**BANK MARKETING EFFECTIVENESS PREDICTION**

**Jayanth V, Gowthaam Kumarasamy,**

**Data science trainees,**

**Almabetter, Bangalore.**

**Abstract:**

The Portuguese banking institution is effectively conducting marketing campaigns in order to make people subscribe to their term deposit through direct phone calls. We are provided with various independent features which is the reason for the results of the marketing campaigns in the dataset.

Our experiment can help understand what could be the reason for the successful and unsuccessful marketing campaigns influenced by the different features by data analysis, feature selection and engineering, and implementing machine learning models, by taking into account previous trends to find the correct prediction (classification).

**Keywords: Machine learning, classification, marketing, feature engineering, data cleaning, data exploration.**

**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).

You are provided with bank marketing data of 45000+ records. The task is to predict whether a client will subscribe the term deposit for the test set.

### **Bank Client data:**

### age (numeric)

### 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: 'basic.4y','basic.6y','basic.9y','high. school','illiterate','professional. Course','university. Degree','unknown')

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

### housing: has a housing loan? (Categorical: 'no','yes','unknown')

### loan: has a personal loan? (Categorical: 'no','yes','unknown')

### **Related to 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', 'Thus’, 'Fri)

### duration: last contact duration, in seconds (numeric). Important note: this attribute highly affects the output target (e.g., if duration=0 then y='no'). Yet, the duration is not known before a call is performed. Also, after the end of the call y is obviously known. Thus, this input should only be included for benchmark purposes and should be discarded if the intention is to have a realistic predictive model.

### **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; 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').

**2. Introduction:**

The banking institutions conduct marketing campaigns for their new policies and schemes regularly for their customers (people). Here a Portuguese banking institution conducting direct marketing campaigns through the phone calls to their clients. Often the bank approaches to more than one contact for a client. This is to ensure whether the client will subscribe to their term deposit or not.

Our goal here is to build a predictive model (classification) which could help the Portuguese bank to predict whether their customer will subscribe term deposit or not in the future proactively.

**3. Types of term deposits:**

* Fixed deposit
* Recurring deposit

A Fixed Deposit is kept for a longer period and hence it earns a higher rate of interest. A Recurring Deposit takes a defined sum and invests it every defined period. This means each instalment earns interest for a lesser period than the previous instalment. The interest on a Fixed Deposit for the same maturity is more than that on a Recurring Deposit.

However, a Recurring Deposit is a convenient way of investment for people who have a fixed investment amount per month. As such, the investment type depends on the goals and funds available.

**4. Major Factors influencing subscription:**

The reasons for the clients to subscribe for this bank’s term deposit are:

* Age
* Job
* Loans
* Contact
* Previous outcome

**5. How Term deposit works and factors influences:**

Financial institutions e.g., Banks generate their revenue through lending and borrowing. Lending generates profits in for of interest from customers but some level of risk is involved, that is why machine learning algorithms come in handy in predicting clients who are eligible for loans. Another form that generates revenue for financial institutions is borrowing or attracting public' savings into the bank which is a bit less risky than lending.

Borrowing works this: the bank invests the client’s long-term deposits into other sectors which brings better returns, where some is paid to the customers. However when a client does fixed-term deposit, the company gets good returns than savings account as the customer or the client is deprived off the rights to access the money prior to the maturity unless the client is ready to compensate the bank.

Due to this reason, there is stiff competition between banks to convince clients to do term deposits in their banks, and due to these marketing campaigns, a huge amount of money is spent by the banks in reaching out to clients, prospective subscribers and non-prospective ones since the bank doesn’t know who is and who is not. With advancements in data science and machine learning and the availability of data, the banks are adapting to data-driven decisions and this will help in reducing the cost of marketing thus increasing the revenue of the bank.

**6. Steps Involved:**

* **Data Overview and Null values Treatment:**

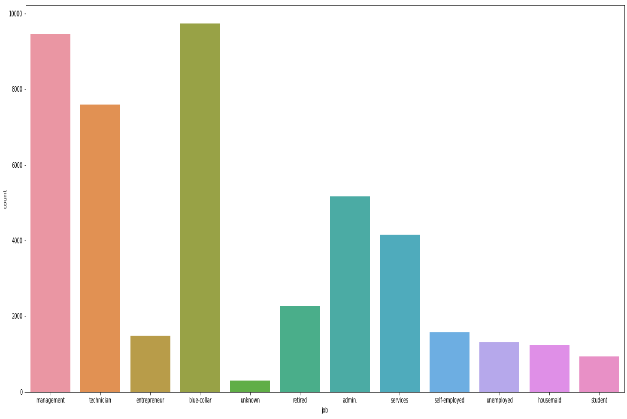
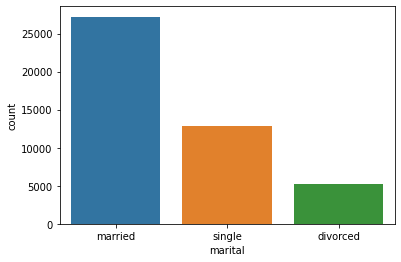
Our dataset contains more than 45000 records and it has almost 17 features of which there are 16 independent features and 1 dependent feature which is the client’s answer (yes/no). The dataset has zero null values and it also has zero duplicated values, thus there is no need for the treatment of the null value.

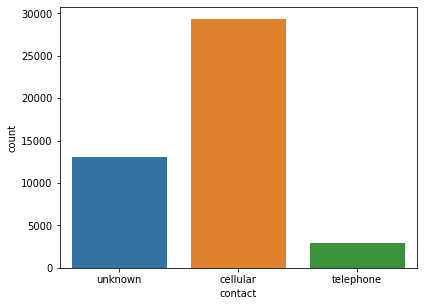
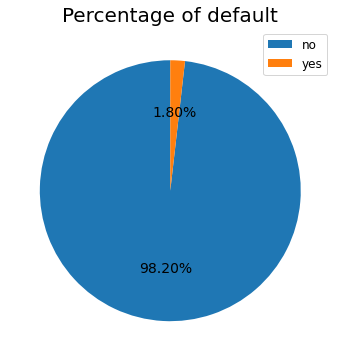
* **Exploratory Data Analysis:**

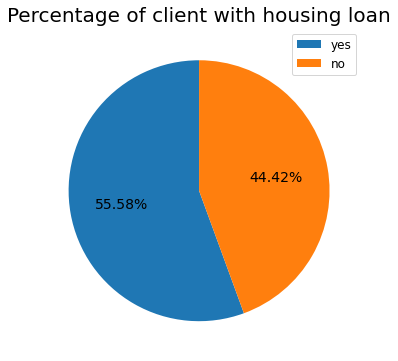
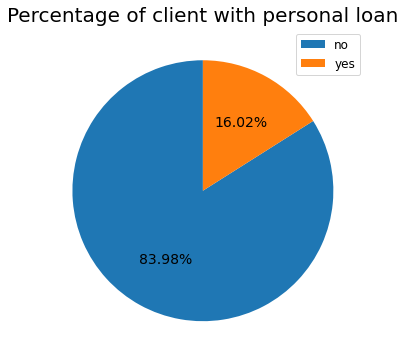
After importing the dataset, we performed univariate and multivariate analysis on columns that has a major impact on our output column. There are various factors influencing the subscription of a term deposit but age, Job, Marital status, Contact, Loans, and defaults make a major impact in availing the term deposit so we tend to perform the univariate and bivariate analysis on them for example :

**a) Univariate Analysis :**

The graph is put out to show how the marital status of a person affects the chance of him getting a term deposit likewise we tend to take a graphical representation of individual columns like for the Job, Contact, Loans and defaults

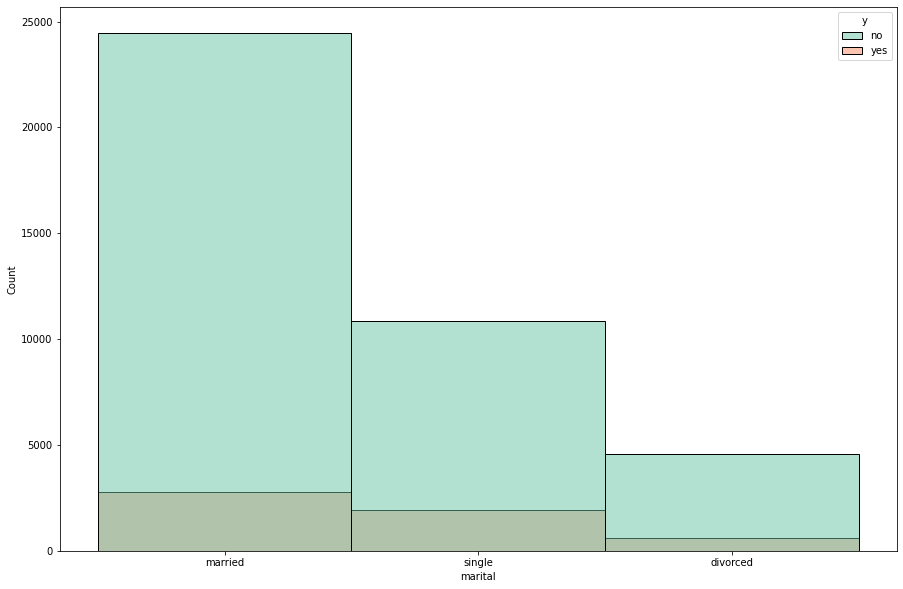
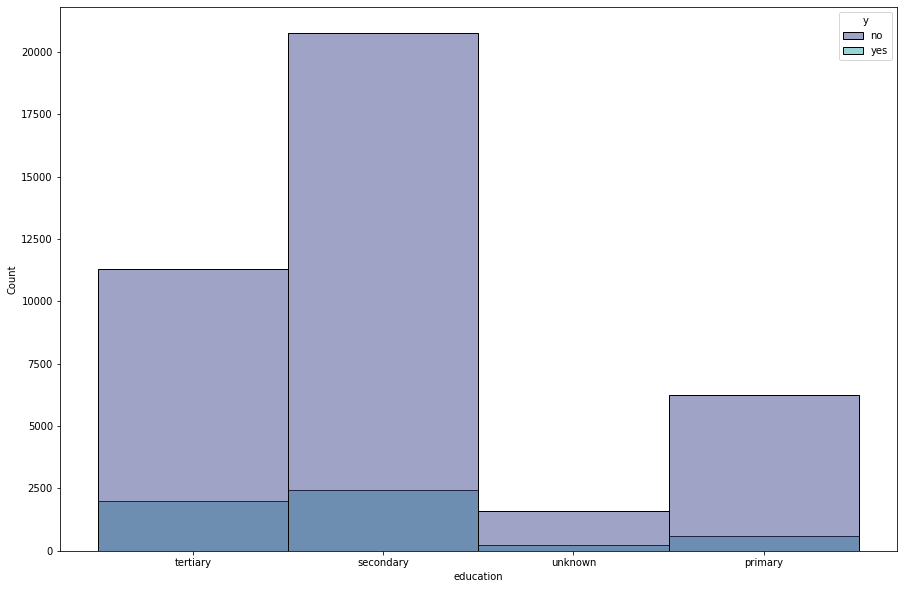
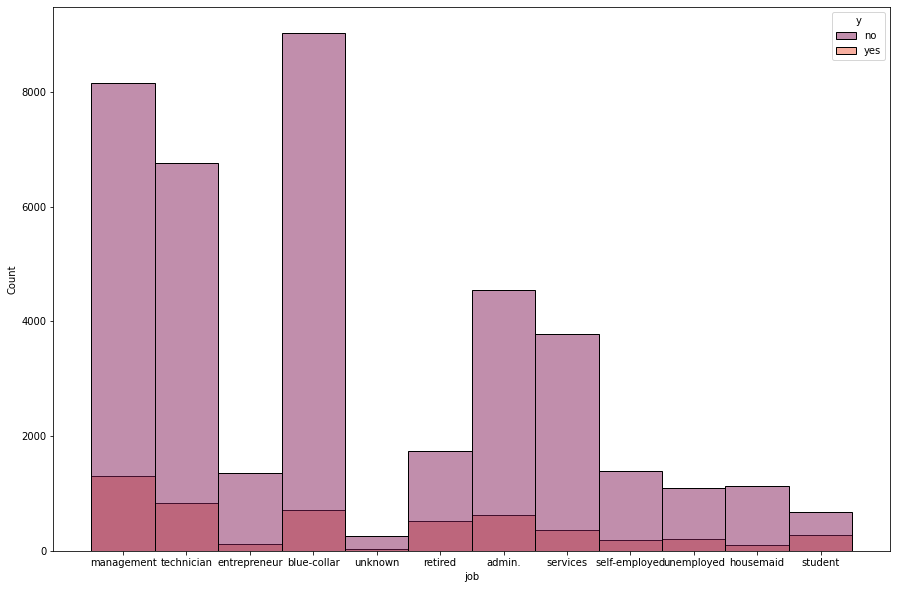


** **

**** ****

**a) Bivariate Analysis :**

Bivariate analysis is one of the simplest forms of quantitative analysis. It involves the analysis of two variables, for the purpose of determining the empirical relationship between them. Bivariate analysis can be helpful in testing simple hypotheses of association. some samples of the analysis done in our project are given below in the form of charts.

********

* **Outlier Treatment**

One of the most important steps as part of data preprocessing is detecting and treating the outliers as they can negatively affect the statistical analysis and the training process of a machine learning algorithm resulting in lower accuracy.

We all have heard of the idiom ‘odd one out’ which means something unusual in comparison to the others in a group.

Similarly, an Outlier is an observation in a given dataset that lies far from the rest of the observations. That means an outlier is vastly larger or smaller than the remaining values in the set.

If our dataset is small, we can detect the outlier by just looking at the dataset. But what if we have a huge dataset, how do we identify the outliers then? We need to use visualization and mathematical techniques.

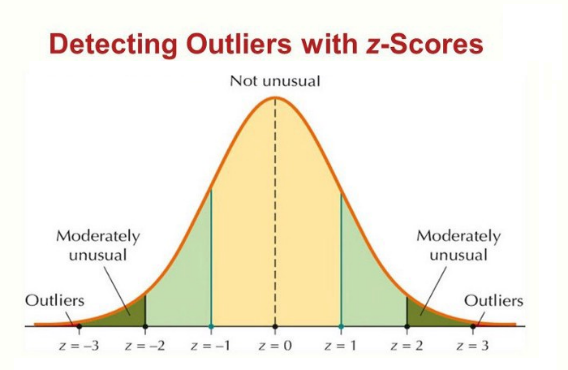
Below are some of the techniques for detecting outliers

* Boxplots
* Z-score
* Inter Quantile Range(IQR)

In our analysis, we tend to use the boxplot for outlier detection and the z-score method to treat the outliers

**Z score and Outliers:**

If the z score of a data point is more than 3, it indicates that the data point is quite different from the other data points. Such a data point can be an outlier.  
For example, In our features the day, previous, campaign, and balance columns are taken into consideration and the outliers were detected and treated.



* **Feature Engineering**

We have used Feature engineering for Manipulation i.e we have combined certain columns that have a high correlation with each other and by doing so we have significantly reduced the number of features in our dataset which will help us to improve our machine learning model training, leading to better performance and greater accuracy for example :

Job feature is been split and combined into various categories such as.

cat\_1 – working-class clients

cat\_2 – self-employed, entrepreneurship

cat\_3 – retired

cat\_4 - not working

and also age has been split and combined into :

age < 25: 'Fresher'

age < 50: 'Stable'

age < 59: 'Retirement stage'

age < 75: 'old age'

age > 75 : 'Counting last days'

* **Standardization of features**

Our main motive through this step was to scale our data into a uniform format that would allow us to utilize the data in a better way while performing fitting and applying different algorithms to it.

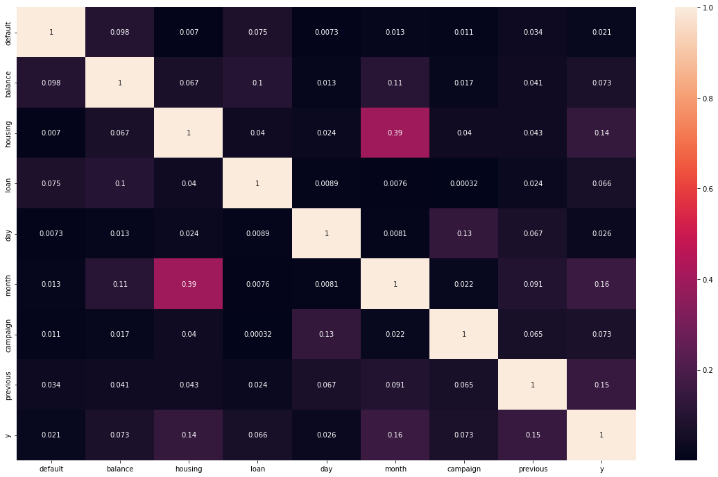
The basic goal was to enforce a level of consistency or uniformity to certain practices or operations within the selected environment since numerical values are essential to carry out ML model implementation we have not performed one hot encoding in the month feature instead we have allocated random numbers for identification. We have removed the other features since they have already been manipulated and stored in a separate column.

Converting minimum categorical values into numerical values for model implementation.

Creating dummy variables for categorical variables- season, month, weekdays, year, holidays, functional day.

* **Multicollinearity**

Multicollinearity occurs when there are two or more independent variables in a multiple regression model, which have a high correlation among themselves. When some features are highly correlated, we might have difficulty distinguishing between their individual effects on the dependent variable. In order to check the correlation f column, we have implemented a technique called VIF (**Variance Inflation Factor)**. Any feature that has a VIF of more than 5 is removed from your training dataset. It is important to note that VIF only works on continuous variables, and not categorical.



* **Handling Class Imbalance**

Before implementing our models we follow one more step that is to check the class imbalance if present then **the machine learning classifier tends to be more biased towards the majority class, causing bad classification of the minority class** hence we tend to balance it by oversampling by a method called **SMOTE (Synthetic Minority Oversampling Technique) and this method helps us** by generating new instances from existing minority cases that we supply as input.

* **Fitting different models**

Before fitting different models we have split the data into samples one for training our data and an another one for training the data

For modeling, we tried various classification algorithms like:

1. **Naïve Bayes Classifier**
2. **KNN Classifier**
3. **SVM Classifier**
4. **Random Forest Classifier**
5. **XG Boost classifier**

* **Tuning the hyperparameters for better accuracy**

Tuning the hyperparameters of respective algorithms is necessary for getting better accuracy and avoiding overfitting in the case of tree-based models like Random Forest Classifier and XG Boost classifier. In our case we have done hyperparameter tuning for XG boost alone sine it has shown more accuracy when it came to predicting the output.

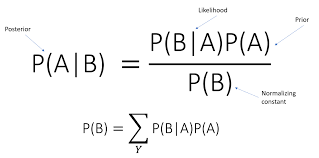
* **SHAP Values for features**

We have applied SHAP value plots on the XG Boost model to determine the features that were most important while model building and the features that didn’t put much weight on the performance of our model.

**7.1. Algorithms:**

**1. Naïve Bayes Classifier :**

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

****

In machine learning, we are often interested in selecting the best hypothesis (h) given data (d).

In a classification problem, our hypothesis (h) may be the class to assign for a new data instance (d).

One of the easiest ways of selecting the most probable hypothesis is given the data that we have that we can use as our prior knowledge about the problem. Bayes’ Theorem provides a way that we can calculate the probability of a hypothesis given our prior knowledge.

Bayes’ Theorem is stated as:

P(h|d) = (P(d|h) \* P(h)) / P(d)

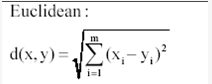
Where

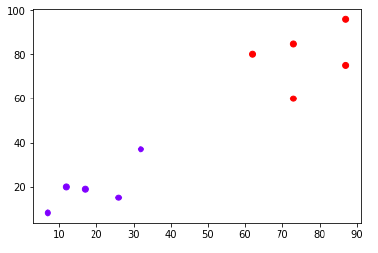
* **P(h|d)** is the probability of hypothesis h given the data d. This is called the posterior probability.
* **P(d|h)** is the probability of data d given that the hypothesis h was true.
* **P(h)** is the probability of hypothesis h being true (regardless of the data). This is called the prior probability of h.
* **P(d)** is the probability of the data (regardless of the hypothesis).

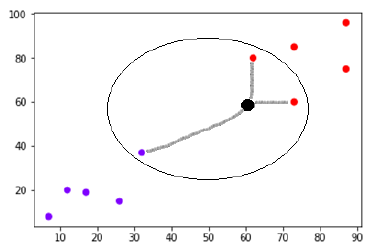
**2. KNN (K- Nearest Neighbour) Classifier :**

K-nearest neighbors (KNN) algorithm uses ‘feature similarity’ to predict the values of new data points which further means that the new data point will be assigned a value based on how closely it matches the points in the training set.

The most used distance metric is the **Euclidean Distance.** This is the geometrical distance that we are using in our daily life. It’s calculated as the square root of the sum of the squared differences between the two points of interest.



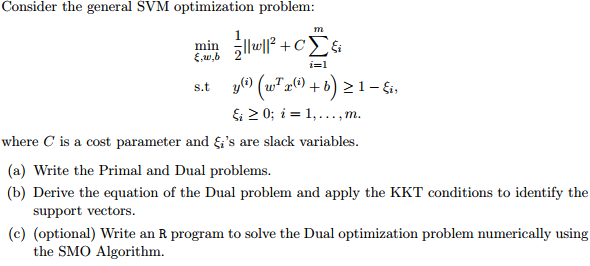


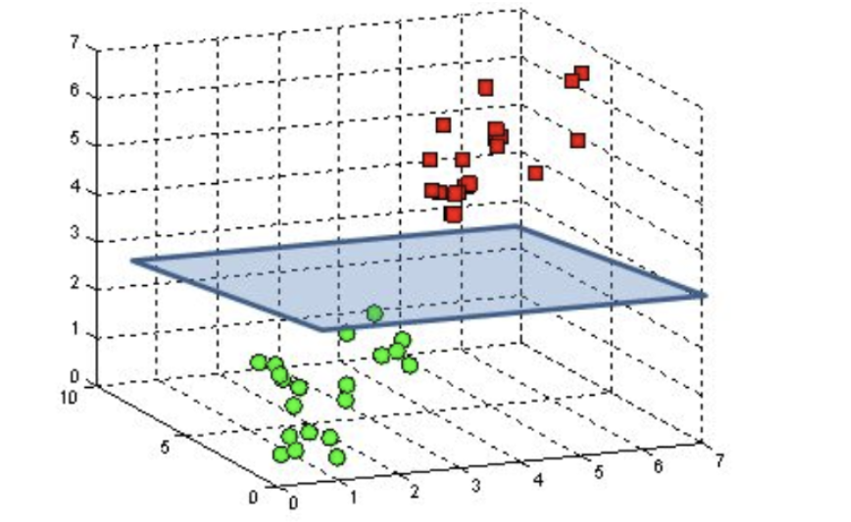


**3. Support Vector Machine Classifier:**

SVM is used mostly when the data cannot be linearly separated by logistic regression and the data has noise. This can be done by separating the data with a hyperplane at a higher-order dimension.

In SVM we use the optimization algorithm as:





We use hinge loss to deal with the noise when the data isn’t linearly separable. Kernel functions can be used to map data to higher dimensions when there is inherent nonlinearity.

**4. Random Forest Classifier:**

Random Forest is a bagging type of Decision Tree Algorithm that creates a number of decision trees from a randomly selected subset of the training set, collects the labels from these subsets, and then averages the final prediction depending on the most number of times a label has been predicted out of all.



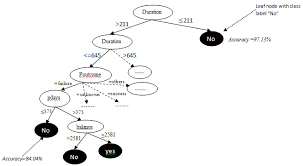
**5. XGBoost :**

**XGBoost** is one of the fastest implementations of gradient boosting. trees. It does this by tackling one of the major inefficiencies of gradient-boosted trees: considering the potential loss for all possible splits to create a new branch (especially if you consider the case where there are thousands of features, and therefore thousands of possible splits). XGBoost tackles this inefficiency by looking at the distribution of features across all data points in a leaf and using this information to reduce the search space of possible feature splits.

To understand XGBoost we have to know gradient boosting beforehand.

* **Gradient Boosting-**

Gradient-boosted trees consider the special case where the simple model is a decision tree

****

In this case, there are going to be 5 kinds of parameters P: the learning rate at each leaf, maximum depth, and Minimum child weight in each tree and some more but the most important one is the learning rate and the number of leaves for our decision tree parameters=[{'learning\_rate': [0.05,0.10,0.15,0.20,0.25,0.30], 'max\_depth'range(3,15),'min\_child\_weight' : [1,3,5,7],'gamma':[0.0,0.1,0.2,0.3,0.4],'colsample\_bytree' : [0.3,0.4,0.5,0.7]}]

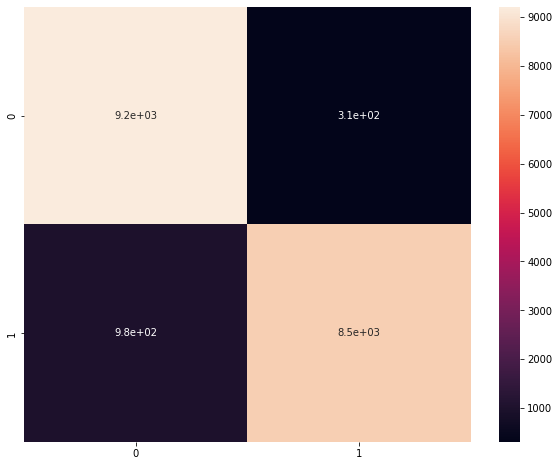
After applying the best\_param function to determine our ideal learning rate value and depth of our leaves we arrive at a conclusion that the 'min child weight': 7, 'max depth': 13, 'learning rate': 0.25, 'gamma': 0.0, 'colsample bytree': 0.3 are the best values for our model to effectively and accurately predict our output values

**7.2. Model performance:**

The model can be evaluated by various metrics such as:

1. **Confusion Matrix**-

The confusion matrix is a table that summarizes how successful the classification model is at predicting examples belonging to various classes. One axis of the confusion matrix is the label that the model predicted, and the other axis is the actual label.



1. **Precision/Recall**-

Precision is the ratio of correct positive predictions to the overall number of positive predictions: TP/TP+FP

The recall is the ratio of correct positive predictions to the overall number of positive examples in the set: TP/FN+TP

1. **Accuracy**-

Accuracy is given by the number of correctly classified examples divided by the total number of classified examples. In terms of the confusion matrix, it is given by: TP+TN/TP+TN+FP+FN

1. **Area under ROC Curve(AUC)**-

ROC curves use a combination of the true positive rate (the proportion of positive examples predicted correctly, defined exactly as recall) and the false positive rate (the proportion of negative examples predicted incorrectly) to build up a summary picture of the classification performance.

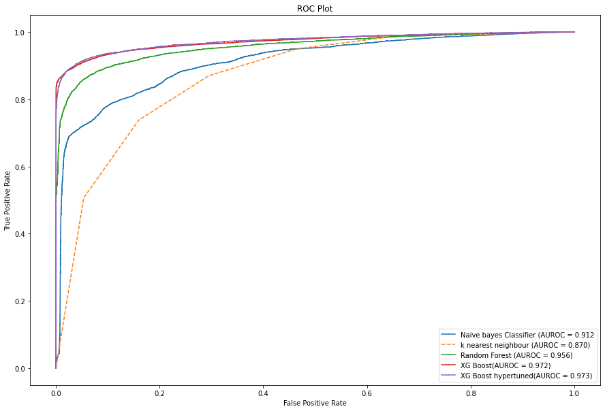
Naive bayes : AUROC = 0.912

k nearest neighbour: AUROC = 0.870

Random Forest: AUROC = 0.956

XG boost: AUROC = 0.972

XG boost hypertunedAUROC = 0.973



**7.3. Hyperparameter tuning:**

Hyperparameters are sets of information that are used to control the way of learning an algorithm. Their definitions impact the parameters of the models, are seen as a way of learning, and change from the new hyperparameters. This set of values affects the 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 generate outputs.

We used Randomized Search CV for hyperparameter tuning. This also results in cross-validation and in our case we divided the dataset into different folds. The best performance improvement among the three was by Bayesian Optimization.

**8. Conclusion:**

That's it! We reached the end of our exercise.

Starting with loading the data so far we have done EDA , null values treatment, encoding of categorical columns, feature selection, and then model building.

In all of these models, our accuracy revolves in the range of 78 to 93%.

And there was a slight improvement in the accuracy score after applying hyperparameter tuning for XG Boost. We have applied hyper parameter tuning only for the best accuracy model. In our case it was the XG boost which had more accuracy in terms of prediction compared to remaining models

# **Gaussian Naive Bayes model accuracy(in %):** 81.75943089023596

# **KNN model accuracy(in %):** 78.80867691999399

**SVM model accuracy(in %):** 61.78216781379709

# **Random Forest Test Accuracy Score(in %):** 0.9049545526191352

# **XG Boost Test Accuracy Score(in %):** 0.9318026585404298

# **XG Boost hypertuned Test accuracy(in %) Score:** 0.9338517312036988

So the accuracy of our best model is 93% which can be said to be good for this large dataset.

**References-**

1. MachineLearningMastery
2. GeeksforGeeks
3. Analytics Vidhya
4. towardsdatascience