: (18728, 23)
After OverSampling the shape of y: (18728,)
After OverSampling counts of label '1': 9364
After OverSampling counts of label '0': 9364
```

TABLE 2 SHAPE AFTER RESAMPLING

Train Test Split of Resampled Data

```
Number of rows and columns of the training set for the independent vari
ables: (13109, 23)
Number of rows and columns of the training set for the dependent variab
le: (13109,)
Number of rows and columns of the test set for the independent variable
s: (5619, 23)
Number of rows and columns of the test set for the dependent variable:
(5619,)
```

TABLE 3 TRAIN TEST SHAPE AFTER RESAMPLING

Modelling

The given case is having variable "Charn" as target variable which is either "0" or "1" so we apply all the models of classification to get the high accuracy in modelling as outcome.

Classifier No 1: Logistic Regression and its Tuning with Parameters

Logistic Regression is statistical method to make predictions on binary classes, the target/outcome variable of these models can only have two possible classes. In examples where the tarhet variable is of categorical nature, the model uses a log of odds as the dependent variable. Logistic regression there after compute the probability of an event occurrences. For this project we can add in the OneVsRestClassifer described in the beginning of this chapter to make our multiclass problem binary in order to fit to a logistics regression model.

Logical reason behind the selection of different values for the parameters involved in each model. That we use parameter C as our regularization parameter. Parameter C = 1/λ. Lambda (λ) controls the trade-off between allowing the model to increase it's complexity as much as it wants with trying to keep it simple. For example, if λ is very low or 0, the model will have enough power to increase it's complexity (overfit) by assigning big values to the weights for each parameter. If, in the other hand, we increase the value of λ, the model will tend to underfit, as the model will become too simple. Parameter C will work the other way around. For small values of C, we increase the regularization strength which will create simple models which underfit the data. For big values of C, we low the power of regularization which implies the model is allowed to increase it's complexity, and therefore, overfit the data.

The equation of the Logistic Regression by which we predict the corresponding probabilities and then go on predict a discrete target variable is

$$y = \frac{1}{1+e-z}$$

Note: $z = \beta 0 + \sum ni=1(\beta i X1)$

Classification Report on Train dataset LR

```
Classification Report
              precision    recall  f1-score   support

           0       0.90      0.97      0.93      6555
           1       0.76      0.47      0.58      1327

    accuracy                           0.89      7882
   macro avg       0.83      0.72      0.76      7882
weighted avg       0.88      0.89      0.87      7882
```

TABLE 4 CLASSIFICATION MATRIX LR

Confusion matrix with the percentage of Values on train data

```
Confusion Matrix
[[6358  197]
[ 703  624]]
```

TABLE 5 CONFUSION MATRIX LR

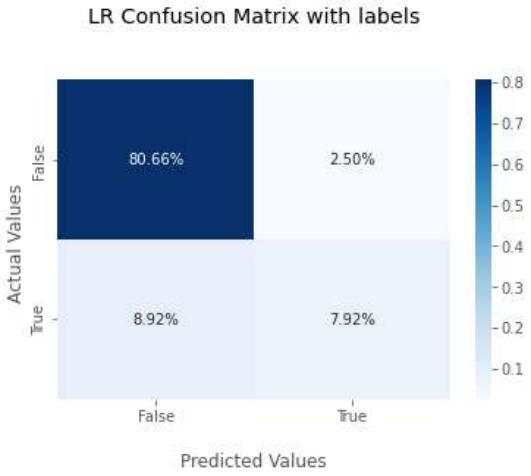

FIGURE 1 CONFUSION MATRIX PERCENTAGE

Classification Report on Test dataset LR

```
Classification Report
              precision    recall  f1-score   support

           0       0.90      0.97      0.94      2809
           1       0.77      0.48      0.59       569

    accuracy                           0.89      3378
   macro avg       0.84      0.73      0.76      3378
weighted avg       0.88      0.89      0.88      3378
```

TABLE 6 CLASSIFICATION REPORT LR TEST

Confusion matrix with the percentage of Values on Test dataset

```
Confusion Matrix
 [[2727   82]
 [ 296  273]]
```

TABLE 7 CONFUSION MATRIX LT TEST

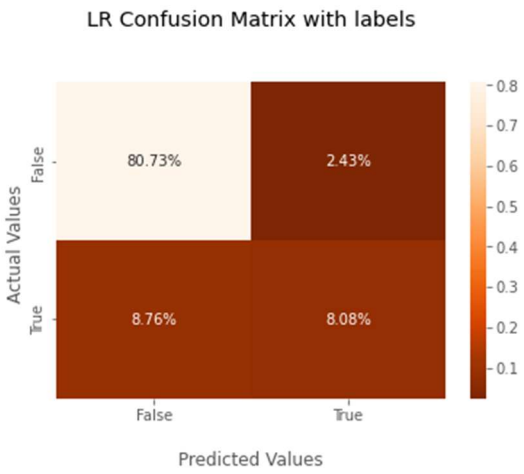

FIGURE 2 CONFUSION MATRIX PERCENTAGE

AUC Score on Train and Test Dataset

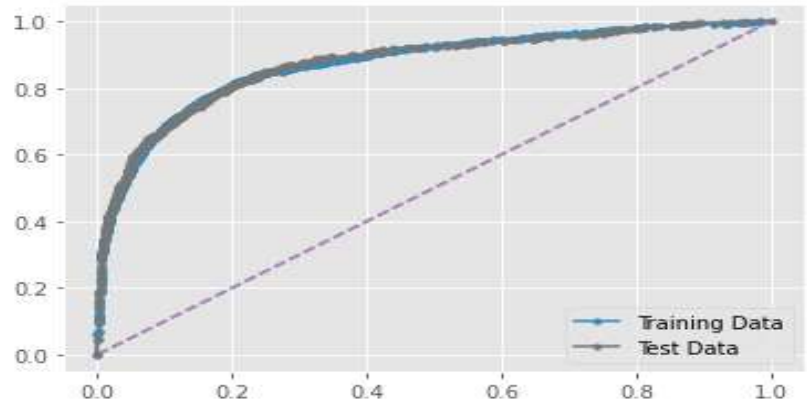

FIGURE 3 RO CURVE

```
AUC for the Training Data: 0.875
AUC for the Test Data: 0.876
```

TABLE 8 AUC VALUES FOR TRAIN AND TEST

Accuracy of Logistic Regression

```
Accuracy of Logistic regression on train set: 0.886
Accuracy of Logistic regression on test set:  0.888
```

TABLE 9 ACCURACY OF LOGISTIC TRAIN AND TEST

Fivefold K-Fold cross Validation of Logistic Regression Basic model

```
[0.8875074008288928, 0.8907637655417406, 0.8880994671403197, 0.88869153
34517466, 0.8883955002960332]
Mean of testing accuracy over 5 folds = 0.89 with std = 0.00
```

TABLE 10 FIVEFOLD K-FOLD CROSS VALIDATION

Hyperparameter Tuning of Logistic regression Model with GridSearchCV

GridSearchCV is a module of the Sklearn model selection package that is used for Hyperparameter tuning. Given a set of different hyperparameters, GridSearchCV loops through all possible values and combinations of the hyperparameter and fits the model on the training dataset. In this process, it is able to identify the best values and combination of hyperparameters (from the given set) that produces the best accuracy.

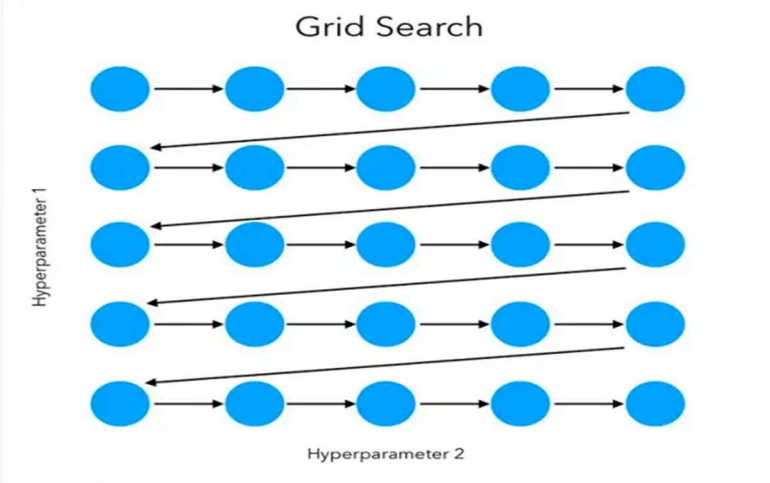

FIGURE 4 SHOWING GRID SEARCH

```
Best Hyperparameters:{'C': 0.1, 'penalty': 'l1', 'solver': 'liblinear'}
```

LogisticRegression

```
LogisticRegression(C=0.1, penalty='l1', solver='liblinear')
```

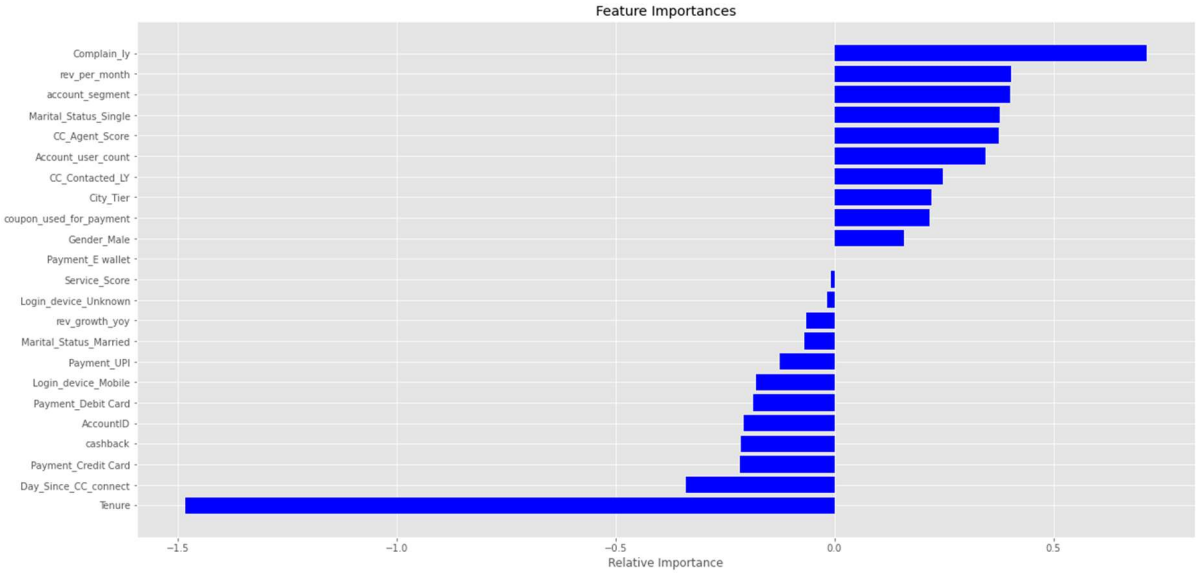

FIGURE 5 FEATURE IMPORTANCE

Classification Report on Train dataset of GridSearchCV

```
Classification Report
              precision    recall  f1-score   support

           0       0.90      0.97      0.93      7471
           1       0.78      0.46      0.58      1537

    accuracy                           0.89      9008
   macro avg       0.84      0.72      0.76      9008
weighted avg       0.88      0.89      0.87      9008
```

TABLE 11 CLASSIFICATION REPORT LRGS

Confusion matrix with the percentage on train data GridSearchCV

```
Confusion Matrix
 [[7271  200]
 [ 825  712]]
```

TABLE 12 CONFUSION MATRIX LRGS

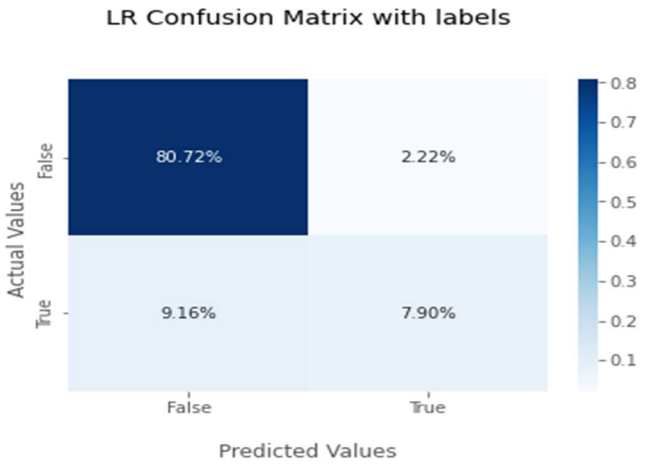

FIGURE 6 CONFUSION MATRIX PERCENTAGE

Classification Report on Test dataset of GridSearchCV

```
Classification Report
              precision    recall  f1-score   support

           0       0.90      0.97      0.94      2809
           1       0.78      0.47      0.59       569

    accuracy                           0.89      3378
   macro avg       0.84      0.72      0.76      3378
weighted avg       0.88      0.89      0.88      3378
```

TABLE 13 CLASSIFICATION REPORT LR GS

Confusion matrix with the percentage on Test data GridSearchCV

```
Confusion Matrix
 [[2732   77]
 [ 299  270]]
```

TABLE 14 CONFUSION MATRIX LRGS

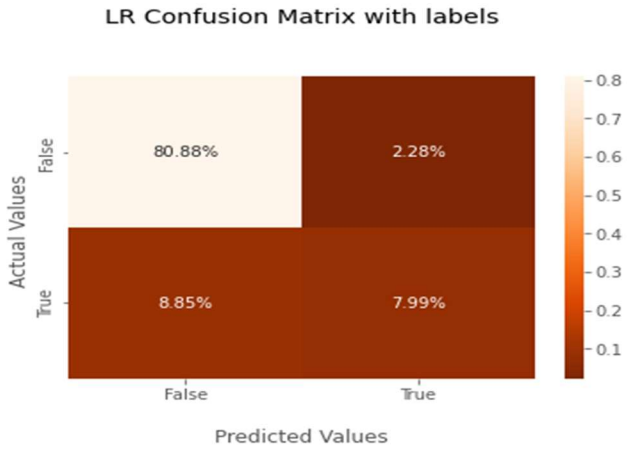

FIGURE 7 CONFUSION MATRIX PERCENTAGE

AUC Score on Train and Test Dataset on data GridSearchCV

```
AUC for the Training Data: 0.874
AUC for the Test Data:     0.877
```

TABLE 15 AUC SCORE LRGS

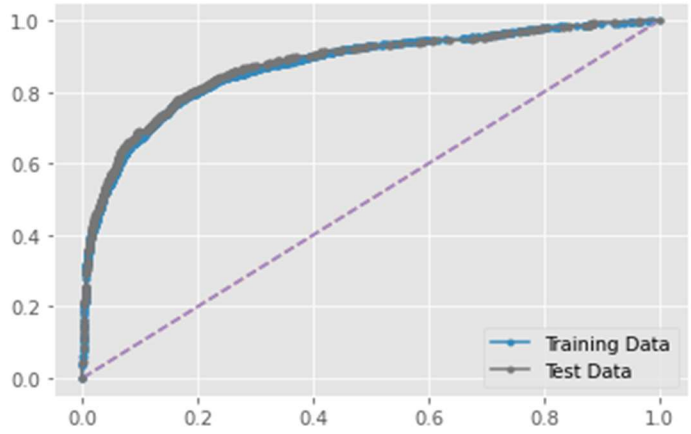

FIGURE 8 RO CURVE

Fivefold K-Fold cross Validation of Logistic Regression GridSearchCV

```
[0.8878034339846063, 0.8907637655417406, 0.8878034339846063, 0.88869153
34517466, 0.886915334517466]
Mean of testing accuracy over 5 folds = 0.89 with std = 0.00
```

TABLE 16 FIVEFOLD K-FOLD CROSS VALIDATION

Accuracy of Logistic Regression GridSearchCV

```
Accuracy of Tuned Logistic regression on train set: 0.884
Accuracy of Tuned Logistic regression on test set:  0.887
```

TABLE 17 ACCURACY OF LOGISTIC REGRESSION GRIDSEARCHCV

Building Logistic Regression model using SMOTE

LogisticRegression

```
LogisticRegression(random_state=1)
```

Classification Report on Train dataset SMOTE

```
Classification Report
              precision    recall  f1-score   support

           0       0.82      0.79      0.80      6554
           1       0.79      0.82      0.81      6555

    accuracy                           0.80     13109
   macro avg       0.81      0.80      0.80     13109
weighted avg       0.81      0.80      0.80     13109
```

TABLE 18 CLASSIFICATION REPORT

Confusion matrix with the percentage on train SMOTE

```
Confusion Matrix
 [[5149 1405]
 [1154 5401]]
```

TABLE 19 CONFUSION MATRIX

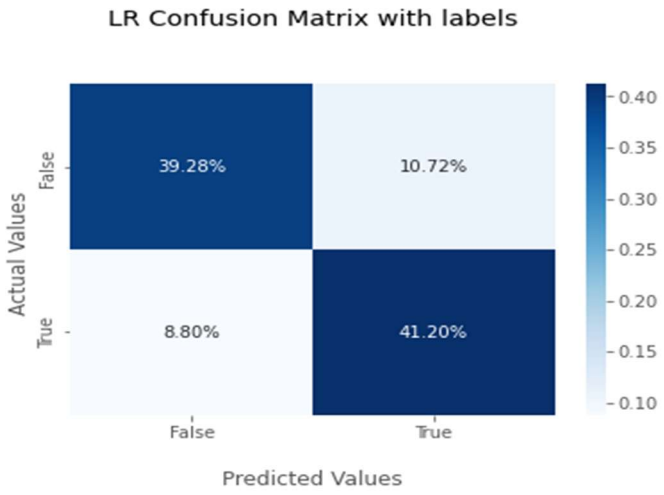

FIGURE 9 CONFUSION MATRIX PERCENTAGE

Classification Report on Test dataset SMOTE

	precision	recall	f1-score	support
0	0.81	0.77	0.79	2810
1	0.78	0.82	0.80	2809
accuracy			0.79	5619
macro avg	0.79	0.79	0.79	5619
weighted avg	0.79	0.79	0.79	5619

TABLE 20 CLASSIFICATION REPORT SMOTE

Confusion matrix with the percentage on test SMOTE

```
Confusion Matrix
 [[2162  648]
 [ 509 2300]]
```

TABLE 21CONFUSION MATRIX LR SMOTE

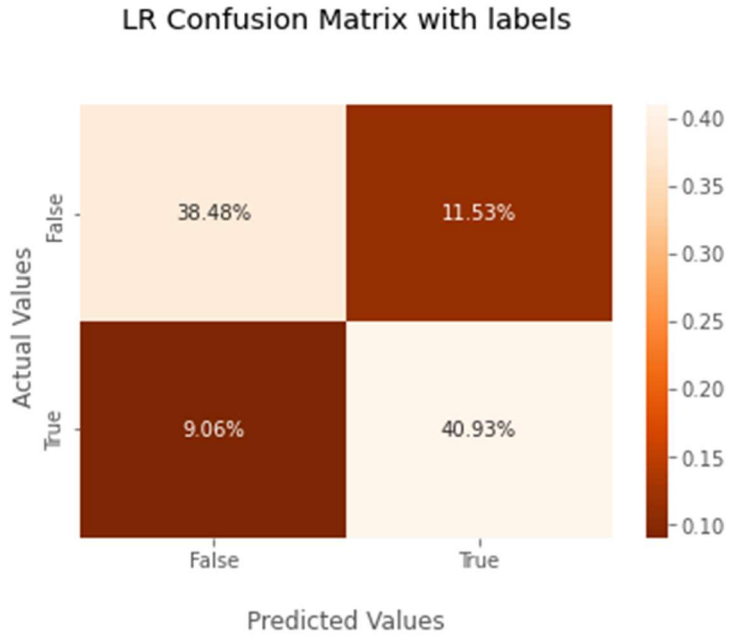

FIGURE 10 CONFUSION MATRIX PERCENTAGE

Fivefold K-Fold cross Validation of Logistic Regression SMOTE

```
[0.7191671115856914, 0.7184552411461114, 0.7189891439757964, 0
.7166755650471615, 0.7175654030966364]
Mean of testing accuracy over 5 folds = 0.72 with std = 0.00
```

TABLE 22 FIVEFOLD K-FOLD CROSS VALIDATION

Accuracy of Logistic Regression SMOTE

```
Accuracy of resampled Logistic regression on train set: 0.886
Accuracy of Resampled Logistic regression on test set:  0.718
```

TABLE 23ACCURACY OF LOGISTIC REGRESSION SMOTE

Insights:

As we performing the logistic regression Modelling on SMOTE data the accuracy level gone down and the case of overfitting can be identified. It is on the basis of difference in more than 10% between train and test accuracy.

Classifier No 2 : Linear Discriminant Analysis and its Tuning with Parameters

Linear Discriminant Analysis (LDA) is a supervised learning algorithm used for classification tasks in machine learning. It is a technique used to find a linear combination of features that best separates the classes in a dataset.

In pattern recognition and machine learning, linear discriminant analysis (LDA), also called canonical Variate Analysis (CVA), is a way to study differences between objects. This sorting method uses a linear combination of features to characterize classes.In other words, it finds the directions in the feature space that best separate the different classes of data.

LDA is similar to logistic regression and probit regression, and also, to some degree, analysis of variance (ANOVA). Although it has the term “linear” in the title, it can be expanded to the analysis of non-linear systems, using nonlinear spline basis functions.

LDA works by projecting the data onto a lower-dimensional space that maximizes the separation between the classes. It does this by finding a set of linear discriminants that maximize the ratio of between-class variance to within-class variance.

LDA assumes that the data has a Gaussian distribution and that the covariance matrices of the different classes are equal. It also assumes that the data is linearly separable, meaning that a linear decision boundary can accurately classify the different classes.

The model uses Bayes Theorem to estimate the probabilities. Briefly Bayes’ Theorem can be used to estimate the probability of the output class (k) given the input (x) using the probability of each class and the probability of the data belonging to each class:

```
P(Y=x|X=x) = (PIk * fk(x)) / sum(PIl * fl(x))
```

Where PIk refers to the base probability of each class (k) observed in your training data (e.g. 0.5 for a 50-50 split in a two class problem). In Bayes’ Theorem this is called the prior probability.

```
PIk = nk/n
```

The f(x) above is the estimated probability of x belonging to the class. A Gaussian distribution function is used for f(x). Plugging the Gaussian into the above equation and simplifying we end up with the equation below. This is called a discriminate function and the class is calculated as having the largest value will be the output classification (y):

```
Dk(x) = x * (muk/siga^2) – (muk^2/(2*sigma^2)) + ln(PIk)
```

Dk(x) is the discriminate function for class k given input x, the muk, sigma^2 and PIk are all estimated from data.

Building linear Discriminant Analysis model

Basic LDA Model Classification Report on Train Dataset

```
classification Report
              precision    recall  f1-score   support

           0       0.89      0.97      0.93      7471
           1       0.76      0.42      0.54      1537

    accuracy                           0.88      9008
   macro avg       0.83      0.70      0.74      9008
weighted avg       0.87      0.88      0.86      9008
```

TABLE 24 CLASSIFICATION REPORT LDA TRAIN

Basic LDA Model Confusion matrix with the percentage on train

```
Confusion Matrix
 [[7268  203]
 [ 889  648]]
```

TABLE 25 CONFUSION MATRIX LDA

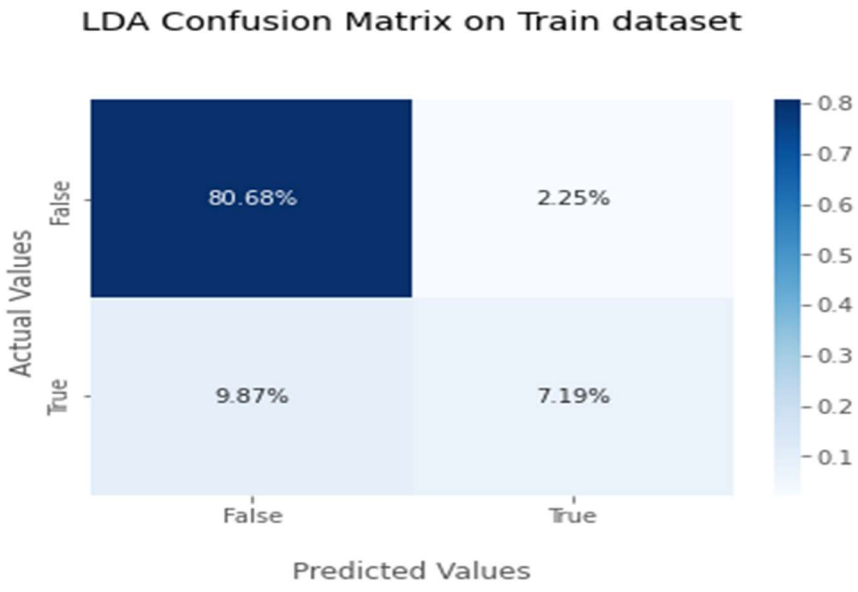

FIGURE 11 CONFUSION MATRIX PERCENTAGE

Basic LDA Model Classification Report on Test dataset

```
classification Report
precision    recall  f1-score   support

           0       0.90      0.97      0.93      2809
           1       0.77      0.44      0.56       569

    accuracy                           0.88      3378
   macro avg       0.83      0.71      0.75      3378
weighted avg       0.87      0.88      0.87      3378
```

TABLE 26 CLASSIFICATION REPORT LDA

Basic LDA Model Confusion matrix with the percentage on test

```
Confusion Matrix
 [[2734   75]
 [ 319  250]]
```

TABLE 27 CLASSIFICATION REPORT LDA TEST

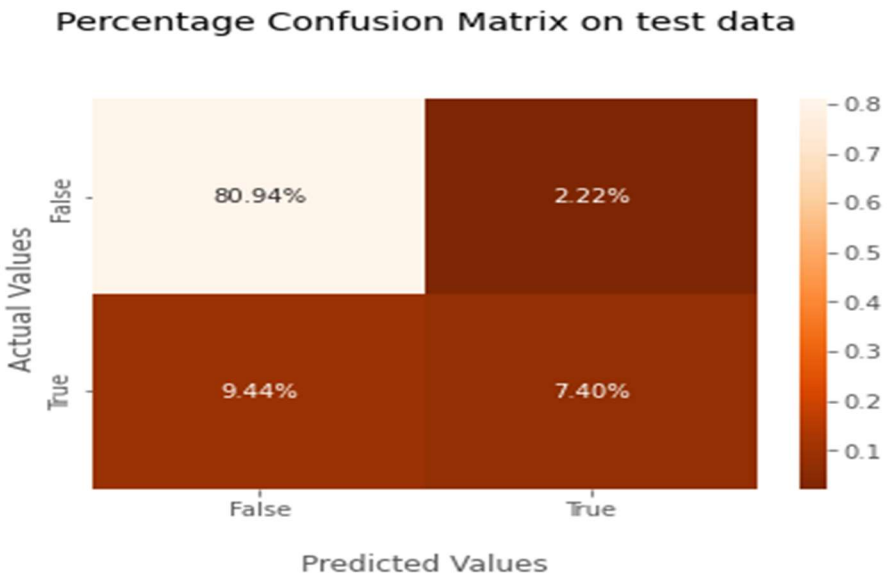

FIGURE 12 CONFUSION MATRIX PERCENTAGE

AUC Score and curve on Train and Test DataSet

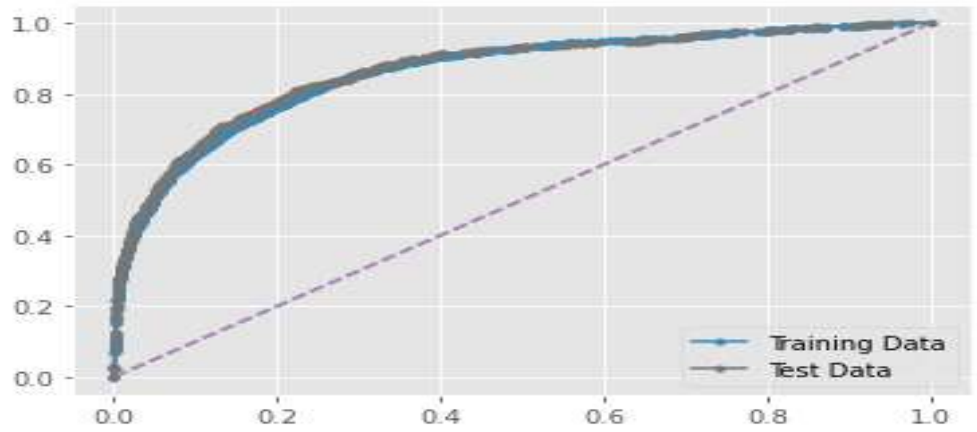

FIGURE 13 RO CURVE

```
AUC for the Training Data: 0.864
AUC for the Test Data:     0.869
```

TABLE 28 AUC SCORES LDA

BASIC LDA MODEL FIVEFOLD K-FOLD CROSS VALIDATION ON TEST DATASET

```
[0.8833629366489046, 0.8833629366489046, 0.8833629366489046, 0.88336293
66489046, 0.8833629366489046]
Mean of testing accuracy over 5 folds = 0.88 with std = 0.00
```

TABLE 29 FIVEFOLD K-FOLD CROSS VALIDATION LDA

LDA basic Model Accuracy on train and Test dataset

```
Accuracy of Linear discriminant Analysis on train set: 0.879
Accuracy of Linear discriminant Analysis on test set:  0.883
```

TABLE 30 ACCURACY SCORES LDA

Building LDA Model on GridSearchCV

```
Best _gride is {'shrinkage': 'auto', 'solver': 'lsqr', 'tol': 0.0001}
```

TABLE 31 BEST GRID LDA

LDA Model Classification Report on Train Dataset on GridSearchCV

```
Classification Report
              precision    recall  f1-score   support

           0       0.89      0.97      0.93      7471
           1       0.76      0.42      0.54      1537

    accuracy                           0.88      9008
   macro avg       0.83      0.70      0.74      9008
weighted avg       0.87      0.88      0.86      9008
```

TABLE 32 CLASSIFICATION REPORT LDA GS

LDA Confusion matrix with the percentage on train on GridSearchCV

```
Confusion Matrix
 [[7270  201]
 [ 890  647]]
```

TABLE 33 CONFUSION MATRIX LDA GS

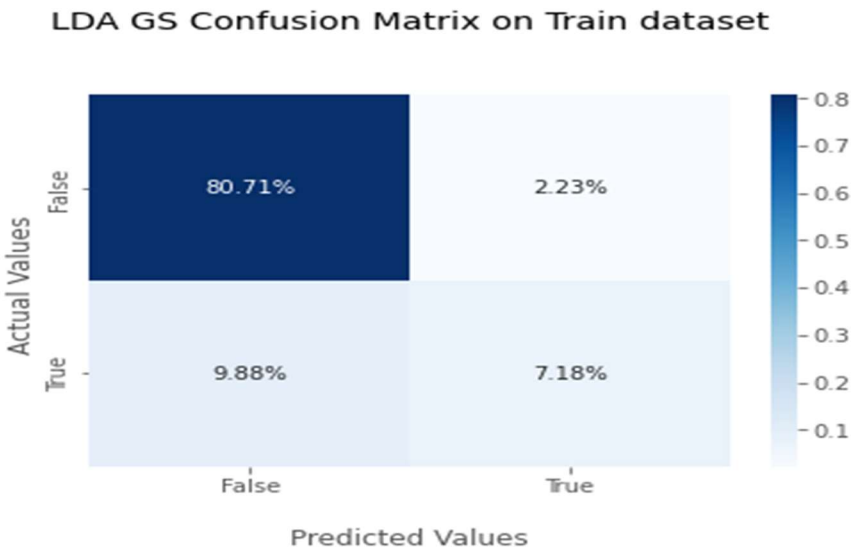

FIGURE 14 CONFUSION MATRIX PERCENTAGE

LDA Model Classification Report on Test dataset on GridSearchCV

```
Classification Report
  precision    recall  f1-score   support

           0       0.90      0.97      0.93      2809
           1       0.77      0.44      0.56       569

    accuracy                           0.88      3378
   macro avg       0.83      0.71      0.75      3378
weighted avg       0.87      0.88      0.87      3378
```

TABLE 34 CLASSIFICATION REPORT LDA GS

LDA Confusion matrix with the percentage on test on GridSearchCV

```
Confusion Matrix
 [[2735   74]
 [ 319  250]]
```

TABLE 35 CONFUSION MATRIX LDA GS

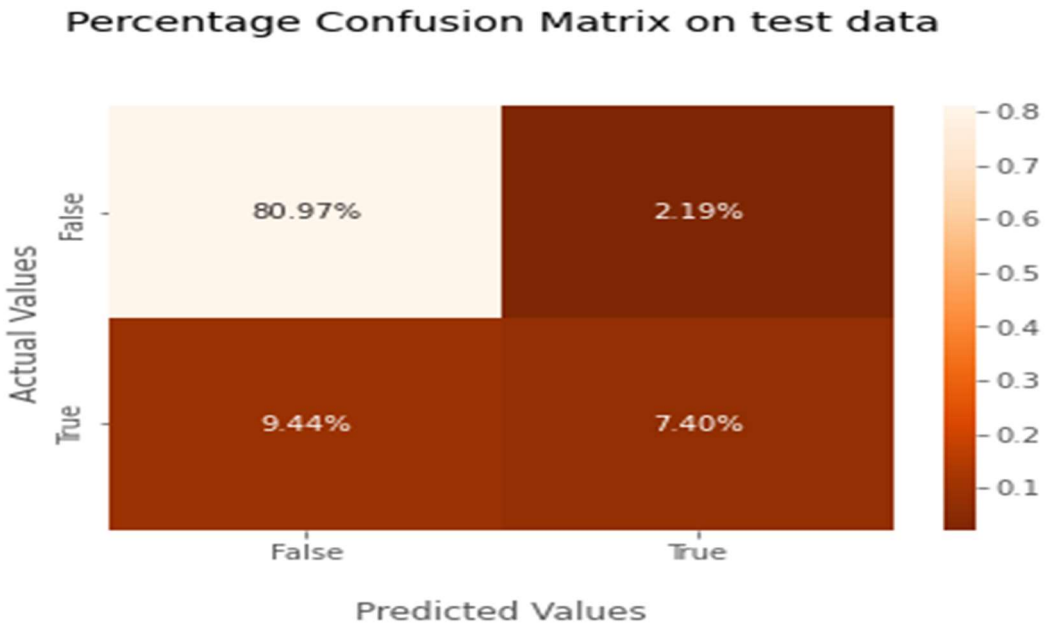

FIGURE 15 CONFUSION MATRIX PERCENTAGE

LDA AUC Score and curve on Train and Test DataSet on GridSearchCV

```
AUC for the Training Data: 0.864
AUC for the Test Data:     0.869
```

TABLE 36 AUC SCORE LDA GS

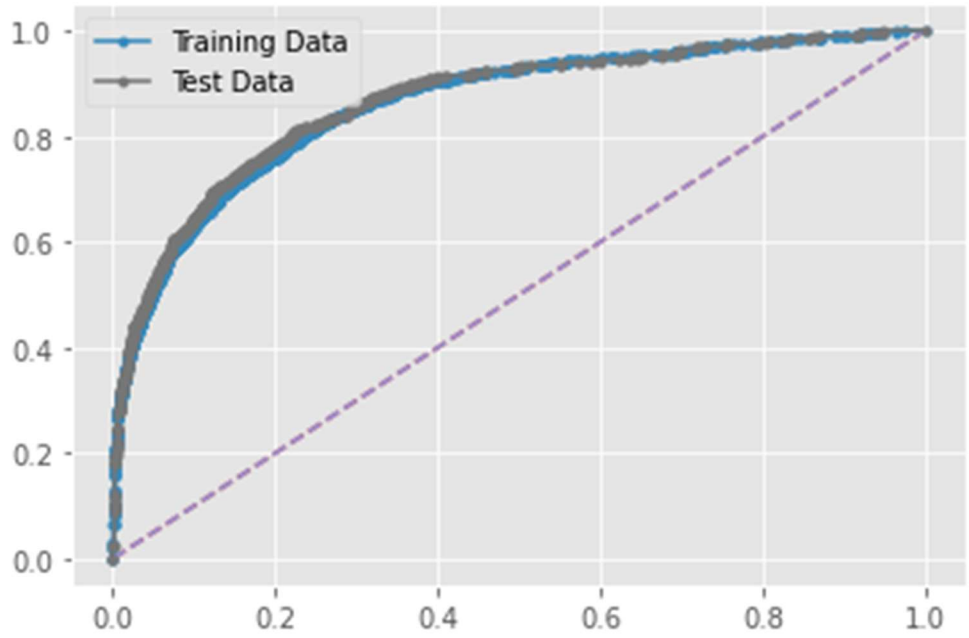

FIGURE 16 RO CURVE

LDA Fivefold K-Fold cross Validation on test dataset on GridSearchCV

```
[0.8836589698046181, 0.8836589698046181, 0.8836589698046181, 0.8836589698046181, 0.8836589698046181]
Mean of testing accuracy over 5 folds = 0.88 with std = 0.00
```

TABLE 37 FIVEFOLD K-FOLD CROSS VALIDATION

LDA basic Model Accuracy on train and Test dataset on GridSearchCV

```
Accuracy of GridSearchCV LDA  on train set: 0.879
Accuracy of GridSearchCV LDA on test set:   0.884
```

TABLE 38 ACCURACY SCORE LDA GS

Building LDA model on SMOTE

Classification Report on SMOTE Train Dataset

```
Classification Report
              precision    recall  f1-score   support

           0       0.89      0.97      0.93      7471
           1       0.76      0.42      0.54      1537

    accuracy                           0.88      9008
   macro avg       0.83      0.70      0.74      9008
weighted avg       0.87      0.88      0.86      9008
```

TABLE 39CLASSIFICATION REPORT LDA SMOTE

Confusion matrix with the percentage on SMOTE train

```
Confusion Matrix
 [[7268  203]
 [ 889  648]]
```

TABLE 40 CONFUSION MATRIX LDA SMOTE

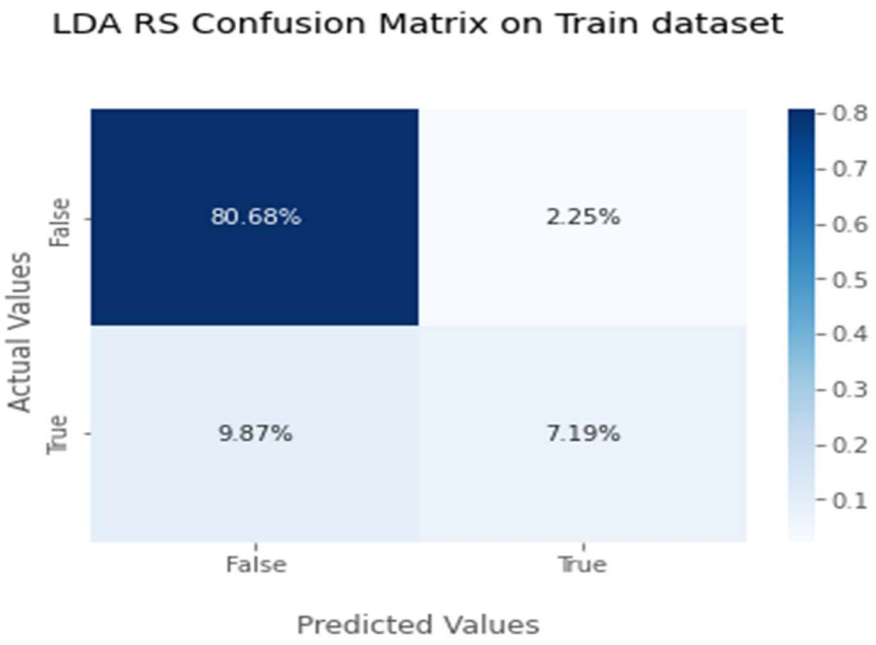

FIGURE 17 CONFUSION MATRIX PERCENTAGE

Classification Report on SMOTE Test dataset

```
Classification Report
              precision    recall  f1-score   support

           0       0.63      0.97      0.76      2810
           1       0.94      0.42      0.58      2809

    accuracy                           0.70      5619
   macro avg       0.78      0.70      0.67      5619
weighted avg       0.78      0.70      0.67      5619
```

TABLE 41 CLASSIFICATION REPORT ON SMOTE

Confusion matrix with the percentage on SMOTE test

```
Confusion Matrix
 [[2738   72]
 [1635 1174]]
```

TABLE 42 CONFUSION MATRIX LDA SMOTE

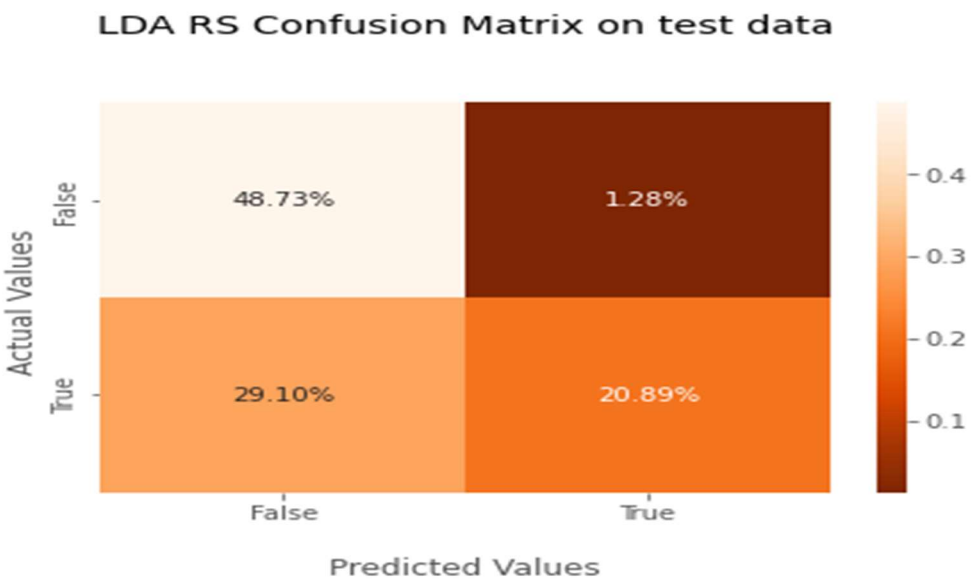

FIGURE 18 CONFUSION MATRIX PERCENTAGE

LDA Fivefold K-Fold cross Validation of SMOTE

```
[0.6965652251290265, 0.6974550631785015, 0.6956753870795516, 0.69389571
09806015, 0.6962092899092365]
Mean of testing accuracy over 5 folds = 0.70 with std = 0.00
```

TABLE 43 FIVEFOLD K-FOLD CROSS VALIDATION OF SMOTE LDA

LDA Accuracy on SMOTE

```
Accuracy of resampled Linear discriminant Analysis on train set: 0.879
Accuracy of Resampled Linear discriminant Analysis on test set:  0.696
```

TABLE 44 ACCURACY ON LDA SMOTE

Insight: The models of LDA are not performing well with the given dataset also, not good at classify the variable to target variable towards high accuracy level.

Classifier No 3 CART and its Tuning with Parameters

Decision tree (DT) is a non-parametric supervised learning method commonly used for classification and regression. The goal is to create a model that predicts the value of a target variable based on several input variables. It is a simple representation for classifying examples. Decision trees learn from data to approximate a sine curve with a set of if-then-else decision rules. The deeper the tree, the more complex the decision rules and the fitter the model. We will see this in an example below where we define a max depth for the decision tree.

Advantages of decision trees:

Require little data preparation: does not need data normalization, dummy variable creation and removing blank values. Handles both numeric and categorical data.

Disadvantages of decision trees:

Prone to overfitting: creating over-complex trees that do not generalise the data well. Decision trees can be unstable because small variations in the data might result in a different tree.

CART Regularized Model Feature Importance

	Imp
Tenure	0.50
Complain_ly	0.11
CC_Agent_Score	0.06
City_Tier	0.05
Day_Since_CC_connect	0.04
account_segment	0.04
rev_per_month	0.04
Marital_Status_Single	0.04
rev_growth_yoy	0.03
CC_Contacted_LY	0.03
Login_device_Mobile	0.02
Account_user_count	0.01
cashback	0.01
Marital_Status_Married	0.01
Payment_Credit Card	0.01
Payment_E wallet	0.00
AccountID	0.00
Gender_Male	0.00
Payment_Debit Card	0.00
coupon_used_for_payment	0.00
Login_device_Unknown	0.00
Service_Score	0.00
Payment_UPI	0.00

TABLE 45 MODEL FEATURE IMPORTANCE CART

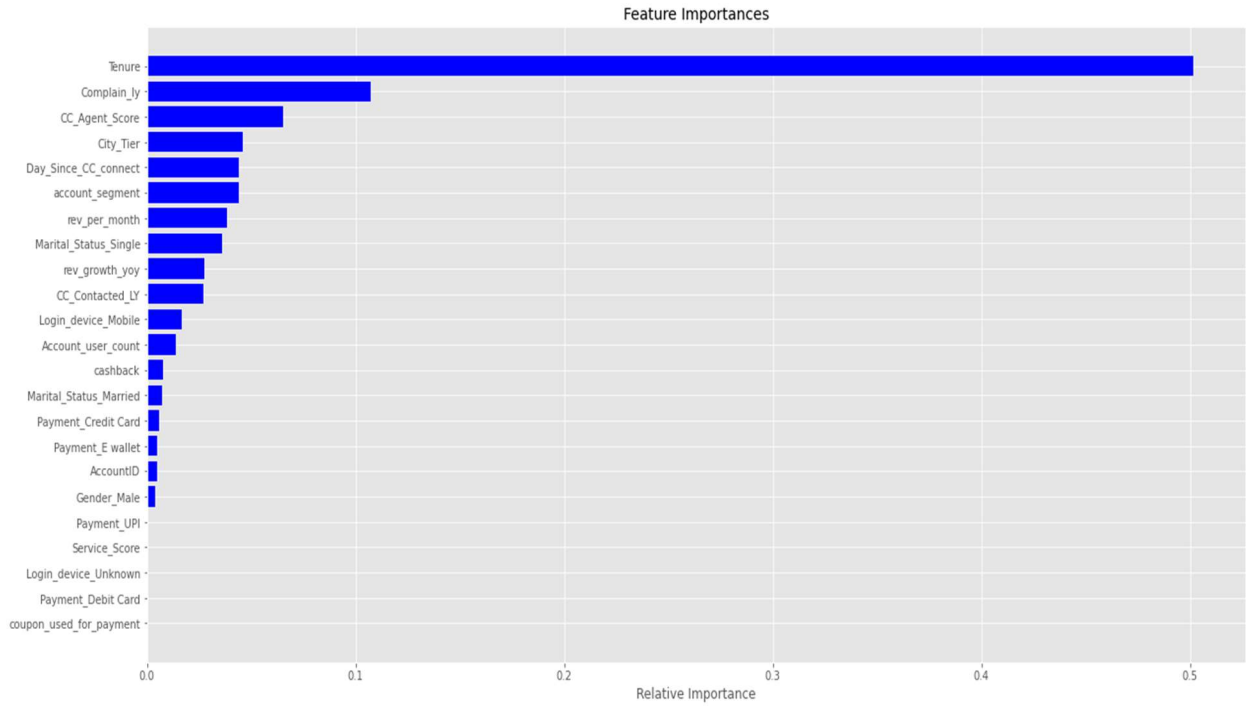

FIGURE 19 FEATURE IMPORTANCE

Regularising the Decision Tree

DecisionTreeClassifier

```
DecisionTreeClassifier(max_depth=7, min_samples_leaf=10, min_samples_s
plit=30)
```

TABLE 46 DECISION TREE PARAMETERS

CART Regularized Model Classification Report on Train Dataset

```
Classification Report
              precision    recall  f1-score   support

           0       0.92      0.97      0.94      7471
           1       0.78      0.62      0.69      1537

    accuracy                           0.91      9008
   macro avg       0.85      0.79      0.82      9008
weighted avg       0.90      0.91      0.90      9008
```

TABLE 47 CLASSIFICATION REPORT DECISION TREE

CART Regularized Model Confusion matrix with the percentage on train

```
Confusion Matrix
 [[7211  260]
 [ 589  948]]
```

TABLE 48 CONFUSION MATRIX DTR

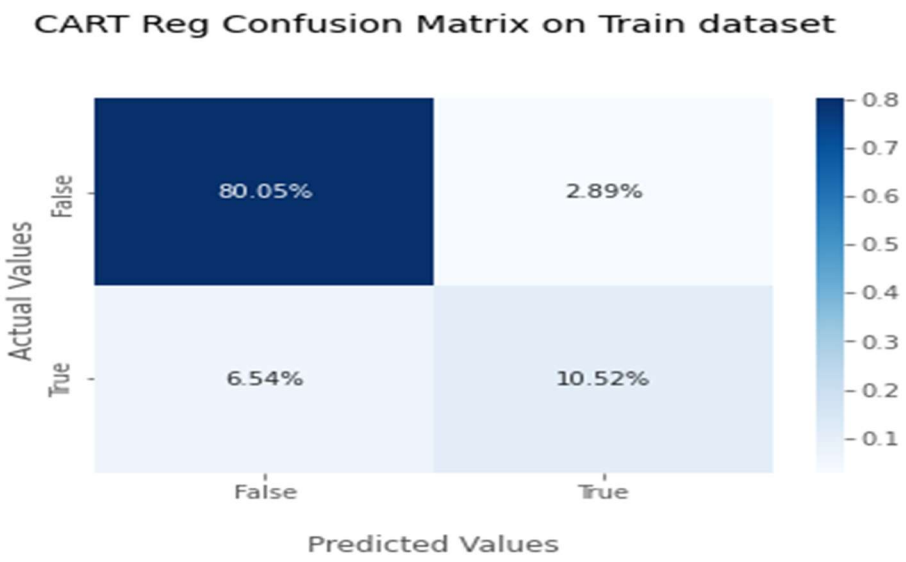

FIGURE 20 CONFUSION MATRIX PERCENTAGE

CART Regularized Model Classification Report on Test dataset

```
Classification Report
precision    recall  f1-score   support

           0       0.93      0.96      0.95      2809
           1       0.78      0.64      0.70       569

    accuracy                           0.91      3378
   macro avg       0.86      0.80      0.83      3378
weighted avg       0.90      0.91      0.91      3378
```

TABLE 49 CLASSIFICATION REPORT DTR

CART Regularized Model Confusion matrix with the percentage on test

```
Confusion Matrix
 [[2708  101]
 [ 205  364]]
```

TABLE 50 CONFUSION MATRIX DTR

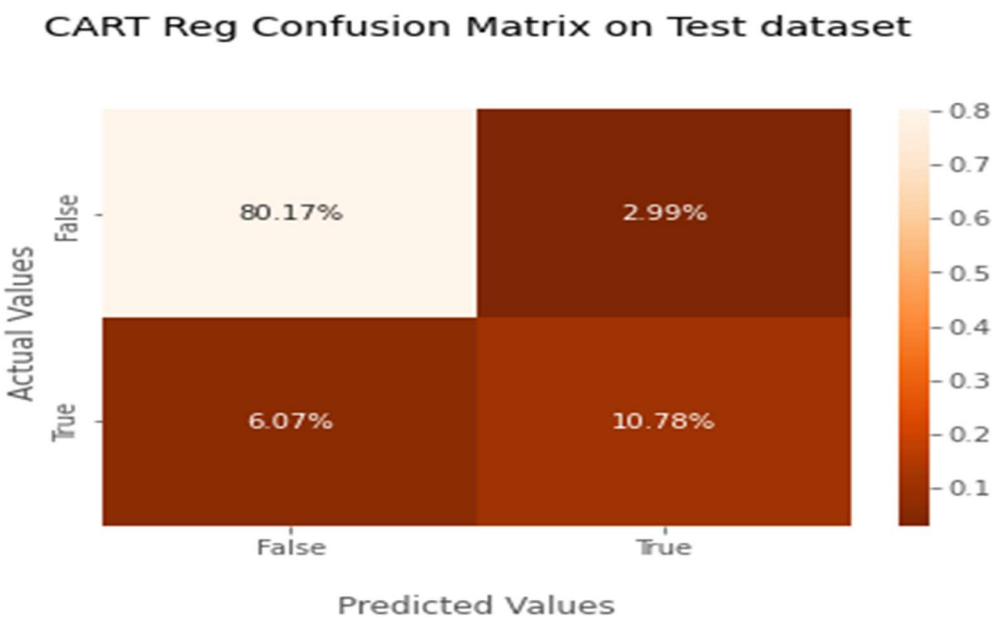

FIGURE 21 CONFUSION MATRIX PERCENTAGE

CART Regularized Model AUC Score and curve on Train and Test DataSet

```
AUC for the Training Data: 0.929
AUC for the Test Data:     0.933
```

TABLE 51 AUC SCORE DTR

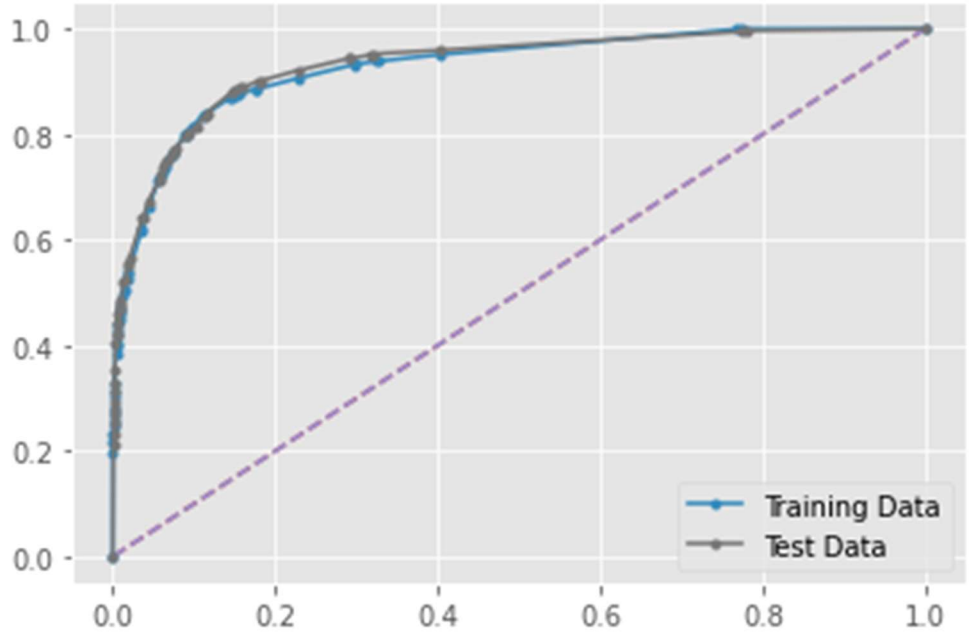

FIGURE 22 RO CURVE

Fivefold K-Fold cross Validation of CART Regularized Model

```
[0.9094138543516874, 0.9094138543516874, 0.9094138543516874, 0.90941385
43516874, 0.9094138543516874]
Mean of testing accuracy over 5 folds = 0.91 with std = 0.00
```

TABLE 52 FIVEFOLD K-FOLD CROSS VALIDATION OF CART

CART Regularized Model Accuracy on Train and Test dataset

```
Accuracy of CART Regularised on train set: 0.906
Accuracy of CART Regularised on test set:  0.909
```

TABLE 53 ACCURACY ON TRAIN AND TEST DATASET DTR

Building CART GridSearchCV model

DecisionTreeClassifier

```
DecisionTreeClassifier(max_depth=20, min_samples_leaf=3, min_samples_s
plit=15,
                       random_state=1)
```

TABLE 54 BEST GRID OF DT GS

CART GridSearchCV model Feature Importance

	Imp
Tenure	0.35
Complain_ly	0.07
CC_Agent_Score	0.07
Day_Since_CC_connect	0.06
CC_Contacted_LY	0.06
rev_growth_yoy	0.05
cashback	0.04
account_segment	0.04
City_Tier	0.04
rev_per_month	0.04
Marital_Status_Single	0.03
Account_user_count	0.03
Payment_Credit Card	0.02
AccountID	0.02
Login_device_Mobile	0.02
Gender_Male	0.02
Marital_Status_Married	0.01
coupon_used_for_payment	0.01
Payment_E wallet	0.01
Payment_Debit Card	0.01
Login_device_Unknown	0.00
Service_Score	0.00
Payment_UPI	0.00

TABLE 55 CART GRIDSEARCHCV MODEL FEATURE IMPORTANCE

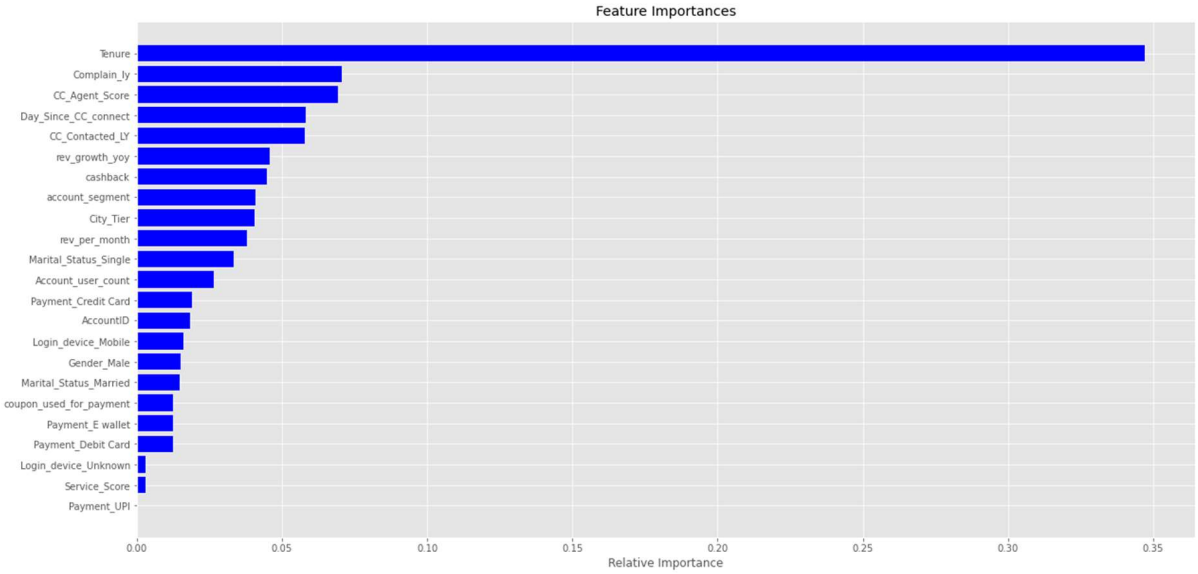

FIGURE 23 FEATURE IMPORTANCE

CART GridSearchCV model Classification Report on Train Dataset

```
Classification Report
              precision    recall  f1-score   support

           0       0.97      0.99      0.98      7471
           1       0.93      0.88      0.90      1537

    accuracy                           0.97      9008
   macro avg       0.95      0.93      0.94      9008
weighted avg       0.97      0.97      0.97      9008
```

TABLE 56 CLASSIFICATION REPORT DT GS

CART GridSearchCV model Confusion matrix with the percentage on train

```
Confusion Matrix
 [[7362  109]
 [ 191 1346]]
```

TABLE 57 CONFUSION MATRIX DT GS

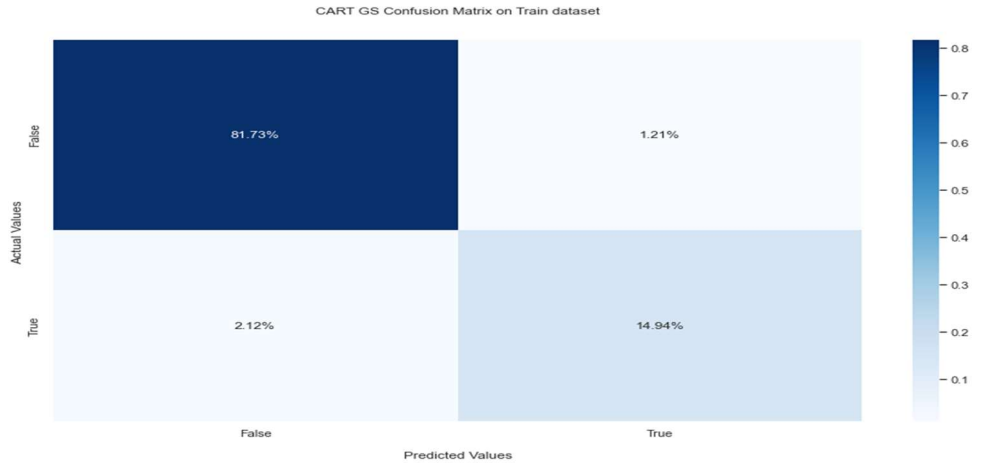

FIGURE 24 CONFUSION MATRIX PERCENTAGE

CART GridSearchCV model Classification Report on Test dataset

```
Classification Report
                precision    recall  f1-score   support

           0       0.97      0.98      0.98      2809
           1       0.90      0.86      0.88       569

    accuracy                           0.96      3378
   macro avg       0.94      0.92      0.93      3378
weighted avg       0.96      0.96      0.96      3378
```

TABLE 58 CLASSIFICATION REPORT DTGS

CART GridSearchCV model Confusion matrix with the percentage on test

```
Confusion Matrix
 [[2753   56]
 [  78  491]]
```

TABLE 59 CONFUSION MATRIX DTGS

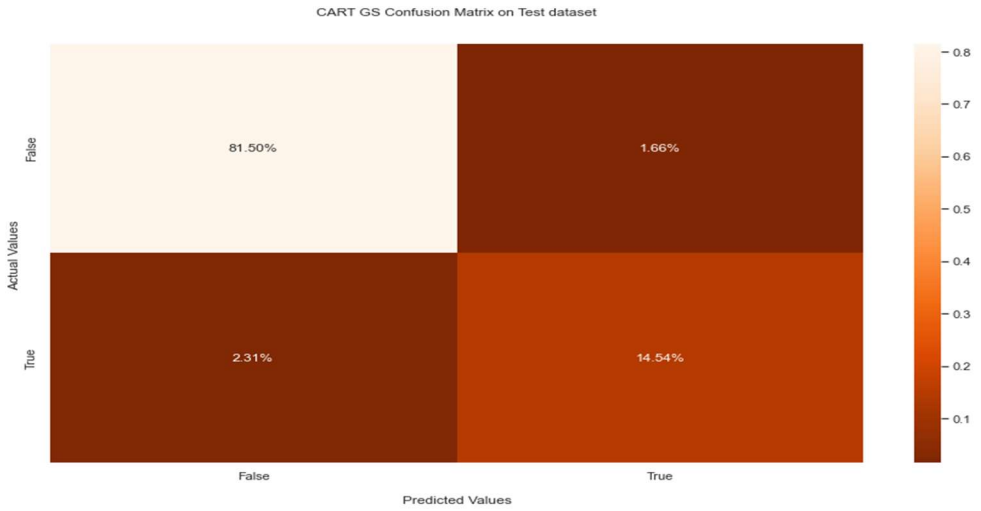

FIGURE 25 CONFUSION MATRIX PERCENTAGE

CART GridSearchCV model AUC Score and curve on Train and Test DataSet

```
AUC for the Training Data: 0.995
AUC for the Test Data:     0.984
```

TABLE 60 GRIDSEARCHCV MODEL AUC SCORE

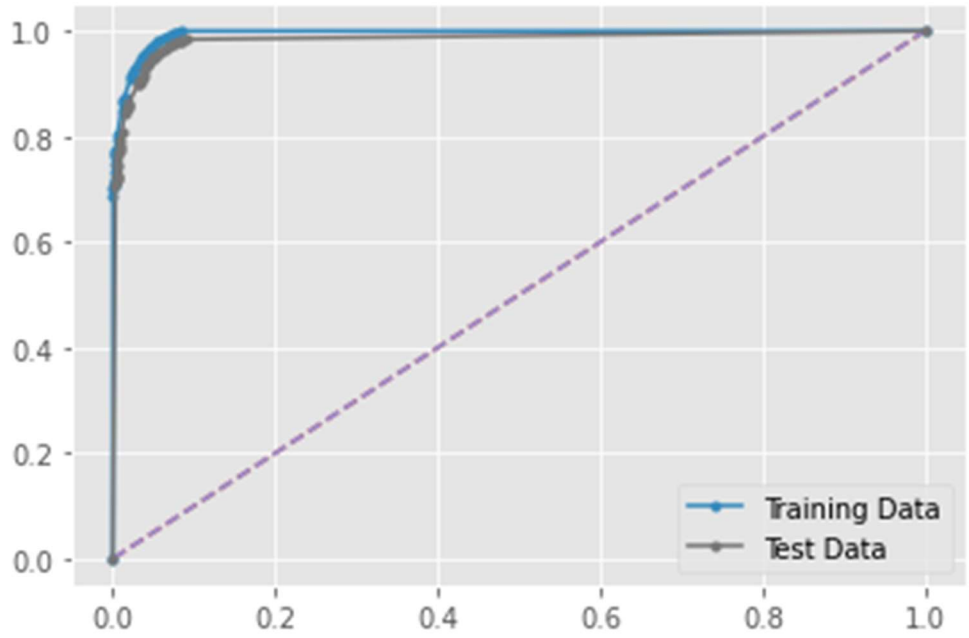

FIGURE 26 RO CURVE

Fivefold K-Fold cross Validation of CART GridSearchCV model

```
[0.9603315571343991, 0.9603315571343991, 0.9603315571343991, 0.96033155
71343991, 0.9603315571343991]
Mean of testing accuracy over 5 folds = 0.96 with std = 0.00
```

TABLE 61 FIVEFOLD K-FOLD CROSS VALIDATION OF CART GRIDSEARCHCV

CART GridSearchCV model Accuracy on train and test dataset

```
Accuracy of GridSearchCV CART on train set: 0.967
Accuracy of GridSearchCV CART on test set:  0.960
```

TABLE 62 CART GRIDSEARCHCV MODEL ACCURACY

CART model Building on SMOTE

DecisionTreeClassifier

```
DecisionTreeClassifier(random_state=1)
```

TABLE 63 DECISION TREE CLASSIFICATION ON SMOTE

CART SMOTE model Feature Importance

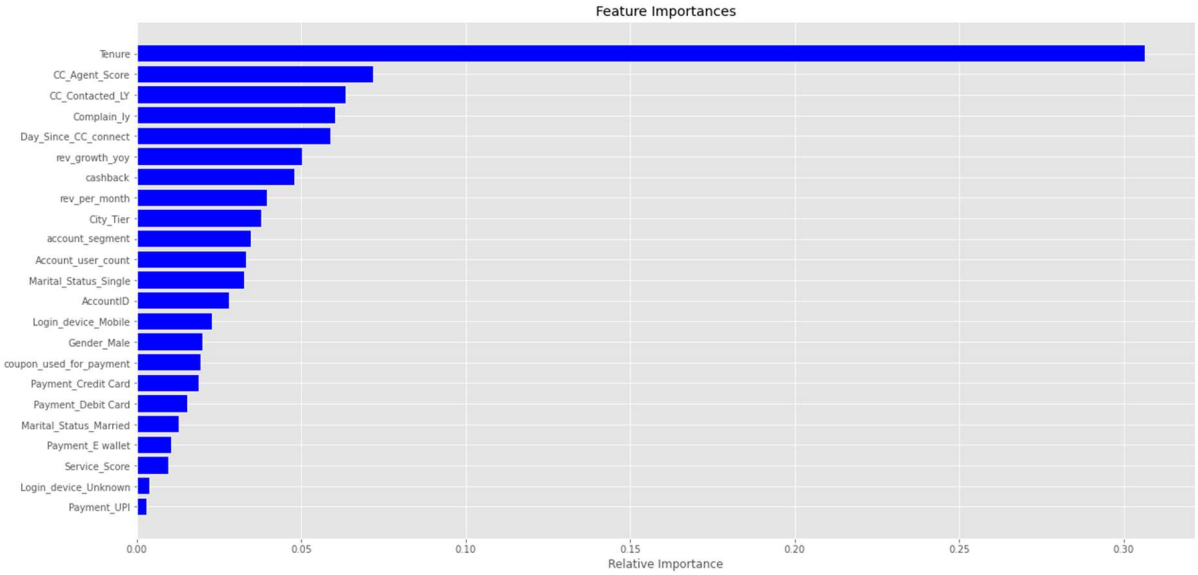

FIGURE 27 FEATURE IMPORTANCE

CART model Classification Report on SMOTE Train Dataset

```
Classification Report
              precision    recall  f1-score   support

           0       1.00      1.00      1.00      7471
           1       1.00      1.00      1.00      1537

    accuracy                           1.00      9008
   macro avg       1.00      1.00      1.00      9008
weighted avg       1.00      1.00      1.00      9008
```

TABLE 64 CLASSIFICATION REPORT ON DT SMOTE

CART model Confusion matrix with the percentage onSMOTE train

```
Confusion Matrix
 [[7471    0]
 [   0 1537]]
```

TABLE 65 CONFUSION MATRIX DT SMOTE

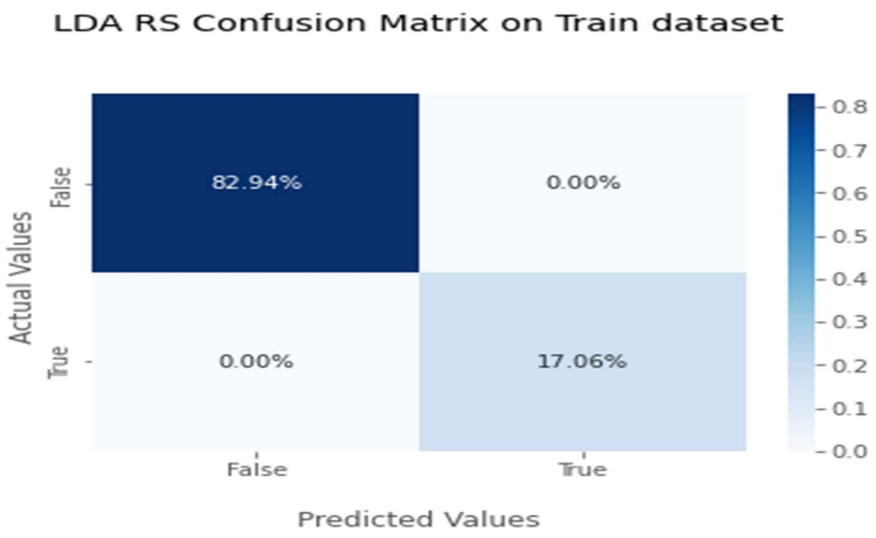

FIGURE 28 CONFUSION MATRIX PERCENTAGE

CART model Classification Report on SMOTE Test dataset

```
Classification Report
              precision    recall  f1-score   support

           0       0.93      0.99      0.96      2810
           1       0.99      0.92      0.96      2809

    accuracy                           0.96      5619
   macro avg       0.96      0.96      0.96      5619
weighted avg       0.96      0.96      0.96      5619
```

TABLE 66 CART MODEL CLASSIFICATION REPORT ON SMOTE TEST DATASET

CART model Confusion matrix with the percentage on SMOTE test

```
Confusion Matrix
 [[2792   18]
 [ 212 2597]]
```

TABLE 67 CART MODEL CONFUSION MATRIX WITH THE PERCENTAGE ON SMOTE TEST

TABLE 68 CART MODEL CONFUSION MATRIX SMOTE

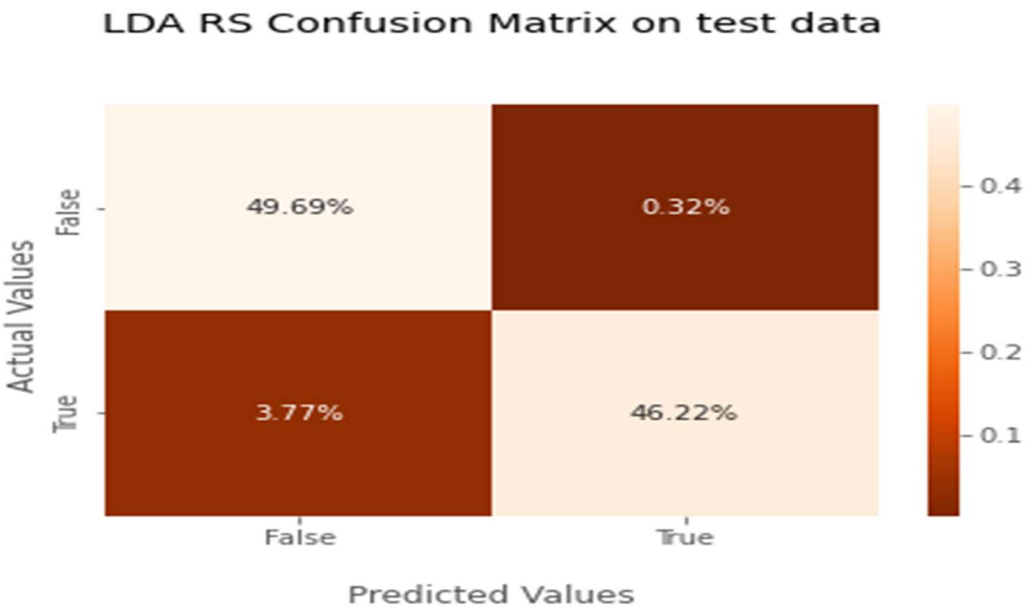

FIGURE 29 CONFUSION MATRIX PERCENTAGE

Fivefold K-Fold cross Validation on SMOTE

```
[0.9588894821142552, 0.9510589072788752, 0.9533724862075103, 0.95675387
07955152, 0.9590674497241503]
Mean of testing accuracy over 5 folds = 0.96 with std = 0.00
```

TABLE 69 FIVEFOLD K-FOLD CROSS VALIDATION ON SMOTE

CART model Accuracy on train and test dataset

```
Accuracy of resampled Linear discriminant Analysis on train set: 1.000
Accuracy of Resampled Linear discriminant Analysis on test set:  0.959
```

TABLE 70 CART MODEL ACCURACY SMOTE

Business Insights of Best Model of CART Resampled :

Based on the provided confusion matrix and classification report:

1.The model shows high accuracy and performs well in classifying both classes (0 and 1).
2.The model has high precision for both classes, indicating a low false positive rate.
3.The recall for class 0 is high (0.99), indicating that the model correctly identifies a large majority of instances of class 0. The recall for class 1 is also good (0.93), indicating a reasonably good ability to identify positive instances.
4.The F1-scores for both classes are high, indicating a good balance between precision and recall.
5.The number of false negatives (194) is relatively higher than the false positives (18), suggesting that the model tends to miss some positive instances more than falsely predicting negatives.
6.Class 0 and class 1 have similar support, indicating a balanced dataset.
7.Overall, the model shows strong performance in classifying both positive and negative

instances. The high precision, recall, and F1-scores indicate that the model is reliable in distinguishing between the two classes

Classifier No 4 Random Forest and its Tuning with Parameters

Model Building on Random Forest

The Random forests classifier is a supervised learning algorithm and is known as the most flexible and easy to use algorithm. A forest is comprised of decision trees. 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.

Advantages of random forest:

Random forests is considered as a highly accurate and robust method because of the number of decision trees participating in the process. It does not suffer (that much) from the overfitting problem. The main reason is that it takes the average of all the predictions and therefore cancels out the biases. Disadvantages of random forest:

The model is more difficult to interpret compared to a decision tree. Random forests process is time-consuming. This is because the model is slow in generating predictions due to the multiple decision trees.

Building Model on Random Forest classifier

RandomForestClassifier

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

TABLE 71 RANDOM FOREST CLASSIFICATION

Random Forest Feature Importance

	Imp
Tenure	0.24
cashback	0.08
CC_Contacted_LY	0.07
Complain_ly	0.07
Day_Since_CC_connect	0.06
rev_growth_yoy	0.06
CC_Agent_Score	0.05
rev_per_month	0.05
AccountID	0.05
account_segment	0.04
Account_user_count	0.04
coupon_used_for_payment	0.03
Marital_Status_Single	0.03
City_Tier	0.03
Gender_Male	0.02

Service_Score	0.02
Login_device_Mobile	0.02
Payment_Credit Card	0.02
Payment_Debit Card	0.01
Marital_Status_Married	0.01
Payment_E wallet	0.01
Payment_UPI	0.01
Login_device_Unknown	0.00

TABLE 72 RANDOM FOREST FEATURE IMPORTANCE

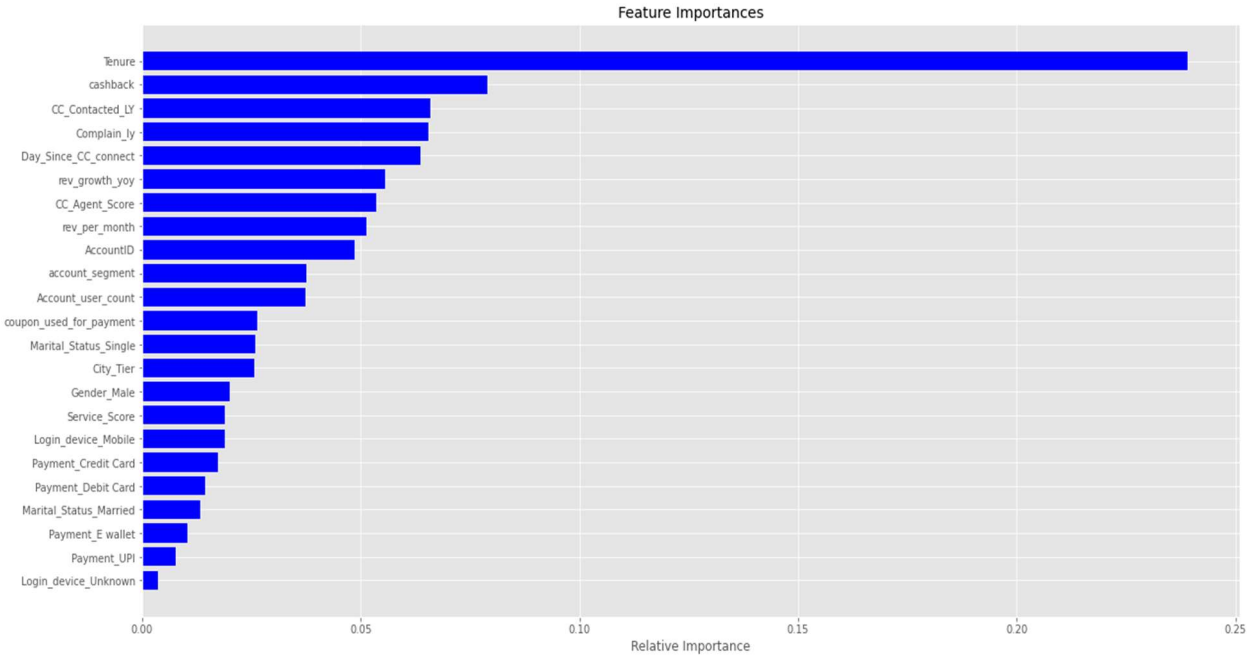

FIGURE 30 FEATURE IMPORTANCE

Random Forest Classification Report on Train Dataset

Classification Report

	precision	recall	f1-score	support
0	1.00	1.00	1.00	7471
1	1.00	1.00	1.00	1537
accuracy			1.00	9008
macro avg	1.00	1.00	1.00	9008
weighted avg	1.00	1.00	1.00	9008

TABLE 73 RANDOM FOREST CLASSIFICATION REPORT

Random Forest Confusion matrix with the percentage on train

Confusion Matrix

```
 [[7471    0]
 [   0 1537]]
```

TABLE 74 CONFUSION MATRIX RF

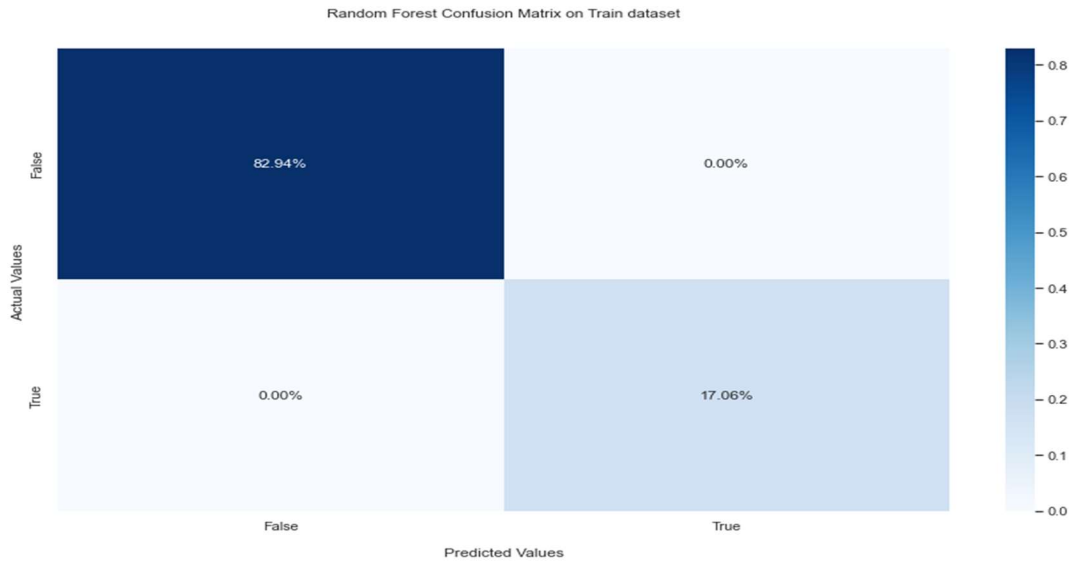

FIGURE 31 CONFUSION MATRIX PERCENTAGE

Random Forest Classification Report on Test dataset

Classification Report

	precision	recall	f1-score	support
0	0.99	1.00	1.00	2809
1	0.99	0.97	0.98	569
accuracy			0.99	3378
macro avg	0.99	0.98	0.99	3378
weighted avg	0.99	0.99	0.99	3378

TABLE 75 CLASSIFICATION REPORT

RANDOM FOREST CONFUSION MATRIX WITH THE PERCENTAGE ON TEST

```
Confusion Matrix
 [[2805    4]
 [  19  550]]
```

TABLE 76 CONFUSION MATRIX RANDOM FOREST

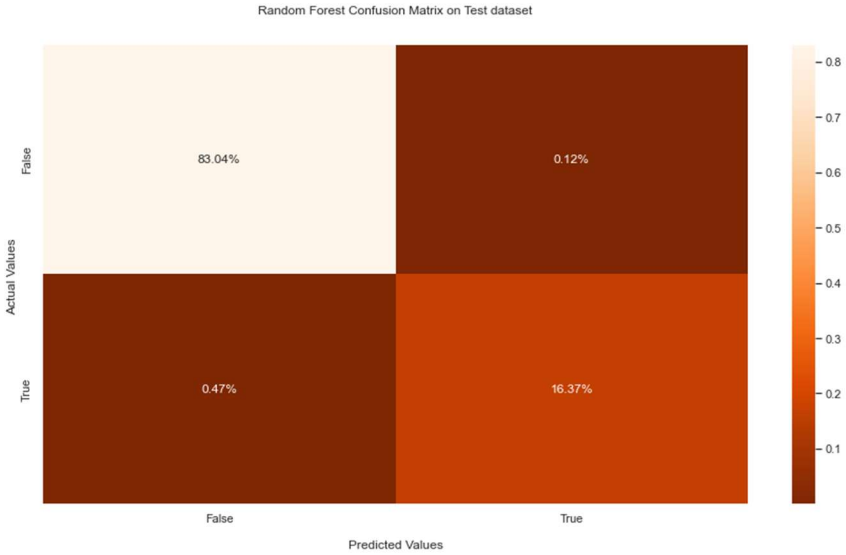

FIGURE 32 CONFUSION MATRIX PERCENTAGE

Random Forest AUC Score and curve on Train and Test DataSet

```
AUC for the Training Data: 1.000
AUC for the Test Data:     1.000
```

TABLE 77 AUC SCORE RANDOM FOREST

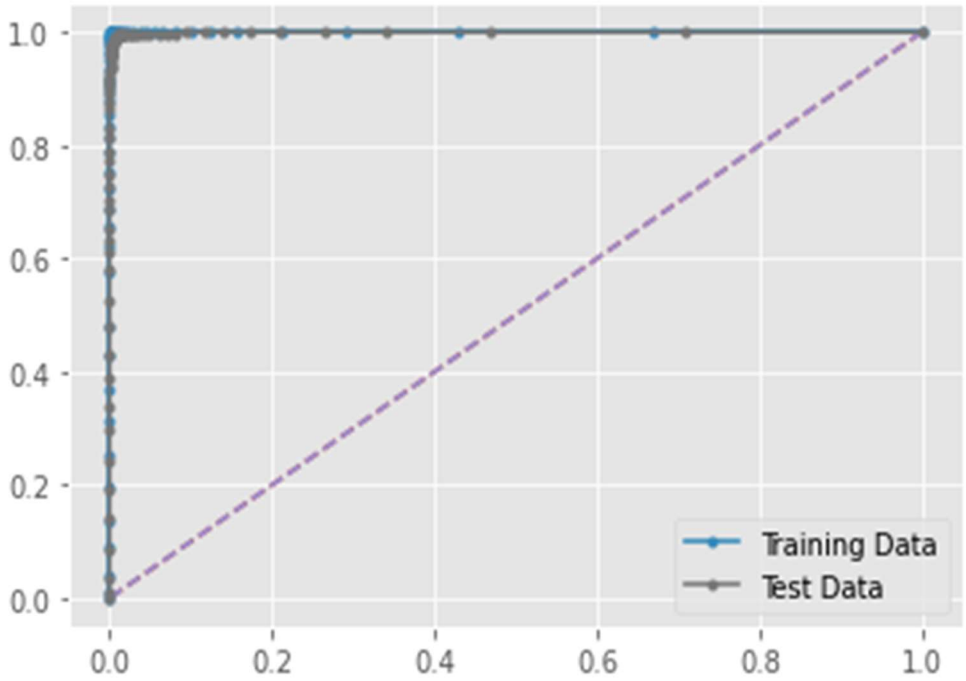

FIGURE 33 RO CURVE

Fivefold K-Fold cross Validation of Random Forest

```
[0.9931912374185908, 0.9931912374185908, 0.9931912374185908, 0.99319123
74185908, 0.9931912374185908]
Mean of testing accuracy over 5 folds = 0.99 with std = 0.00
```

TABLE 78 FIVEFOLD K-FOLD CROSS VALIDATION OF RANDOM FOREST

Random Forest Accuracy on train and test dataset

```
Accuracy of Random Forest on train set: 1.000
Accuracy of Random Forest on test set:  0.994
```

TABLE 79 ACCURACY SCORE RF

Building Model on Random Forest GridSearchCV classifier

Best grid= RandomForestClassifier

```
RandomForestClassifier(max_depth=8, max_features=7, min_samples_leaf=2
,
                       min_samples_split=30, oob_score=True, random_state=0)
```

TABLE 80 BEST GRID RFGS

Random Forest GridSearchCV Feature Importance

	Imp
Tenure	0.42
Complain_ly	0.11
Day_Since_CC_connect	0.06
CC_Agent_Score	0.04
account_segment	0.04
cashback	0.04
CC_Contacted_LY	0.04
rev_per_month	0.04
Marital_Status_Single	0.03
Account_user_count	0.03
City_Tier	0.03
rev_growth_yoy	0.02
AccountID	0.02
Marital_Status_Married	0.01
coupon_used_for_payment	0.01
Payment_Credit Card	0.01
Login_device_Mobile	0.01
Payment_E wallet	0.01
Gender_Male	0.01
Payment_Debit Card	0.01
Service_Score	0.00
Payment_UPI	0.00
Login_device_Unknown	0.00

TABLE 81 RANDOM FOREST GRIDSEARCHCV FEATURE IMPORTANCE

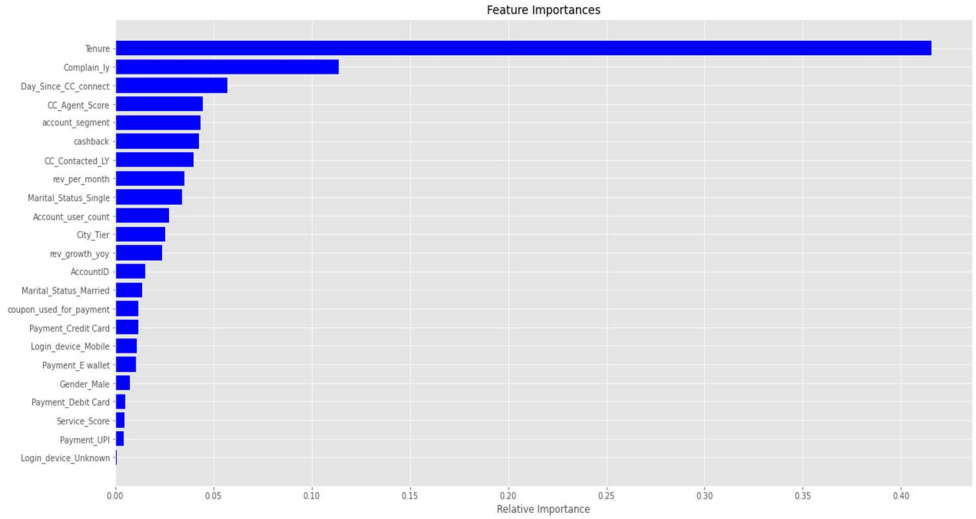

FIGURE 34 FEATURE IMPORTANCE

Random Forest GridSearchCV Classification Report on Train Dataset

Classification Report

	precision	recall	f1-score	support
0	0.94	0.99	0.96	7471
1	0.91	0.67	0.77	1537
accuracy			0.93	9008
macro avg	0.92	0.83	0.87	9008
weighted avg	0.93	0.93	0.93	9008

TABLE 82 RANDOM FOREST GRIDSEARCHCV CLASSIFICATION REPORT

Random Forest GridSearchCV Confusion matrix with the percentage on train

```
Confusion Matrix
 [[7365  106]
 [ 507 1030]]
```

TABLE 83GRIDSEARCHCV CONFUSION MATRIX RF

FIGURE 35 CONFUSION MATRIX PERCENTAGE

Random Forest GridSearchCV Classification Report on Test dataset

```
Classification Report
precision    recall  f1-score   support

           0       0.94      0.98      0.96      2809
           1       0.89      0.66      0.76       569

    accuracy                           0.93      3378
   macro avg       0.91      0.82      0.86      3378
weighted avg       0.93      0.93      0.93      3378
```

TABLE 84 RANDOM FOREST GRIDSEARCHCV CLASSIFICATION REPORT

Random Forest GridSearchCV Confusion matrix with the percentage on test

```
Confusion Matrix
 [[2763   46]
 [ 191  378]]
```

TABLE 85 RANDOM FOREST GRIDSEARCHCV CONFUSION MATRIX

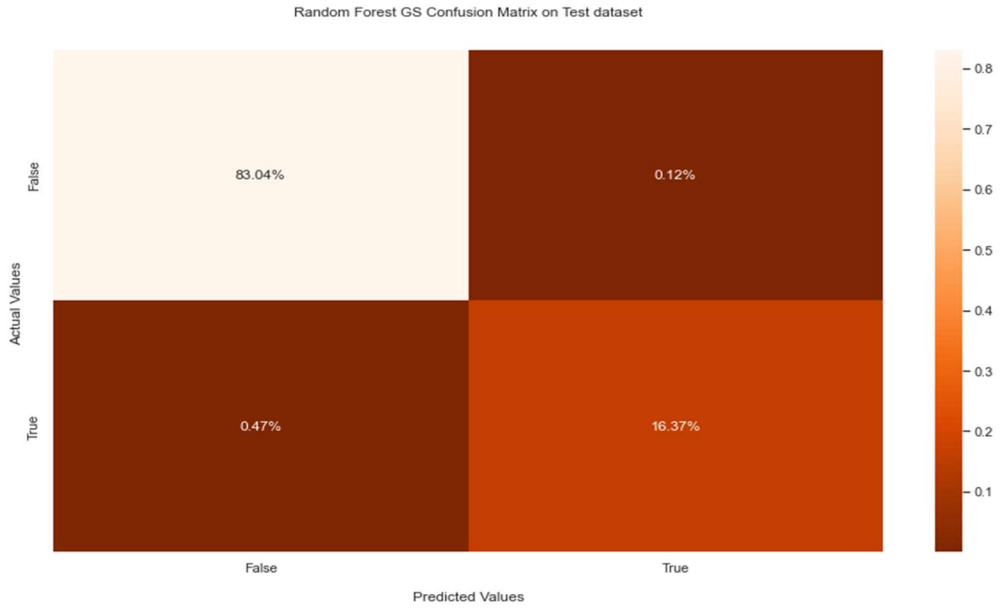

FIGURE 36 CONFUSION MATRIX PERCENTAGE

Random Forest GridSearchCV AUC Score and curve on Train and Test DataSet

```
AUC for the Training Data: 0.969
AUC for the Test Data:     0.969
```

TABLE 86 RANDOM FOREST GRIDSEARCHCV AUC SCORE

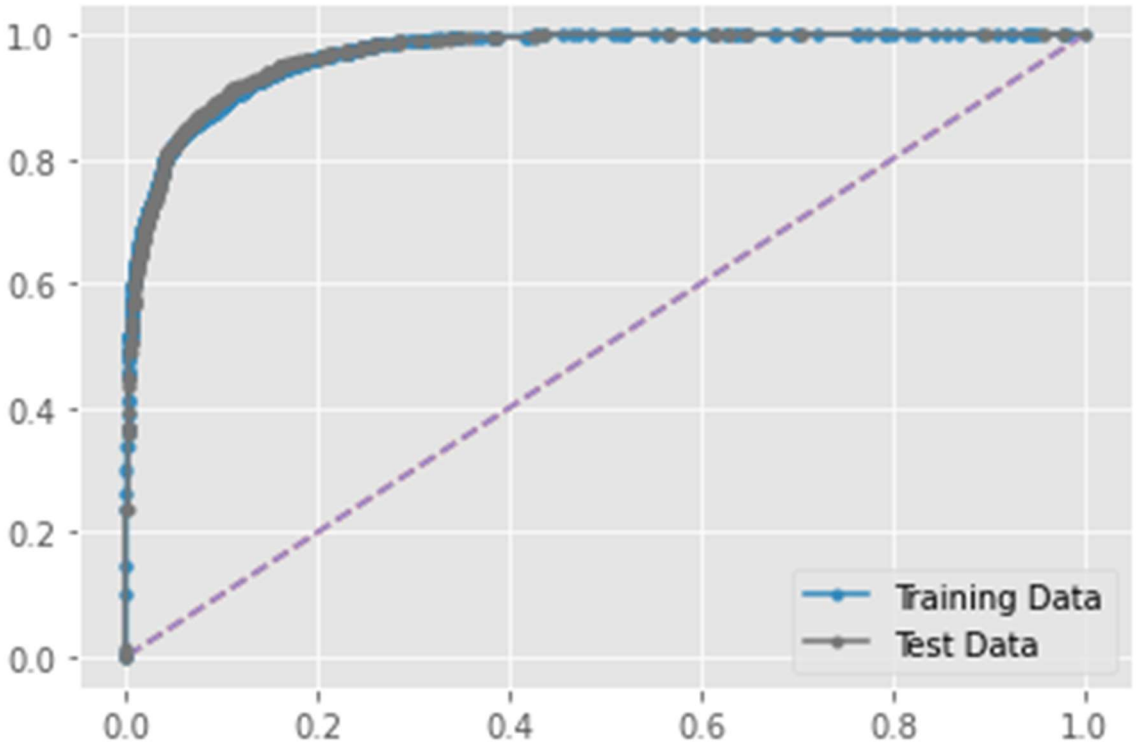

FIGURE 37 RO CURVE

Fivefold K-Fold cross Validation of Random Forest GridSearchCV

```
[0.9298401420959147, 0.9298401420959147, 0.9298401420959147, 0.92984014
20959147, 0.9298401420959147]
Mean of testing accuracy over 5 folds = 0.93 with std = 0.00
```

TABLE 87 FIVEFOLD K-FOLD CROSS VALIDATION OF RANDOM FOREST GRIDSEARCHCV

Random Forest GridSearchCV Accuracy on train and test dataset

```
Accuracy of GridSearchCV Random Forest on train set: 0.932
Accuracy of GridSearchCV Random Forest on test set:  0.930
```

TABLE 88 RANDOM FOREST GRIDSEARCHCV ACCURACY

Building Model on Random Forest SMOTE classifier

RandomForestClassifier

```
RandomForestClassifier(random_state=1
```

TABLE 89 RANDOM FOREST SMOTE CLASSIFIER

Random Forest SMOTE Feature Importance

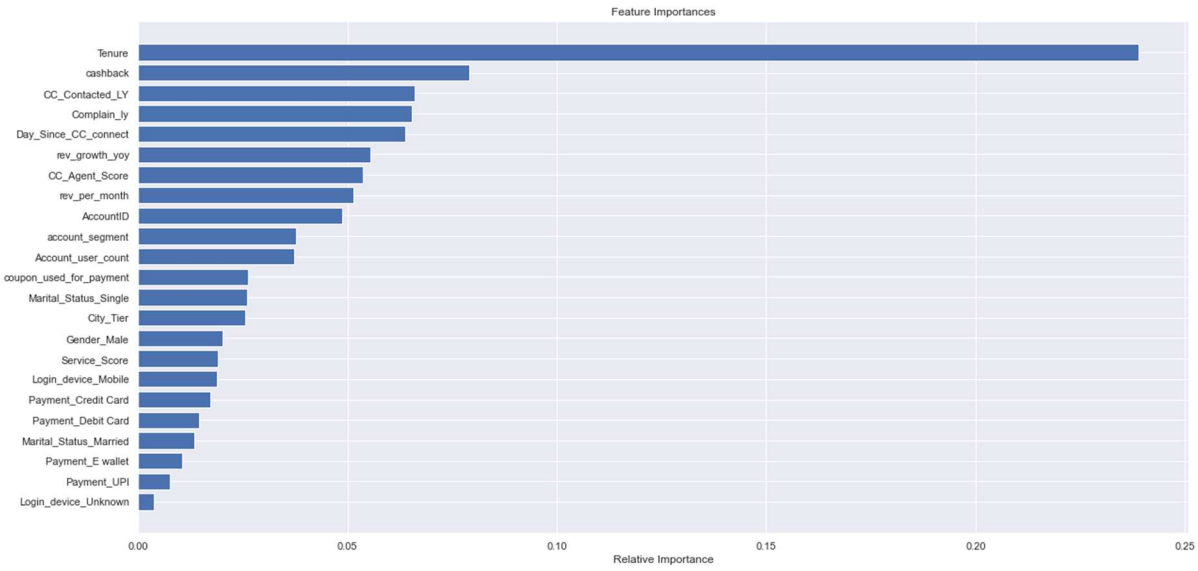

FIGURE 38 FEATURE IMPORTANCE

Random Forest SMOTE Classification Report on Train Dataset

```
Classification Report
              precision    recall  f1-score   support

           0       1.00      1.00      1.00      7471
           1       1.00      1.00      1.00      1537

    accuracy                           1.00      9008
   macro avg       1.00      1.00      1.00      9008
weighted avg       1.00      1.00      1.00      9008
```

TABLE 90 RANDOM FOREST SMOTE CLASSIFICATION REPORT

Random Forest SMOTE Confusion matrix with the percentage on train

```
Confusion Matrix
 [[7471    0]
 [   0 1537]]
```

TABLE 91 RANDOM FOREST SMOTE CONFUSION MATRIX

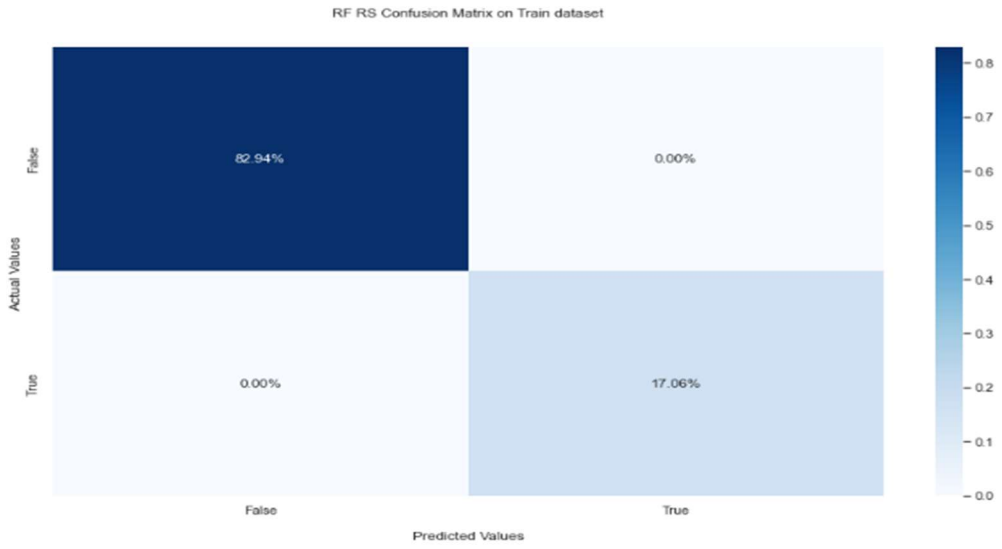

FIGURE 39 CONFUSION MATRIX PERCENTAGE

Random Forest SMOTE Classification Report on Test dataset

Classification Report

	precision	recall	f1-score	support
0	0.93	1.00	0.96	2810
1	1.00	0.92	0.96	2809
accuracy			0.96	5619
macro avg	0.96	0.96	0.96	5619
weighted avg	0.96	0.96	0.96	5619

TABLE 92 RANDOM FOREST SMOTE CLASSIFICATION REPORT

Random Forest SMOTE Confusion matrix with the percentage on test

```
Confusion Matrix
 [[2803    7]
 [ 222 2587]]
```

TABLE 93 RANDOM FOREST SMOTE CONFUSION MATRIX

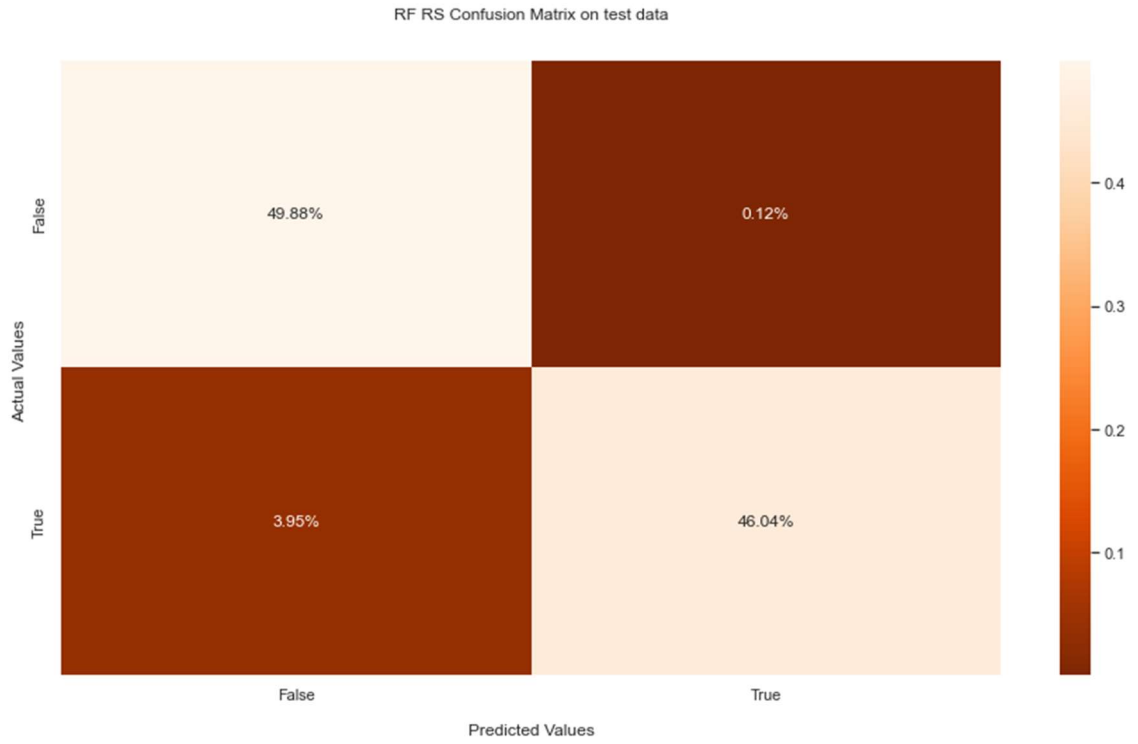

FIGURE 40 CONFUSION MATRIX PERCENTAGE

Random Forest SMOTE AUC Score and curve on Train and Test DataSet

```
AUC for the Training Data: 1.000
AUC for the Test Data:     0.999
```

TABLE 94 RANDOM FOREST SMOTE AUC SCORE

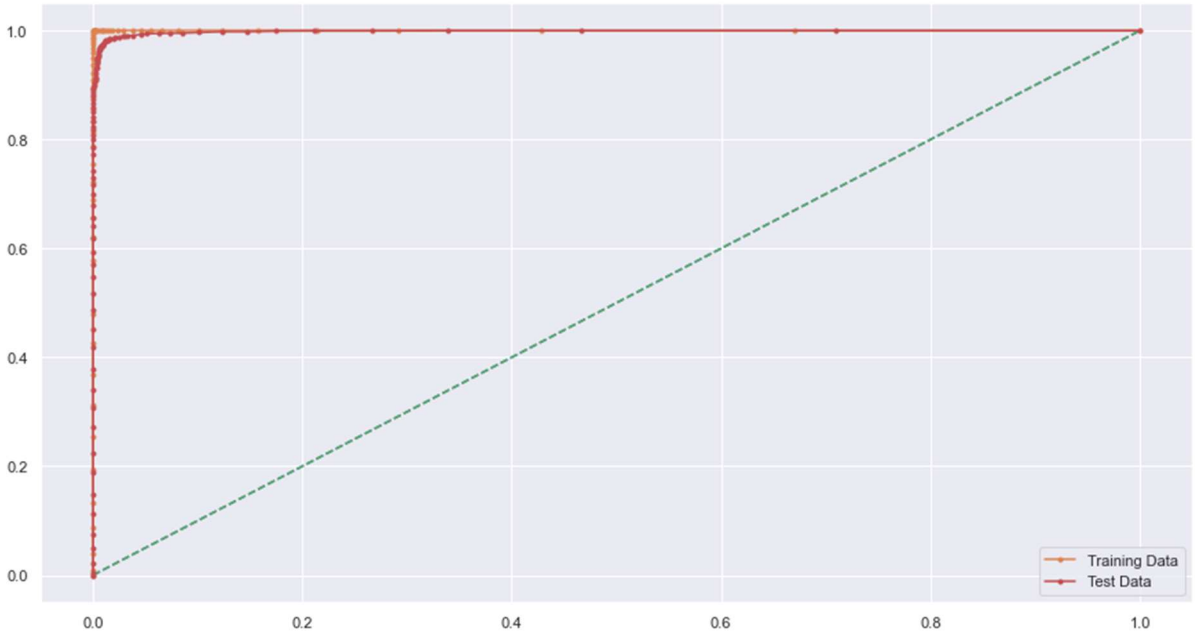

FIGURE 41 RO CURVE

Fivefold K-Fold cross Validation of Random Forest SMOTE

```
[0.9631607047517352, 0.9592454173340452, 0.9588894821142552, 0.96066915
82132052, 0.9592454173340452]
Mean of testing accuracy over 5 folds = 0.96 with std = 0.00
```

TABLE 95 FIVEFOLD K-FOLD CROSS VALIDATION OF RANDOM FOREST SMOTE

Random Forest SMOTE Accuracy on train and test dataset

```
Accuracy of Random Forest SMOTE on train set: 1.000
Accuracy of Random Forest SMOTE on test set:  0.959
```

TABLE 96 ACCURACY OF RANDOM FOREST SMOTE

Best Random Forest Model is Base Model the business Insights:

On the basis of model performance on both Train and Test data we can make our infer on which are as follows:

True Positive (TP): 2805 This represents the number of instances that are actually positive (class 1) and are correctly predicted as positive by the model.

False Positive (FP): 4 This indicates the number of instances that are actually negative (class 0) but are incorrectly predicted as positive by the model.

False Negative (FN): 18 This represents the number of instances that are actually positive (class 1) but are incorrectly predicted as negative by the model.

True Negative (TN): 553 This indicates the number of instances that are actually negative (class 0) and are correctly predicted as negative by the model.

Classification Report: Precision, Recall, F1-score, Support, and Accuracy metrics are provided for each class (0 and 1), as well as macro and weighted averages.

Precision: Precision measures the model's ability to correctly identify positive instances out of the total instances predicted as positive. For class 0, the precision is 0.99, indicating that 99% of the instances predicted as class 0 are correct. For class 1, the precision is 0.99, meaning that 99% of the instances predicted as class 1 are correct.

Recall: Recall (also known as sensitivity or true positive rate) measures the model's ability to correctly identify positive instances out of the actual positive instances. For class 0, the recall is 1.0, indicating that the model identifies 100% of the instances of class 0. For class 1, the recall is 0.97, meaning that the model identifies 97% of the instances of class 1.

F1-score: The F1-score is the harmonic mean of precision and recall. It provides a balance between precision and recall. For class 0, the F1-score is 1.0, indicating good performance. For class 1, the F1-score is also 0.98, indicating good performance.

Support: Support represents the number of instances in each class. For class 0, the support is 2809, and for class 1, the support is 569.

Accuracy: The overall accuracy of the model is 0.99, meaning that it correctly predicts 99% of the instances.

AUC score: AUC score of 1.000 for the test data means that the model achieved a perfect classification performance on the test set. It suggests that the model achieved a perfect balance between the true positive rate and the false positive rate, resulting in a flawless ROC curve

Based on the confusion AUC score, matrix and classification report:

1.The model shows high accuracy and performs well in classifying both classes (0 and 1).
2.The model has high precision for both classes, indicating a low false positive rate.
3.The recall for class 0 is high (0.99), indicating that the model correctly identifies a large majority of instances of class 0. The recall for class 1 is also good (0.97), indicating a reasonably good ability to identify positive instances.
4.The F1-scores for both classes are high, indicating a good balance between precision and recall.
5.The number of false negatives (16) is relatively higher than the false positives (4), suggesting that the model tends to miss some positive instances more than falsely predicting negatives.
6.Class 0 and class 1 have similar support, indicating a balanced dataset.
7.Overall, the model shows strong performance in classifying both positive and negative instances. The high precision, recall, and F1-scores indicate that the model is reliable in distinguishing between the two classes
8.This high AUC score indicates that the model has excellent discriminative ability in distinguishing between the positive and negative classes in the test data

Classifier No 5 KNN and its Tuning with Parameters

For naive bayes algorithm while calculating likelihoods of numerical features it assumes the feature to be normally distributed and then we calculate probability using mean and variance of that feature only and also it assumes that all the predictors are independent to each other. Scale doesn't matter. Performing a features scaling in this algorithm may not have much effect.

To tackle the high variance of the Hold-out method, the k-fold method is used. The idea is simple, divide the whole dataset into 'k' sets preferably of equal sizes. Then the first set is selected as the test set and the rest 'k-1' sets are used to train the data. Error is calculated for this particular dataset. Then the steps are repeated, i.e., the second set is selected as the test data, and the remaining 'k-1' sets are used as the training data. Again, the error is calculated. Similarly, the process continues for 'k' times. In the end, the CV error is given as the mean of the total errors calculated individually.

Building Logistic KNN model

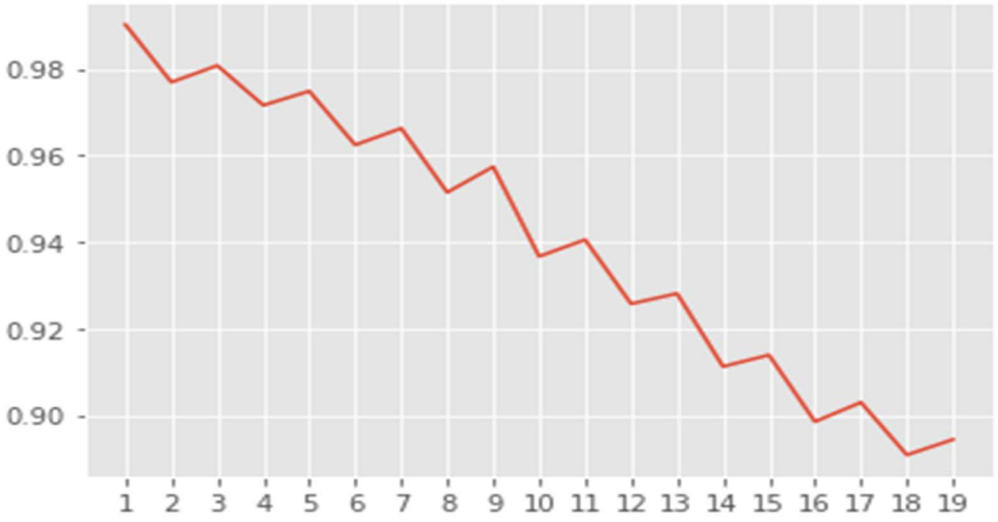

FIGURE 42 FINDING K

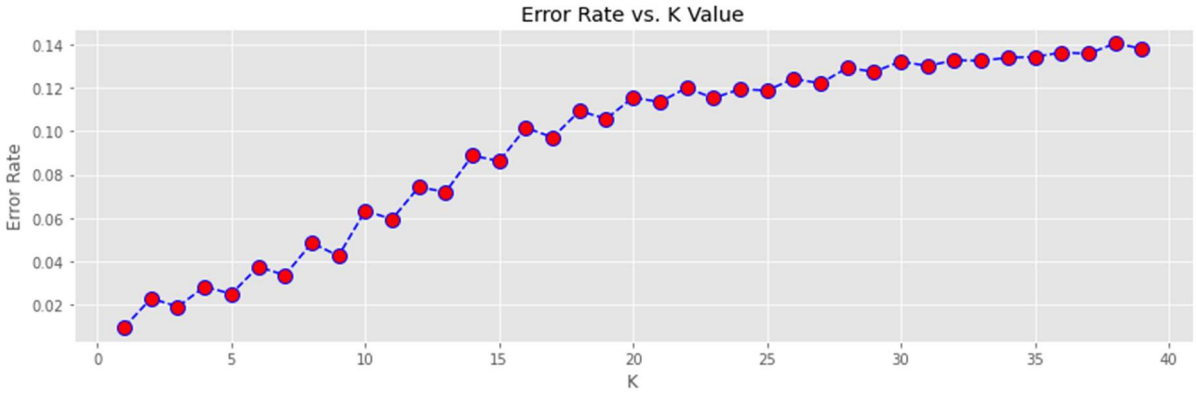

FIGURE 43 FINDING K BY ERROR METHOD

KNN basic model Classification Report on Train Dataset

```
Classification Report
              precision    recall  f1-score   support

           0       0.98      0.99      0.99      7471
           1       0.96      0.90      0.93      1537

    accuracy                           0.98      9008
   macro avg       0.97      0.95      0.96      9008
weighted avg       0.98      0.98      0.98      9008
```

TABLE 97 KNN BASIC MODEL CLASSIFICATION REPORT

KNN basic model Confusion matrix with the percentage on train

```
Confusion Matrix
 [[7408   63]
 [ 150 1387]]
```

TABLE 98 KNN BASIC MODEL CONFUSION MATRIX

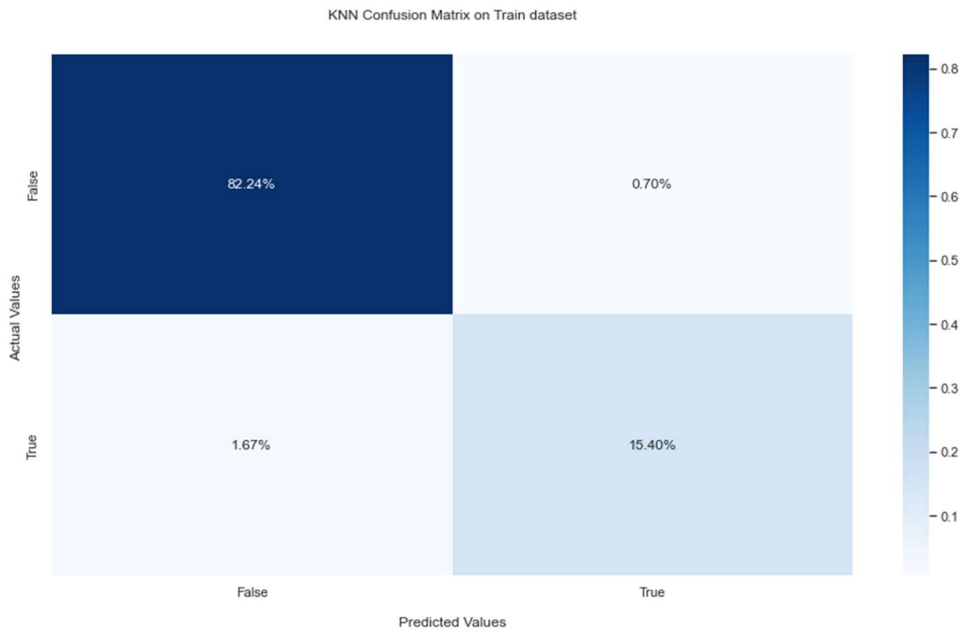

FIGURE 44 CONFUSION MATRIX PERCENTAGE

KNN basic model Classification Report on Test dataset

```
Classification Report
                precision    recall  f1-score   support

           0       0.98      0.99      0.98      2809
           1       0.95      0.90      0.92       569

    accuracy                           0.97      3378
   macro avg       0.96      0.94      0.95      3378
weighted avg       0.97      0.97      0.97      3378
```

TABLE 99 KNN BASIC MODEL CLASSIFICATION REPORT

KNN basic model Confusion matrix with the percentage on test

```
Confusion Matrix
 [[2782   27]
 [  58  511]]
```

TABLE 100 KNN BASIC MODEL CONFUSION MATRIX

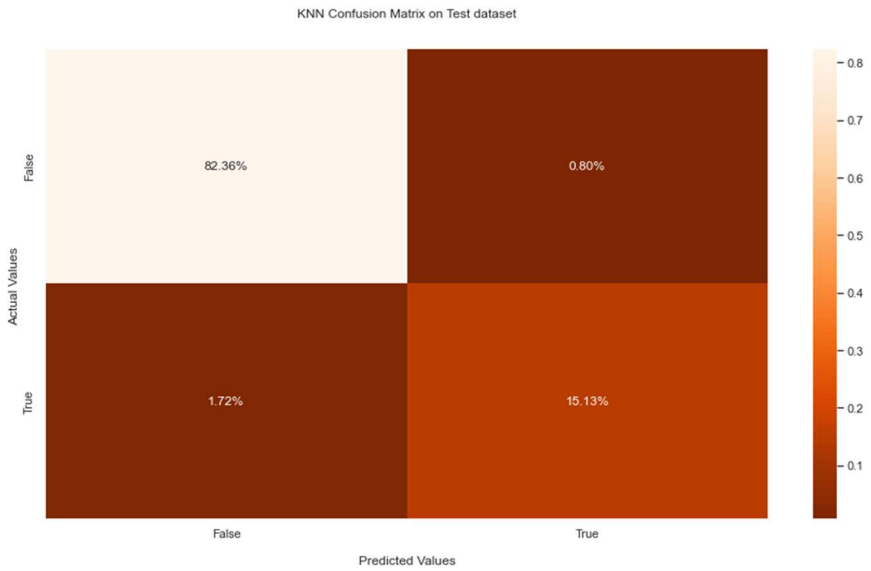

FIGURE 45 CONFUSION MATRIX PERCENTAGE

KNN basic model AUC Score and curve on Train and Test DataSet

```
AUC for the Training Data: 0.995
AUC for the Test Data:     0.995
```

TABLE 101 KNN BASIC MODEL AUC SCORE

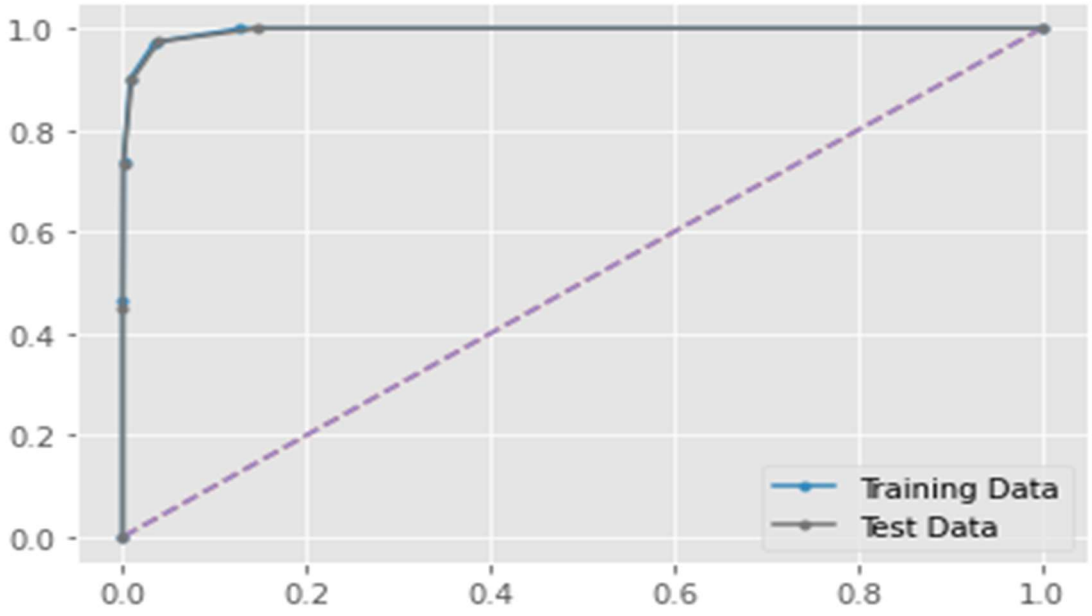

FIGURE 46 RO CURVE

Fivefold K-Fold cross Validation of KNN Basic model

```
[0.9748371817643576, 0.9748371817643576, 0.9748371817643576, 0.97483718
17643576, 0.9748371817643576]
Mean of testing accuracy over 5 folds = 0.97 with std = 0.00
```

TABLE 102 FIVEFOLD K-FOLD CROSS VALIDATION OF KNN BASIC MODEL

KNN basic model Accuracy on Train and Test dataset

```
Accuracy of KNN on train set: 0.976
Accuracy of KNN on test set:  0.975
```

TABLE 103 KNN BASIC MODEL ACCURACY

Building GridSearchCV KNN model

KNeighborsClassifier

```
KNeighborsClassifier(metric='manhattan', n_jobs=1, n_neighbors=1, p=1)
```

TABLE 104 KNEIGHBORSCLASSIFIER BEST GRID

KNN GridSearchCV model Classification Report on Train Dataset

```
Classification Report
              precision    recall  f1-score   support

           0       1.00      1.00      1.00      7471
           1       1.00      1.00      1.00      1537

    accuracy                           1.00      9008
   macro avg       1.00      1.00      1.00      9008
weighted avg       1.00      1.00      1.00      9008
```

TABLE 105 KNN GRIDSEARCHCV MODEL CLASSIFICATION

KNN GridSearchCV model Confusion matrix with the percentage on train

```
Confusion Matrix
 [[7471    0]
 [   0 1537]]
```

TABLE 106 KNN GRIDSEARCHCV MODEL CONFUSION MATRIX

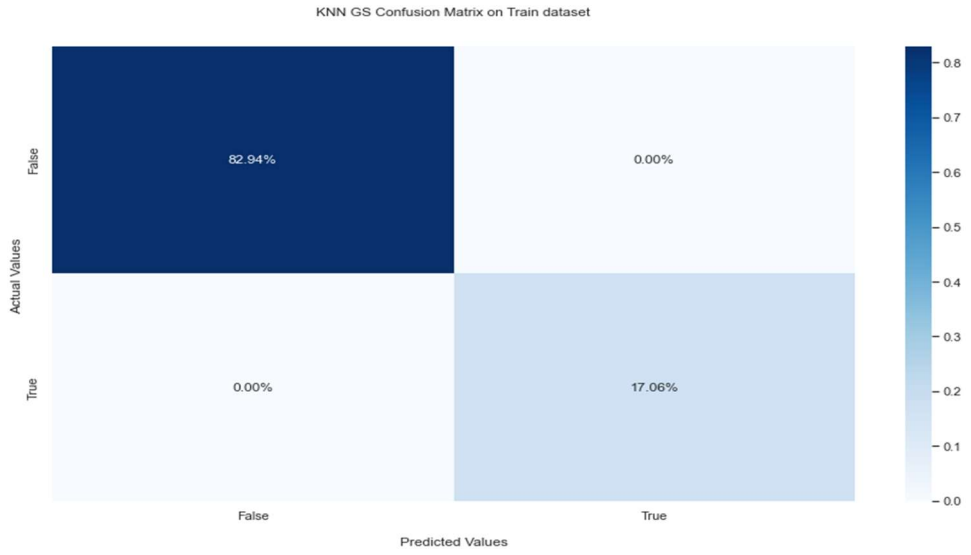

FIGURE 47 CONFUSION MATRIX PERCENTAGE

KNN GridSearchCV model Classification Report on Test dataset

```
Classification Report
              precision    recall  f1-score   support

           0       1.00      1.00      1.00      2809
           1       0.98      0.99      0.99       569

    accuracy                           1.00      3378
   macro avg       0.99      0.99      0.99      3378
weighted avg       1.00      1.00      1.00      3378
```

TABLE 107 KNN GRIDSEARCHCV MODEL CLASSIFICATION REPORT

KNN GridSearchCV model Confusion matrix with the percentage on test

```
Confusion Matrix
 [[2800    9]
 [   6  563]]
```

TABLE 108 KNN GRIDSEARCHCV MODEL CONFUSION MATRIX

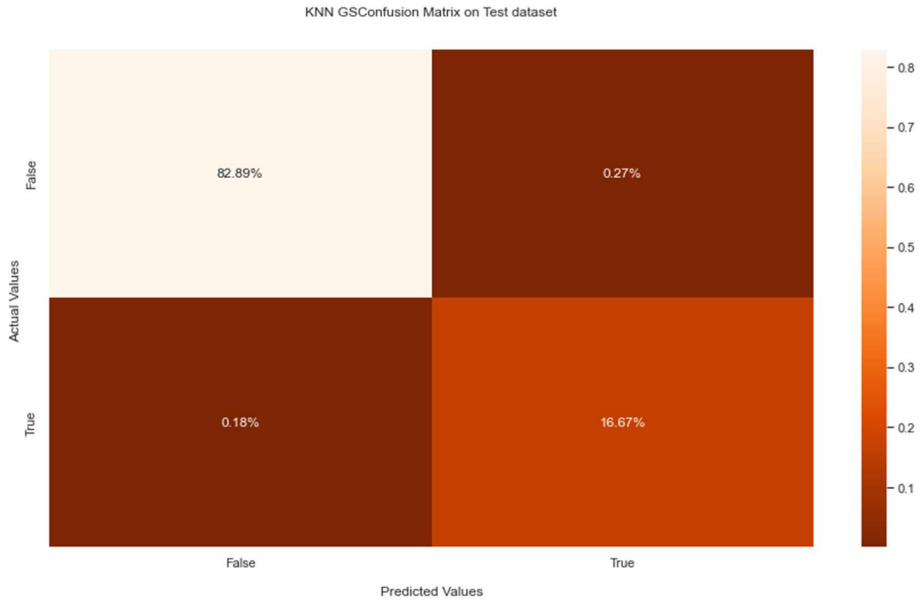

FIGURE 48 CONFUSION MATRIX PERCENTAGE

KNN GridSearchCV model AUC Score and curve on Train and Test DataSet

```
AUC for the Training Data: 1.000
AUC for the Test Data:     0.993
```

TABLE 109 KNN GRIDSEARCHCV MODEL AUC SCORE

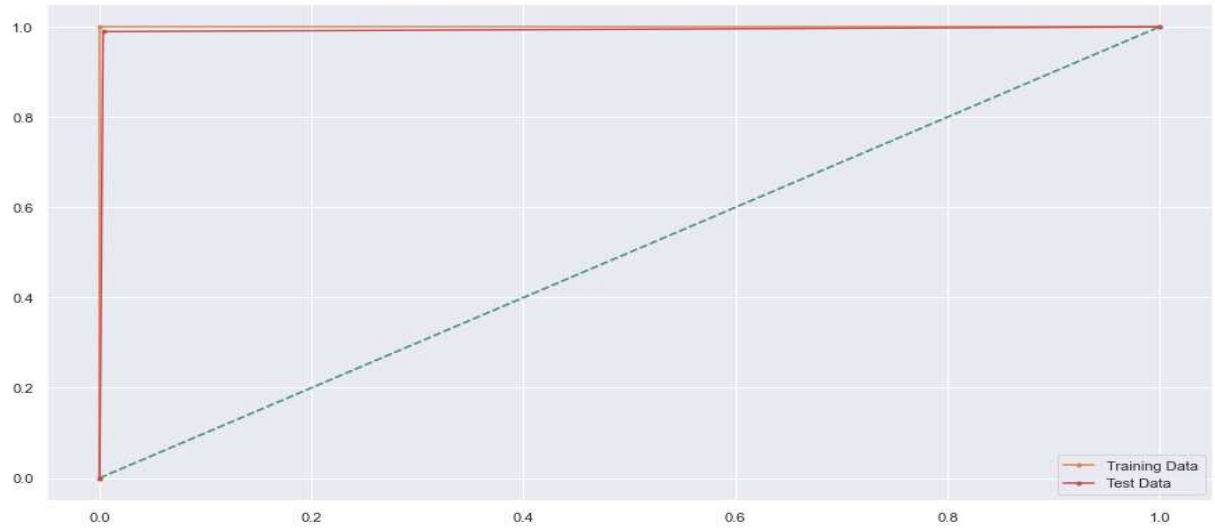

FIGURE 49 RO CURVE

Fivefold K-Fold cross Validation of KNN GridSearchCV model

```
[0.9955595026642984, 0.9955595026642984, 0.9955595026642984, 0.99555950
26642984, 0.9955595026642984]
Mean of testing accuracy over 5 folds = 1.00 with std = 0.00
```

TABLE 110 FIVEFOLD K-FOLD CROSS VALIDATION OF KNN GRIDSEARCHCV MODEL

KNN GridSearchCV model Accuracy on Train and Test dataset

```
Accuracy of KNN GridSearchCV on train set: 1.000
Accuracy of KNN GridSearchCV on test set:  0.996
```

TABLE 111 KNN GRIDSEARCHCV MODEL ACCURACY

Building SMOTE KNN model

KNN SMOTE model Classification Report on Train Dataset

```
Classification Report
              precision    recall  f1-score   support

           0       0.98      0.99      0.99      7471
           1       0.96      0.90      0.93      1537

    accuracy                           0.98      9008
   macro avg       0.97      0.95      0.96      9008
weighted avg       0.98      0.98      0.98      9008
```

TABLE 112 KNN SMOTE MODEL CLASSIFICATION REPORT

KNN SMOTE model Confusion matrix with the percentage on train

```
Confusion Matrix
 [[7408   63]
 [ 150 1387]]
```

TABLE 113 KNN SMOTE MODEL CONFUSION MATRIX

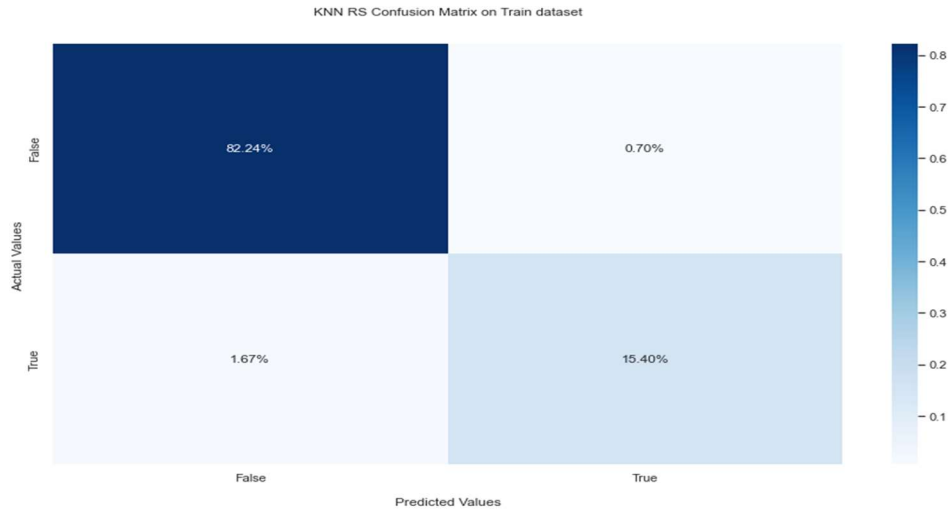

FIGURE 50 CONFUSION MATRIX PERCENTAGE

KNN SMOTE model Classification Report on Test dataset

```
Classification Report
              precision    recall  f1-score   support

           0       0.93      0.99      0.96      2810
           1       0.99      0.93      0.96      2809

    accuracy                           0.96      5619
   macro avg       0.96      0.96      0.96      5619
weighted avg       0.96      0.96      0.96      5619
```

TABLE 114 KNN SMOTE MODEL CLASSIFICATION REPORT

KNN SMOTE model Confusion matrix with the percentage on test

```
Confusion Matrix
 [[2785   25]
 [ 205 2604]]
```

TABLE 115 KNN SMOTE MODEL CONFUSION MATRIX

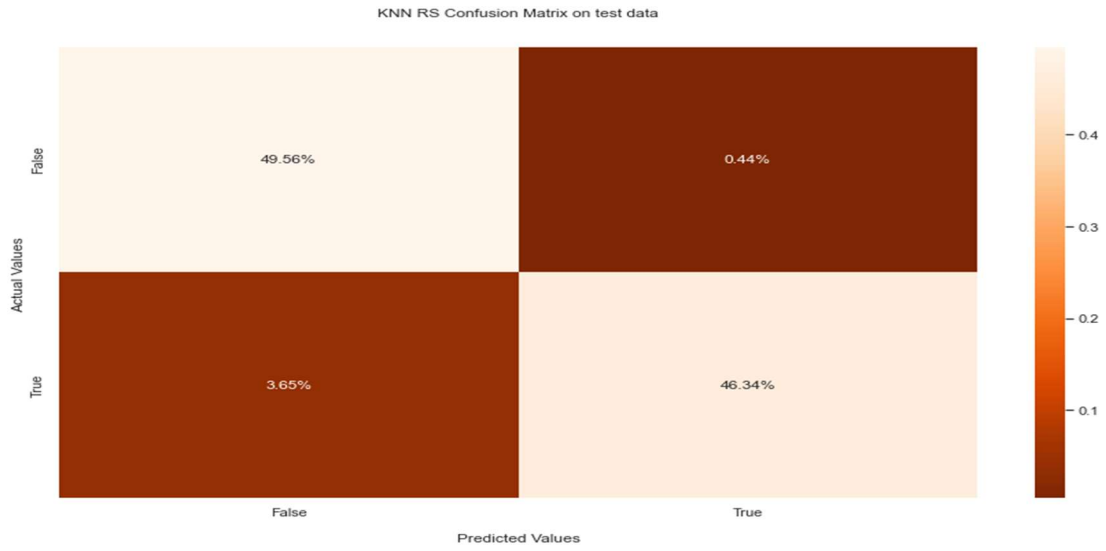

FIGURE 51 CONFUSION MATRIX PERCENTAGE

KNN SMOTE model AUC Score and curve on Train and Test DataSet

```
AUC for the Training Data: 0.995
AUC for the Test Data:     0.996
```

TABLE 116 KNN SMOTE MODEL AUC SCORE

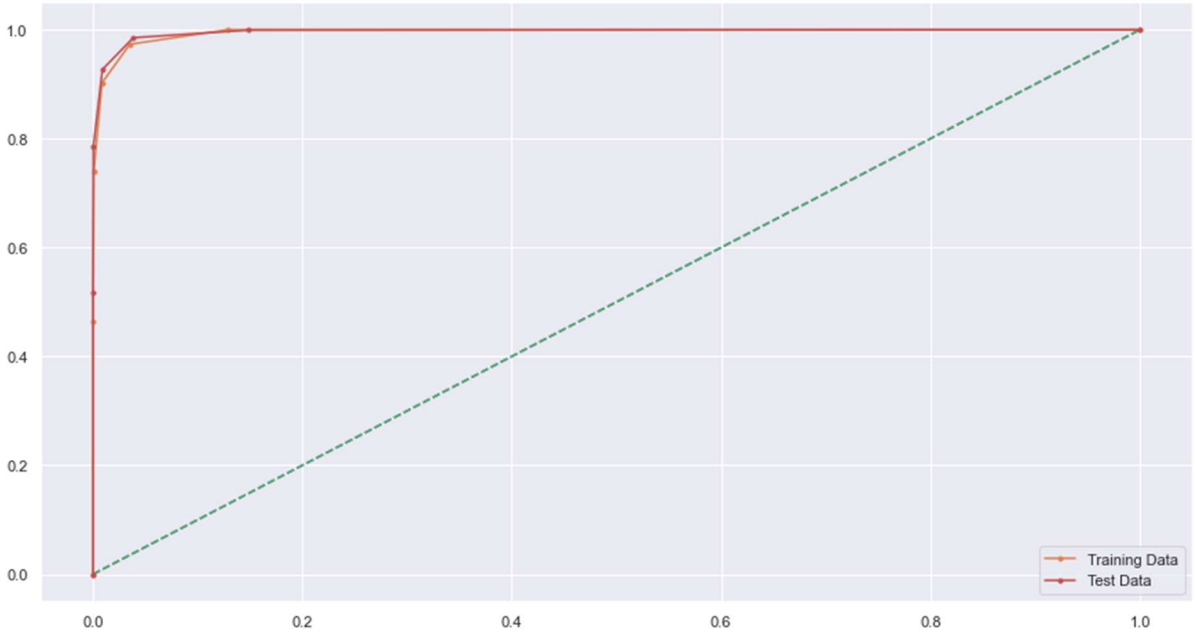

FIGURE 52 RO CURVE

Fivefold K-Fold cross Validation of KNN SMOTE model

```
[0.9578216764548853, 0.9567538707955152, 0.9501690692294003, 0.9510589072788752, 0.9590674497241503]
Mean of testing accuracy over 5 folds = 0.95 with std = 0.00
```

TABLE 117 FIVEFOLD K-FOLD CROSS VALIDATION OF KNN SMOTE MODEL

KNN SMOTE model Accuracy on Train and Test dataset

```
Accuracy of KNN Resampled on train set: 0.976
Accuracy of KNN Resampled on test set:  0.959
```

TABLE 118 KNN SMOTE MODEL ACCURACY

Insights KNN Best Model: KNN GridSearchCV

1.The model achieved a high accuracy of 1.00, indicating that it predicted the correct class for the majority of instances.
2.The precision for class 0 is 1.00, which means that all predicted class 0 instances were actually class 0.
3.The recall for class 0 is 1.00, indicating that the model correctly identified all instances of class 0.
4.The precision for class 1 is 0.93, suggesting that the model predicted class 1 correctly in the majority of cases.
5.The recall for class 1 is 0.99, indicating that the model correctly identified most instances of class 1.

Business Insights:

1.The model shows excellent performance in distinguishing between the two classes, with high precision and recall values. With an accuracy of 1.00, the model can be considered highly reliable in predicting the target variable.

2.The high precision and recall for class 0 suggest that the model is effective in identifying instances of class 0 correctly, which may be of significant importance in certain business scenarios.

3.The model's ability to accurately predict instances of class 1 (although slightly lower precision and recall compared to class 0) can still provide valuable insights for decision-making or risk assessment.

Overall, the model demonstrates strong predictive capabilities and can be considered reliable for making predictions in this specific context

Classifier No 6: Naive Bayes and its Tuning with Parameters

Naïve Bayes Classifier is one of the simple and most effective Classification algorithms which helps in building the fast machine learning models that can make quick predictions. Naïve Bayes Classifier is a probabilistic classifier and is based on Bayes Theorem. which means it predicts on the basis of the probability of an object.

Principle: all naive Bayes classifiers assume that the value of a particular feature is independent of the value of any other feature, given the class variable. Bayes theorem gives us the probability of Event A to happen given that event B has occurred.

Improvement of Model Accuracy Logic Total number of fits is 1000 since the cv is defined as 10 and there are 100 candidates (var_smoothing has 100 defined parameters). Therefore, the calculation for a total number of fits → 10 x [100] = 1000.

Estimator is the machine learning model of interest, provided the model has a scoring function; in this case, the model assigned is GaussianNB(). param_grid is a dictionary with parameters names (string) as keys and lists of parameter settings to try as values; this enables searching over any sequence of parameter settings. verbose is the verbosity: the higher, the more messages; in this case, it is set to 1. cv is the cross-validation generator or an iterable, in this case, there is a 10-fold cross-validation. n_jobs is the maximum number of concurrently running workers; in this case, it is set to -1 which implies that all CPUs are used.

Building Naïve Bayes Classifier

Naïve Bayes Classifier Classification Report on Train Dataset

```
Classification Report
               precision    recall  f1-score   support

           0       0.90      0.93      0.91      7471
           1       0.59      0.51      0.55      1537

    accuracy                           0.86      9008
   macro avg       0.75      0.72      0.73      9008
weighted avg       0.85      0.86      0.85      9008
```

TABLE 119 NAÏVE BAYES CLASSIFIER CLASSIFICATION REPORT

Naïve Bayes Classifier Confusion matrix with the percentage on train

```
Confusion Matrix
 [[6921  550]
 [ 748  789]]
```

TABLE 120 NAÏVE BAYES CLASSIFIER CONFUSION MATRIX

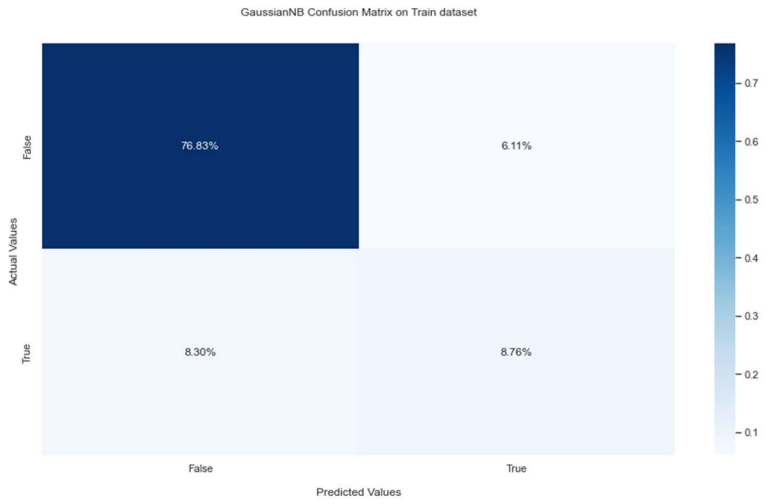

FIGURE 53 CONFUSION MATRIX PERCENTAGE

Naïve Bayes Classifier Classification Report on Test dataset

```
Classification Report
              precision    recall  f1-score   support

           0       0.91      0.90      0.91      2809
           1       0.54      0.58      0.56       569

    accuracy                           0.85      3378
   macro avg       0.73      0.74      0.73      3378
weighted avg       0.85      0.85      0.85      3378
```

TABLE 121 NAÏVE BAYES CLASSIFIER CLASSIFICATION REPORT

Naïve Bayes Classifier Confusion matrix with the percentage on test

```
Confusion Matrix
 [[2532  277]
 [ 241  328]]
```

TABLE 122 NAÏVE BAYES CLASSIFIER CONFUSION MATRIX

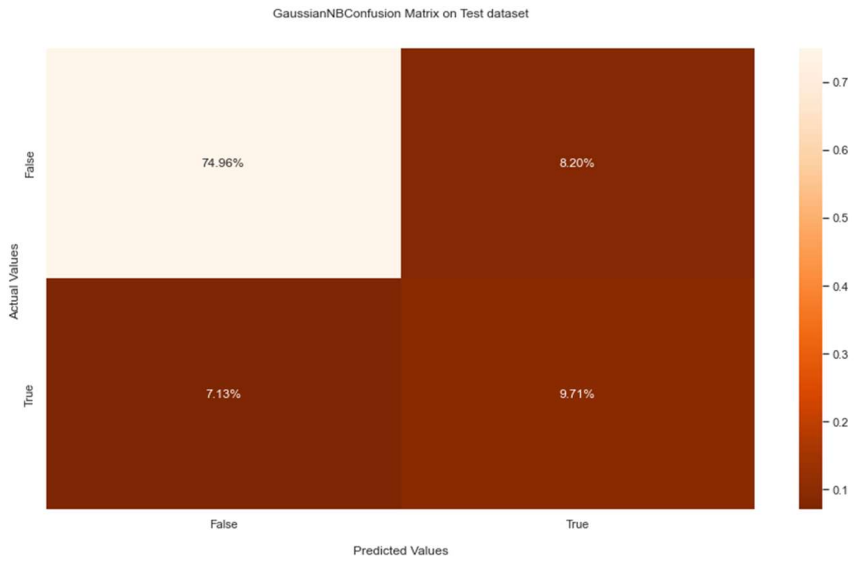

FIGURE 54 CONFUSION MATRIX PERCENTAGE

Naïve Bayes Classifier AUC Score and curve on Train and Test DataSet

```
AUC for the Training Data: 0.822
AUC for the Test Data:     0.823
```

TABLE 123 NAÏVE BAYES CLASSIFIER AUC SCORE

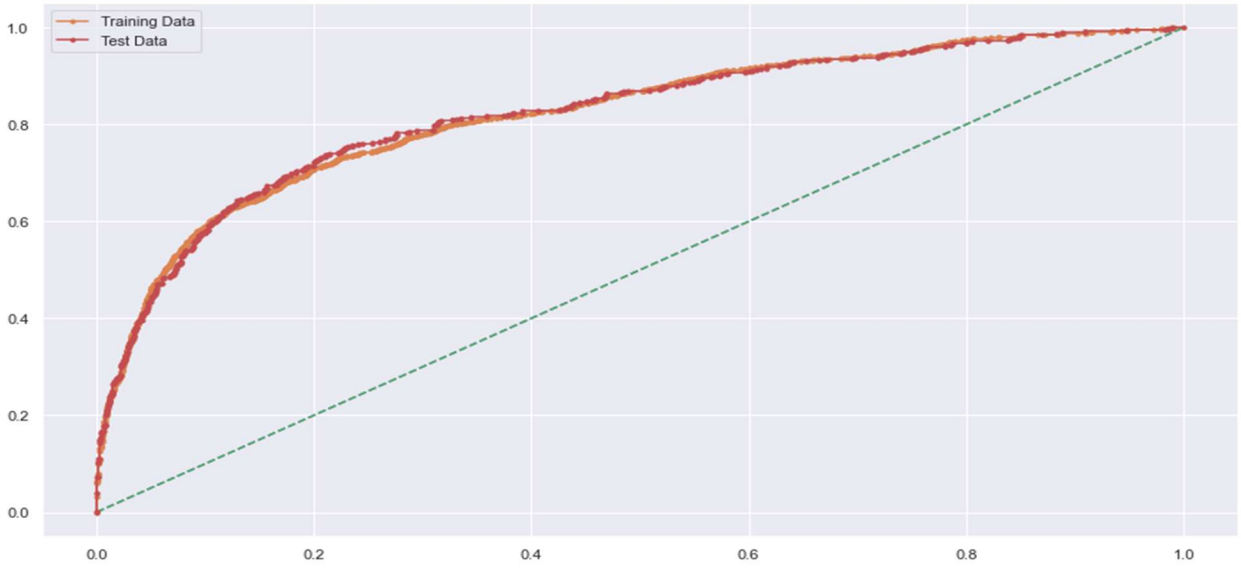

FIGURE 55 RO CURVE

Accuracy of Naïve Bayes Classifier

```
AUC for the Training Data: 0.822
AUC for the Test Data:     0.823
```

TABLE 124 ACCURACY OF NAÏVE BAYES CLASSIFIER

Building Naïve Bayes Classifier GridSearchCV

GaussianNB

```
GaussianNB(var_smoothing=1e-08)
```

TABLE 125 GAUSSIANNB GS

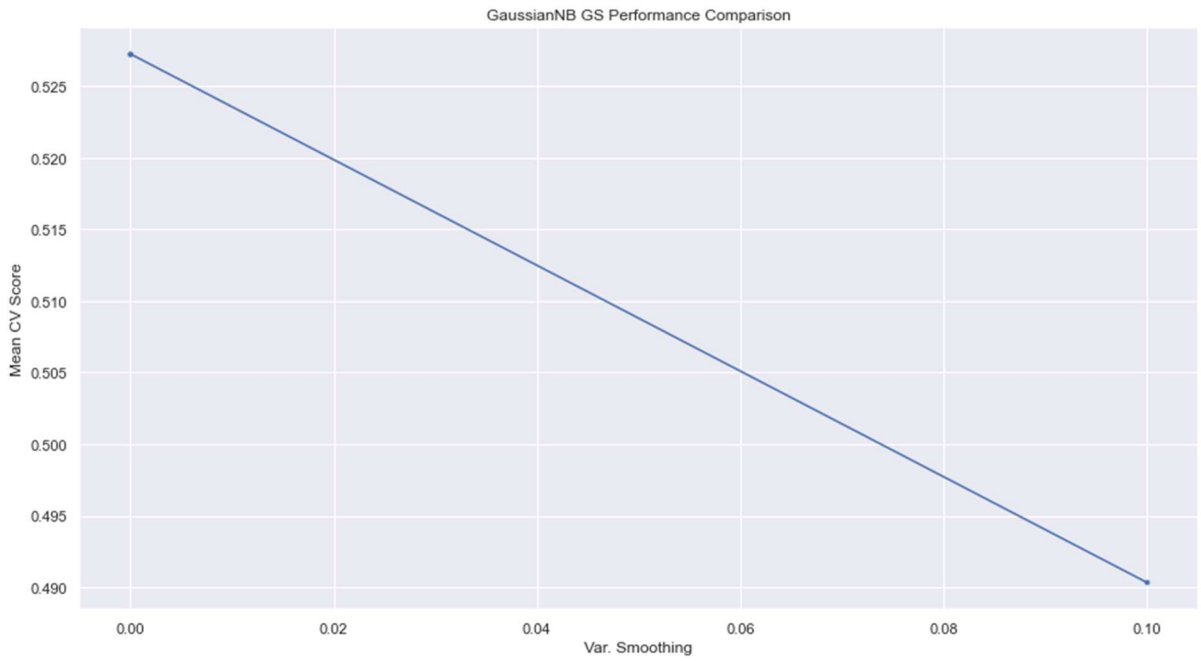

FIGURE 56 VAR SMOOTHING

Naïve Bayes Classifier GridSearchCV Classification Report on Train Dataset

```
Classification Report
              precision    recall  f1-score   support

           0       0.90      0.93      0.91      7471
           1       0.59      0.51      0.55      1537

    accuracy                           0.86      9008
   macro avg       0.75      0.72      0.73      9008
weighted avg       0.85      0.86      0.85      9008
```

TABLE 126 NAÏVE BAYES CLASSIFIER GRIDSEARCHCV CLASSIFICATION REPORT

Naïve Bayes Classifier GridSearchCV Confusion matrix with the % on train

```
Confusion Matrix
 [[6921  550]
 [ 748  789]]
```

TABLE 127 NAÏVE BAYES CLASSIFIER GRIDSEARCHCV CONFUSION MATRIX

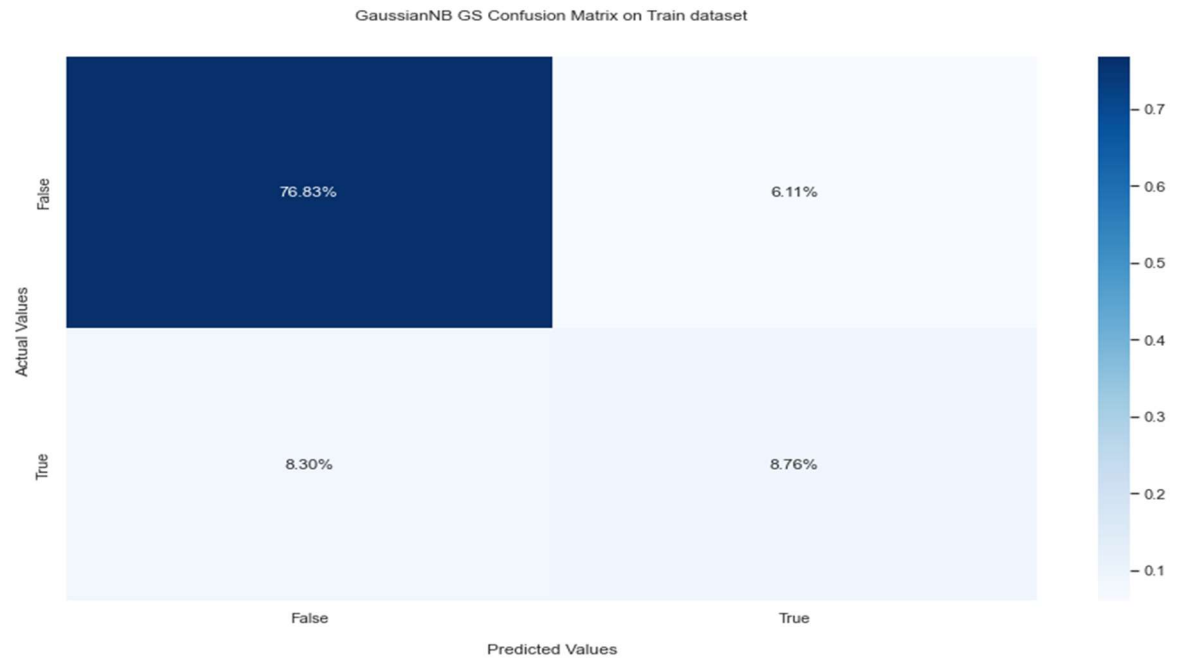

FIGURE 57 CONFUSION MATRIX PERCENTAGE

Naïve Bayes Classifier GridSearchCV Classification Report on Test dataset

```
Classification Report
               precision    recall  f1-score   support

           0       0.91      0.90      0.91      2809
           1       0.54      0.58      0.56       569

    accuracy                           0.85      3378
   macro avg       0.73      0.74      0.73      3378
weighted avg       0.85      0.85      0.85      3378
```

TABLE 128 NAÏVE BAYES CLASSIFIER GRIDSEARCHCV CLASSIFICATION REPORT

Naïve Bayes Classifier GridSearchCV Confusion matrix with the % on test

```
Confusion Matrix
 [[2532  277]
 [ 241  328]]
```

TABLE 129 NAÏVE BAYES CLASSIFIER GRIDSEARCHCV CONFUSION MATRIX

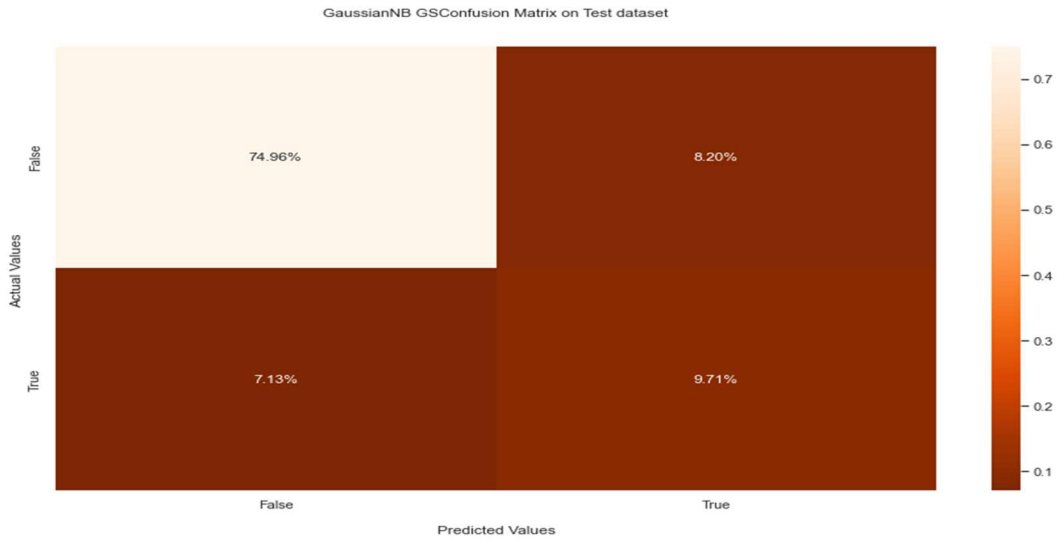

FIGURE 58 CONFUSION MATRIX PERCENTAGE

Naïve Bayes Classifier GridSearchCV AUC Score and curve on Train and Test DataSet

```
AUC for the Training Data: 0.827
AUC for the Test Data:     0.837
```

TABLE 130 NAÏVE BAYES CLASSIFIER GRIDSEARCHCV AUC SCORE

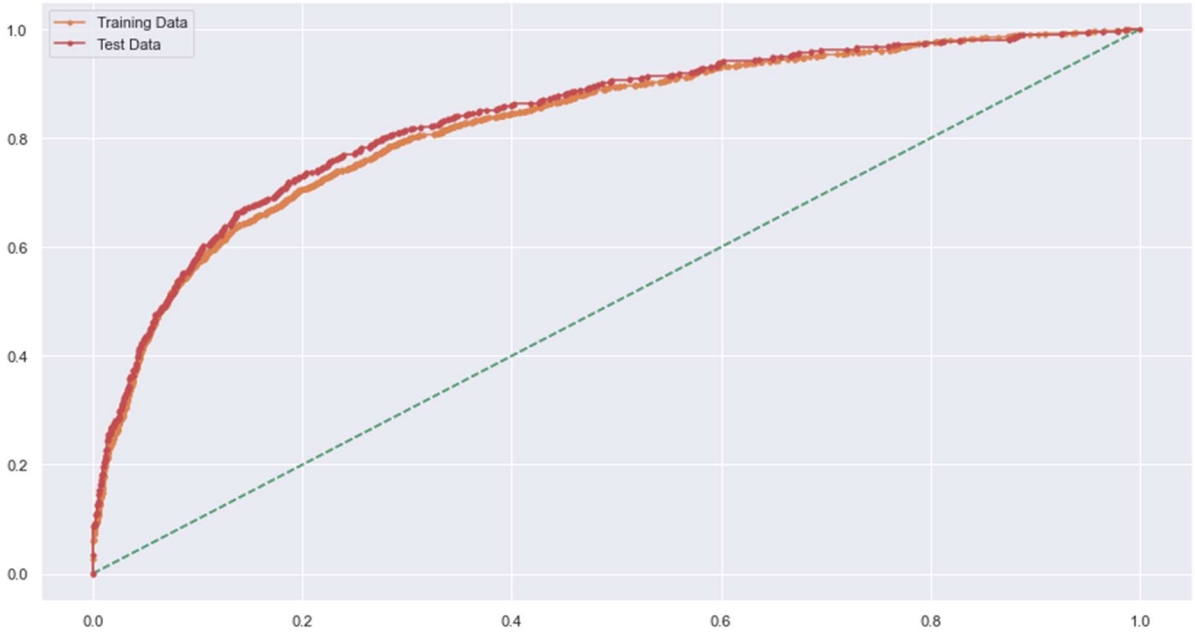

FIGURE 59 RO CURVE

Accuracy of Naïve Bayes Classifier GridSearchCV

```
Accuracy of GridSearchCV  GaussianNB on train set: 0.856
Accuracy of GridSearchCV  GaussianNB on test set:  0.855
```

TABLE 131 ACCURACY OF NAÏVE BAYES CLASSIFIER GRIDSEARCHCV

Building Naïve Bayes Classifier SMOTE

GaussianNB

```
GaussianNB()
```

TABLE 132 NAÏVE BAYES CLASSIFIER SMOTE

Accuracy of Naïve Bayes Classifier SMOTE

```
Accuracy of GaussianNB Resampled on train set: 0.853
Accuracy of GaussianNB Resampled on test set:  0.734
```

TABLE 133 ACCURACY OF NAÏVE BAYES CLASSIFIER SMOTE

Ensemble Techniques

The ensemble methods in machine learning combine the insights obtained from multiple learning models to facilitate accurate and improved decisions. These methods follow the same principle as the example of buying an air-conditioner cited above. There are mainly Two Types of Advance Ensemblling methods

1.**Bagging (Bootstrap Aggregating):** The primary goal of "bagging" or "bootstrap aggregating" ensemble method is to minimize variance errors in decision trees.

2.**Boosting:** An iterative ensemble technique, "boosting," adjusts an observation's weight based on its last classification.

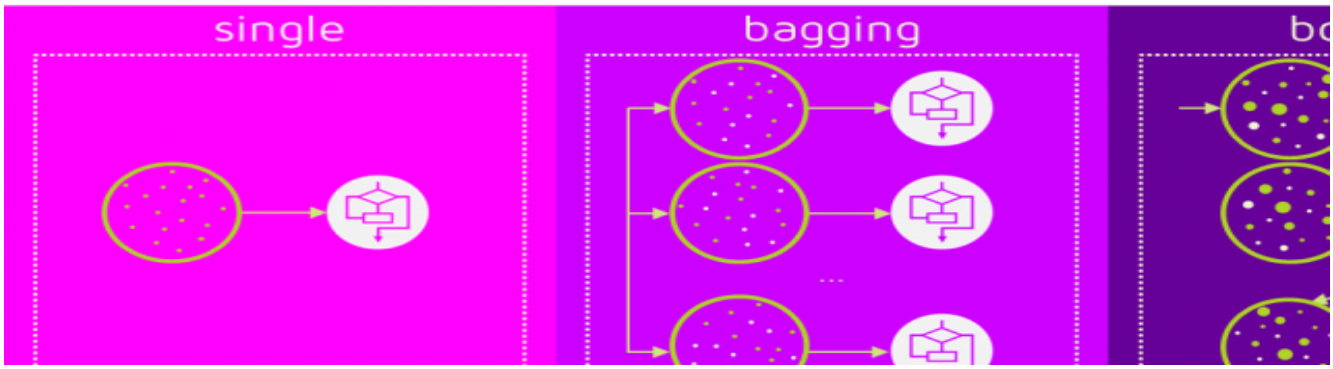

FIGURE 60 WORKING OF TECHNIQUES

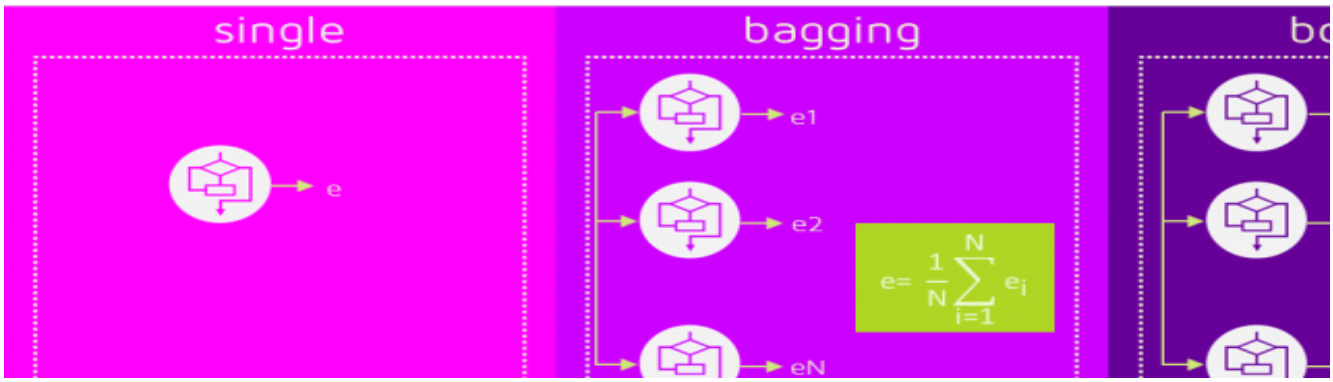

FIGURE 61 EXPRESSING OF TECHNIQUES

Classifier No 7: Bagging

Bagging performs best with algorithms that have high variance. A popular example are decision trees, often constructed without pruning.

A **Bagging classifier** is an ensemble meta-estimator that fits base classifiers each on random subsets of the original dataset and then aggregate their individual predictions (either by voting or by averaging) to form a final prediction. Such a meta-estimator can typically be used as a way to reduce the variance of a black-box estimator (e.g., a decision tree), by introducing randomization into its construction procedure and then making an ensemble out of it.

Bagging is a special case of the model averaging approach; in case of regression problem, we take mean of the output and in case of classification we take the majority vote.

Building Random Forest Classifier for Bagging

Bagging, short for bootstrap aggregating, is an ensemble method that combines multiple models trained on different subsets of the training data. The RandomForestClassifier is a popular algorithm that uses bagging to create an ensemble of decision tree classifiers.

RandomForestClassifier

```
RandomForestClassifier()
```

TABLE 134 RANDOMFORESTCLASSIFIER

Forest Classifier (Bagging) Feature Importance

	Imp
Tenure	0.23
cashback	0.08
CC_Contacted_LY	0.07
Complain_ly	0.07
Day_Since_CC_connect	0.06
rev_growth_yoy	0.06
CC_Agent_Score	0.06
rev_per_month	0.05
AccountID	0.05
account_segment	0.04
Account_user_count	0.04
coupon_used_for_payment	0.03
City_Tier	0.03
Marital_Status_Single	0.03
Gender_Male	0.02
Login_device_Mobile	0.02
Service_Score	0.02
Payment_Credit Card	0.02
Payment_Debit Card	0.01
Marital_Status_Married	0.01
Payment_E wallet	0.01
Payment_UPI	0.01
Login_device_Unknown	0.00

TABLE 135 FOREST CLASSIFIER (BAGGING) FEATURE IMPORTANCE

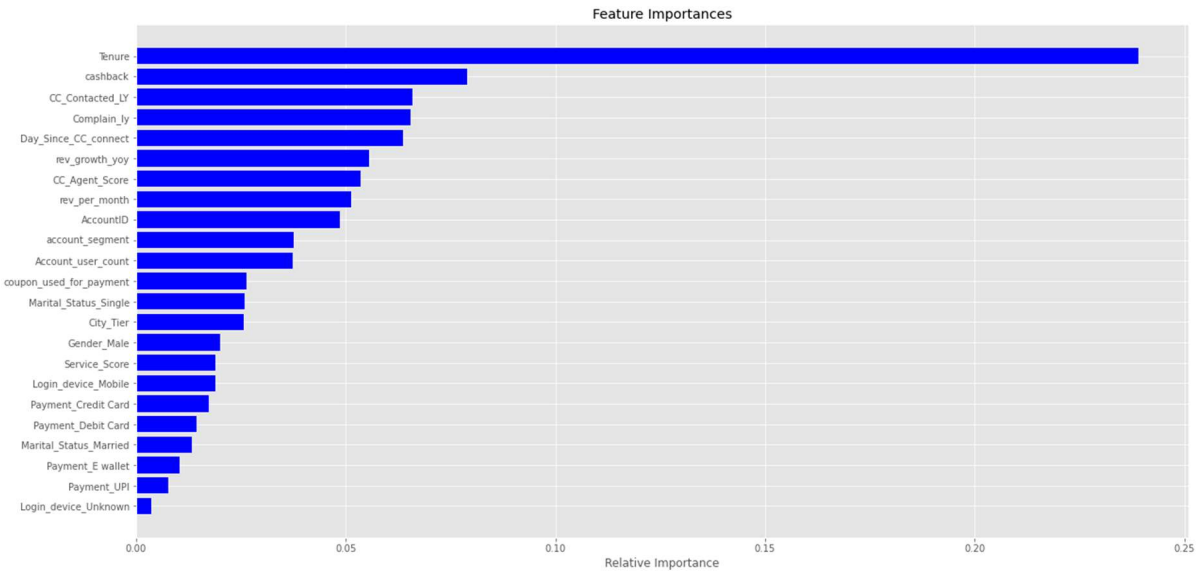

FIGURE 62 FEATURE IMPORTANCE

Interpretations of Feature Importance

To interpret the feature importance in terms of churning, we can analyse the percentage importance of each feature insights on how they relate to the likelihood of churn. Here's an interpretation based on the provided percentage importance values:

Tenure: This feature has the highest importance with 24.00%. It suggests that customers with longer tenure are less likely to churn. Longer tenure could indicate customer loyalty, satisfaction, or a higher level of engagement.

Cashback: With 8.00% importance, this feature suggests that offering cashback incentives may contribute to reducing churn. Cashback programs can incentivize customers to stay and engage with the company's offerings.

CC_Contacted_LY: The 7.00% importance of this feature implies that frequent customer care interactions in the past year may influence churn. Higher contact frequency might indicate dissatisfaction or issues that need resolution.

Complain_ly: With 6.00% importance, this feature suggests that customers who lodged complaints in the last year might be at a higher risk of churn. Addressing complaints effectively and promptly could help mitigate churn.

Day_Since_CC_connect: This feature, also with 6.00% importance, indicates that more recent customer care interactions may impact churn. Regular touchpoints with customers can help maintain engagement and satisfaction.

Rev_growth_yoy: The 6.00% importance of this feature suggests that positive year-on-year revenue growth may contribute to reducing churn. It implies that a thriving business and positive financial performance can attract and retain customers.

CC_Agent_Score: With 5.00% importance, this feature implies that customer care agent performance and satisfaction scores may influence churn. Well-trained and efficient agents can enhance the customer experience and reduce churn.

Rev_per_month: The 5.00% importance of this feature indicates that customers with higher monthly revenues are less likely to churn. It suggests that customers who spend more are more invested in the company's offerings.

These interpretations provide insights into the relationship between each feature and the likelihood of churn. They can guide business decisions and strategies focused on customer retention, such as improving customer care interactions, optimizing cashback programs, addressing complaints, and fostering customer loyalty through personalized engagement and revenue growth.

Forest Classifier (Bagging) Classification Report on Train Dataset

```
Classification Report
              precision    recall  f1-score   support

           0       1.00      1.00      1.00      7471
           1       1.00      1.00      1.00      1537

    accuracy                           1.00      9008
   macro avg       1.00      1.00      1.00      9008
weighted avg       1.00      1.00      1.00      9008
```

TABLE 136 FOREST CLASSIFIER (BAGGING) CLASSIFICATION REPORT

Forest Classifier (Bagging) Confusion matrix with the percentage on train

```
Confusion Matrix
 [[7471    0]
 [   0 1537]]
```

TABLE 137 FOREST CLASSIFIER (BAGGING) CONFUSION MATRIX

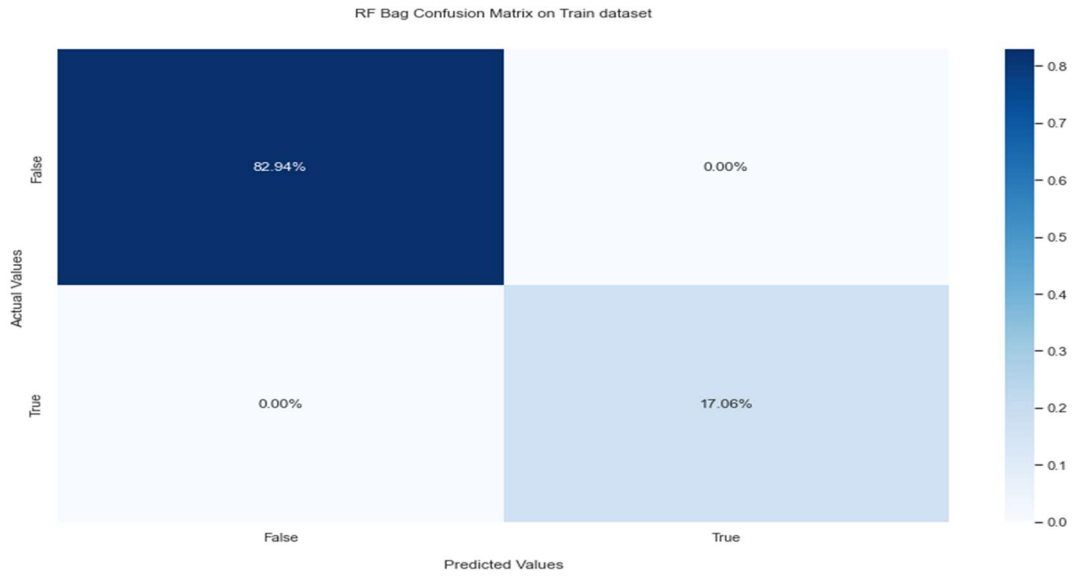

FIGURE 63 CONFUSION MATRIX PERCENTAGE

Forest Classifier (Bagging) Classification Report on Test dataset

```
Classification Report
              precision    recall  f1-score   support

           0       0.99      1.00      1.00      2809
           1       0.99      0.96      0.98       569

    accuracy                           0.99      3378
   macro avg       0.99      0.98      0.99      3378
weighted avg       0.99      0.99      0.99      3378
```

TABLE 138 FOREST CLASSIFIER (BAGGING) CLASSIFICATION REPORT

Forest Classifier (Bagging) Confusion matrix with the percentage on test

```
Confusion Matrix
 [[2806    3]
 [  21  548]]
```

TABLE 139 FOREST CLASSIFIER (BAGGING) CONFUSION MATRIX

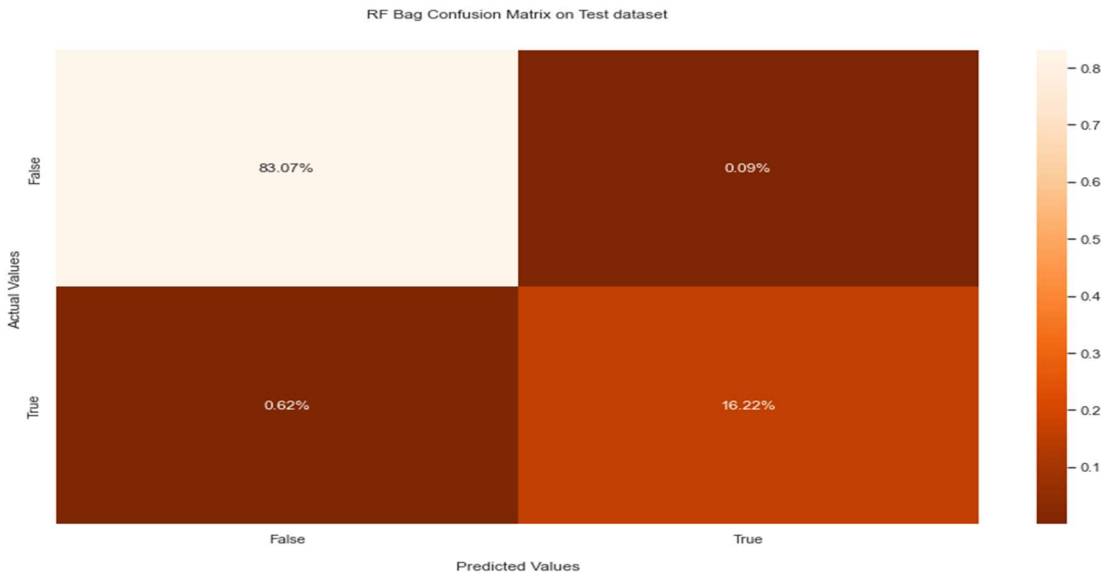

FIGURE 64 CONFUSION MATRIX PERCENTAGE

Forest Classifier (Bagging) AUC Score and curve on Train and Test DataSet

```
AUC for the Training Data: 1.000
AUC for the Test Data:     1.000
```

TABLE 140 FOREST CLASSIFIER (BAGGING) AUC SCORE

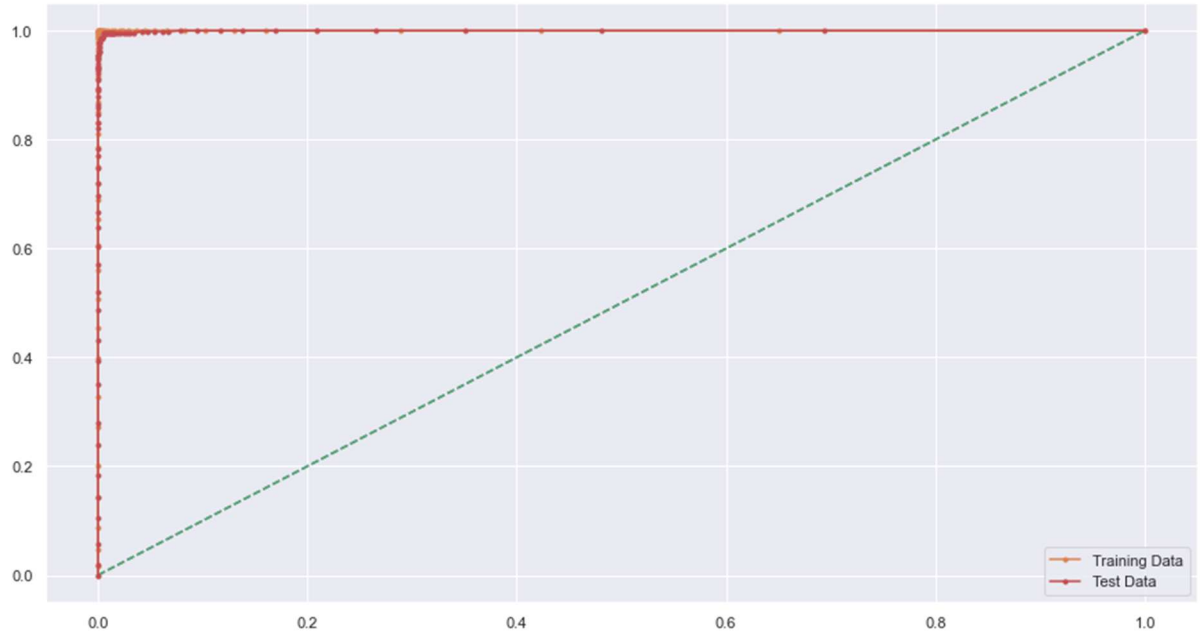

FIGURE 65 RO CURVE

Fivefold K-Fold cross Validation of Forest Classifier (Bagging)

```
[0.9920071047957372, 0.9946714031971581, 0.9931912374185908, 0.99407933
68857312, 0.9946714031971581]
Mean of testing accuracy over 5 folds = 0.99 with std = 0.00
```

TABLE 141 FIVEFOLD K-FOLD CROSS VALIDATION OF FOREST CLASSIFIER (BAGGING)

Forest Classifier (Bagging) Accuracy on Train and Test

```
Accuracy of Bagging on train set: 1.000
Accuracy of Bagging on test set:  0.995
```

TABLE 142 FOREST CLASSIFIER (BAGGING) ACCURACY

Insights on RF Bagging of Test data:

On the basis of Performing the bagging on base of random forest classifier the following insights has been observed

True Positive (TP): 2806 This represents the number of instances that are actually positive (class 1) and are correctly predicted as positive by the model.

False Positive (FP): 3 This indicates the number of instances that are actually negative (class 0) but are incorrectly predicted as positive by the model.

False Negative (FN): 21 This represents the number of instances that are actually positive (class 1) but are incorrectly predicted as negative by the model.

True Negative (TN): 548 This indicates the number of instances that are actually negative (class 0) and are correctly predicted as negative by the model.

Classification Report: Precision, Recall, F1-score, Support, and Accuracy metrics are provided for each class (0 and 1), as well as macro and weighted averages.

Precision: Precision measures the model's ability to correctly identify positive instances out of the total instances predicted as positive. For class 0, the precision is 0.99, indicating that 99% of the instances predicted as class 0 are correct. For class 1, the precision is also 0.99, meaning that 99% of the instances predicted as class 1 are correct.

Recall: Recall (also known as sensitivity or true positive rate) measures the model's ability to correctly identify positive instances out of the actual positive instances. For class 0, the recall is 1.00, indicating that the model correctly identifies all instances of class 0. For class 1, the recall is 0.96, meaning that the model identifies 96% of the instances of class 1.

F1-score: The F1-score is the harmonic mean of precision and recall. It provides a balance between precision and recall. For class 0, the F1-score is 1.00, indicating excellent performance. For class 1, the F1-score is 0.98, indicating very good performance.

Support: Support represents the number of instances in each class. For class 0, the support is 2809, and for class 1, the support is 569.

Accuracy: The overall accuracy of the model is 0.99, meaning that it correctly predicts 99% of the instances.

Business Insights:

- The confusion matrix and classification report:
- The model shows high accuracy and performs well in classifying both classes (0 &1).
- The model has high precision for both classes, indicating a low false positive rate.
- The recall for class 0 is perfect (1.00), indicating that the model correctly identifies all instances of class 0. The recall for class 1 is also high (0.96), indicating a good ability to identify positive instances.
- The F1-scores for both classes are high, indicating a good balance between precision and recall. The number of false negatives (15) is relatively low, suggesting that the model is effective in identifying positive instances correctly. Class 0 (negative instances) has a slightly larger support compared to class 1 (positive instances).
- Overall, the model shows strong performance in classifying both positive and negative instances. The high precision, recall, and F1-scores indicate that the model is reliable in distinguishing between the two classes

Boosting

1.Boosting ensemble algorithms creates a sequence of models that attempt to correct the mistakes of the models before them in the sequence.
2.Machine learning models are used one after the other and the predictions made by first layer models are used as input to next layer models. The last layer of models will use the predictions from all previous layers to get the final predictions. 3.Boosting is more helpful if we have biased base models.
4.Boosting can be used to solve regression and classification problems. Here we have to solve the classification problem.

We will construct an AdaBoost model and Gradient Boost model for classification using the AdaBoostClassifier and Gradient Boost Classifier class

Classifier No 8 : AdaBoost Classifier

An AdaBoost classifier is a meta-estimator that begins by fitting a classifier on the original dataset and then fits additional copies of the classifier on the same dataset but where the weights of incorrectly classified instances are adjusted such that subsequent classifiers focus more on difficult cases. AdaBoost was perhaps the first successful boosting ensemble algorithm. It generally works by weighting instances in the dataset by how easy or difficult they are to classify, allowing the algorithm to pay or or less attention to them in the construction of subsequent models.

Building AdaBoost Classifier (Boosting)

AdaBoostClassifier

```
AdaBoostClassifier(n_estimators=30)
```

TABLE 143 ADABOOSTCLASSIFIER

AdaBoost Classifier (Boosting) Feature Importance

	Imp
Tenure	0.23
rev_per_month	0.10
AccountID	0.07
CC_Contacted_LY	0.07
Account_user_count	0.07
CC_Agent_Score	0.07
Day_Since_CC_connect	0.07
Payment_Debit Card	0.03
Login_device_Mobile	0.03
Marital_Status_Single	0.03
cashback	0.03
coupon_used_for_payment	0.03
rev_growth_yoy	0.03
Complain_ly	0.03
account_segment	0.03

```
City_Tier                0.03
Payment_UPI              0.03
Gender_Male              0.00
Marital_Status_Married   0.00
Service_Score            0.00
Login_device_Unknown     0.00
Payment_Credit Card      0.00
Payment_E wallet         0.00
```

TABLE 144 ADABOOST CLASSIFIER (BOOSTING) FEATURE IMPORTANCE

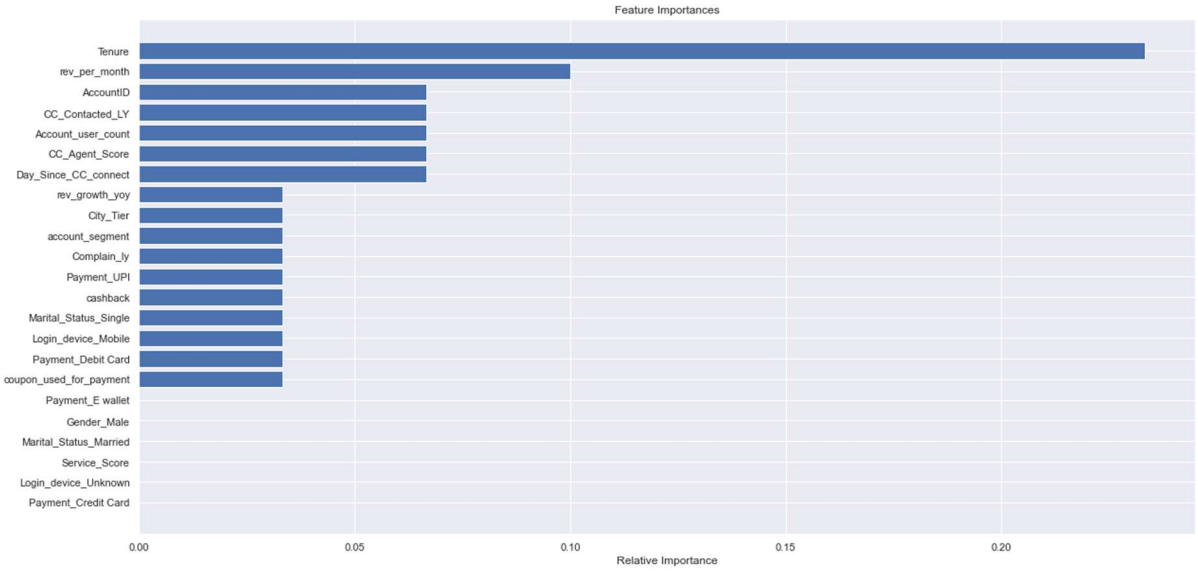

FIGURE 66 FEATURE IMPORTANCE

AdaBoost Classifier (Boosting) Classification Report on Train Dataset

```
Classification Report
              precision    recall  f1-score   support

           0       0.92      0.96      0.94      7471
           1       0.74      0.58      0.65      1537

    accuracy                           0.89      9008
   macro avg       0.83      0.77      0.79      9008
weighted avg       0.89      0.89      0.89      9008
```

TABLE 145 ADABOOST CLASSIFIER (BOOSTING) CLASSIFICATION REPORT

AdaBoost Classifier (Boosting) Confusion matrix with the percentage on train

```
Confusion Matrix
 [[7162  309]
 [ 650  887]]
```

TABLE 146 ADABOOST CLASSIFIER (BOOSTING) CONFUSION MATRIX

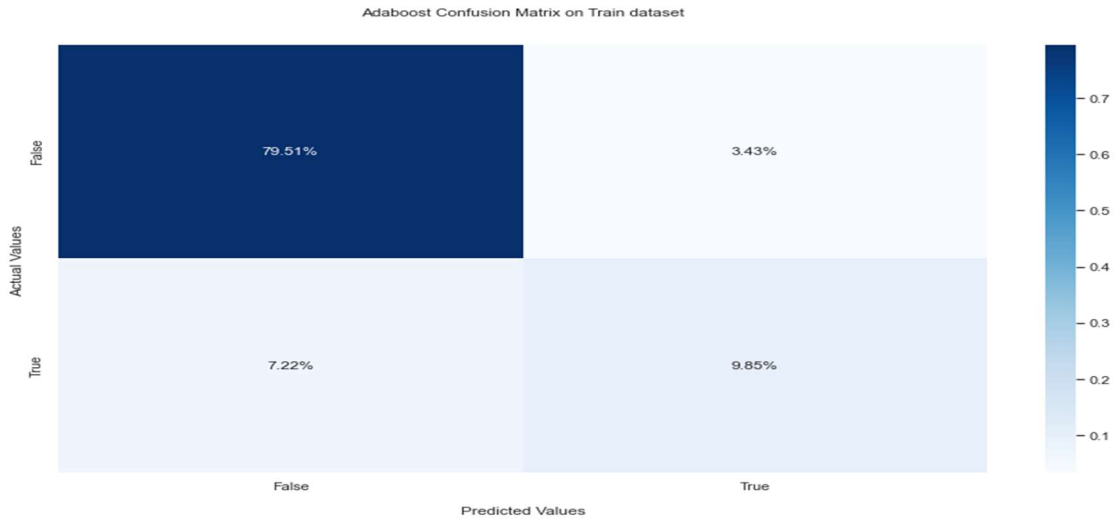

FIGURE 67 CONFUSION MATRIX PERCENTAGE

AdaBoost Classifier (Boosting) Classification Report on Test dataset

```
Classification Report
              precision    recall  f1-score   support

           0       0.92      0.95      0.94      2809
           1       0.72      0.59      0.65       569

    accuracy                           0.89      3378
   macro avg       0.82      0.77      0.79      3378
weighted avg       0.89      0.89      0.89      3378
```

TABLE 147 ADABOOST CLASSIFIER (BOOSTING) CLASSIFICATION REPORT

AdaBoost Classifier (Boosting) Confusion matrix with the percentage on test

```
Confusion Matrix
 [[2679  130]
 [ 232  337]]
```

TABLE 148 ADABOOST CLASSIFIER (BOOSTING) CONFUSION MATRIX

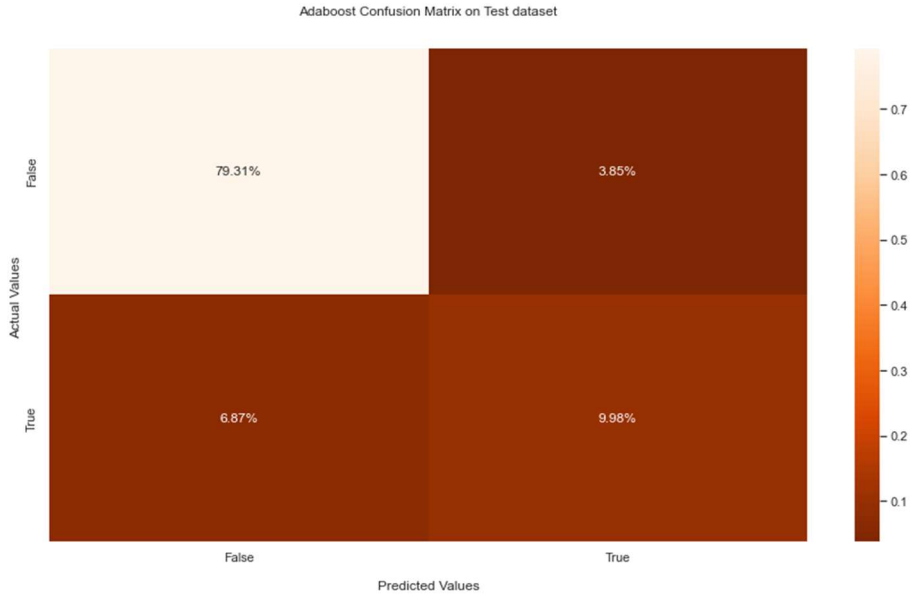

FIGURE 68 CONFUSION MATRIX PERCENTAGE

AdaBoost Classifier (Boosting) AUC Score and curve on Train and Test DataSet

```
AUC for the Training Data: 0.914
AUC for the Test Data:     0.917
```

TABLE 149 ADABOOST CLASSIFIER (BOOSTING) AUC SCORE

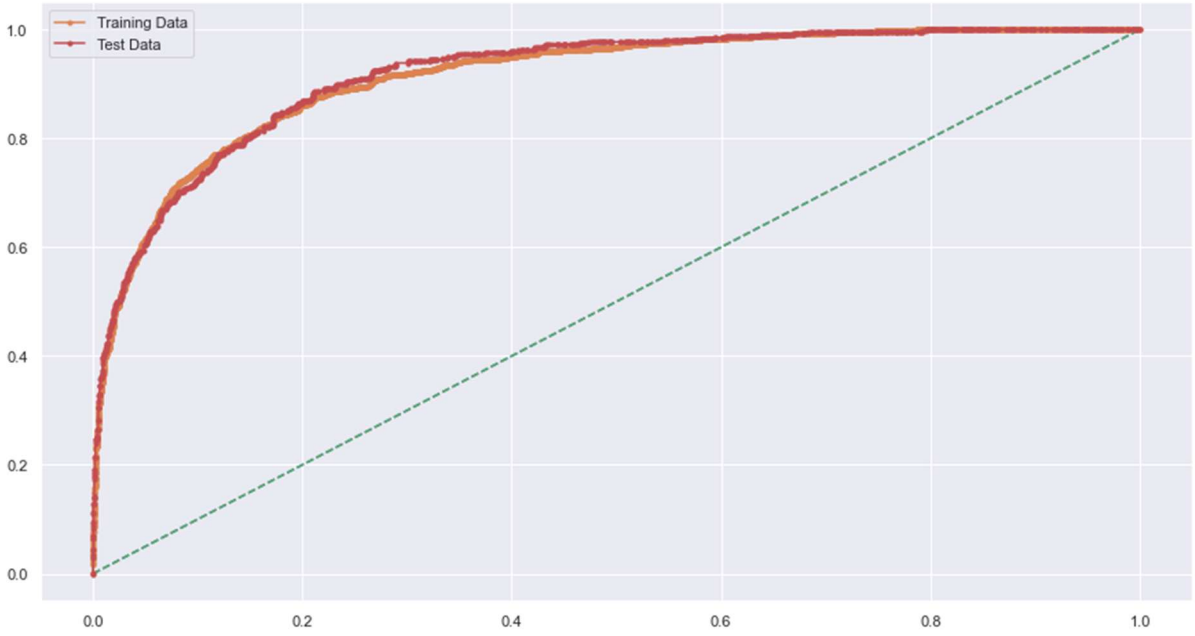

FIGURE 69 RO CURVE

Fivefold K-Fold cross Validation of AdaBoost Classifier (Boosting)

```
[0.8984606275902901, 0.8949082297217288, 0.8984606275902901, 0.90112492
59917111, 0.8928359976317347]
Mean of testing accuracy over 5 folds = 0.90 with std = 0.00
```

TABLE 150 FIVEFOLD K-FOLD CROSS VALIDATION OF ADABOOST CLASSIFIER

AdaBoost Classifier (Boosting) Accuracy on Train and Test

```
Accuracy of Addaboost Classifier on train set: 0.89
Accuracy of Addaboostt Classifier  on test set: 0.89
```

TABLE 151 ADABOOST CLASSIFIER (BOOSTING) ACCURACY

Building AdaBoost Classifier (Boosting) GS

```
Best Grid  {'algorithm': 'SAMME.R',
 'learning_rate': 1.0,
 'n_estimators': 100,
 'random_state': 0}
```

TABLE 152 ADABOOST CLASSIFIER BEST GRID

AdaBoostClassifier

```
AdaBoostClassifier(n_estimators=100, random_state=0)
```

TABLE 153 ADABOOSTCLASSIFIER

AdaBoost Classifier (Boosting) GS Feature Importance

	Imp
cashback	0.22
Tenure	0.18
Day_Since_CC_connect	0.10
AccountID	0.09
rev_growth_yoy	0.08
rev_per_month	0.05
Account_user_count	0.04
Payment_Debit Card	0.03
CC_Contacted_LY	0.03
Complain_ly	0.03
Payment_Credit Card	0.02
Payment_UPI	0.02
CC_Agent_Score	0.02
account_segment	0.02
Gender_Male	0.01
Marital_Status_Married	0.01
Marital_Status_Single	0.01
Login_device_Mobile	0.01
Service_Score	0.01
City_Tier	0.01
coupon_used_for_payment	0.01
Login_device_Unknown	0.00
Payment_E wallet	0.00

TABLE 154 ADABOOST CLASSIFIER (BOOSTING) GS FEATURE IMPORTANCE

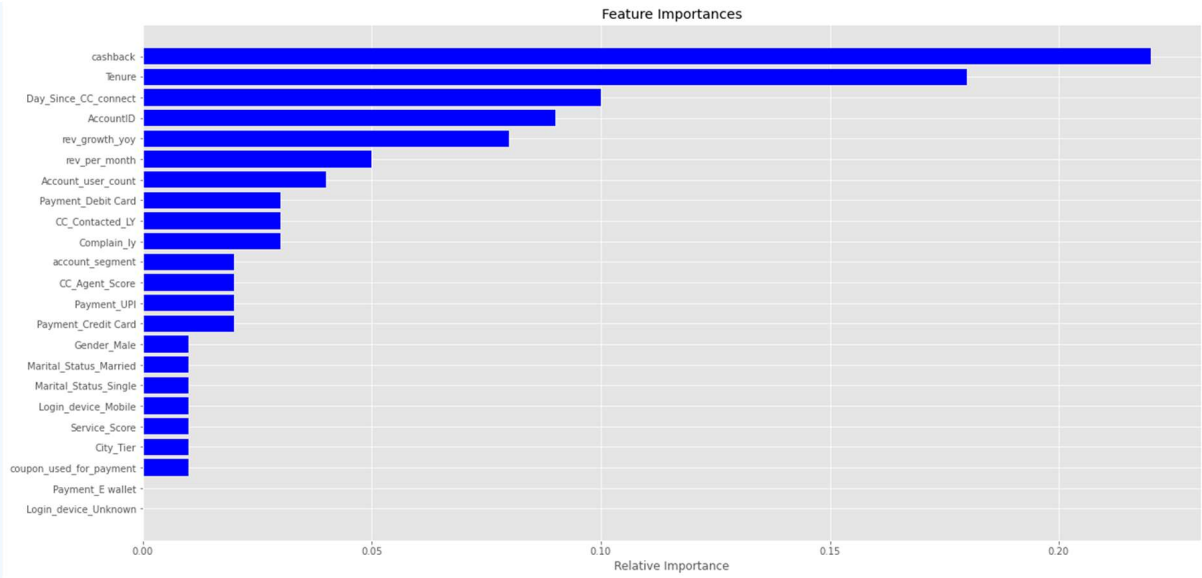

FIGURE 70 FEATURE IMPORTANCE

AdaBoost Classifier (Boosting) GS Classification Report on Train Dataset

```
Classification Report
              precision    recall  f1-score   support

           0       0.92      0.96      0.94      7471
           1       0.75      0.60      0.66      1537

    accuracy                           0.90      9008
   macro avg       0.83      0.78      0.80      9008
weighted avg       0.89      0.90      0.89      9008
```

TABLE 155 ADABOOST CLASSIFIER (BOOSTING) GS CLASSIFICATION REPORT

AdaBoost Classifier (Boosting) Classification Report on Test dataset

```
Classification Report
              precision    recall  f1-score   support

           0       0.92      0.96      0.94      2809
           1       0.74      0.61      0.67       569

    accuracy                           0.90      3378
   macro avg       0.83      0.79      0.81      3378
weighted avg       0.89      0.90      0.90      3378
```

TABLE 156 ADABOOST CLASSIFIER (BOOSTING) CLASSIFICATION REPORT

AdaBoost Classifier (Boosting) Confusion matrix with the percentage on test

```
Confusion Matrix
 [[2689  120]
 [ 220  349]]
```

TABLE 157 ADABOOST CLASSIFIER CONFUSION MATRIX

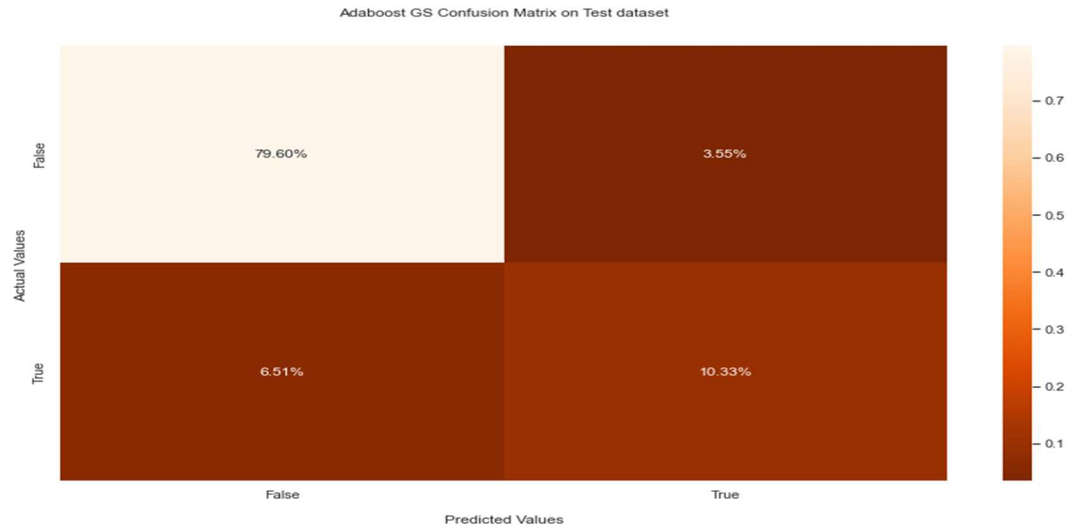

FIGURE 71 CONFUSION MATRIX PERCENTAGE

Fivefold K-Fold cross Validation AdaBoost Classifier (Boosting)

```
[0.9020130254588514, 0.9014209591474245, 0.9026050917702783, 0.90438129
07045589, 0.8993487270574304]
Mean of testing accuracy over 5 folds = 0.90 with std = 0.00
```

TABLE 158 FIVEFOLD K-FOLD CROSS VALIDATION ADABOOST CLASSIFIER

Accuracy of AdaBoost Classifier (Boosting) GS on Train and Test dataset

```
Accuracy of GridSearchCV Addaboost Classifier on train set: 0.90
Accuracy of GridSearchCV Addaboost Classifier  on test set: 0.90
```

TABLE 159 ACCURACY OF ADABOOST CLASSIFIER GS

Classifier No 9 : Gradient Boost Classifier:

Gradient Boosting is an analytical technique that is designed to explore the relationship between two or more variables (X,and y)
Its analytical output identifies important factors (Xi) impacting the dependent variable (y) and the nature of the relationship between each of these factors and the dependent variable.

The main differences, the Gradient Boosting is a generic algorithm to find approximate solutions to the additive modelling problem, while AdaBoost can be seen as a special case with a particular loss function. Hence, Gradient Boosting is much more flexible.

Building Gradient Boost Classifier

GradientBoostingClassifier

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

TABLE 160 GRADIENTBOOSTINGCLASSIFIER

Gradient Boost Classifier Feature Importance

	Imp
Tenure	0.53
Complain_ly	0.11
CC_Agent_Score	0.06
rev_per_month	0.05
Day_Since_CC_connect	0.05
Marital_Status_Single	0.03
Account_user_count	0.03
account_segment	0.03
AccountID	0.02
CC_Contacted_LY	0.02
City_Tier	0.01
Payment_E wallet	0.01
cashback	0.01
coupon_used_for_payment	0.01
rev_growth_yoy	0.01
Payment_Credit Card	0.01
Login_device_Mobile	0.01
Gender_Male	0.00
Marital_Status_Married	0.00
Service_Score	0.00
Payment_Debit Card	0.00
Payment_UPI	0.00
Login_device_Unknown	0.00

TABLE 161 GRADIENT BOOST CLASSIFIER FEATURE IMPORTANCE

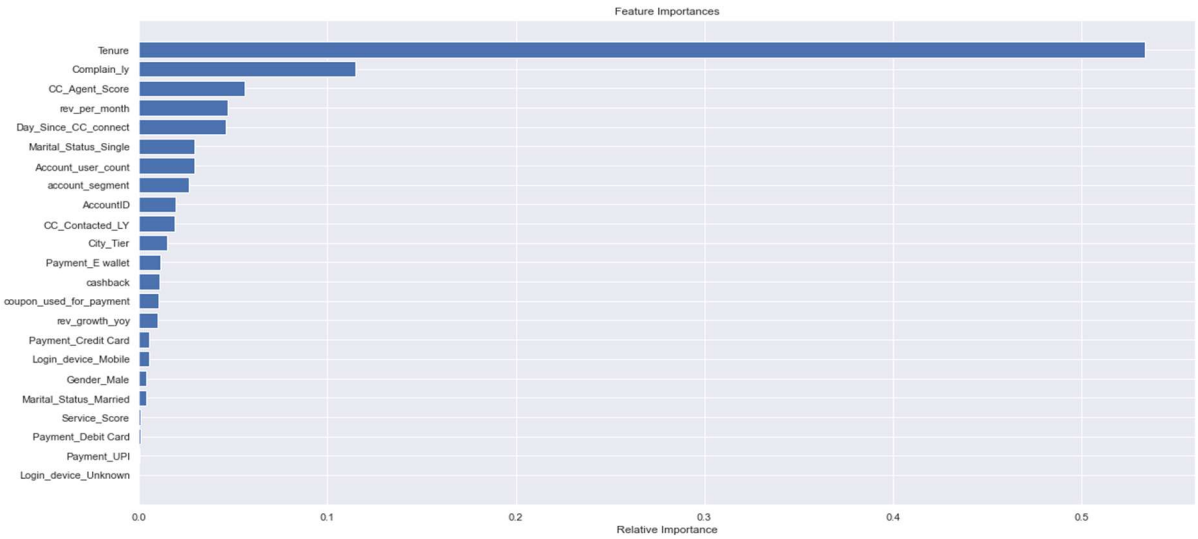

FIGURE 72 FEATURE IMPORTANCE

Gradient Boost Classifier Classification Report on Train Dataset

	precision	recall	f1-score	support
0	0.93	0.97	0.95	7471
1	0.84	0.64	0.73	1537
accuracy			0.92	9008
macro avg	0.88	0.81	0.84	9008
weighted avg	0.91	0.92	0.91	9008

TABLE 162 GRADIENT BOOST CLASSIFIER CLASSIFICATION REPORT

Gradient Boost Classifier Classification Report on Test dataset

```
Classification Report
              precision    recall  f1-score   support

           0       0.93      0.97      0.95      2809
           1       0.83      0.64      0.72       569

    accuracy                           0.92      3378
   macro avg       0.88      0.81      0.84      3378
weighted avg       0.91      0.92      0.91      3378
```

TABLE 163 GRADIENT BOOST CLASSIFIER CLASSIFICATION REPORT

Gradient Boost Classifier Confusion matrix with the percentage on test

```
Confusion Matrix
 [[2734   75]
 [ 206  363]]
```

TABLE 164 GRADIENT BOOST CLASSIFIER CONFUSION MATRIX

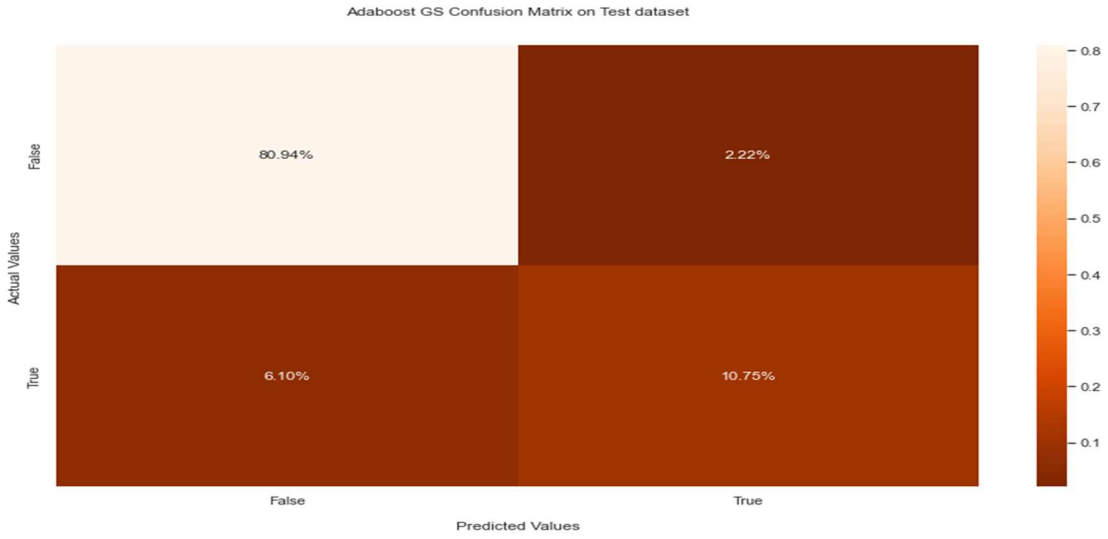

FIGURE 73 CONFUSION MATRIX PERCENTAGE

Fivefold K-Fold cross Validation of Gradient Boost Classifier

```
[0.9200710479573713, 0.9153345174659562, 0.9162226169330965, 0.9150384843102427, 0.9168146832445234]
Mean of testing accuracy over 5 folds = 0.92 with std = 0.00
```

TABLE 165 FIVEFOLD K-FOLD CROSS VALIDATION OF GRADIENT BOOST CLASSIFIER

Gradient Boost Classifier Accuracy on Train and Test dataset

```
Accuracy of Gradient boost Classifier on train set: 0.92
Accuracy of Gradient boost Classifier on test set:  0.92
```

TABLE 166 GRADIENT BOOST CLASSIFIER ACCURACY

Building Gradient Boost Classifier GS

GradientBoostingClassifier

```
GradientBoostingClassifier(learning_rate=1.0, loss='deviance', n_estim
ators=200)
```

TABLE 167 GRADIENT BOOST CLASSIFIER GS

Gradient Boost Classifier GS Feature Importance

	Imp
Tenure	0.46
Complain_ly	0.10
cashback	0.08
AccountID	0.05
Day_Since_CC_connect	0.05
CC_Agent_Score	0.04
rev_per_month	0.03
CC_Contacted_LY	0.03
account_segment	0.03
Account_user_count	0.02
rev_growth_yoy	0.02
Marital_Status_Single	0.02
City_Tier	0.01
Login_device_Mobile	0.01
Payment_E wallet	0.01
coupon_used_for_payment	0.01
Gender_Male	0.01
Payment_Credit Card	0.00
Service_Score	0.00
Payment_Debit Card	0.00
Marital_Status_Married	0.00
Payment_UPI	0.00
Login_device_Unknown	0.00

TABLE 168 GRADIENT BOOST CLASSIFIER GS FEATURE IMPORTANCE

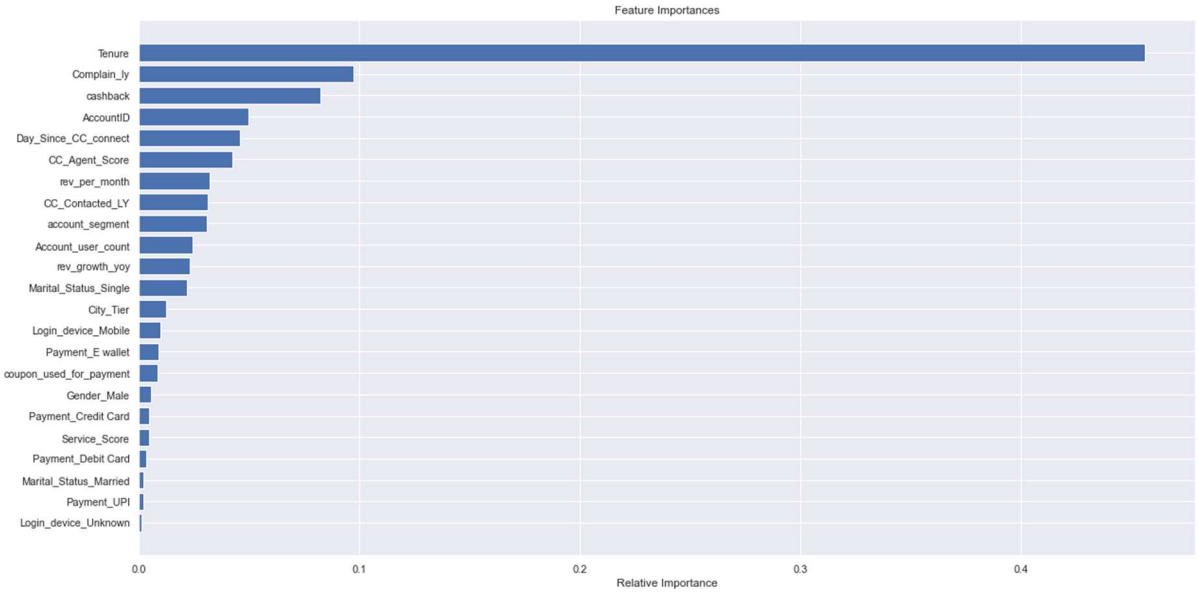

FIGURE 74 FEATURE IMPORTANCE

Gradient Boost Classifier GS Classification Report on Train Dataset

```
Classification Report
              precision    recall  f1-score   support

           0       1.00      1.00      1.00      7471
           1       1.00      0.99      0.99      1537

    accuracy                           1.00      9008
   macro avg       1.00      1.00      1.00      9008
weighted avg       1.00      1.00      1.00      9008
```

TABLE 169 GRADIENT BOOST CLASSIFIER GS CLASSIFICATION REPORT

Gradient Boost Classifier GS Classification Report on Test dataset

```
Classification Report
              precision    recall  f1-score   support

           0       0.99      1.00      0.99      2809
           1       0.97      0.95      0.96       569

    accuracy                           0.99      3378
   macro avg       0.98      0.97      0.98      3378
weighted avg       0.99      0.99      0.99      3378
```

TABLE 170 GRADIENT BOOST CLASSIFIER GS CLASSIFICATION REPORT

Gradient Boost Classifier GS Confusion matrix with the percentage on test

```
Confusion Matrix
 [[2795   14]
 [  26  543]]
```

TABLE 171 GRADIENT BOOST CLASSIFIER GS CONFUSION MATRIX

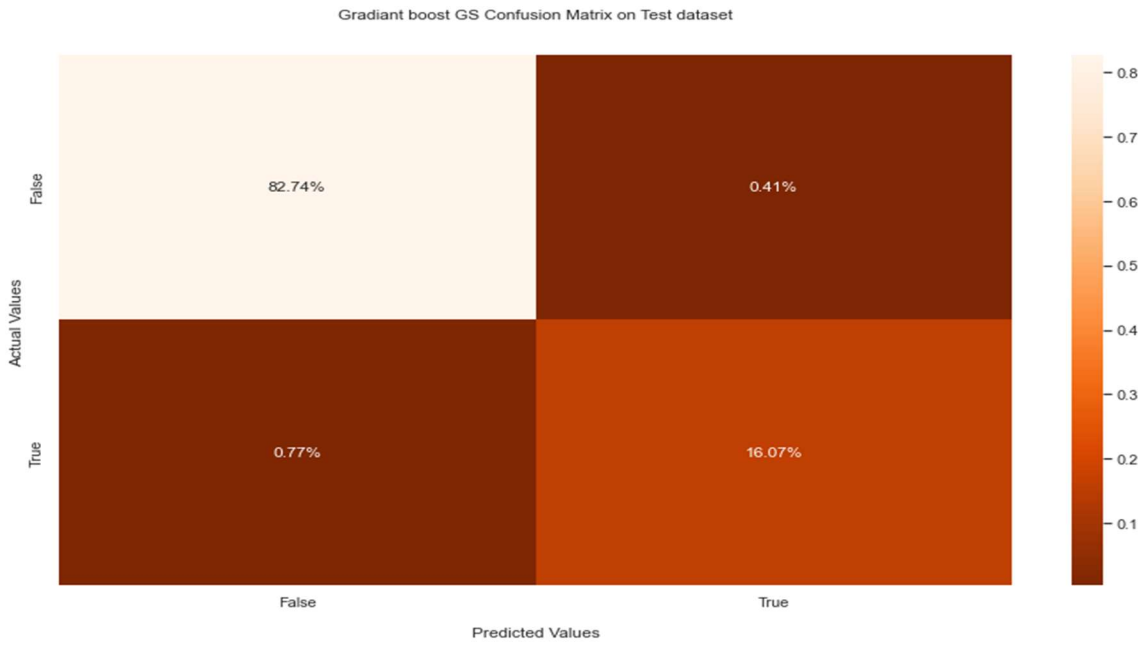

FIGURE 75 CONFUSION MATRIX PERCENTAGE

Fivefold K-Fold cross Validation of Gradient Boost Classifier GS

```
[0.9854943753700415, 0.9869745411486086, 0.9887507400828893, 0.98727057
43043221, 0.9881586737714624]
Mean of testing accuracy over 5 folds = 0.99 with std = 0.00
```

TABLE 172 FIVEFOLD K-FOLD CROSS VALIDATION OF GRADIENT BOOST CLASSIFIER GS

Gradient Boost Classifier GS Accuracy on Train and Test dataset

```
Accuracy of grid search Gradient boost Classifier on train set: 1.00
Accuracy of grid search Gradient boost Classifier  on test set: 0.99
```

TABLE 173 GRADIENT BOOST CLASSIFIER GS ACCURACY

Gradient Boost Classifier RS Accuracy on Train and Test dataset

```
Accuracy of Resampled Gradient boost Classifier on train set: 0.92
Accuracy of Resampled Gradient boost Classifier  on test set: 0.79
```

TABLE 174 GRADIENT BOOST CLASSIFIER RS ACCURACY

Addaboost Boost Classifier RS Accuracy on Train and Test dataset

```
Accuracy of Resampled Addaboost Classifier on train set: 0.90
Accuracy of Resampled Addaboost Classifier  on test set: 0.77
```

TABLE 175 ADDABOOST BOOST CLASSIFIER RS ACCURACY

Building Random Forest Classifier (Bagging) RS

RandomForestClassifier

```
RandomForestClassifier(n_estimators=1000, random_state=1)
```

TABLE 176 RANDOM FOREST CLASSIFIER (BAGGING) RS

Random forest Classifier (Bagging) RS Classification Report on Train Dataset

```
Classification Report
              precision    recall  f1-score   support

           0       1.00      1.00      1.00      7471
           1       1.00      1.00      1.00      1537

    accuracy                           1.00      9008
   macro avg       1.00      1.00      1.00      9008
weighted avg       1.00      1.00      1.00      9008
```

TABLE 177 RANDOM FOREST CLASSIFIER (BAGGING) RS CLASSIFICATION REPORT

Random forest Classifier (Bagging) RS Confusion matrix with the % on train

```
Confusion Matrix
 [[7471    0]
 [   0 1537]]
```

TABLE 178 RANDOM FOREST CLASSIFIER (BAGGING) RS CONFUSION MATRIX

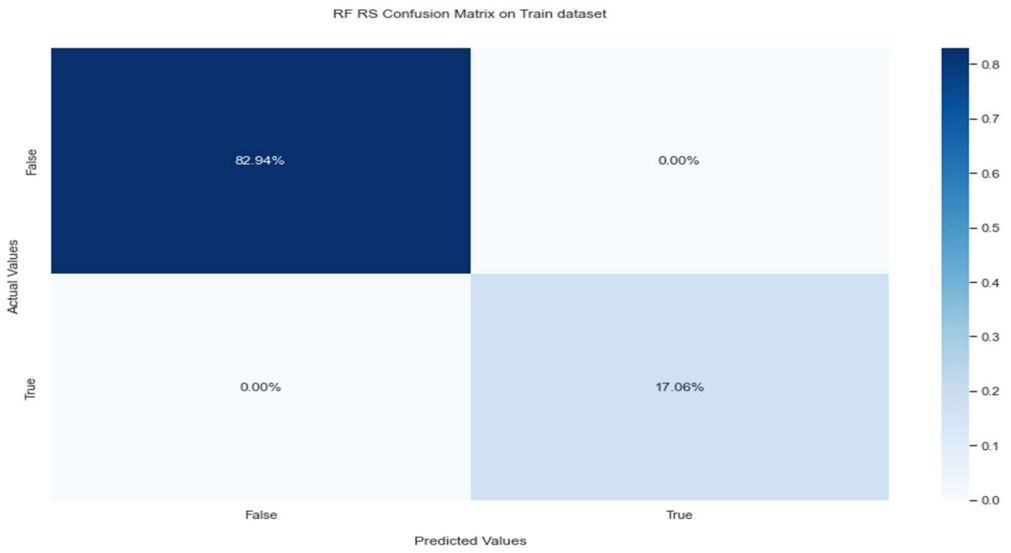

FIGURE 76 CONFUSION MATRIX PERCENTAGE

Random forest Classifier (Bagging) RS Classification Report on Test dataset

```
Classification Report
              precision    recall  f1-score   support

           0       0.93      1.00      0.96      2810
           1       1.00      0.93      0.96      2809

    accuracy                           0.96      5619
   macro avg       0.96      0.96      0.96      5619
weighted avg       0.96      0.96      0.96      5619
```

TABLE 179 RANDOM FOREST CLASSIFIER (BAGGING) RS CLASSIFICATION REPORT

Random forest Classifier (Bagging) RS Confusion matrix with the % on test

```
Confusion Matrix
 [[2803    7]
 [ 209 2600]]
```

TABLE 180 RANDOM FOREST CLASSIFIER (BAGGING) RS CONFUSION MATRIX

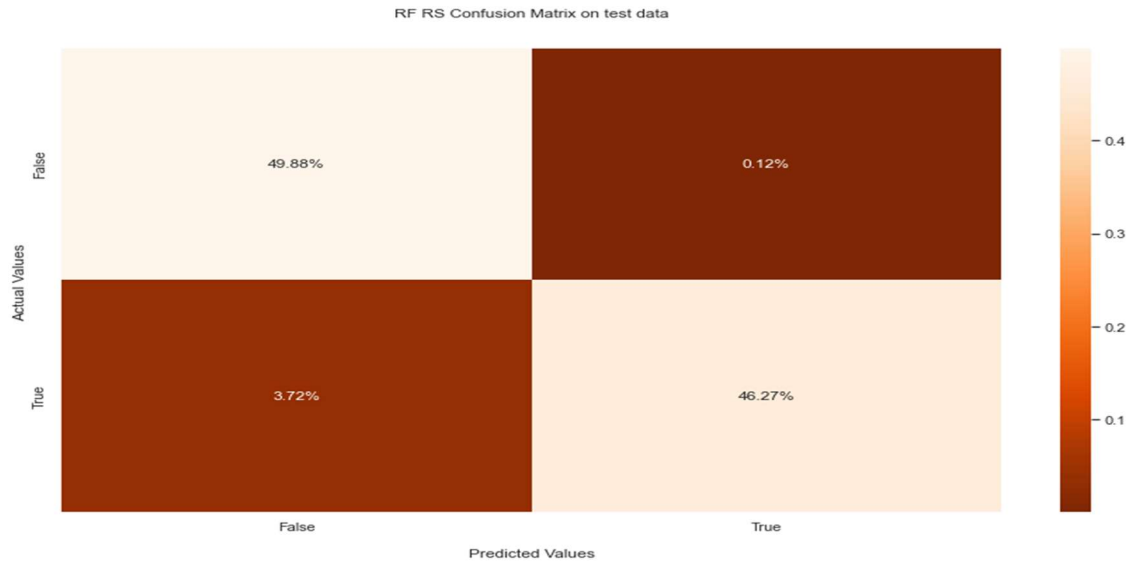

FIGURE 77 CONFUSION MATRIX PERCENTAGE

Random forest Classifier (Bagging) RS AUC Score and curve on Train and Test DataSet

```
AUC for the Training Data: 0.995
AUC for the Test Data:     0.996
```

TABLE 181 RANDOM FOREST CLASSIFIER (BAGGING) RS AUC SCORE

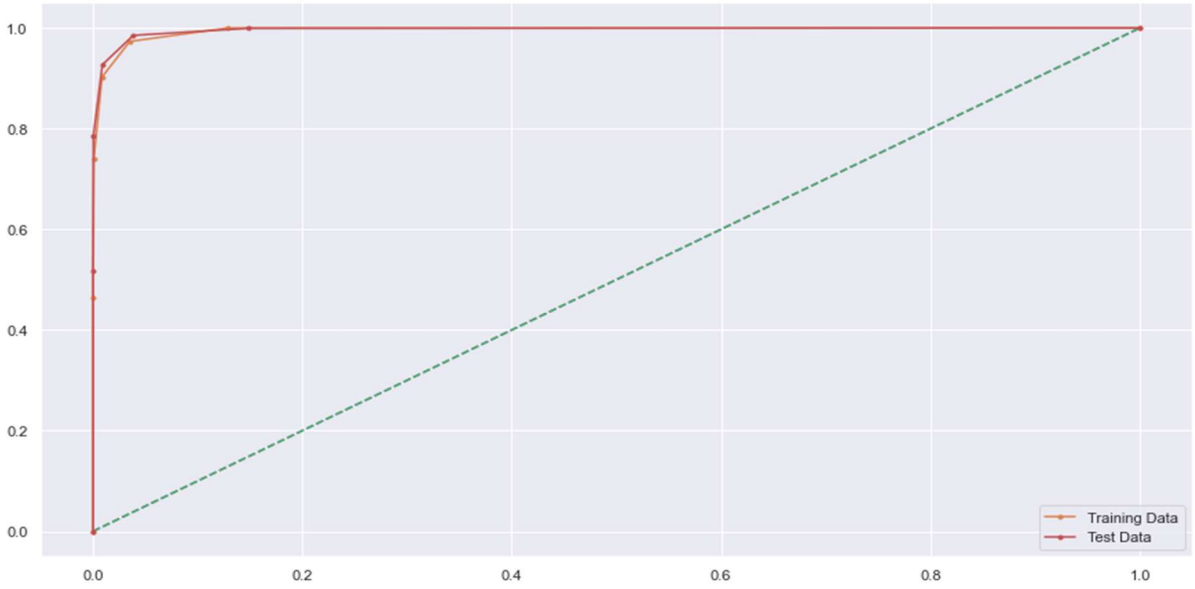

FIGURE 78 RO CURVE

Fivefold K-Fold cross Validation of Random Forest Classifier (Bagging) RS

Accuracy Random Forest Classifier (Bagging) RS

```
Accuracy of Resampled Random Forest (Bagging) on train set: 1.00
Accuracy of Resampled Random Forest (Bagging)  on test set: 0.96
```

TABLE 182 ACCURACY RANDOM FOREST CLASSIFIER (BAGGING) RS

Model Comparison

Comparison on Accuracy, Precision , Recall and F1 and AUC score on Test and Train Dataset

Conclusion : Picking the best performing Model

Based on the generated multiple model scores, the Random Forest Classifier and the Decision Tree Classifier (without grid search) appear to be the best models. Both models achieve high scores across multiple evaluation metrics, including accuracy, precision, recall, AUC score, and F1 score.

Comparing the two models, the Random Forest Classifier and the Decision Tree Classifier have the same scores, indicating similar performance. Both models achieve an accuracy score, precision score, and AUC score of 0.99, indicating that they are able to correctly classify the majority of the test dataset. Additionally, the models have high recall scores of 0.99, suggesting that they can effectively identify positive instances.

The Gaussian Naive Bayes model also performs well, although slightly lower than the Random Forest and Decision Tree models. It achieves an accuracy score, precision score, AUC score, and F1 score of 0.97, indicating good overall performance. However, it has a slightly lower recall score of 0.95, indicating a slightly higher number of false negatives compared to the Random Forest and Decision Tree models.

The Decision Tree Classifier with grid search and the Decision Tree Classifier with regularization have the lowest scores among the models provided. They achieve lower accuracy, precision, recall, AUC score, and F1 score compared to the other models.

Overall, the Random Forest Classifier and the Decision Tree Classifier (without grid search) are the top models based on their consistently high scores across multiple evaluation metrics. They demonstrate strong performance in terms of accuracy, precision, recall, AUC score, and F1 score, making them the best choices for classification tasks on the given test dataset.

Model Comparison on Train Dataset

Model Comparison Training dataset					
Models	Accuracy Score	Precision Score	Recall	AUC score	F1 Score
Random Forest Classifier Bag	1	1	1	1	1
DecisionTreeClassifier	1	1	1	1	1
Decision Tree Classifiers GS	1	1	1	1	1
RandomForestClassifier	1	1	1	1	1
GaussianNB	0.98	0.9	0.96	0.97	0.98
GaussianNB GS	0.98	0.9	0.96	0.97	0.98
DecisionTreeClassifier REG	0.97	0.88	0.93	0.95	0.97
RandomForestClassifier GS	0.93	0.67	0.91	0.92	0.93
GradientBoostingClassifier GS	0.92	0.64	0.84	0.88	0.92
GradientBoostingClassifier	0.92	0.64	0.84	0.88	0.92
AdaBoostClassifier GS	0.9	0.6	0.75	0.83	0.9
AdaBoostClassifier	0.89	0.58	0.74	0.83	0.89
LogisticRegression	0.89	0.48	0.77	0.83	0.89
LogisticRegression GS	0.89	0.48	0.77	0.83	0.89
Linear Discriminant Analysis GS	0.88	0.42	0.76	0.83	0.88
Linear Discriminant Analysis	0.88	0.42	0.76	0.83	0.88
KNeighborsClassifier GS	0.85	0.57	0.57	0.74	0.85
KNeighborsClassifier	0.85	0.57	0.57	0.74	0.85

TABLE 183 MODEL COMPARISON ON TRAIN DATASET

TOP 5 Models on SMOTE Train dataset					
	Accuracy Score	**Precision Score**	**Recall**	**AUC score**	**F1 Score**
DecisionTreeClassifier RS	1	1	1	1	1
RandomForestClassifier RS	1	1	1	1	1
RandomForestClassifier (Bag) RS	1	1	1	1	1
GaussianNB RS	0.98	0.90	0.96	0.97	0.98

TABLE 184 TOP 5 MODELS ON SMOTE TRAIN DATASET

Comparison of AUC & ROC on Train Data

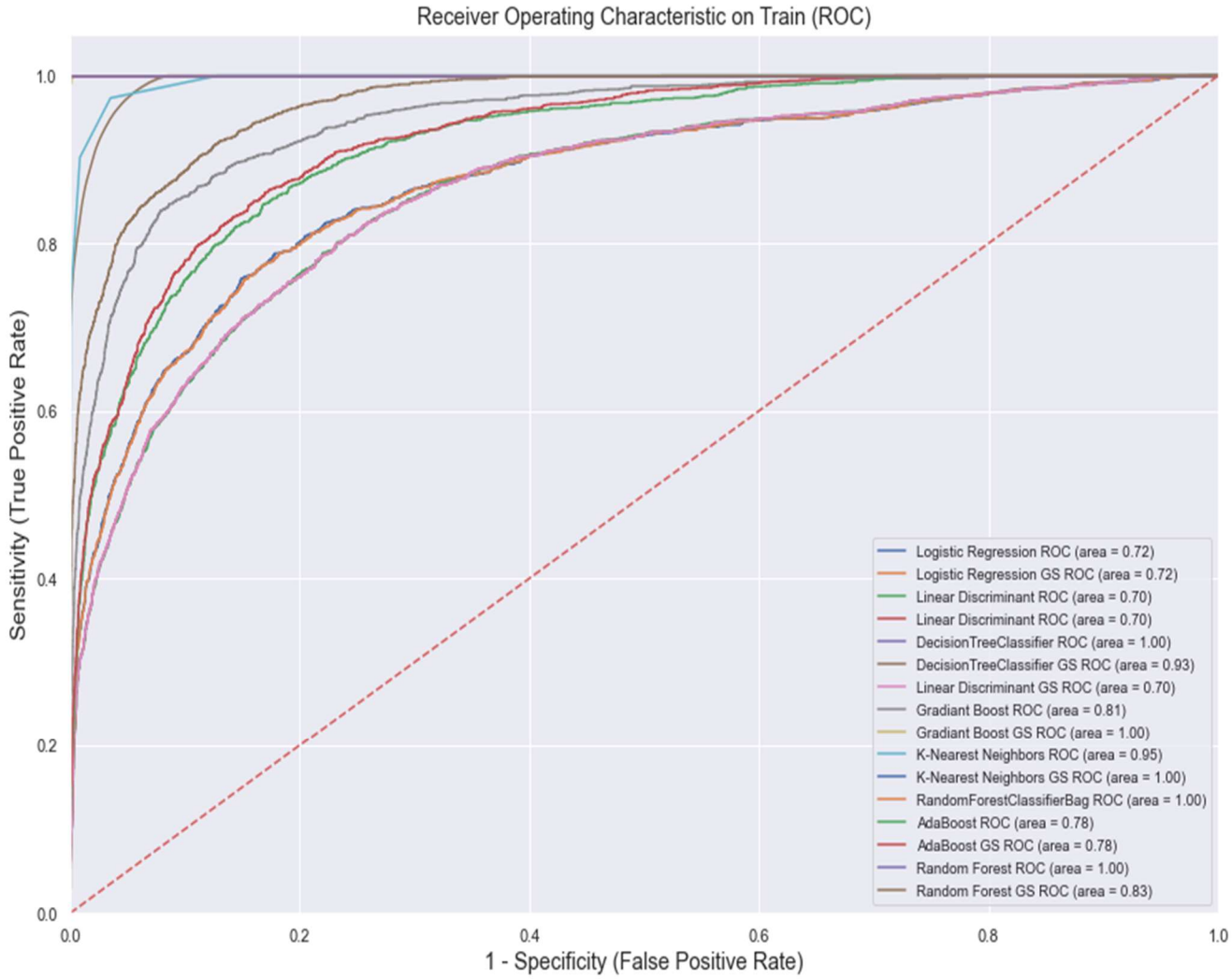

FIGURE 79 RO CURVE COMPARISON ON TRAIN DATASET

Overall Best Model on Train Dataset

TOP 5 Model on Train dataset					
	Accuracy Score	**Precision Score**	**Recall**	**AUC score**	**F1 Score**
Random Forest Classifier Bag	1	1	1	1	1
DecisionTreeClassifier	1	1	1	1	1
DecisionTreeClassifier GS	1	1	1	1	1
RandomForestClassifier	1	1	1	1	1
DecisionTreeClassifier RS	1	1	1	1	1
RandomForestClassifier RS	1	1	1	1	1
RandomForestClassifier RS	1	1	1	1	1

TABLE 185 MODEL COMPARISON ON TRAIN DATASET

Model Comparison on Test Dataset

Model Comparison Test dataset					
Models	Accuracy Score	Precision Score	Recall	AUC score	F1 Score
RandomForestClassifier Bag	0.99	0.97	0.99	0.99	0.99
RandomForestClassifier	0.99	0.97	0.99	0.99	0.99
DecisionTreeClassifier GS	0.99	0.97	0.97	0.98	0.99
DecisionTreeClassifier	0.99	0.97	0.97	0.98	0.99
GaussianNB	0.97	0.9	0.95	0.96	0.97
GaussianNB GS	0.97	0.9	0.95	0.96	0.97
DecisionTreeClassifier REG	0.96	0.86	0.9	0.94	0.96
RandomForestClassifier GS	0.93	0.66	0.89	0.91	0.93
GradientBoostingClassifier GS	0.92	0.64	0.83	0.88	0.92
GradientBoostingClassifier	0.92	0.64	0.83	0.88	0.92
AdaBoostClassifier GS	0.9	0.61	0.74	0.83	0.9
AdaBoostClassifier	0.89	0.59	0.72	0.82	0.89
LogisticRegression	0.89	0.49	0.76	0.83	0.89
LogisticRegression GS	0.89	0.49	0.76	0.83	0.89
Linear Discriminant Analysis GS	0.88	0.44	0.77	0.83	0.88
Linear Discriminant Analysis	0.88	0.44	0.77	0.83	0.88
KNeighborsClassifier GS	0.85	0.58	0.54	0.73	0.85
KNeighborsClassifier	0.85	0.58	0.54	0.73	0.85

TABLE 186 MODEL COMPARISON ON TEST DATASET

TOP 5 Model on SMOTE Test dataset					
	Accuracy Score	**Precision Score**	**Recall**	**AUC score**	**F1 Score**
RandomForestClassifier RS	0.96	0.93	1	0.96	0.96
RandomForestClassifier RS	0.96	0.92	1	0.96	0.96
DecisionTreeClassifier RS	0.96	0.92	0.99	0.96	0.96
GaussianNB RS	0.96	0.93	0.99	0.96	0.96

TABLE 187 TOP 5 MODEL ON SMOTE TEST DATASET

TOP Models on Test dataset					
Models	**Accuracy Score**	**Precision Score**	**Recall**	**AUC score**	**F1 Score**
Random Forest Classifier Bag	0.99	0.97	0.99	0.99	0.99
RandomForestClassifier	0.99	0.97	0.99	0.99	0.99
DecisionTreeClassifier GS	0.99	0.97	0.97	0.98	0.99
DecisionTreeClassifier	0.99	0.97	0.97	0.98	0.99
RandomForestClassifier RS	0.96	0.93	1	0.96	0.96
RandomForestClassifier RS	0.96	0.92	1	0.96	0.96
DecisionTreeClassifier RS	0.96	0.92	0.99	0.96	0.96
GaussianNB RS	0.96	0.93	0.99	0.96	0.96

TABLE 188 TOP MODELS ON TEST DATASET

Comparison of AUC & ROC on Test Data

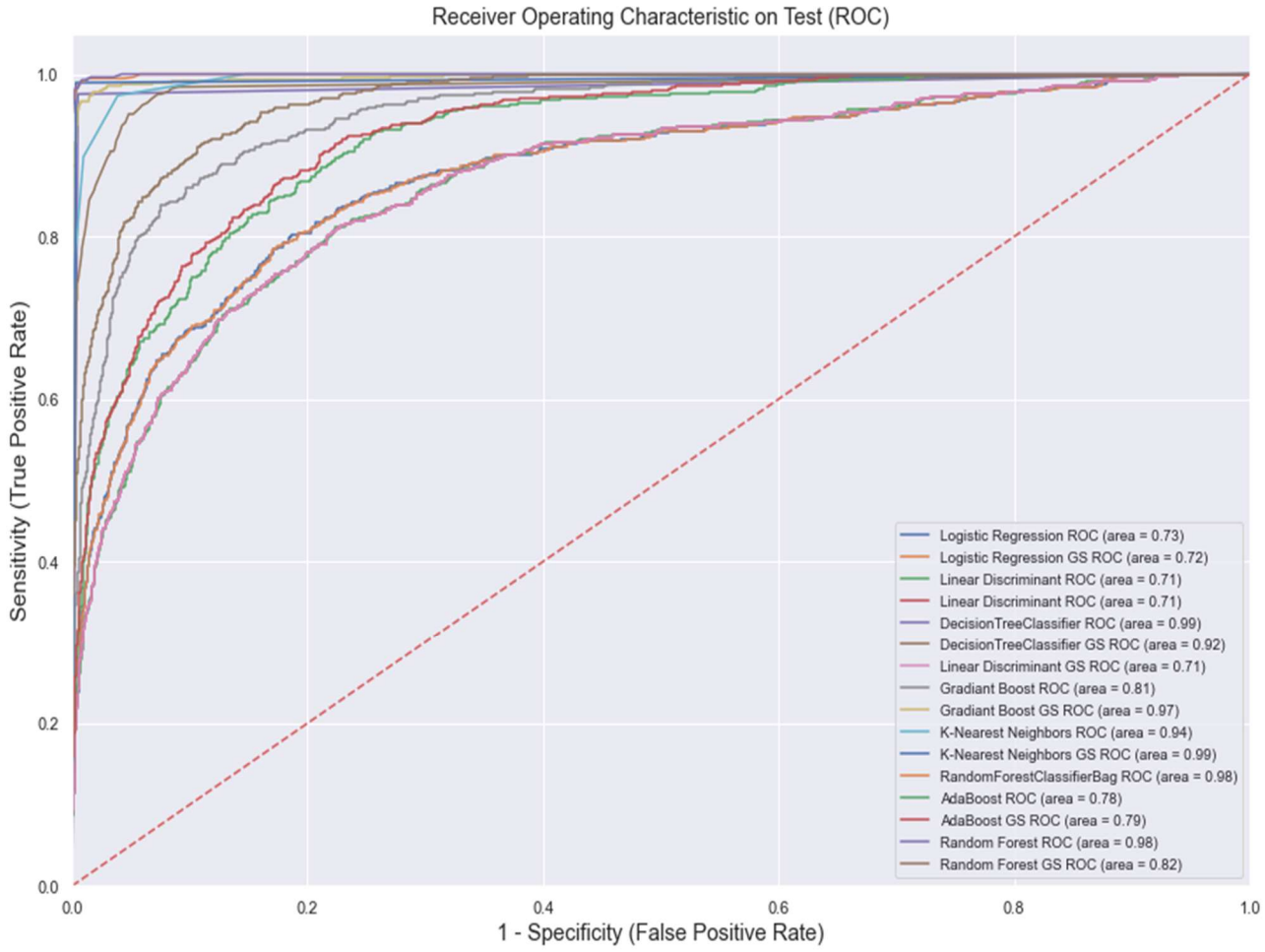

FIGURE 80 RO CURVE COMPARISON ON TEST DATASET

Plotting of Accuracy of Models on Test Data

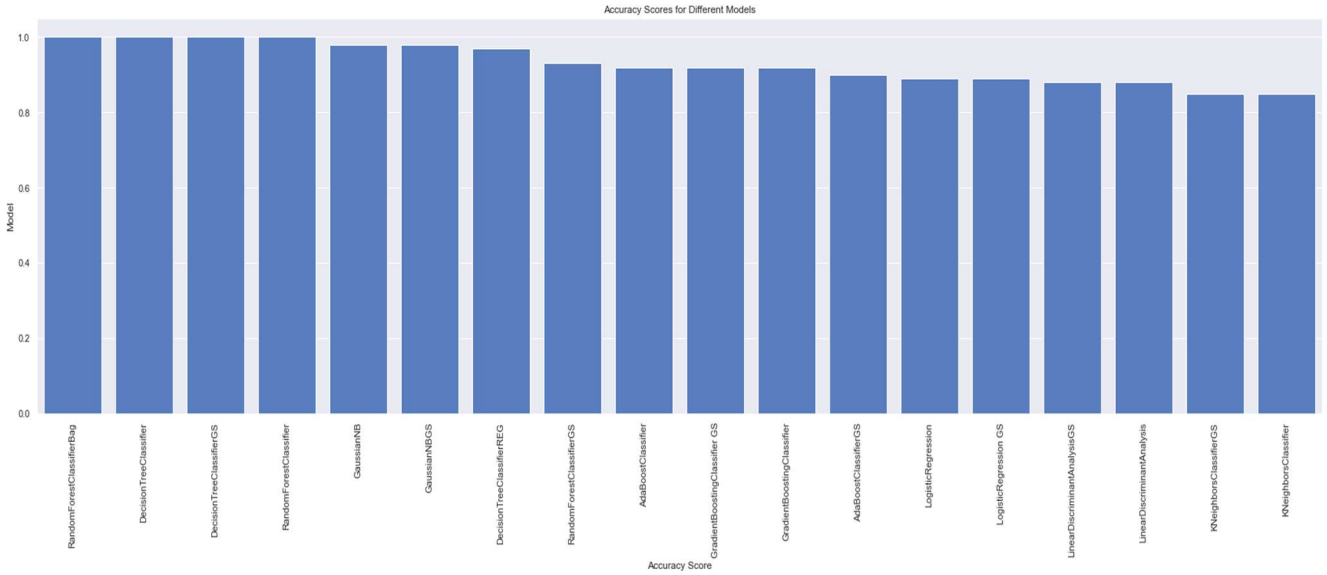

FIGURE 81 ACCURACY SCORES PLOTTING

The Best Performed Model

TOP performing Model both on Train and Test dataset						
Model	Dataset	Accuracy Score	Precision Score	Recall	AUC score	F1 Score
RandomForestClassifier Bag	**Test**	**0.99**	**0.97**	**0.99**	**0.99**	**0.99**
RandomForestClassifier Bag	**Train**	**1**	**1**	**1**	**1**	**1**

TABLE 189 THE BEST PERFORMED MODEL

Reason for Selection Random Forest bagging as best model:

Choosing the Best Model among Random Forest and Decision Tree Classifier

The parameters scores are almost close to each other in RF & DT , then we will analysis the Random Forest classifier and a Decision Tree classifier to choose best performing.

Random Forest Classifier:

The Random Forest classifier is an ensemble model that combines multiple decision trees to make predictions. It builds a collection of decision trees and aggregates their predictions through voting or averaging.

Advantages:

1.Random Forests are known for their high accuracy and robustness. They tend to perform well across various datasets and handle a wide range of input features.
2.Random Forests can handle a large number of input features and can handle both numerical and categorical data.
3.They are less prone to overfitting compared to a single Decision Tree as they reduce variance by averaging predictions from multiple trees.
4.Random Forests provide feature importance rankings, which can help identify the most influential features in the model.

Disadvantages:
1.Random Forests can be computationally expensive and may require more memory compared to a single Decision Tree due to multiple trees being built.
2.The interpretability of a Random Forest model is lower compared to a single Decision Tree since it involves an ensemble of trees.

Decision Tree Classifier:

The Decision Tree classifier is a single tree-based model that makes predictions based on a series of hierarchical decision rules.

Advantages:

1.Decision Trees are relatively easy to understand and interpret. They provide explicit decision rules that can be visualized and explained.

2.They can handle both numerical and categorical data.
3.Decision Trees can handle feature interactions and non-linear relationships.
4.Decision Trees can be computationally efficient, especially for smaller datasets.

Disadvantages:

1.Decision Trees are prone to overfitting, especially when the tree becomes too deep and complex. This can result in poor generalization on unseen data.
2.They can be sensitive to small variations in the training data, leading to different tree structures and potentially different predictions.
3.Decision Trees may not perform as well as Random Forests in situations where there are complex relationships and interactions between features.

In summary, on prioritize high accuracy, robustness, and the ability to handle complex relationships between features, and considering disadvantages decision tree, the Random Forest classifier is a best choice as [er our dataset.

Key Outcomes of Random Forest(Bagging)

10 influencing Feature of Importance:

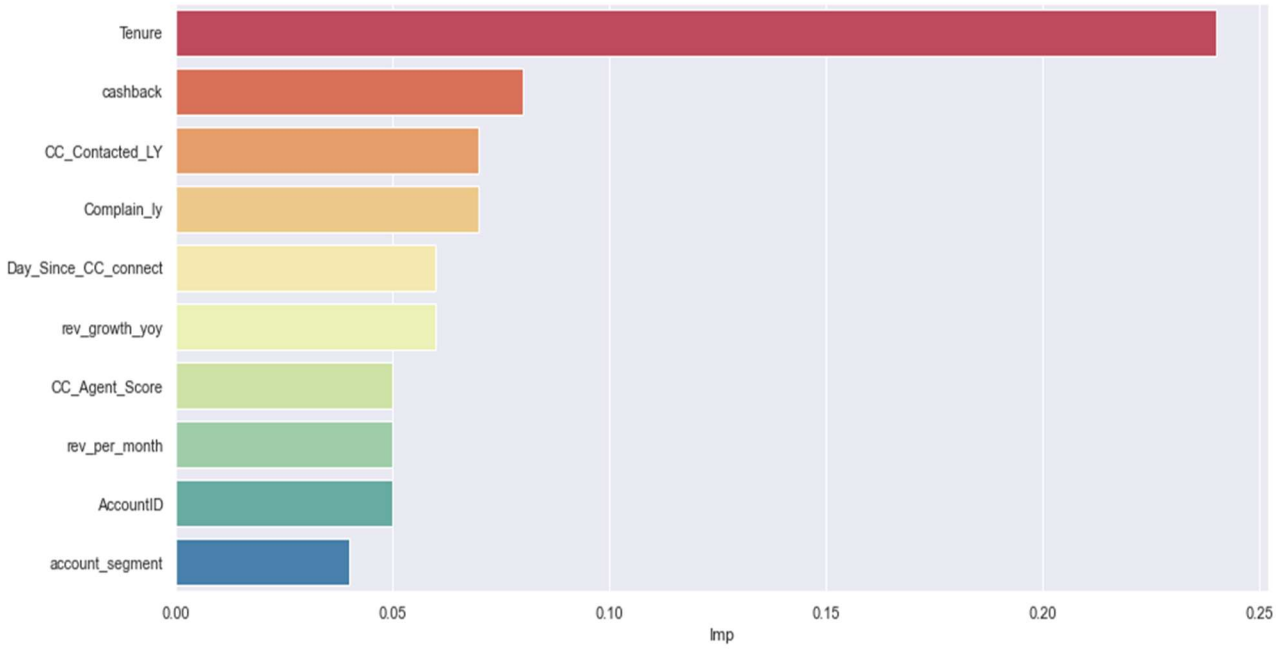

FIGURE 82 FEATURE IMPORTANCE

Feature Importance Business Interpretations:

Based on the feature importance values obtained from the Random Forest model, the following insights can be derived:

Tenure: The tenure of the account has the highest importance (0.24). This suggests that the length of time a customer has been associated with the company plays a significant role in

predicting churn. Providing incentives or rewards to long-term customers may help in retaining them.

Cashback: Cashback (0.08) is also an important feature, indicating that customers who receive cashback offers are less likely to churn. Offering attractive cashback schemes or rewards can help in retaining customers.

CC_Contacted_LY: The frequency of customer care contact in the last year (0.07) is another important factor. Customers who have had more interactions with customer care are more likely to churn. Improving customer service and addressing customer concerns promptly can help in reducing churn.

Complain_ly: The number of complaints in the last year (0.07) is also significant. Customers who have lodged more complaints are at a higher risk of churning. It is important to address customer complaints effectively and work towards improving the overall customer experience.

Day_Since_CC_connect: The number of days since the last contact with the customer care (0.06) is a relevant factor. Customers who have not had recent interactions with customer care are more prone to churn. Proactive outreach or personalized communication with customers can help in maintaining engagement and reducing churn.

These insights suggest that focusing on customer satisfaction, personalized communication, and addressing complaints and concerns can play a crucial role in reducing churn. Tailoring retention strategies around these important features can help in retaining customers and improving overall customer loyalty

Predicting the probabilities of target variables Random Forest Classifier (Bagging)

Depicting the symbolic top 5 values of probabilities of Churn

	0	1
0	0.98	0.02
1	0.99	0.01
2	0.98	0.02
3	1.00	0.00
4	0.88	0.12

TABLE 190 TOP 5 VALUES OF PROBABILITIES OF CHURN

Predicting the AccountID in light of actual and Predicted Churn by Random Forest Classifier (Bagging)

Depicting the symbolic top 10 values

	Churn	AccountID	Predicted
0	1.00	20012	1.00
1	1.00	20029	1.00
2	1.00	20031	1.00
3	1.00	20034	1.00
4	1.00	20049	1.00
5	1.00	20061	1.00
6	1.00	20069	1.00
7	1.00	20083	1.00
8	1.00	20093	1.00
9	1.00	20110	1.00

TABLE 191 SYMBOLIC TOP 10 ACCOUNTS

Observations on key Variables & Suggestions:

Tenure: The tenure of the customer is a key factor in predicting churn, with a high dependency of 30% to 51% weight across different models. Customers are more prone to churn during their early days of 0 to 5.
To address this, the company should formulate policies and strategies to engage new accounts within this age group and provide them with incentives or personalized offers to increase their retention rate.

Customer Segment: Among the customer segments, the company has a significant base of Super customers, followed by Regular Plus customers, with approximately 3,100 customers in this segment. However, around 30% of these customers churn,
To address this issue, Need for targeted efforts to retain them. Implementing promotional offers or exclusive benefits for this segment can help in improving customer loyalty and reducing churn.

City Tier: In terms of geographical distribution, the company has a large customer base in Tier 1 cities, with over 6,200 customers. However, around 1,100 customers from this segment churn.
To address this, the company should launch specific campaigns or initiatives in Tier 1 cities to retain accounts and prevent churn. Additionally, while Tier 2 and Tier 3 cities have fewer customers, they exhibit a similar churn trend to Tier 1, suggesting the need for retention strategies in these cities as well.

Account_user_count : Among the user account categories, User account 4 has the highest number of customers, approximately 4,200, with a churn rate of nearly 800. This is followed by categories 3 and 5. Interestingly, category 5 has the highest churn rate in terms of percentage, with approximately 25% of the tagged customers churning.
To address this, the company should focus on providing enhanced services and personalized attention to customers in these account categories to reduce churn.

Cashback: Specifically, customers within the cashback range of 125 to 150 and 150 to 175 are more prone to churn.
To address this issue, Company should pay close attention to these cashback bands and implement strategies to reduce churn within these ranges. The company can consider offering additional incentives or exclusive cashback offers to customers falling within this range.

Login Device: In terms of login device usage, approximately 6% of customers using either mobile or other devices churn. However, in terms of absolute numbers, customers using mobile devices for login have the highest turnover, with around 1,200 out of 6,200 customers churning.
To address this issue, The company should pay attention to providing a seamless and optimized experience for customers using mobile devices to reduce churn in this segment.it is most prevailing device in today's scenario.

Payment: It has come out in analysis threat the Debit card and Credit card payment having some issue probably technical one.
To address this issue, Possible reasons could include technical difficulties during the payment process, limited acceptance of credit cards, or customer preferences for other payment methods.

CC_Agent_Score :Customers who rated the company between 2.5 to 3.5 which can be treated as not good in today's scenarios were churned. It indicates that satisfaction levels are low and reflecting in candid rating scores.
To address this issue, The company should focus on analysing the feedback type and reasons of complaint and its resolution address these concern account should be treated empathically and implement measures to address their concerns and improve overall customer satisfaction index.

Overall, by focusing on engaging new accounts, targeting specific customer segments, launching campaigns in key cities, providing enhanced services to high-churn account categories, optimizing the mobile experience, and addressing the concerns of customers with varying rating scores, the company can effectively reduce churn and improve customer retention.

Recommendations to Business:

Focus on Retention: The analysis highlights several variables that are crucial for improving customer retention. Tenure plays a significant role, with customers being more prone to churn during their early days of 0 to 5. The company should formulate policies and strategies to engage new accounts within this age group and provide personalized offers to increase retention. Additionally, focusing on customer segments with higher churn rates, such as Super customers, and implementing targeted efforts to retain them can significantly improve overall retention.

Personalized Campaigns: The findings indicate that different customer segments have varying churn rates. By leveraging this information, the company can design personalized campaigns tailored to each segment's needs and preferences. For example, offering promotional offers or exclusive benefits to Super customers or customers falling within specific cashback bands can enhance their loyalty and reduce churn. Personalized campaigns based on factors like city tier, account user count, or customer preferences can also contribute to higher engagement and retention.

Customer Engagement Program: To increase customer engagement, the company should focus on initiatives that enhance the overall customer experience. This can include providing seamless login experiences across different devices, optimizing mobile app functionality, and ensuring smooth payment processes for both debit and credit cards. Engaging customers through loyalty programs, rewards, and personalized communication can also strengthen their connection with the company and reduce churn.

Feedback and Issue Resolution: Customer feedback is crucial for identifying areas of improvement and resolving issues. By analysing customer ratings, particularly those falling in the 2.5 to 3.5 range, the company can gain insights into customer satisfaction levels and areas requiring attention. Implementing measures to address customer concerns and improve issue resolution can significantly impact customer retention. Regularly collecting feedback, conducting surveys, and proactively addressing customer grievances can demonstrate the company's commitment to customer satisfaction.

In summary, to improve retention, the company should focus on personalized campaigns, customer engagement programs, feedback analysis, and issue resolution. By leveraging variables such as tenure, customer segments, cashback, city tier, account user count, login device usage, and payment methods, the company can tailor its strategies to enhance customer loyalty, drive engagement, and reduce churn.

----------End of the Report----------