Bank Marketing Effectiveness Prediction – Classification

Tushar Gautam, Data Science Trainee,AlmaBetter, Bangalore.

Abstract:

Data from a marketing campaign run by Portugal is examined. The campaign's aim was to increase customers' subscription rates to fixed-term deposit products. Using knowledge from the course, a number of machine learning algorithms are implemented to answer the question: How can banks successfully market these products in the most efficient way possible and with the highest possible rate if success?

Problem Statement:

The data is related with direct marketing campaigns (phone calls) of a Portuguese banking institution. The marketing campaigns were based on phone calls. Often, more than one contact to the same client was required, in order to access if the product (bank term deposit) would be ('yes') or not ('no') subscribed. The classification goal is to predict if the client will subscribe a term deposit (variable y).

Introduction:

The banking industry provides different types of banking and allied services to its clients, so bank marketing is known for its nature of developing a unique brand image, which is treated as the capital reputation of the financial academy. It is very important for a bank to develop good relationship with valued customers accompanied by innovative ideas which can be used as measures to meet their requirements.

Variables description:

Input variables

age:(numeric): age in years.

job: type of job (categorical): ['admin', 'blue-collar', 'entrepreneur', 'housemaid', 'management', 'retired' , 'self-employed' ,'services' , 'student' , 'technician' , 'unemployed' , 'unknown']

marital: marital status (categorical): 'divorced', 'married', 'single', 'unknown', note: 'divorced' means divorced or widowed.

education :(categorical): ['tertiary', 'secondary', 'primary', 'unknown']

default: has credit in default? (categorical): 'no', 'yes', 'unknown'

housing: has housing loan? (categorical): 'no', 'yes', 'unknown'

loan: has personal loan? (categorical): 'no', 'yes', 'unknown') Related with the last contact of the current campaign

contact: contact communication type (categorical): 'cellular', 'telephone'.

month: last contact month of year (categorical): 'jan', 'feb', 'mar', ..., 'nov', 'dec'

day_of_week: last contact day of the week (categorical): 'mon','tue','wed', 'thu', 'fri'.

duration: last contact duration, in seconds (Numeric).

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; 999 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): 'failure', 'nonexistent', 'success')

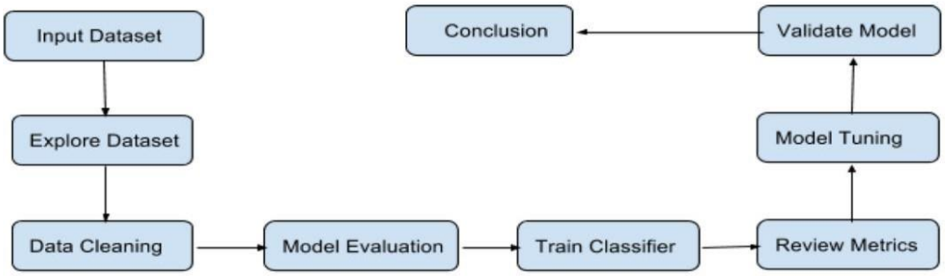
Output variable (desired target): **y** - has the client subscribed a term deposit? (Binary: 'yes', 'no')

This Photo is licensed under CC BY-SA-NC

This Photo is licensed under CC BY

The Bank's Offer Campaign Prediction

Steps Involved:



Data Collection:

Data collection is the process of collecting, measuring and analysing different types of information using a set of standard validated techniques. The main objective of data collection is to gather information-rich and reliable data, and analyse them to make critical business decisions. Once the data is collected, it goes through a rigorous process of data cleaning and data processing to make this data truly useful for businesses. It refers to the process of finding and loading data into our system.

Pandas library is used to loading our data in our system in python. Using pandas we can manipulate data easily.

Data Cleaning:

Data cleaning refers to the process of removing unwanted variables and values from your dataset and getting rid of any irregularities in it. Such anomalies can disproportionately skew the data and hence adversely affect the results. Some steps that can be done to clean data are:

- Handling missing values: There are always some missing values in dataset. If we don't remove or handle those missing values then that can cause a trouble in our analysis. Removing or replacing those missing values with something meaningful is very important so that our data will have no missing values.
- Removing duplicates: Drop the duplicates rows.
- Formatting data to proper dtype.
- Adding or removing columns required for analysis.

Exploratory Data Analysis (EDA):

Exploratory Data Analysis is a data analytics process to understand the data in depth and learn the different data characteristics, often with visual means. This allows you to get a better feel of your data and find useful patterns in it.

It is crucial to understand it in depth before you perform data analysis and run your data through an algorithm. You need to know the patterns in your data and determine which variables are important and which do not play a significant role in the output. Further, some variables may have correlations with other variables. You also need to recognize errors in your data.

All of this can be done with Exploratory Data Analysis. It helps you gather insights and make better sense of the data, and removes irregularities and unnecessary values from data.

Feature Engineering:

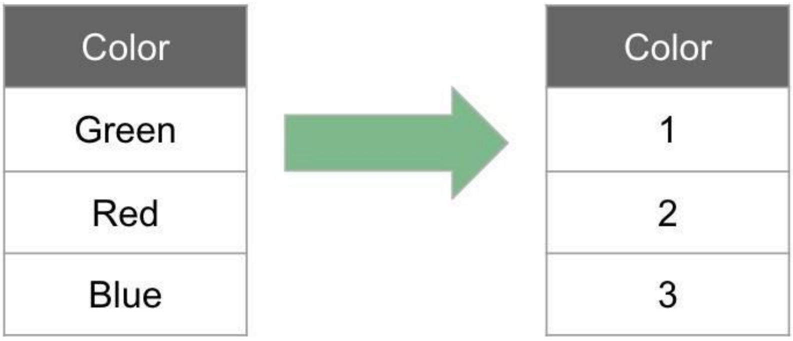
Feature engineering is a machine learning technique that leverages data to create new variables that aren't in the training set. It can produce new features for both supervised and unsupervised learning, with the goal of simplifying and speeding up data transformations while also enhancing model accuracy. Feature engineering is required when working with machine learning models. Regardless of the data or architecture, a terrible feature will have a direct impact on your model.

Feature Engineering is a very important step in machine learning. Feature engineering refers to the process of designing artificial features into an algorithm. These artificial features are then used by that algorithm in order to improve its performance, or in other words reap better results. Data scientists spend most of their time with data, and it becomes important to make models accurate. When feature engineering activities are done correctly, the resulting dataset is optimal and contains all of the important factors that affect the business problem. As a result of these datasets, the most accurate predictive models and the most useful insights are produced.

Encoding Categorical Variables:

In simple words encoding means converting data into required format. Since ML models takes only numerical data to do computation, we will convert all cat variable into numerical data. We used two methods to encode data.

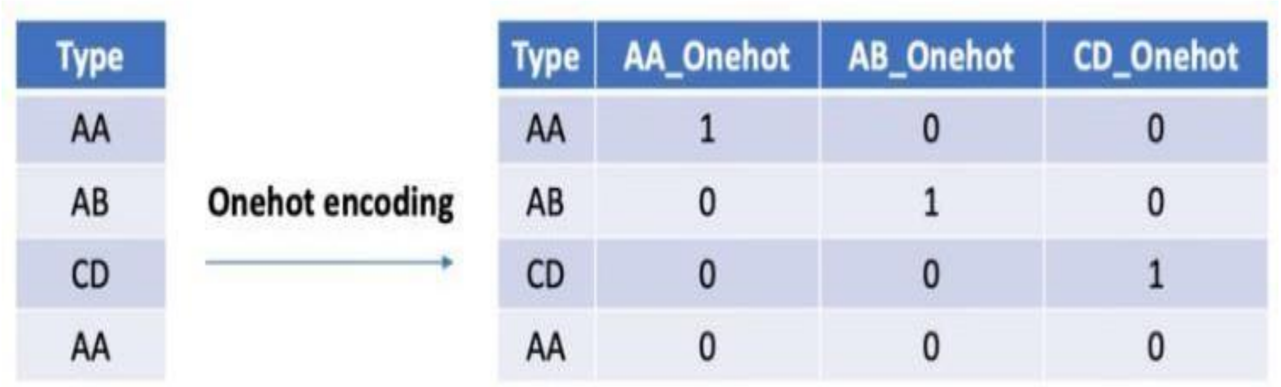
Label Encoding: Label Encoding refers to converting labels to numeric form.



Color
Green
Red
Blue

Color
1
2
3

One Hot Encoding: It is also the process of converting categorical data into numeric data but here we don't give labels to each category instead we create new columns for each category and gives binary values



Type
AA
AB
CD
AA

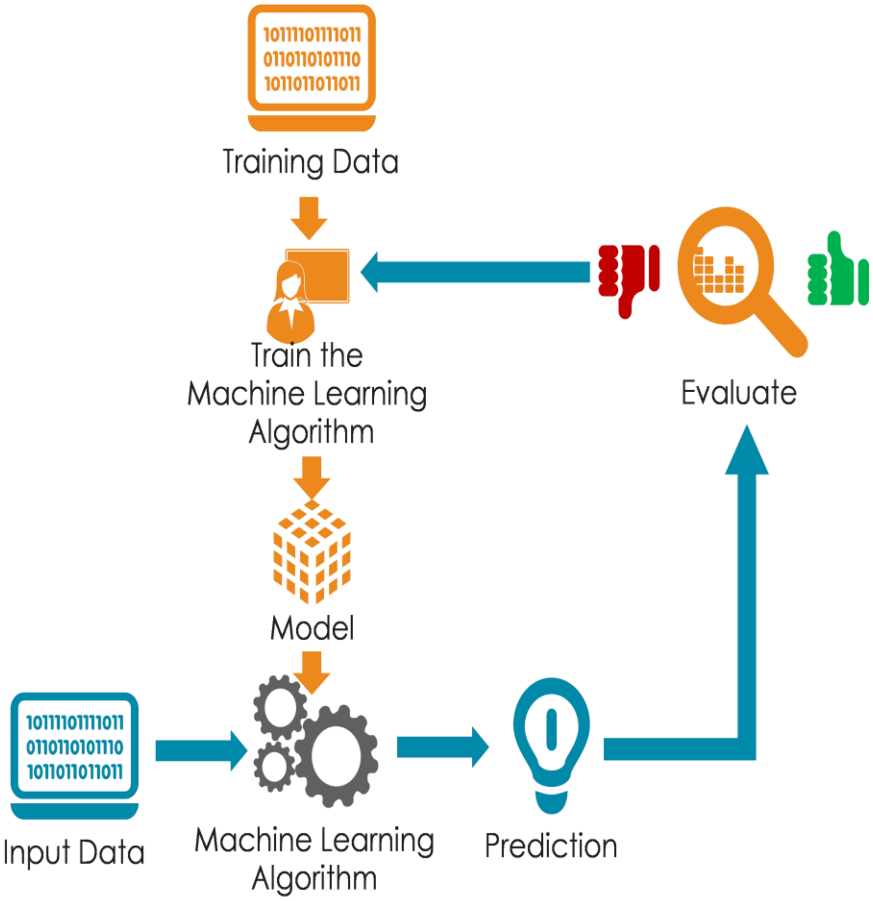
Onehot encoding

Type	AA_Onehot	AB_Onehot	CD_Onehot
AA	1	0	0
AB	0	1	0
CD	0	0	1
AA	0	0	0

Model Training:

Model training is the process of fitting a data into machine learning model from which model learns the patterns in data to predict the dependent variable. Model do it so by assigning a weight to each variable. After our model is trained, we test our model on test data to check how our model is performing.

In our Project, we need to solve a binary Classification problem. For this we use Supervised Learning Binary Classification Model, to predict the Customer's acceptance of term deposit, to train and test our data. Classification is a process of categorizing a given set of data into classes. The process starts with predicting the class of given data points. The classes are often referred to as target, label or categories.
The classification predictive modelling is the task of approximating the mapping function from input variables to discrete output variables. The main goal is to identify which class/category the new data will fall into.

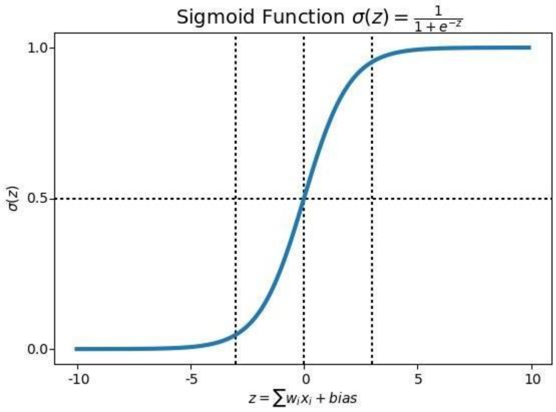


We used four different types of models to train and test performances.

1. Logistic Regression
2. Decision Tree Classifier
3. Random Forest Classifier
4. XGBoost Classifier

Logistic Regression:

Logistic regression is basically a supervised classification algorithm. In a classification problem, the target variable (or output), y, can take only discrete values for a given set of features (or inputs), X.
Contrary to popular belief, logistic regression IS a regression model. The model builds a regression model to predict the probability that a given data entry belongs to the category numbered as “1”. Just like Linear regression assumes that the data follows a linear function, Logistic regression models the data using the sigmoid function.

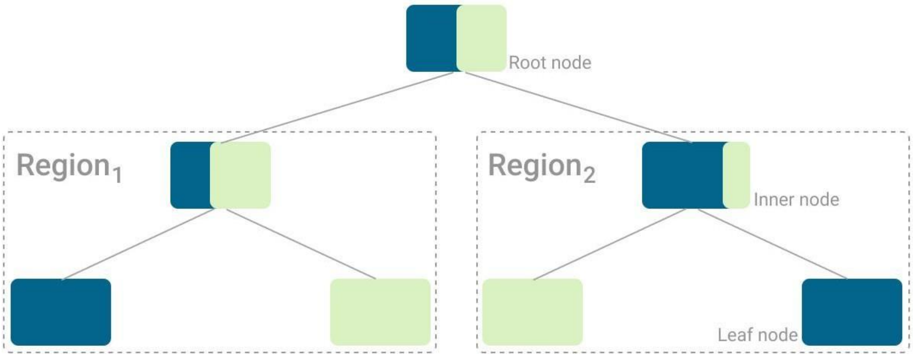


Decision Tree Classifier:

Decision Tree is a Supervised Machine Learning Algorithm that uses a set of rules to make decisions, similarly to how humans make decisions.

The intuition behind Decision Trees is that you use the dataset features to create *yes/no* questions and continually split the dataset until you isolate all data points belonging to each class.

With this process you’re organizing the data in a tree structure. Every time you *ask a question,* you’re adding a node to the tree. And the first node is called the root node. The result of *asking a question* splits the dataset based on the value of a feature, and creates new nodes. If you decide to stop the process after a split, the last nodes created are called leaf nodes.



Random Forest Classifier:

Random Forest is a supervised machine learning algorithm that is composed of individual decision trees. This type of model is called an ensemble model because an “ensemble” of independent models is used to compute a result.

What is a Decision Tree?

The basis for the Random Forest is formed by many individual decision trees, the so-called Decision Trees. A tree consists of different decision levels and branches, which are used to classify data.

The Decision Tree algorithm tries to divide the training data into different classes so that the objects within a class are as similar as possible and the objects of different classes are as different as possible. This results in multiple decision levels and response paths, as in the following example:

XGBoost:

XGBoost is one of the most popular variants of gradient boosting. It is a decision-tree-based ensemble Machine Learning algorithm that uses a gradient boosting framework. XGBoost is basically designed to enhance the performance and speed of a Machine Learning model. In prediction problems involving unstructured data (images, text, etc.), artificial neural networks tend to outperform all other algorithms or frameworks. However, when it comes to small-to-medium structured/tabular data, decision tree-based algorithms are considered bestin-class right now.

XGBoost uses pre-sorted algorithm & histogram-based algorithm for computing the best split. The histogrambased algorithm splits all the data points for a feature into discrete bins and uses these bins to find the split value of the histogram. Also, in XGBoost, the trees can have a varying number of terminal nodes and left weights of the trees that are calculated with less evidence is shrunk more heavily.

Hyperparameter Tuning:

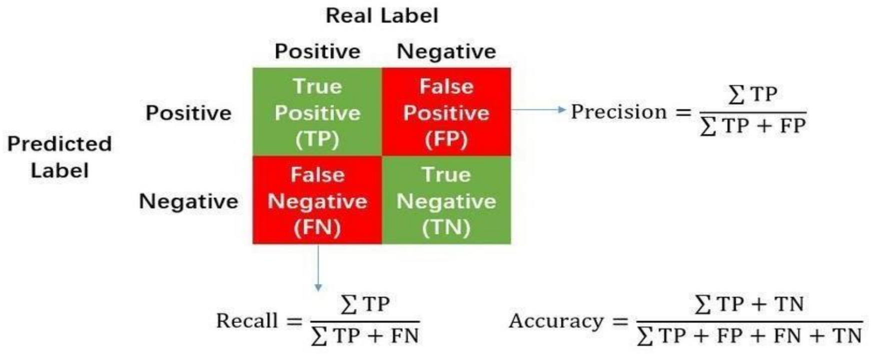
Hyperparameters are sets of information that are used to control the way of learning an algorithm. Their definitions impact parameters of the models, seen as a way of learning, change from the new hyperparameters. This set of values affects performance, stability and interpretation of a model. Each algorithm requires a specific hyperparameters grid that can be adjusted according to the business problem. Hyperparameters alter the way a model learns to trigger this training algorithm after parameters to generate outputs.

Grid Search CV-Grid Search combines a selection of hyperparameters established by the scientist and runs through all of them to evaluate the model’s performance. Its advantage is that it is a simple technique that will go through all the programmed combinations. The biggest disadvantage is that it traverses a specific region of the parameter space and cannot understand which movement or which region of the space is important to optimize the model.

Model Evaluation

Metrics that can provide better insight are:

- Confusion Matrix: A table showing correct predictions and types of incorrect predictions.



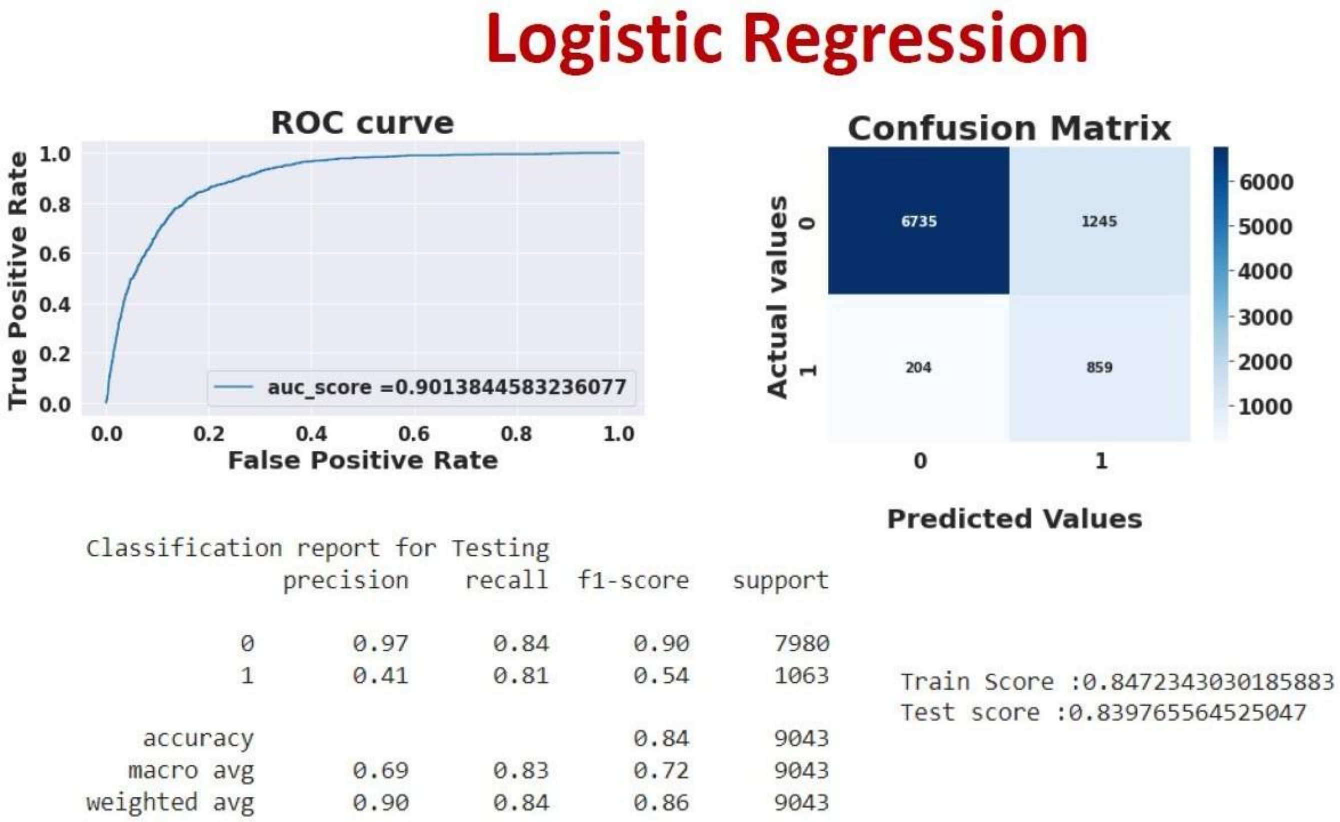
		Real Label	
		Positive	Negative
Predicted Label	Positive	True Positive (TP)	False Positive (FP)
	Negative	False Negative (FN)	True Negative (TN)

$$\text{Precision} = \frac{\sum \text{TP}}{\sum \text{TP} + \text{FP}}$$

$$\text{Recall} = \frac{\sum \text{TP}}{\sum \text{TP} + \text{FN}} \qquad \text{Accuracy} = \frac{\sum \text{TP} + \text{TN}}{\sum \text{TP} + \text{FP} + \text{FN} + \text{TN}}$$

- Precision: the number of true positives divided by all positive predictions. Precision is also called Positive Predictive Value. It is a measure of a classifier’s exactness. Low precision indicates a high number of false positives
- Recall: the number of true positives divided by the number of positive values in the test data. The recall is also called Sensitivity or the True Positive Rate. It is a measure of a classifier’s completeness. Low recall indicates a high number of false negatives.
- F1 Score: the weighted average of precision and recall.
- Area Under ROC Curve (AUC-ROC): AUC-ROC represents the likelihood of your model distinguishing observations from two classes. In other words, if you randomly select one observation from each class, what’s the probability that your model will be able to “rank” them correctly?

Model metrics and Performance.

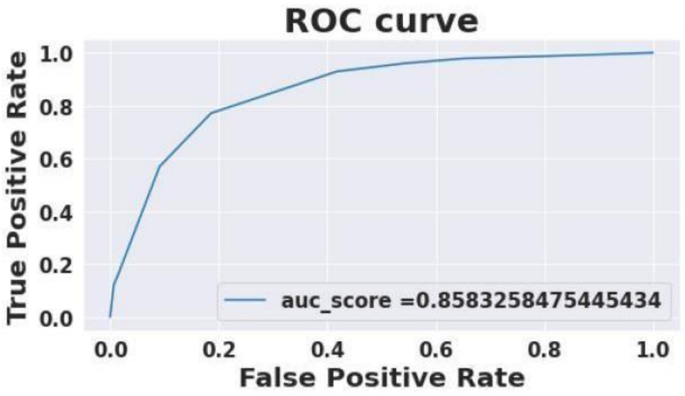


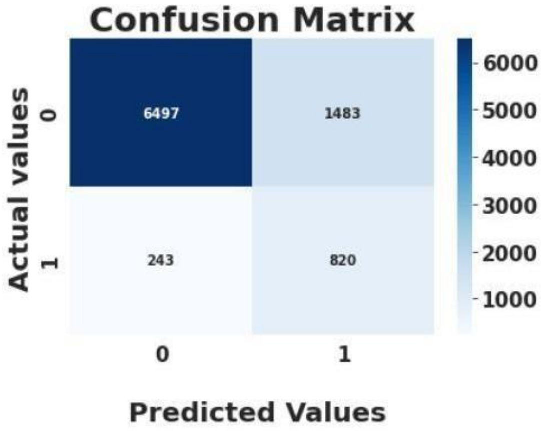
Logistic Regression

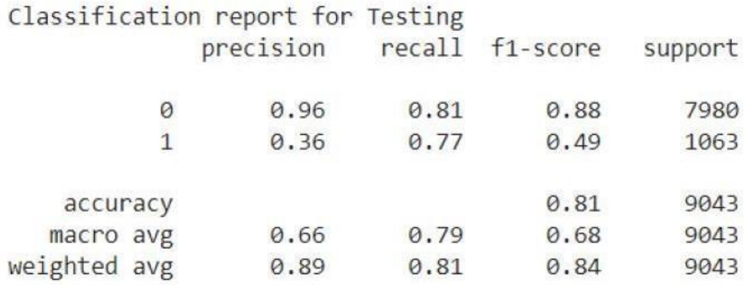
```
Classification report for Testing
              precision    recall  f1-score   support

           0       0.97      0.84      0.90      7980
           1       0.41      0.81      0.54      1063

    accuracy                           0.84      9043
   macro avg       0.69      0.83      0.72      9043
weighted avg       0.90      0.84      0.86      9043
```

```
Train Score :0.8472343030185883
Test score :0.839765564525047
```

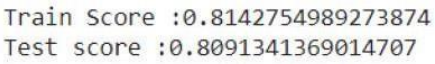
Decision Tree Classifier





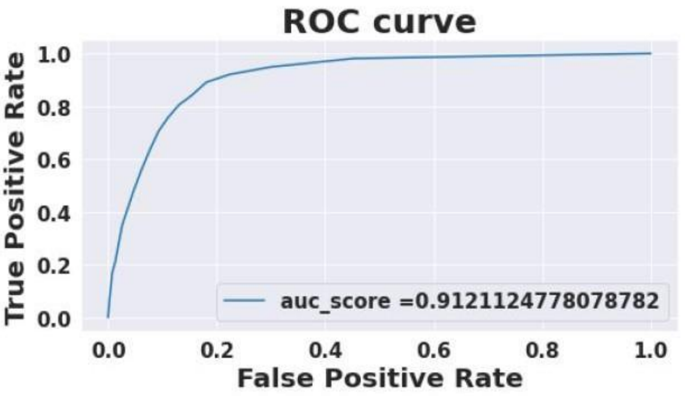


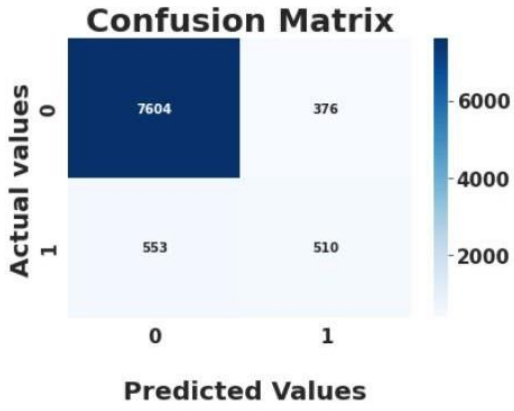
```
Classification report for Testing
              precision    recall  f1-score   support

           0       0.96      0.81      0.88      7980
           1       0.36      0.77      0.49      1063

    accuracy                           0.81      9043
   macro avg       0.66      0.79      0.68      9043
weighted avg       0.89      0.81      0.84      9043
```

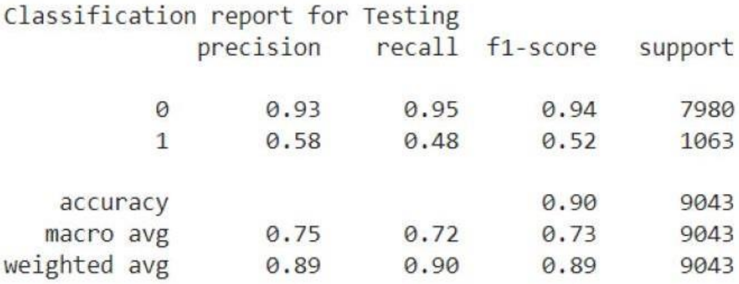


```
Train Score :0.8142754989273874
Test score :0.8091341369014707
```

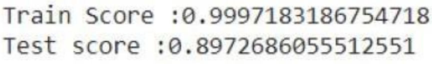
Random Forest Classifier





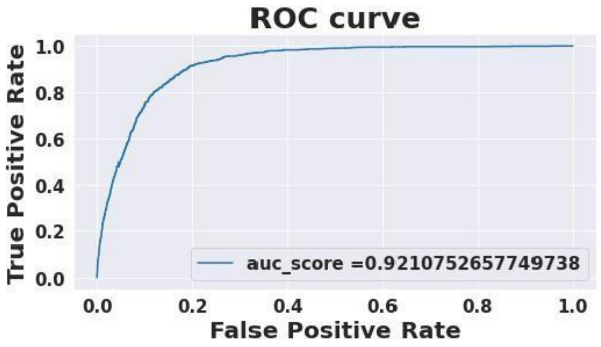


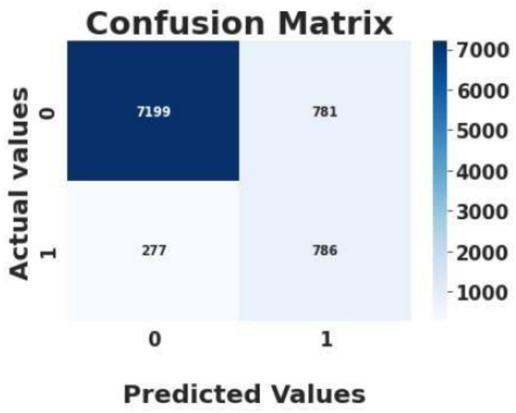
```
Classification report for Testing
              precision    recall  f1-score   support

           0       0.93      0.95      0.94      7980
           1       0.58      0.48      0.52      1063

    accuracy                           0.90      9043
   macro avg       0.75      0.72      0.73      9043
weighted avg       0.89      0.90      0.89      9043
```

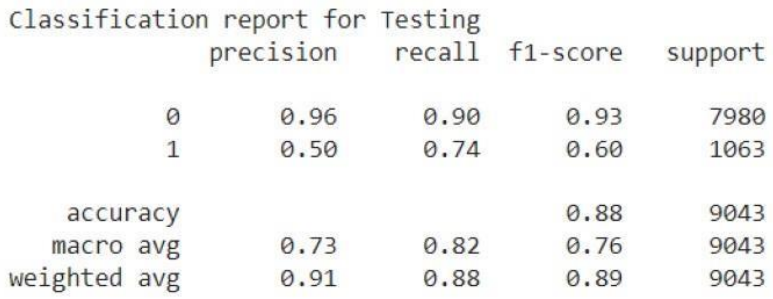


```
Train Score :0.9997183186754718
Test score :0.8972686055512551
```

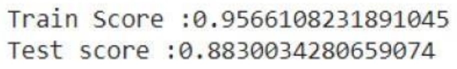
XGBoost Classifier







```
Classification report for Testing
              precision    recall  f1-score   support

           0       0.96      0.90      0.93      7980
           1       0.50      0.74      0.60      1063

    accuracy                           0.88      9043
   macro avg       0.73      0.82      0.76      9043
weighted avg       0.91      0.88      0.89      9043
```



```
Train Score :0.9566108231891045
Test score :0.8830034280659074
```

Conclusion:

- First, we trained our model before handling class imbalance our model performed very good on 0 category and very poor for category 1.
- After solving class imbalance, we trained and compared performances of logistic regression, Decision Tree classifier, Random Forest classifier and Xgboost classifier.
- After tuning hyperparameter Xgboost model gives performance.
  TP = 7199, FP=781, TN = 786 and FN=277) and we got auc score of 0.9210.
- This resulting model can be used by Banking sector to anticipate for further campaigning Techniques. We tuned some parameters and we get XGBoost model gives best results out of all models.

Reference:

https://www.geeksforgeeks.org/understanding-logistic-regression/ https://www.mygreatlearning.com/blog/gradient-boosting/

https://towardsdatascience.com/introduction-to-random-forest-algorithm-fed4b8c8e848