**CAPSTONE PROJECT**

**Prediction to subscribe for a term deposit**

**FINAL REPORT**

**Mentored by:**

**Mr. Saurav Reddy**

**Submitted by: (Group – 12)**

# Ms. Diksha G Nayak

# Ms. Gifty Maria Paul

# Mr. Navneet Sankhla

# Mr. Sai Nithil Gopu

# Mr. Sameer Mannur

**Table of Contents:**

1. **Overview – Industry Background and Key Objectives**

1. **Data Dictionary**

1. **Data Pre-processing:** 
   * Imputation of missing values
   * Grammatical rectification and other cosmetic improvements

1. **Initial Data Exploration** 
   * Analysing Relationships between Variables
     + Univariate analysis
     + Bi-variate analysis
     + Multi-Variate analysis
   * Treatment of Outliers
   * Treatment of Imbalanced data

1. **Model Building and additional data treatments:** 
   * Splitting, Scaling and Encoding data
   * Model training
   * Model performance and scoring

1. **Future Actions**

**Overview:**

# Industry Background and Key Objectives

Term deposits are a major source of income for a bank. A term deposit is a cash investment held at a financial institution. Your money is invested for an agreed rate of interest over a fixed amount of time, or term. The bank has various outreach plans to sell term deposits to their customers such as email marketing, advertisements, telephonic marketing, and digital marketing. Telephonic marketing campaigns still remain one of the most effective ways to reach out to people.

We are an in-house analytics team at this leading ecommerce player, tasked with better understanding order rejection by customers at various stages of delivery.

In particular, we have been assigned a dataset containing orders for women’s clothing, a certain portion of which have been cancelled due to various reasons.

**Business problem statement:**

1. **Business Problem Understanding**

Term deposits are a major source of income for a bank. A term deposit is a cash investment held at a financial institution. Your money is invested for an agreed rate of interest over a fixed amount of time, or term. The bank has various outreach plans to sell term deposits to their customers such as email marketing, advertisements, telephonic marketing, and digital marketing. Telephonic marketing campaigns still remain one of the most effective ways to reach out to people. However, they require huge investment as large call centres are hired to actually execute these campaigns. Hence, it is crucial to identify the customers most likely to convert beforehand so that they can be specifically targeted via call. The data is related to direct marketing campaigns (phone calls) of a Portuguese banking institution. The classification goal is to predict if the client will subscribe to a term deposit (variable y).

1. **Business Objective**

Broadly speaking, recent developments in machine learning techniques and data mining has led to an interest of implementing these techniques in various fields.

We are trying to predict the customers for subscribing to the term deposit of our bank and we can do this by finding those factors which is affecting the customers to not take the Term deposit.

**Data Dictionary:**

The **bank client data:** consists of below mentioned attributes:

* **age** (numeric)
* **job**: type of job (categorical: "admin.","unknown","unemployed","management","housemaid","entrepreneur","student","blue-collar", "self-employed”, "retired", "technician", "services")
* **marital**: marital status (categorical: "married", "divorced", "single"; note: "divorced" means divorced or widowed
* **education**: (categorical: "unknown", "secondary", "primary", "tertiary")
* **default**: has credit in default? (Binary: "yes", "no")
* **balance**: average yearly balance, in euros (numeric)
* **housing**: has housing loan? (Binary: "yes", "no")
* **loan**: has personal loan? (Binary: "yes", "no")  
  # related with the last contact of the current campaign:
* **contact**: contact communication type (categorical: "unknown", "telephone", "cellular")
* **day**: last contact day of the month (numeric)
* **month**: last contact month of year (categorical: "jan", "feb", "mar", …, "nov", "dec")
* **duration**: last contact duration, in seconds (numeric)  
  # other attributes:
* **campaign**: number of contacts performed during this campaign and for this client (numeric, includes last contact)
* **pdays**: number of days that passed by after the client was last contacted from a previous campaign (numeric, -1 means client was not previously contacted)
* **previous**: number of contacts performed before this campaign and for this client (numeric)
* **poutcome**: outcome of the previous marketing campaign (categorical: "unknown", "other", "failure", "success")

**Output variable (desired target):**

* **y** - has the client subscribed a term deposit? (Binary: "yes", "no")

**Data Pre-processing:**

We begin by reading in the default csv file and getting a sense of the overall size and features we will be working with.

**Our default data file includes:**

Number of Columns:17

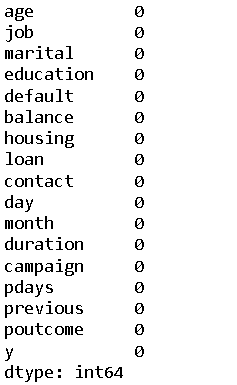
Total Number of Records: 45211

**Imputation of missing values:**

Our initial search helps us identify missing values. None of the columns have any missing values.

Hence, we won’t be imputing null values.

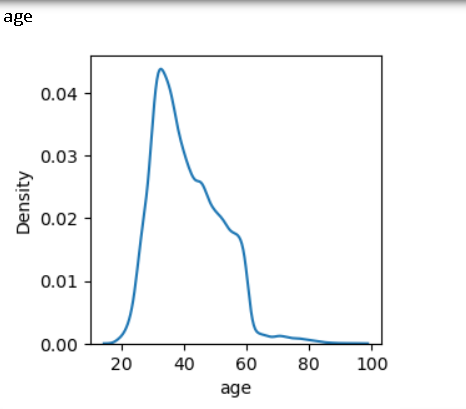
* In case there were null values in numerical variables we could have a look at the presence of outliers depending on the presence and absence we could impute them with either mean or median.
* In case there were null values in categorical variables we could impute them with mode.



**Initial Data Exploration - Analysing Relationships between the Variables:**

**Univariate analysis:**

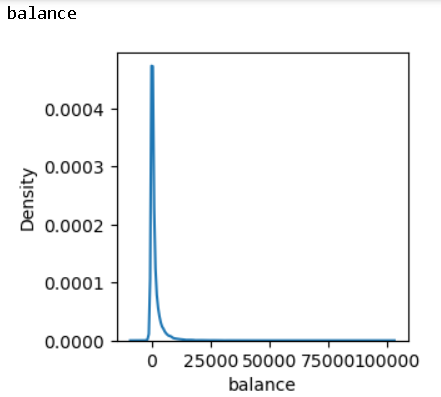
**Let us begin by performing univariate analysis on numerical variables.**

****

age Skewness: 0.6848179257252598, this indicates the data is right skewed.

age Kurtosis: 0.3195703759105042, it indicates that the data is leptokurtic.

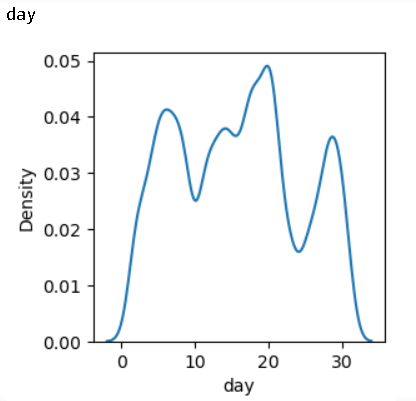
Our data is unimodal and the value which has highest frequency of occurrences are 32.

****

balance Skewness: 8.360308326166326, this indicates the data is right skewed.

balance Kurtosis: 140.75154662504158, it indicates that the data is leptokurtic.

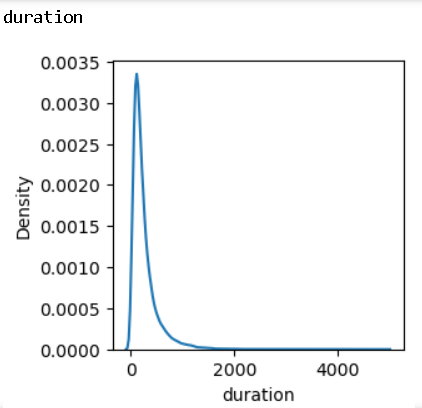
Our data is unimodal and the value which has highest frequency of occurance is 0.

****

day Skewness: 0.09307901402122411, this indicates the data is right skewed.

day Kurtosis: -1.0598973728286003, it indicates that the data is leptokurtic.

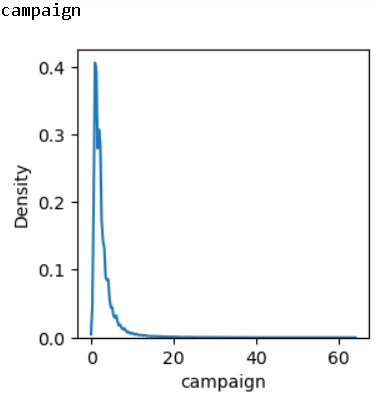
Our data is multimodal and the value which has highest frequency of occurrence is 20.

****

duration Skewness: 3.144318099423456, this indicates the data is right skewed.

duration Kurtosis: 18.153915269019706, it indicates that the data is leptokurtic.

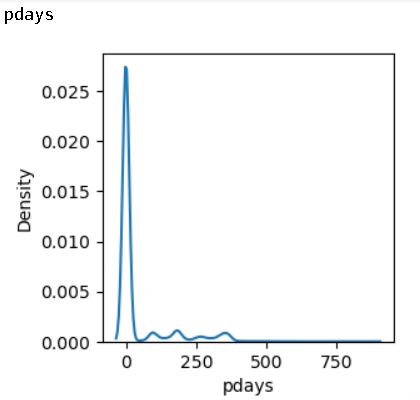
Our data is unimodal and the value which has highest frequency of occurrences 124.



campaign Skewness: 4.898650166179674, this indicates the data is right skewed.

campaign Kurtosis: 39.2496508023021, it indicates that the data is leptokurtic.

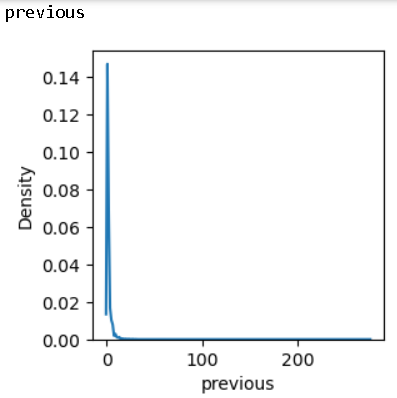
Our data is unimodal and the value which has highest frequency of occurance is 1.



pdays Skewness: 2.6157154736563477, this indicates the data is right skewed.

pdays Kurtosis: 6.935195210422799, it indicates that the data is leptokurtic.

Our data is unimodal and the value which has highest frequency of occurrence is -1.



previous Skewness: 41.84645447266292, this indicates the data is right skewed.

previous Kurtosis: 4506.860660183261, it indicates that the data is leptokurtic.

Our data is unimodal and the value which has highest frequency of occurrence is 0.

**Let us begin by performing univariate analysis on categorical variables.**

variable: job has these categories with count of:

Blue-collar employees and Management employees opt most for term deposit.

|  |  |
| --- | --- |
| blue-collar 9732  management 9458  technician 7597  admin. 5171  services 4154  retired 2264  self-employed 1579  entrepreneur 1487  unemployed 1303  housemaid 1240  student 938  unknown 288 |  |

variable: marital has these categories with count of:

Married people opt the highest for term deposit.

|  |  |
| --- | --- |
| married 27214  single 12790  divorced 5207 |  |

variable: education has these categories with count of:

People who have secondary education opt the highest for term deposit.

|  |  |
| --- | --- |
| secondary 23202  tertiary 13301  primary 6851  unknown 1857 |  |

variable: default has these categories with count of:

A lot of People haven’t opted term deposit by default.

|  |  |
| --- | --- |
| no 44396  yes 815 |  |

variable: housing has these categories with count of:

People who have their own house opt for term deposit.

|  |  |
| --- | --- |
| yes 25130  no 20081 |  |

variable: loan has these categories with count of:

People who don’t have any loan from bank are the highest to opt for term deposit.

|  |  |
| --- | --- |
| no 37967  yes 7244 |  |

variable: contact has these categories with count of:

People who have cell phone are the highest to take these term deposit.

|  |  |
| --- | --- |
| cellular 29285  unknown 13020  telephone 2906 |  |

variable: month has these categories with count of:

People opt to take term deposit in highest numbers during the month of May.

|  |  |
| --- | --- |
| may 13766  jul 6895  aug 6247  jun 5341  nov 3970  apr 2932  feb 2649  jan 1403  oct 738  sep 579  mar 477  dec 214 |  |

variable: poutcome has these categories with count of:

Many people who are not classified as successful or failure in their career are the highest in numbers to take term deposit.

|  |  |
| --- | --- |
| unknown 36959  failure 4901  other 1840  success 1511 |  |

variable: y has these categories with count of:

Our target variable indicates that no of people who don’t take term deposit are the highest.

|  |  |
| --- | --- |
| no 39922  yes 5289 |  |

**Bivariate analysis:**

Here we will attempt to understand how the available columns are impacting each other on a one-to-one basis.

**First, we visualise the relationship between our target column (Status) and numerical features (Quantity, Amount)**

|  |  |
| --- | --- |
|  |  |

|  |  |
| --- | --- |
|  |  |

|  |  |
| --- | --- |
|  |  |

|  |  |
| --- | --- |
|  |  |

|  |  |
| --- | --- |
|  |  |

|  |  |
| --- | --- |
|  |  |

|  |  |
| --- | --- |
|  |  |

**Now, let us look at how categorical column data is distributed vis-à-vis ‘y’**

Broadly speaking, the status of Term deposit come under the ‘No’ -Term deposit segment, and that can be see across all visualisations.

* We have performed crosstab to get the values of every sub category under the category how it is varying with the target variable.
* To visually represent the same, we have used count plot and performed the analysis.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | |  |  |  | | --- | --- | --- | | y | no | yes | | Admin | 4540 | 631 | | Blue-collar | 9024 | 708 | | entrepreneur | 1364 | 123 | | Housemaid | 1131 | 109 | | management | 8157 | 1301 | | retired | 1748 | 516 | | Self- employed | 1392 | 187 | | services | 3785 | 369 | | student | 669 | 269 | | technician | 6757 | 840 | | Unemployed | 1101 | 202 | | unknown | 254 | 34 | |

* We can infer from this data that people who have opted Term deposited are close to 10% of the people who have not opted Term Deposit.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | |  |  |  | | --- | --- | --- | | y | no | Yes | | Divorced | 4585 | 622 | | Married | 24459 | 2755 | | single | 10878 | 1912 | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | |  |  |  | | --- | --- | --- | | y | No | yes | | primary | 6260 | 591 | | Secondary | 20752 | 2450 | | Tertiary | 11305 | 1996 | | unknown | 1605 | 252 | |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | |  |  |  | | --- | --- | --- | | Y | No | Yes | | No | 39159 | 5237 | | Yes | 763 | 52 | |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | |  |  |  | | --- | --- | --- | | y | no | yes | | No | 16727 | 3354 | | yes | 23195 | 1935 | |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | |  |  |  | | --- | --- | --- | | Y | no | Yes | | No | 33162 | 4805 | | yes | 6760 | 482 | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | |  |  |  | | --- | --- | --- | | y | No | yes | | Cellular | 24916 | 4369 | | Telephone | 2516 | 390 | | unknown | 12490 | 530 | |

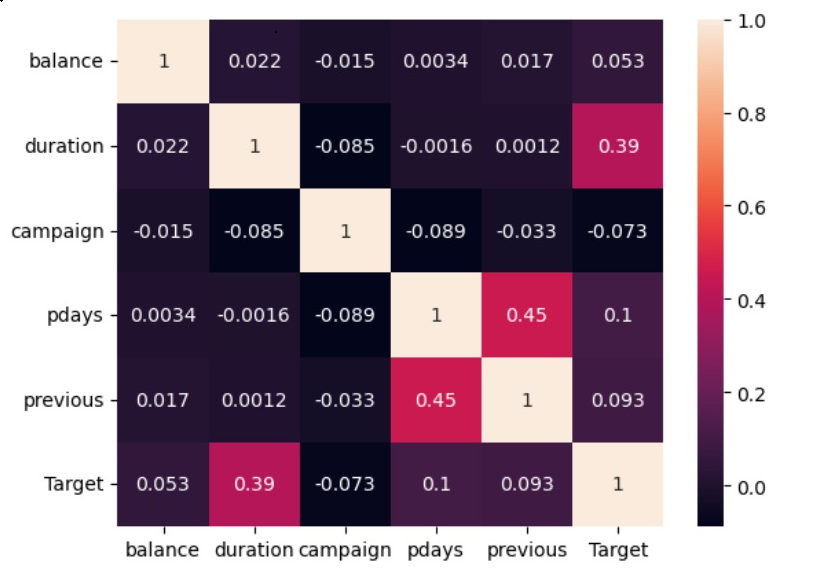
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | |  |  |  | | --- | --- | --- | | y | No | Yes | | Apr | 2355 | 577 | | Aug | 5559 | 688 | | Dec | 114 | 100 | | Feb | 2208 | 441 | | Jan | 1261 | 142 | | Jul | 6268 | 627 | | Jun | 4795 | 546 | | Mar | 229 | 248 | | May | 12841 | 925 | | Nov | 3567 | 403 | | Oct | 415 | 323 | | Sep | 310 | 269 | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | |  |  |  | | --- | --- | --- | | y | no | yes | | Failure | 4283 | 618 | | Other | 1533 | 307 | | Success | 533 | 978 | | unknown | 33573 | 3386 | |

**Multivariate Analysis:**

In this section we aim to visualise the relationship between our target variable and features more broadly.

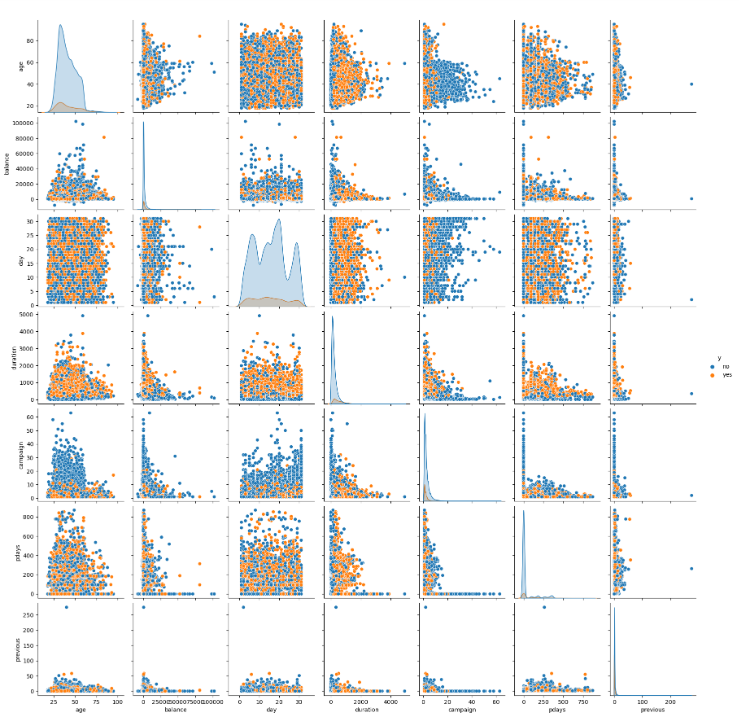
For instance, it might be useful to check how our term deposit is affected with different numerical and categorical variables. In this section we perform heatmap to check the correlation between the numerical variables with our Target variable.



* From our heatmap we can understand that there is high correlation between our numerical variables. We can see that duration is the only variable which is a little correlated with our Target variable.
* There exists very little multicollinearity between the variables pdays and previous, which is very negligible. Rest no other independent variables is exhibiting multicollinearity.

The Inference we get from pair plot is that:

* We can observe correlations between features and the target variable based on patterns and trends in the scatter plots.
* Outliers can be identified as data points deviating significantly from the overall pattern.

****

**Outlier Treatment:**

While we have worked to treat missing values in our dataset, we have taken a different approach when it comes to outliers, given the type of data available and the scope of analysis that we aim to undertake.

We can see that after outlier treatment the number of rows is reduced from 45211 to 28193.

* We can infer that, 'pdays' and 'previous' variables are not showing any significant values.
* So, we cannot do outlier treatment.

**Treatment of Imbalanced Data**

Common consensus suggests that a proportion of at least 30:70 should be present in a binary target variable. However, in case of the ‘Status’ column, the share of values within ‘yes’ are of a lower proportion.

Similar to our decision on outliers, we have made certain considerations for our target variable, and not attempted to treat imbalanced immediately at the outset.

We have decided to use the data in its original proportion for our base model, and based on its performance will make decisions on whether and how to adjust the imbalance at a later stage.

**Model Building and Additional Data Treatments:**

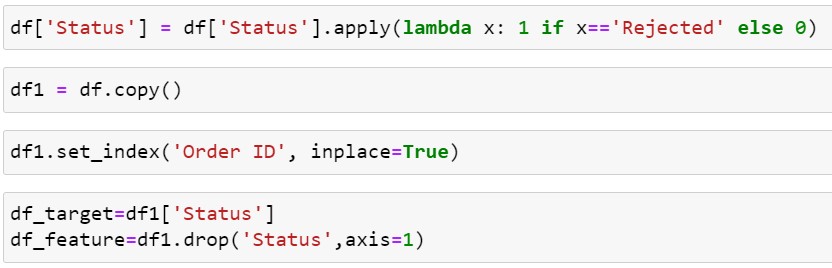
**Classification Focussed Predictive Modelling**

As originally mentioned in our objective, our aim with this analysis is to try and understand trends in order rejections. Accordingly, we will build a model that will help identify which of the features are most likely to affect rejections and consequently be useful in predicting rejections in the future.

As a start, we will build a logistic regression based classifier. We shall use popular metrics such as precision and f1-score among other, to interpret the performance of our base model.

For our logistic model, we shall start with reclassifying our target variable

‘Status’ with values 0 (for ‘Not Rejected’) and 1(for ‘Rejected’).



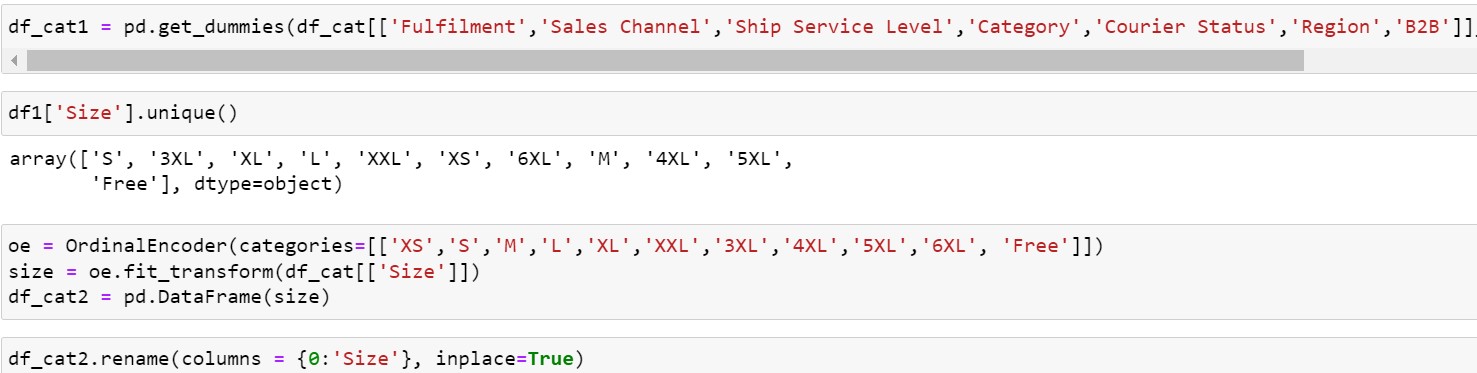
**Feature encoding and scaling:**

We shall scale the numeric features of our dataset, and encode the categorical features. Numeric features are scaled to bring them into the same base level; and categorical features are encoded so that their numeric representations can be used to perform further analysis and be fed into out machine learning model.

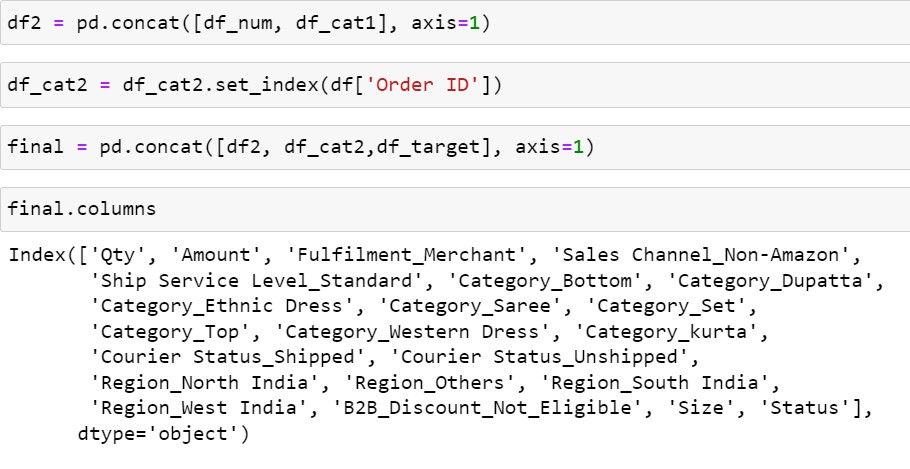
* Scaling the numerical columns using the Standard Scaler function



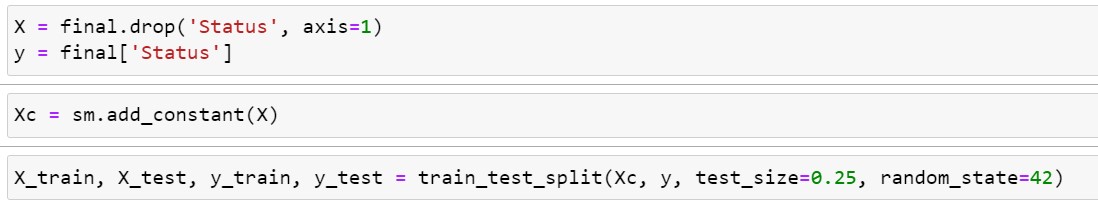
* Encoding the categorical columns using Ordinal Encoding



* Concatenating the scaled numeric and encoded categorical columns to reform our dataset



* Separating our data into target and features sections and using the Train\_Test\_Split function to create appropriate data partitions to be fed into the model



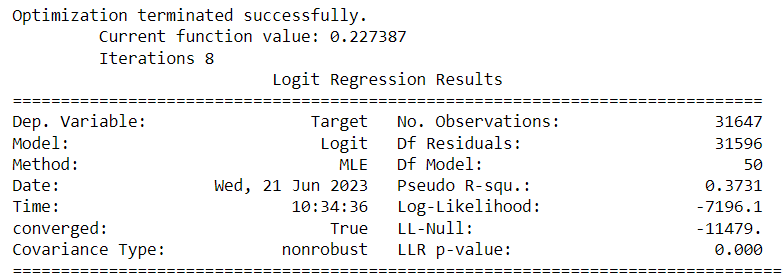
* Feeding the data in to train our model

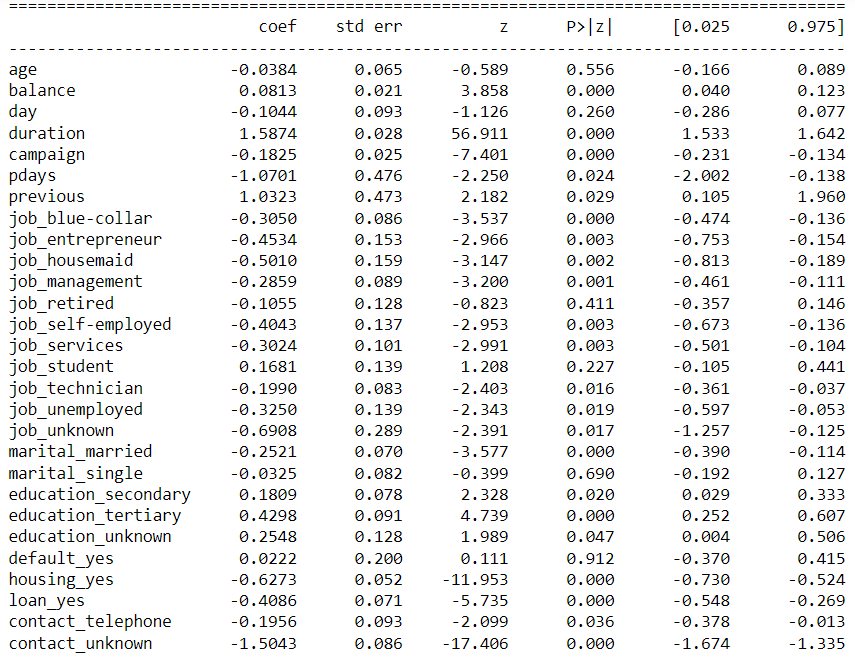


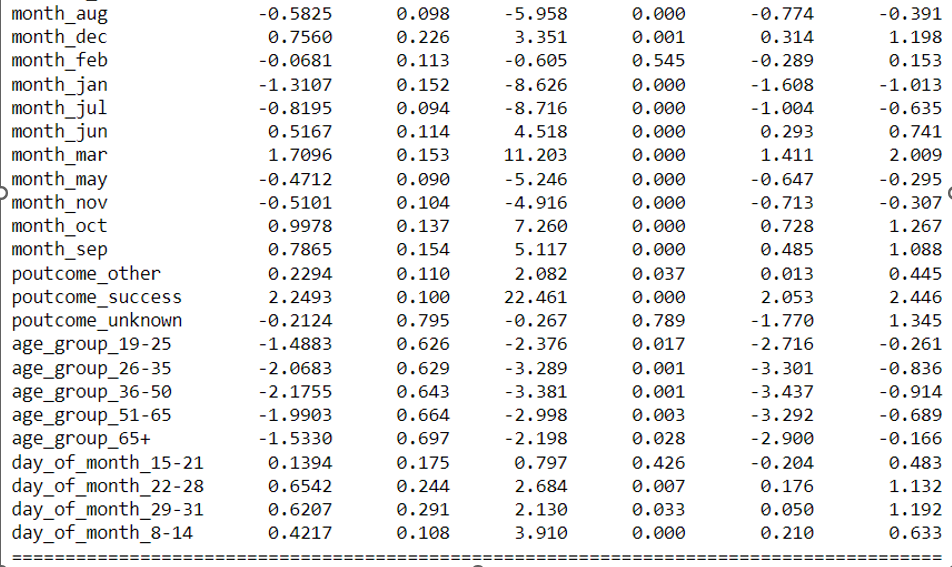
**Analysis and Scoring:**

Now that we have split our model in training and testing segments, and trained our model using the training segment, we shall use various methods to check performance.

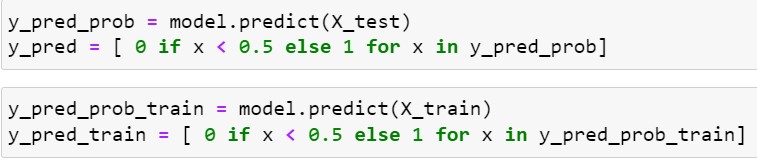
First, we use the model.summary() function to output a table of various results from our Logistic Regression Classifier.



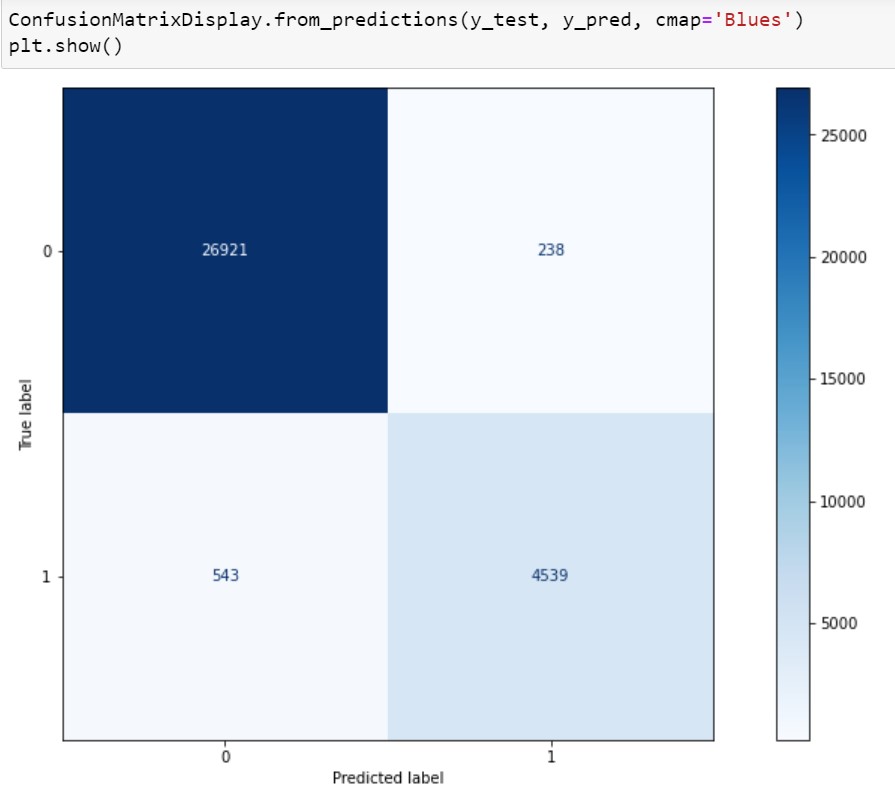




Next, given the classification nature of this model, we shall use a standard cut-off of 0.5 to put predictions into the ‘Rejected’/’Not Rejected’ buckets

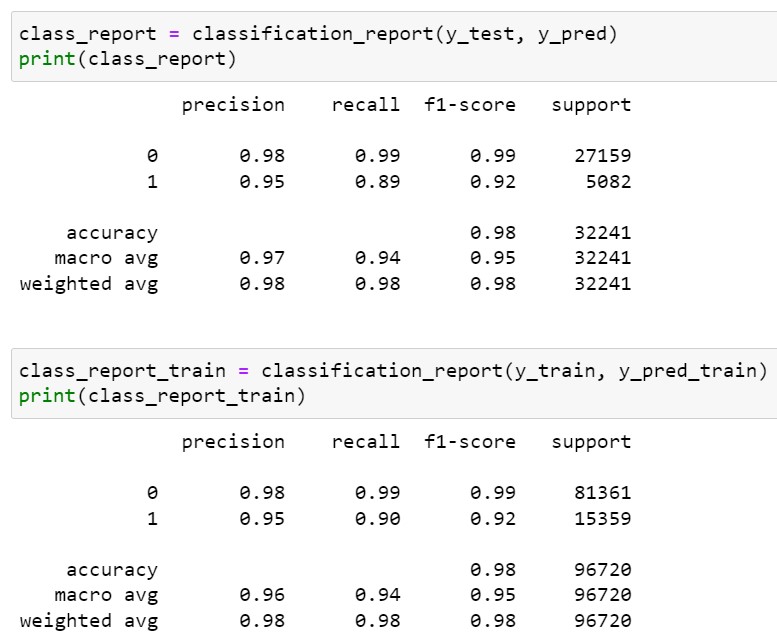


As part of performance scoring, let us look at the confusion matrix, classification report and roc\_auc score.

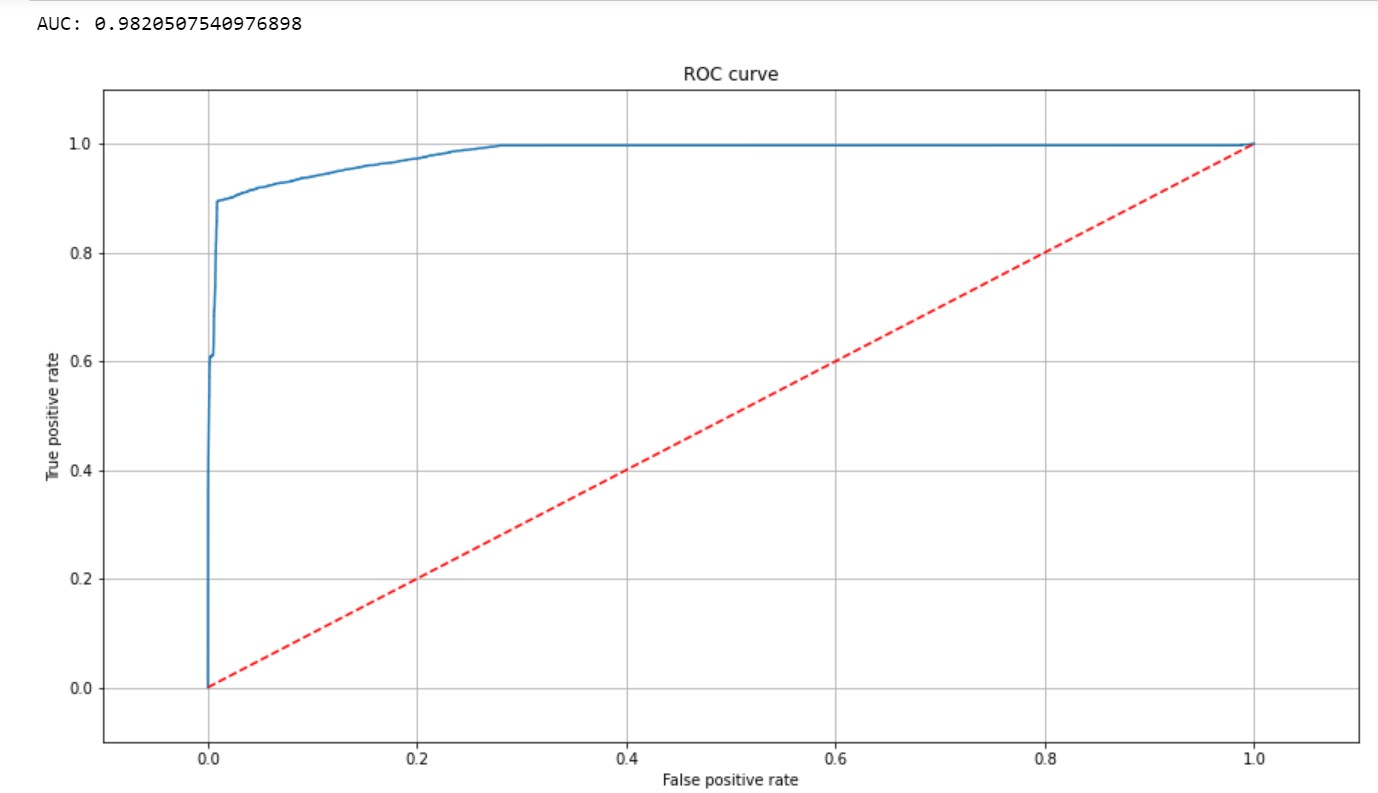


Below we can see that class wise and overall accuracy, precision and f1-score of the model on the training and testing set is broadly similar. However the model’s ability to accurately classify class 1 (‘Rejected’) values seems weaker.

Our base model’s relative weakness in accurately predicting when order rejections might occur are a key reason for considering improvement techniques and alternative classifiers.



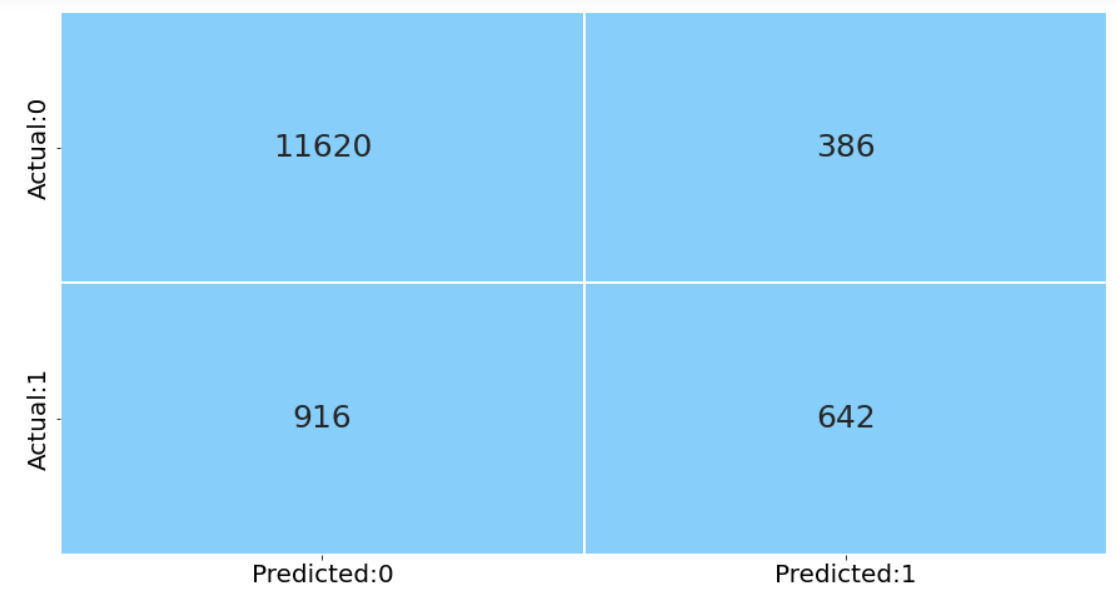
Finally, our model receives an AUC score of .9820. Given the range of AUC from 0 to 1, on this metric, we can conclude that our model has good performance.



**Random Forest**

Random forest is a commonly-used machine learning algorithm, which combines the output of multiple decision trees to reach a single result. Its ease of use and flexibility have fuelled its adoption, as it handles both classification and regression problems.

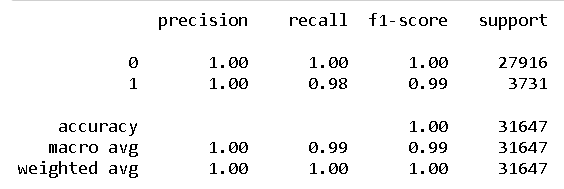
**Confusion Matrix**

****

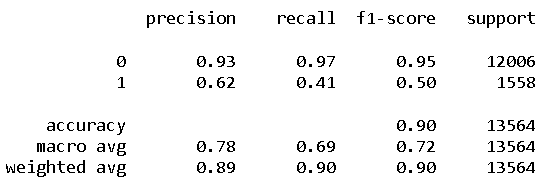
As part of performance scoring, let us look at the confusion matrix, classification report and roc\_auc score.

**We create generalised function to calculate the metrics of the train and test set.**

**Training data**

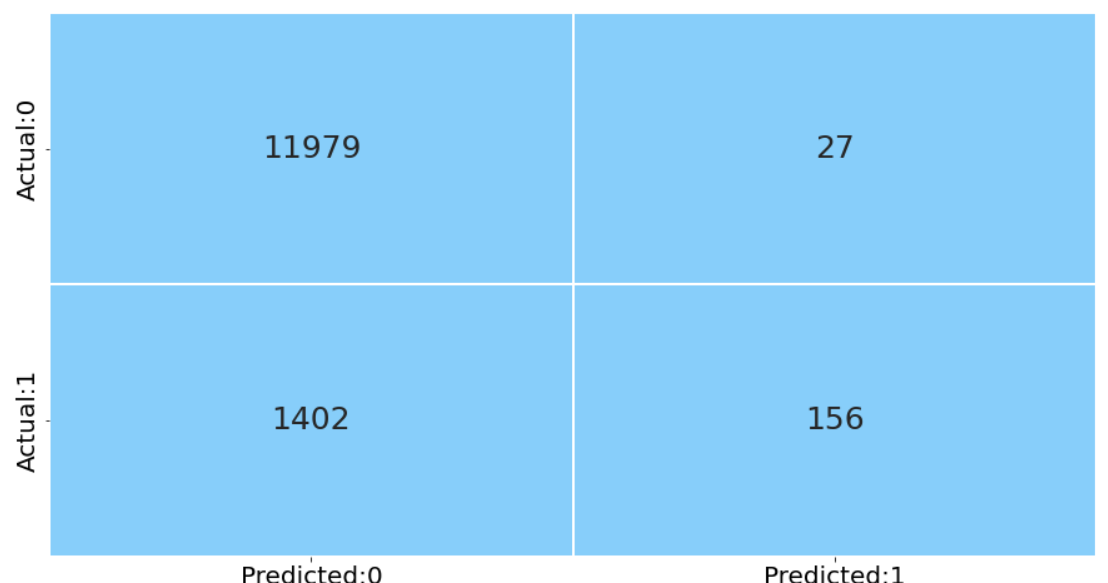
****

**Test data**

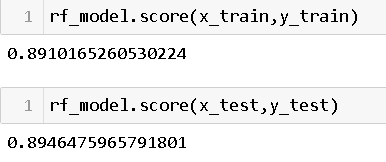
****

We can see that, the accuracy of train and test data are different, hence we need to tune the model and then see the accuracy.

**Tune the Hyperparameter using Grid Search CV [Random Forest]**

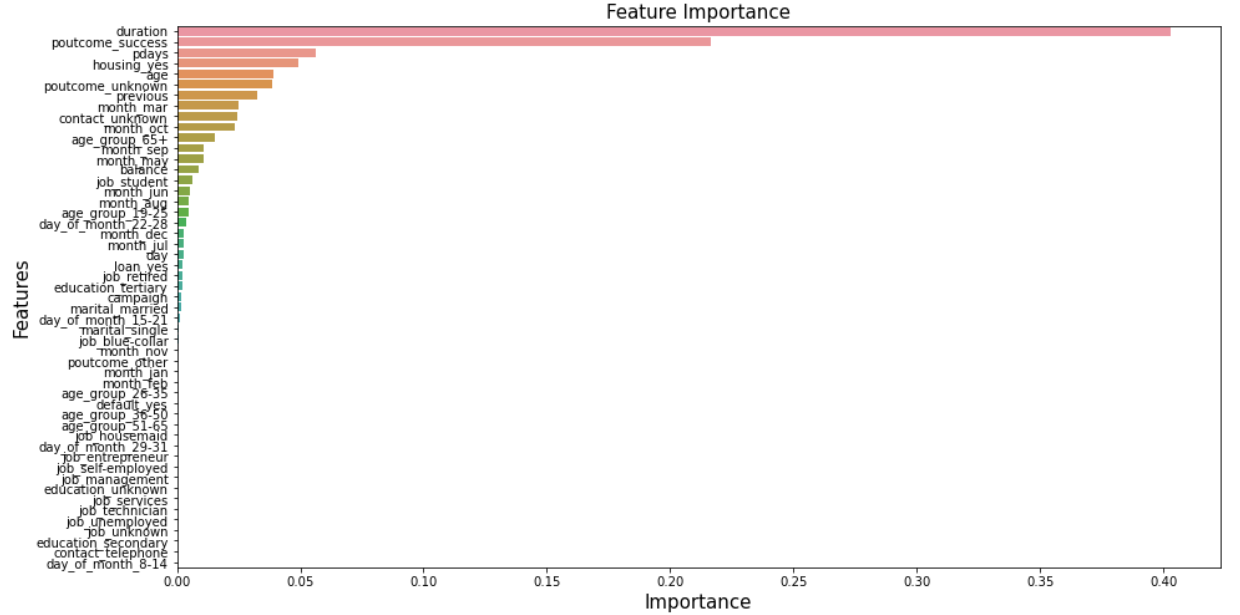
****

As part of performance scoring, let us look at the confusion matrix, classification report and roc\_auc score.

****

We can see that after performing Tuning accuracy of both train and test dataset are almost equal, hence we can proceed with the model for prediction.

Feature Importance graph represents those features on which the dependent variable is dependent the highest.

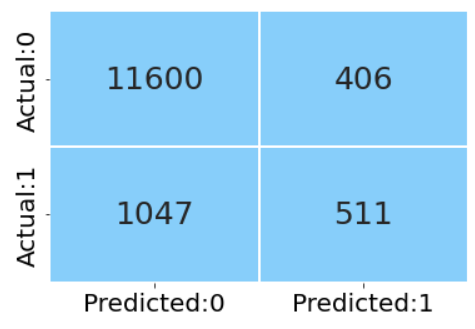


We can see from our outcome that for this prediction model Duration, Poutcome\_success, Pdays are the most important features, on which the dependent variable is relying.

**KNN**

The k-nearest neighbours algorithm, also known as KNN or k-NN, is a non-parametric, supervised learning classifier, which uses proximity to make classifications or predictions about the grouping of an individual data point.

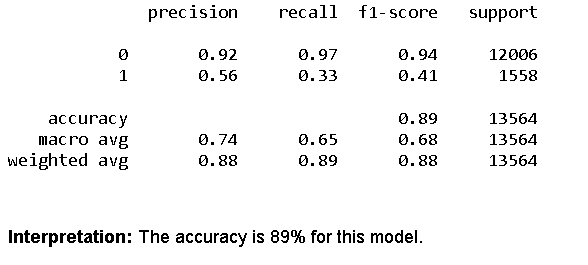
**Confusion Matrix**

****

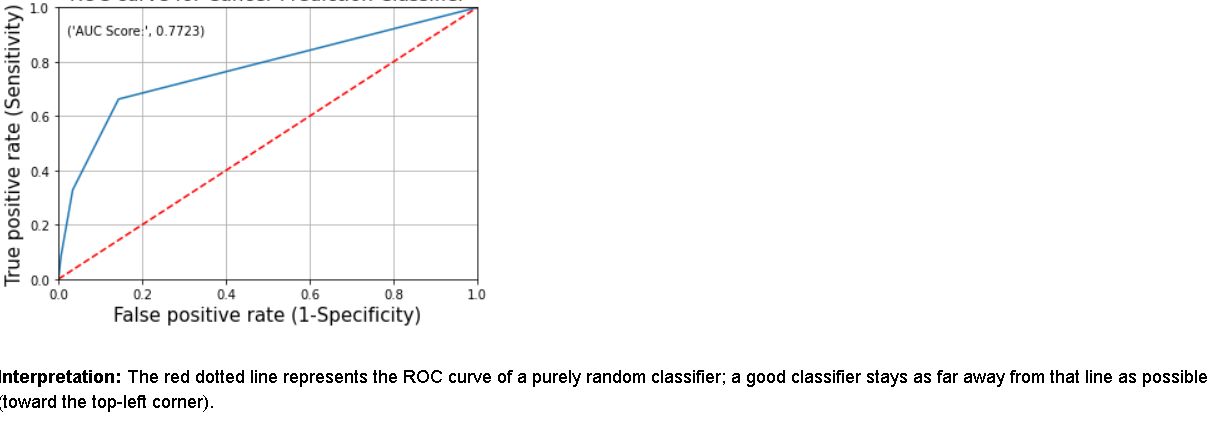
As part of performance scoring, let us look at the confusion matrix, classification report and roc\_auc score.

**We create generalised function to calculate the metrics of the train and test set.**

**Train data**

****

**Test data**

****

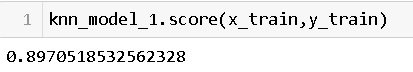
We can see that, the accuracy of train and test data are different, hence we need to tune the model and then see the accuracy.

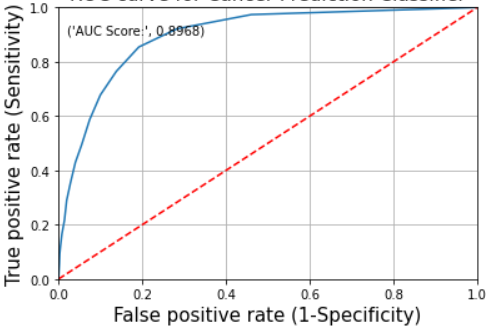
**Optimum value of K using Grid Search**

**Train data**

****

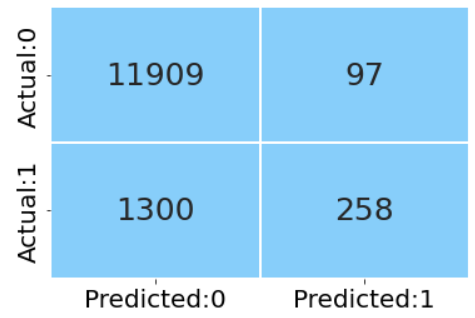
We can see that accuracy of train set is changed to 0.8975.

****

****

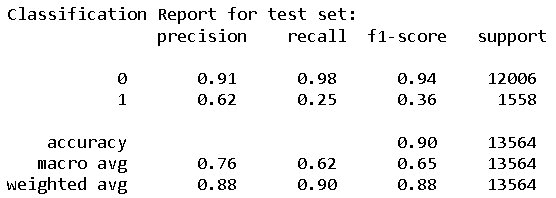
**Test data**

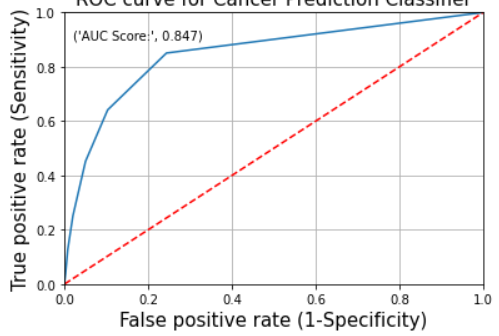
**Confusion Matrix**

****

As part of performance scoring, let us look at the confusion matrix, classification report and roc\_auc score.

**Tune the Hyperparameter using Grid Search CV [Random Forest]**

****

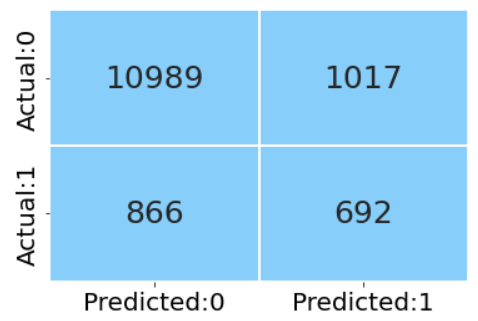
****

We can see that after performing Tuning accuracy of both train and test dataset are almost equal, hence we can proceed with the model for prediction.

**Naives Bayes**

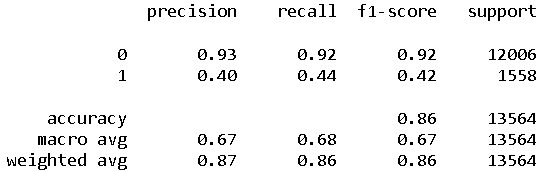
The Naive Bayes Algorithm is one of the crucial algorithms in machine learning that helps with classification problems. It is derived from Bayes' probability theory and is used for text classification, where you train high-dimensional datasets.

**Confusion Matrix**

****

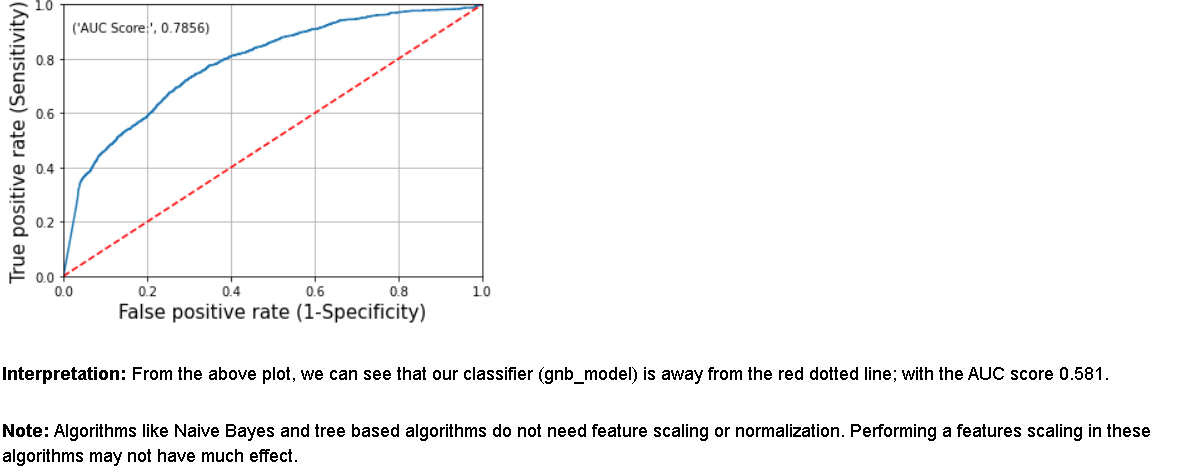
As part of performance scoring, let us look at the confusion matrix, classification report and roc\_auc score.

**We create generalised function to calculate the metrics of the train and test set.**

****

We can see that, the accuracy of train and test data are different, hence we need to tune the model and then see the accuracy.

****

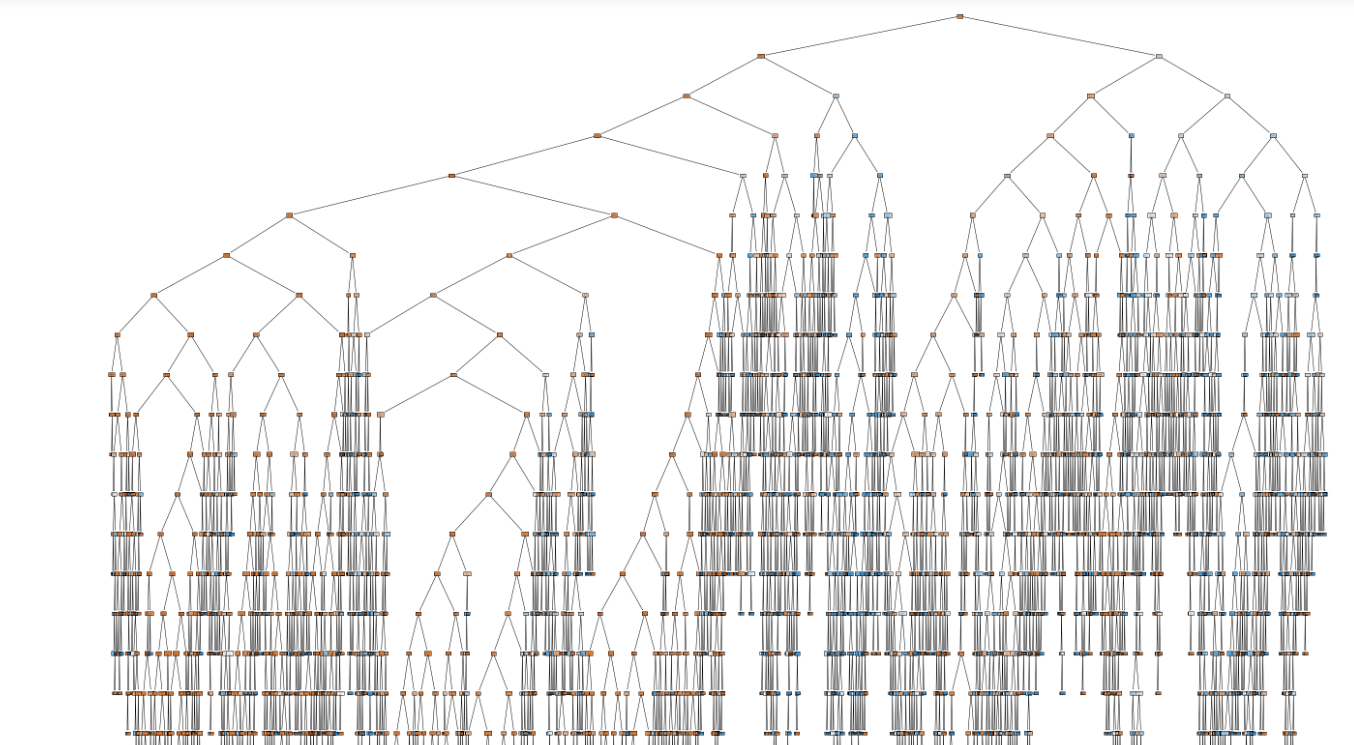
****

We can see that after performing Tuning accuracy of both train and test dataset are almost equal, hence we can proceed with the model for prediction.

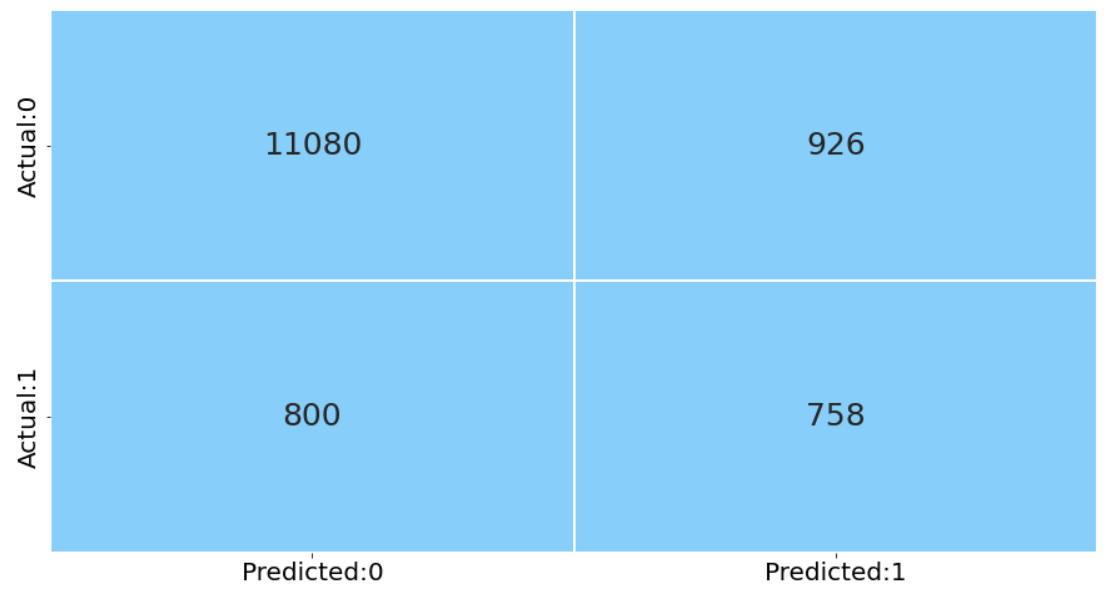
**Decision Tree**

A decision tree is a non-parametric supervised learning algorithm, which is utilized for both classification and regression tasks. It has a hierarchical, tree structure, which consists of a root node, branches, internal nodes and leaf nodes.

**Decision Tree model depicted in the below image.**

****

**Confusion Matrix**

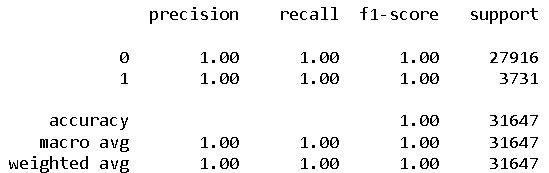
****

As part of performance scoring, let us look at the confusion matrix, classification report and roc\_auc score.

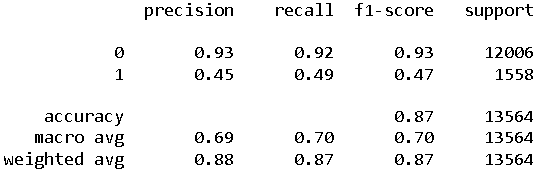
**We create generalised function to calculate the metrics of the train and test set.**

**Overfitting in Decision Tree**

**Train data**

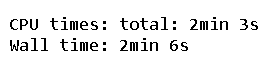
****

**Test data**

****

We can see that, the accuracy of train and test data are different, hence we need to tune the model and then see the accuracy.

**Tune parameters Gridsearch CV**

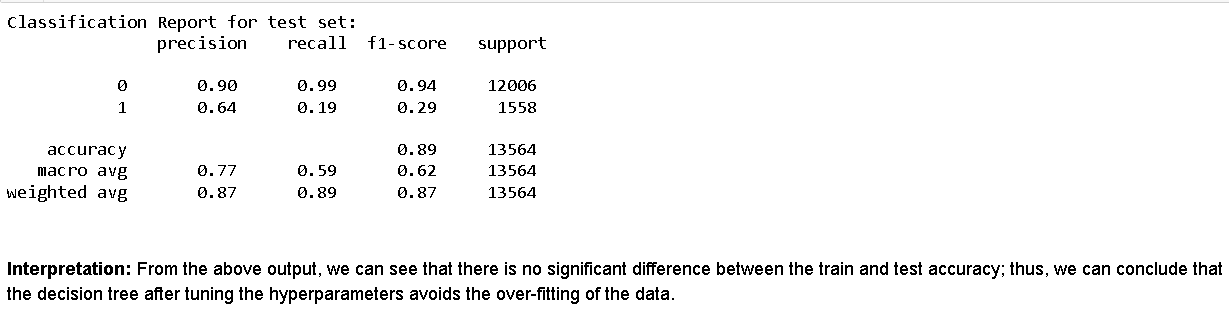
****

**Train data**

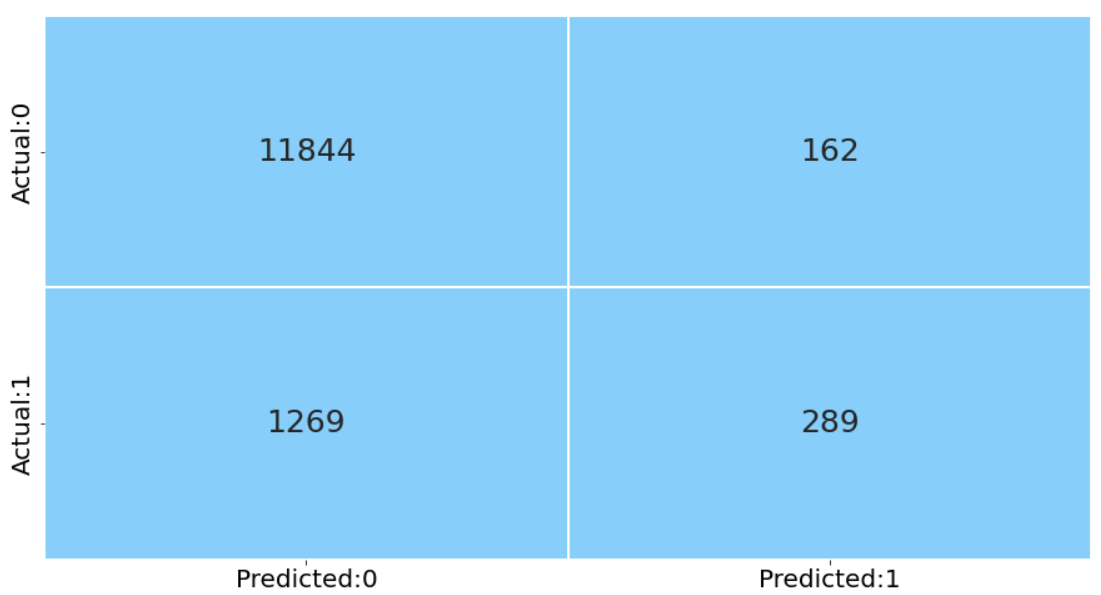
**Tune the Hyperparameter using Grid Search CV [Decision Tree]**

****

**Test data**

****

**Confusion Matrix**

****

As part of performance scoring, let us look at the confusion matrix, classification report and roc\_auc score.

****

We can see that after performing Tuning accuracy of both train and test dataset are almost equal, hence we can proceed with the model for prediction.

**Boosting**

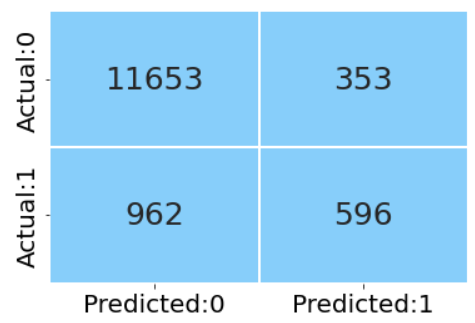
The Ensemble technique considers multiple models for predicting the results. Bagging and Boosting are two of the types of ensembles. The bagging methods construct the multiple models in parallel; whereas, the boosting methods construct the models sequentially.

Earlier, we have studied one of the bagging (bootstrap aggregating) technique i.e. Random Forest. The boosting method fits multiple weak classifiers to create a strong classifier. In this method, the model tries to correct the errors in the previous model. In this section, we learn some of the boosting methods such as AdaBoost, Gradient Boosting and XGBoost.

**ADABOOST**

Let us build the AdaBoost classifier with decision trees. The model creates several stumps (decision tree with only a single decision node and two leaf nodes) on the train set and predicts the class based on these weak learners (stumps). For the first model, it assigns equal weights to each sample. It assigns the higher weight for the wrongly predicted samples and lower weight for the correctly predicted samples. This method continues till all the observations are correctly classified or the predefined number of stumps is created.

**Confusion Matrix**

****

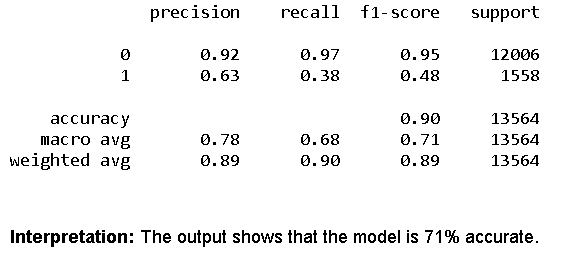
As part of performance scoring, let us look at the confusion matrix, classification report and roc\_auc score.

**We create generalised function to calculate the metrics of the train and test set.**

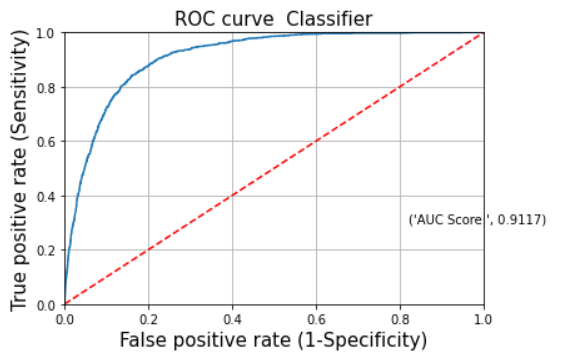
**Train data**

****

**Test data**

****

We can see that, the accuracy of train and test data are different, hence we need to tune the model and then see the accuracy.

****

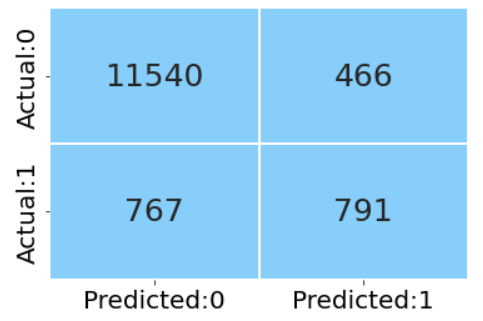
**Interpretation:** The red dotted line represents the ROC curve of a purely random classifier; a good classifier stays as far away from that line as possible (toward the top-left corner).

From the above plot, we can see that the AdaBoost model is away from the dotted line; with the AUC score 0.6532.

**Gradient Boosting**

This method optimizes the differentiable loss function by building the number of weak learners (decision trees) sequentially. It considers the residuals from the previous model and fits the next model to the residuals. The algorithm uses a gradient descent method to minimize the error.

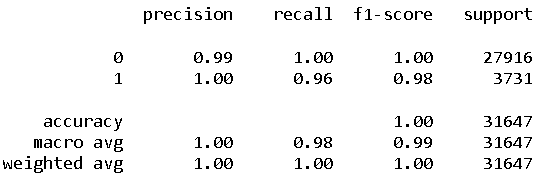
**Confusion Matrix**

****

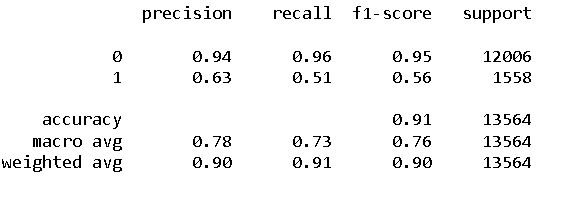
As part of performance scoring, let us look at the confusion matrix, classification report and roc\_auc score.

**We create generalised function to calculate the metrics of the train and test set.**

**Train data**

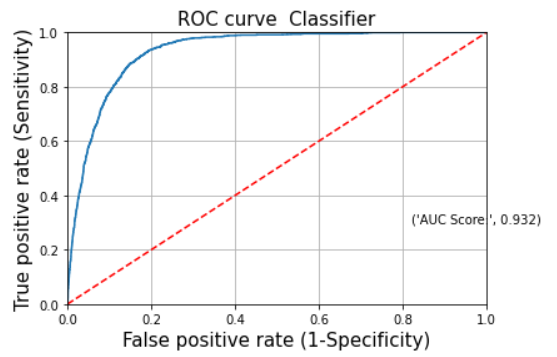
****

**Test data**

****

**Interpretation:** The classification report shows that the model is 71% accurate. Also, the sensitivity and specificity are equal.

We can see that, the accuracy of train and test data are different, hence we need to tune the model and then see the accuracy.

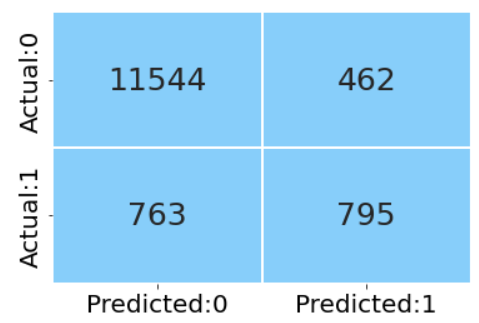
****

**Interpretation:** The red dotted line represents the ROC curve of a purely random classifier; a good classifier stays as far away from that line as possible (toward the top-left corner).  
From the above plot, we can see that the gradient boosting model is away from the dotted line; with the AUC score 0.6554.

**XG boost**

XGBoost (extreme gradient boost) is an alternative form of gradient boosting method. This method generally considers the initial prediction as 0.5 and build the decision tree to predict the residuals. It considers the regularization parameter to avoid overfitting.

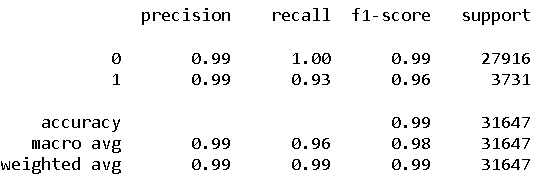
**Confusion Matrix**

****

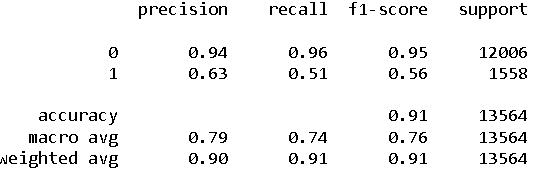
As part of performance scoring, let us look at the confusion matrix, classification report and roc\_auc score.

**We create generalised function to calculate the metrics of the train and test set.**

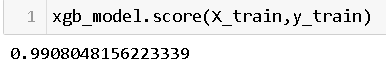
**Train data**

****

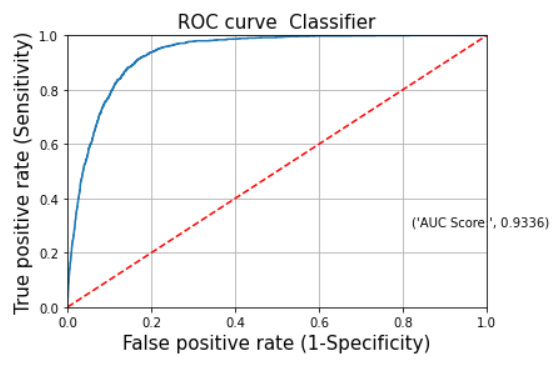
**Test data**

****

We can see that, the accuracy of train and test data are different, hence we need to tune the model and then see the accuracy.

****

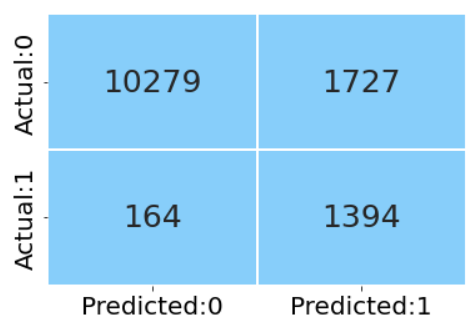
We can see that after performing Tuning accuracy of both train and test dataset are almost equal, hence we can proceed with the model for prediction.

****

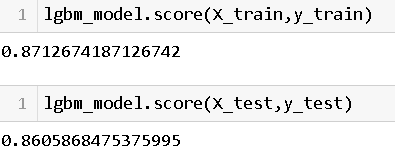
**Interpretation:** The red dotted line represents the ROC curve of a purely random classifier; a good classifier stays as far away from that line as possible (toward the top-left corner).  
From the above plot, we can see that the XGBoost model is away from the dotted line; with the AUC score 0.6461.

**Light Gradient Boost**

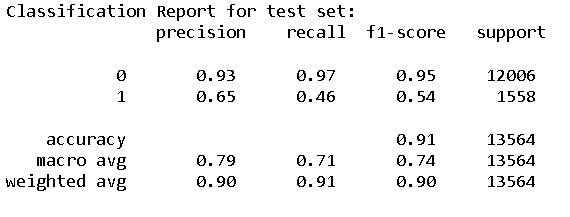
**Confusion Matrix**

****

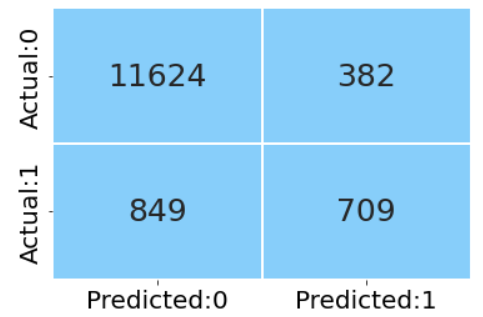
As part of performance scoring, let us look at the confusion matrix, classification report and roc\_auc score.

****

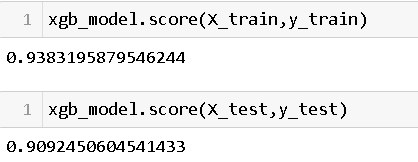
**Tune the Hyperparameter using Grid Search CV [Boost]**

****

**Confusion Matrix**

****

As part of performance scoring, let us look at the confusion matrix, classification report and roc\_auc score.

****

We can see that after performing Tuning accuracy of both train and test dataset are almost equal, hence we can proceed with the model for prediction.

Feature Importance graph represents those features on which the dependent variable is dependent the highest.

**Identify the important features with the help of XG Boost**

****

We can see from our outcome that for this prediction model Poutcome\_success, contact, Housing are the most important features, on which the dependent variable is dependent.

**Future Actions:**

Through the outcome of the models, we understand that the time taken by XG Boost is less and this model is giving highest accuracy. Hence, we proceed to use XG Boost model.

Further, we would like to extend the project by retaining the customers for subscribing to the term deposit. By identifying the features which contribute the most and how we can improve any of the factor with respect to business perspective.