A PROJECT REPORT

ON

Bank Product Marketing Analysis

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE AND ENGINEERING

Submitted By

Abhay Kumar (2001921530001)

Under the Supervision of

Ms. Anju Chandna

G.L. BAJAJ INSTITUTE OF TECHNOLOGY & MANAGEMENT, GREATER NOIDA

Affiliated to

DR. APJ ABDUL KALAM TECHNICAL UNIVERSITY, LUCKNOW

2021-22

Declaration

We hereby declare that the project work presented in this report entitled **Bank Product Marketing Analysis** in partial fulfillment of the requirement for the award of the degree of Bachelor of Technology in Computer Science & Engineering, submitted to A.P.J. Abdul Kalam Technical University, Lucknow, is based on my own work carried out at Department of Computer Science & Engineering, G.L. Bajaj Institute of Technology & Management, Greater Noida. The work contained in the report is original and project work reported in this report has not been submitted by me/us for award of any other degree or diploma.

Signature:		Signature:
Name:	Abhay Kumar	Name:
Roll No:	2001921540001	Roll No:
Signature:		Signature:
Name:		Name:
Roll No:		Roll No:

Date:

Place: Greater Noida

Certificate

This is to certify that the Project report entitled **"Bank Product Marketing Analysis" done Abhay Kumar (2001921530001)** is an original work carried out by them in Department of Computer Science & Engineering, G.L Bajaj Institute of Technology & Management, Greater Noida under my guidance. The matter embodied in this project work has not been submitted earlier for the award of any degree or diploma to the best of my knowledge and belief.

Date:

Ms. Anju Chandna
Signature of the Supervisor

Dr. Sansar Singh Chauhan
Head of the Department

Acknowledgement

The merciful guidance bestowed to us by the almighty made us stick out this project to a successful end. We humbly pray with sincere heart for his guidance to continue forever.

We pay thanks to our project guide Ms**. Anju Chandna** who has given guidance and light to us during this project. Her versatile knowledge has cased us in the critical times during the span of this project.

We pay special thanks to our Head of Department Dr. Sansar Singh Chauhan who has been always present as a support and help us in all possible way during this project.

We also take this opportunity to express our gratitude to all those people who have been directly and indirectly with us during the completion of the project.

We want to thanks our friends who have always encouraged us during this project.

At the last but not least thanks to all the faculty of CSE department who provided valuable suggestions during the period of project.

Abstract

Project: Bank Product Marketing Analysis

In this project we are trying to find the best strategies to improve for the next marketing campaign. How can the financial institution have a greater effectiveness for future marketing campaigns? In order to answer this, we must analyse the last marketing campaign the bank performed and identify the patterns that will help us find conclusions in order to develop future strategies.

This analysis will help the business to deploy proper resources on right areas to maximise the output from their resources. Reduce the cost on wasteful advertisements and client outreach. We can pinpoint factors exactly what affects a product's probability of being subscribed in a marketing campaign.

The used dataset pulled from Kaggle is the classic marketing bank dataset uploaded originally in the UCI Machine Learning Repository. The dataset gives us information about a marketing campaign of a financial institution.

Tools and Modules used: -

Programming Language: Python

Data manipulation: Pandas and NumPy

Data representation: seaborn and Matplotlib

Model Building Module: Scikit Learn

Machine Learning Algorithm: -Logistic Regression, Random Forest Classifier, XGBClassifier (Extreme Gradient Boosting Classifier).

TABLE OF CONTENT

LIST OF FIGURES

LIST OF TABLES

Chapter 1

Introduction

1.1 Problem Statement:

Deposits are the main source of revenue for banks. Many banks offer different types of accounts to attract customers willing to deposit their funds. The terms and conditions of depositing depends on the type of account. For instance, current accounts are held by customers willing to withdraw their funds at any time. On the other hand, fixed deposits are held by customers ready to lock their funds for a given period. The rate of interest is one of the motivating factors which encourage individuals to open fixed deposit accounts. A bank can increase the number of subscribers to term deposits through effective marketing. Banks should have an effective marketing campaign strategy to reach their customers. Customer service is one of the marketing techniques that should be applied by banks. In this regard, the bank should ensure that the customers are treated fairly. The response team should assist customers within the shortest time possible. Video content campaigns are also used by various banks to attract customers. The primary objective of the video content campaigns is to ensure that the customers understand the products offered by the bank. If customers do not understand the terms under a fixed deposit account, they may not subscribe to it. Notably, customers are likely to subscribe to something that they know. The choice of a marketing strategy plays an important role in determining the level of subscription by banks.

With the improvements in technology, banks can use Big Data to collect and analyze customer data. This data can be used to identify the likelihood of customers subscribing to term deposits. If the bank realizes that many customers do not understand the terms and conditions of term deposits, the application of direct marketing strategies would be appropriate. The interaction between the bank officials and customers may increase the number of customers who are willing to subscribe to term deposits. Such communications improve customers' understanding of the bank's products.

Studies show that bank marketing campaigns focus on competitive strategies. Some of the strategies include demographic targeting, customer outreach, loyalty programs, and

technology adoption. These strategies not only help the banks to reach many customers but to sell their products to the general public. By targeting a specific group of customers, banks can achieve their organizational objectives. One of the goals is an increase in the number of subscriptions to term deposits (Grzonka et al., 2016). The literature review tries to understand the findings of various researchers. The focus of the review is the factors that can increase customers' subscription to a term deposit.

1.2 Project question

What is the main marketing campaign factor that can increase the customer's decision to subscribe to a term deposit?

1.3 Project objective

To identify the main factor that can increase the customer's subscription to a term deposit.

Chapter 2

EXISTING SYSTEM

1. **Asare-Frempong and Jayabalan (2017)** reveal that direct marketing enables banks to focus on the customers who show the possibility of subscribing to their products and packages. In this case, the researchers intended to find out the determinants of customer response to a direct marketing campaign by applying various classifiers such as Logic Regression, Decision Tree, Random Forest, and Multiplayer Perception Neutral Network (MLPNN). The evaluation of the classifiers shows that there are specific elements that determine the likelihood of a customer subscribing to the bank's products and packages. According to the study carried out by Asare-Frempong and Jayabalan (2017), the Random Forest classifier is the most appropriate in terms of predictive ability. The research shows that the Random Forest represented 87% of the likelihood of customers to subscribe to the bank's products. However, the classification accuracy analysis determined that the possibility of the customers subscribing to a bank product depends on the strategies applied by the bank.

2. **The study of Elsalamony (2014)**, analysis bank direct marketing with the use of data mining techniques. The study emphasized the fact that every marketing campaign is dependent on the customers' data. This simply implies that the information or customer data to be used in achieving every bank marketing campaign is large enough, thereby making it impossible for analysts to reach good decision making. The study went on to introduce the use of a data mining technique to help in achieving its campaigns. Some of the techniques introduced were multilayer perception neural network (MLPNN), tree augmented Naïve Bayes (TAN), Nominal regression or logistic regression (LR), and Ross Quinlan's new decision tree model. Elsalamony (2014), the purpose of introducing these techniques was to check its performance on real-world data knowing that human data is large to handle and in the long run improve campaign effectiveness by highlighting its success characteristics.

3. **Marinakos and Daskalaki (2017)** found that sampling techniques play an important role in different algorithms and cluster-based methods of determining customer response. The researchers focused on a comparison of machine learning techniques and algorithms in 4 predicting customers' response to term deposits. Based on the results of the study, algorithms play an important role in determining the rate at which customers are likely to subscribe to the term deposits of the bank. However, the sampling technique determines the effectiveness of a mathematical or statistical approach. If the data collected for machine learning is more effective than that for an algorithm, the research may determine that machine learning is the best predictive approach. Therefore, banks should ensure that the sampling technique for the cluster-based

procedure is established effectively. Information is the key determinant of the factors that can predict customers' response to term deposits. By using term deposits and the main point of consideration, Marinakos and Daskalaki (2017) revealed that direct marketing should be applied based on the available dataset. In most cases, banks tend to focus on customers' needs and preferences. Likewise, customers focus on the interest rate concerning term deposits. Therefore, banks should utilize distance-based and cluster-based sampling techniques to understand how customers respond to the products. Also, banks can use publicly available data for direct marketing.

4. A study carried out by **Tekouabou, Cherif, and Silkan (2019)** found that the effectiveness of five machine learning techniques depends on the application of effective bank telemarketing. The study focused on five machine learning techniques, including Logistic Regression, Support Vector Machines, Artificial Neutral Network, Decision Tree, and Naïve Bayes. By using accuracy and f-measure analysis, the study found that Artificial Neutral Network and Decision Tree techniques scored above 93% accuracy. The results showed that ANN and DT are important elements in determining customers' subscription to bank products. The application of machine learning techniques is influenced by the availability of the bank to interpret the results and availability of data.

5. A study on the application of selected supervised classification campaign by **Daniel, Grayzna, and Barbara (2016)** revealed that the complexity of the data used by banks could determine the application of data mining techniques. The form of direct marketing by banks influences the ability to get more information from the customers. The study found that more than 50% of the banks used direct marketing strategies for campaigns (Rubtcova and Pavenkov, 2019). These banks recorded an improvement in the use of the decision tree 8 approach in the determination of customer response. The likelihood of a customer to respond to term deposit depends on the sufficiency of the information provided by the bank through campaigns.

Chapter 3

PROBLEM FORMULATION

3.1 Collection of Dataset:
The most important factor in any ML project is Dataset and its quality and features. We have collected the dataset provided by Kaggle. This is the classic marketing bank dataset uploaded originally in the UCI Machine Learning Repository. The dataset gives us information about a marketing campaign of a financial institution which we will have to analyze in order to find ways to look for future strategies in order to improve future marketing campaigns for the bank.

The data set has 11162 data entries and 17 attributes in which 16 attributes are categorical features and 1 which is 'deposit' is the label or output of all these categories.
Features:

1. **age** | int64 | age in years
2. **job** | object | type of job (categorical: ['admin.' 'technician' 'services' 'management' 'retired' 'blue-collar' 'unemployed' 'entrepreneur' 'housemaid' 'unknown' 'self-employed' 'student'])
3. **marital** | object | marital status (categorical: ['married' 'single' 'divorced'])
4. **education** | Object | education background (categorical: ['secondary' 'tertiary' 'primary' 'unknown'])
5. **default** | Object | has credit in default? (Categorical: ['no' 'yes'])
6. **balance** | int64 | Balance of the individual
7. **housing** | object | has housing loan? (Categorical: ['yes' 'no'])
8. **loan** | object | has personal loan? (Categorical: ['no' 'yes'])
9. **contact** | object | contact communication type (categorical: ['unknown' 'cellular' 'telephone'])
10. **day** | int64 | last contact day of the week (categorical: 'mon','tue','wed','thu','fri')
11. **month** | object | last contact month of year (categorical: ['may' 'jun' 'jul' 'aug' 'oct' 'nov' 'dec' 'jan' 'feb' 'mar' 'apr' 'sep'])
12. **duration** | int64 | last contact duration, in seconds (numeric)
13. **campaign** | int64 | number of contacts performed during this campaign and for this client
14. **pdays** | int64 | number of days that passed by after the client was last contacted from a previous campaign (numeric; 999 means client was not previously contacted)
15. **previous** | int64 | number of contacts performed before this campaign and for this client
16. **poutcome** | object | outcome of the previous marketing campaign (categorical: ['unknown' 'other' 'failure' 'success'])
17. **deposit** | object | has the client subscribed to a term deposit? (Binary: 'yes','no')

3.2 Data Processing:
We will need to perform Exploratory Data Analysis to the given dataset as the dataset has a lot of data entries to go through.

Exploratory Data Analysis refers to the critical process of performing initial investigations on data to discover patterns, to spot anomalies, to test hypothesis and to check assumptions with the help of summary statistics and graphical representations.

Our goal for current project will be:

- Find Unwanted Columns
- Find Missing Values
- Find Features with one value
- Explore the Categorical Features
- Find Categorical Feature Distribution
- Relationship between Categorical Features and Label
- Explore the Numerical Features
- Find Discrete Numerical Features
- Relation between Discrete numerical Features and Labels
- Find Continuous Numerical Features
- Distribution of Continuous Numerical Features
- Relation between Continuous numerical Features and Labels
- Find Outliers in numerical features
- Explore the Correlation between numerical features
- Find Pair Plot
- Check the Data set is balanced or not based on target values in classification

Chapter 4

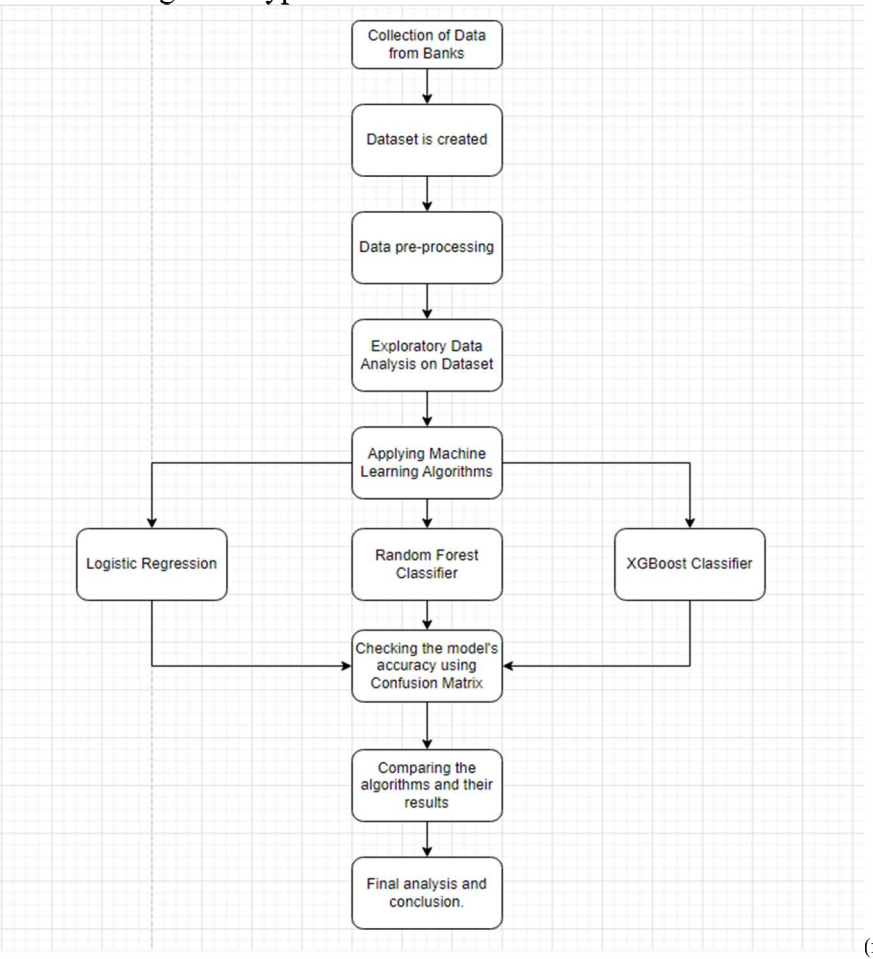
SYSTEM FORMULATION AND DESIGN

4.1 SYSTEM ARCHITECTURE

The system architecture gives an overview of the working of the system, The working of this system is described as follows:

Dataset collection is collecting data which contains patient details. Attributes selection process selects the useful attributes for the prediction of deposits. After identifying the available data resources, they are further selected, cleaned and made into the desired form, Different classification techniques as stated will be applied on preprocessed data to predict the accuracy of deposits. We will use confusion matrix to determine whether the model is classifying the data correctly or not.

The project would include a qualitative comparison of three machine learning models which will be considered as the strategies to be used in this research for the proper implementation of the models outlined for this report. The machine learning models can also be referred to as machine learning classifiers or classification approaches in the study for greater understanding and clarification since the study deals with a classification problem. It could also be said that the research work deals with supervised learning as the dataset to be used as a label for each example and that label is of a categorical type.

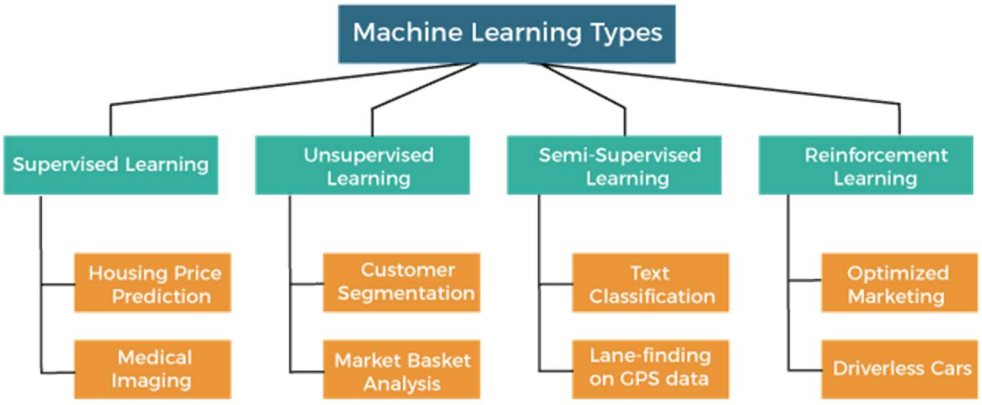


(fig 4.1.a)

4.2 MACHINE LEARNING

Machine learning (ML) is a field of inquiry devoted to understanding and building methods that 'learn', that is, methods that leverage data to improve performance on some set of tasks. It is seen as a part of artificial intelligence, Machine learning algorithms build a model based on sample data, known as training data, in order to make predictions or decisions without being explicitly programmed to do so.

Machine learning programs can perform tasks without being explicitly programmed to do so. It involves computers learning from data provided so that they carry out certain tasks. For simple tasks assigned to computers, it is possible to program algorithms telling the machine how to execute all steps required to solve the problem at hand on the computer's part, no learning is needed. For more advanced tasks, it can be challenging for a human to manually create the needed algorithms, in practice, it can turn out to be more effective to help the machine develop its own algorithm, rather than having human programmers specify every needed step



(fig 4.2.a)

4.2.1 Supervised Learning

Supervised learning algorithms build a mathematical model of a set of data that contains both the inputs and the desired outputs. The data IS known as training data and consists of a set of training examples. Each training example has one or more inputs and the desired output, also known as a supervisory signal in the mathematical model, each training example is represented by an array or vector, sometimes called a feature vector, and the training data is represented by a matrix. Through iterative optimization of an objective function, supervised learning algorithms learn a function that can be used to predict the output associated with new inputs. An optimal function will allow the algorithm to correctly determine the output for inputs that were not a part of the training data, an algorithm that improves the accuracy of its outputs or predictions over time is said to have learned to perform that task.

4.2.2 Unsupervised Learning

Unsupervised learning algorithms take a set of data that contains only inputs, and find structure in the data, like grouping or clustering of data points. The algorithms, therefore, learn from test data that has not been labeled, classified or categorized. Instead of responding to feedback, unsupervised learning algorithms identify commonalities in the data and react based on the presence or absence of such commonalities in each new piece of data. A central application of unsupervised learning is in the field of density estimation in statistics, such as finding the probability density function, though unsupervised learning encompasses other domains involving summarizing and explaining data features,

4.2.3 Semi-supervised Learning

Semi-supervised learning falls between unsupervised learning (without any labeled training data) and supervised learning (with completely labeled training data). Some of the training examples are missing training labels, yet many machine-learning researchers have found that unlabeled data, when used in conjunction with a small amount of labeled data, can produce a considerable improvement in learning accuracy. In weakly supervised learning, the training labels are noisy,

limited, or imprecise; however, these labels are often cheaper to obtain, resulting in larger effective training sets.

4.2.4 Reinforcement Learning

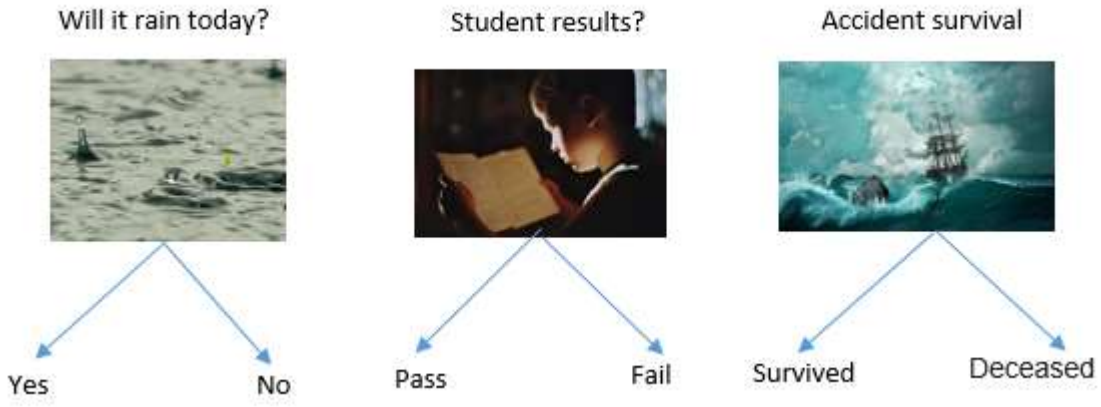
Reinforcement learning is an area of machine learning concerned with how software agents ought to take actions 111 an environment to maximize some notion of cumulative reward. Due to its generality, the field is studied in many other disciplines, such as game theory, control theory, operations research, information theory, simulation-based optimization, multi-agent systems, swarm intelligence, statistics and genetic algorithms. In machine learning, the environment is typically represented as a Markov Decision Process (MDP). Reinforcement learning algorithms do not assume knowledge of an exact mathematical model of the MDP and are used when exact models are infeasible. Reinforcement learning algorithms are used in autonomous vehicles or in learning to play a game against a human opponent.

4.3 ALGORITHMS

4.3.1 Logistic Regression

The logistic regression statistic modeling technique is used when we have a binary outcome variable. For example: given the parameters, will the student pass or fail? Will it rain or not? etc.

So, though we may have continuous or categorical independent variables, we can use the logistic regression modeling technique to predict the outcome when the outcome variable is binary.



(fig 4.3.1.a)

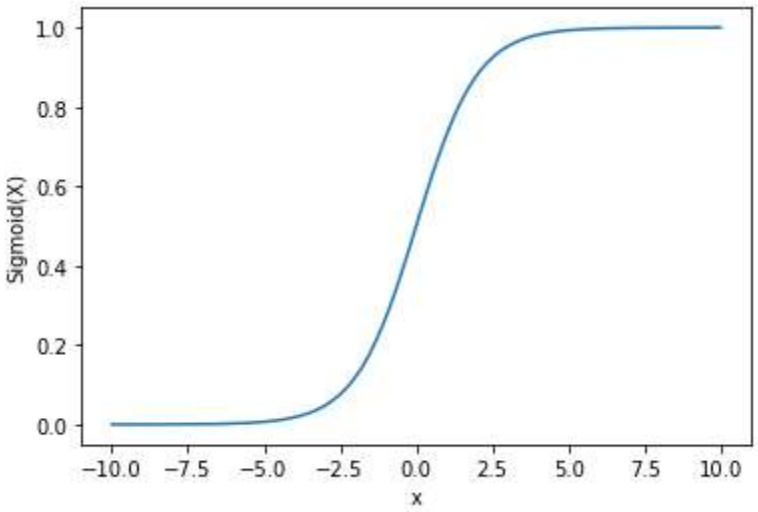
Let's see how the algorithm differs from linear regression (Module 2). The linear regression statistical model is used to predict continuous outcome variables, whereas logistic regression predicts categorical outcome variables. Linear regression model regression line is highly susceptible to outliers. So, it will not be appropriate for logistic regression.

Below is the function for logistic regression:

$$\text{Sig}(x) = \frac{1}{1 + e^{-x}}$$

(fig 4.3.1.b)

- E is log base
- X is the numerical value that needs to be transformed.



(fig 4.3.1.c)

If we feed an output value to the sigmoid function, it will return the probability of the outcome between 0 and 1. If the value is below 0.5, then the output is returned as No/Fail/Deceased (above example). If the value is above 0.5, then the output is returned as Yes/Pass/Deceased.

Assumptions of Logistic regression:

- Independent variables show a linear relationship with the log of output variables.
- Non-Collinearity between independent variables. That is, independent variables are independent of each other.
- The output variable is binary.

4.3.2 Random Forest

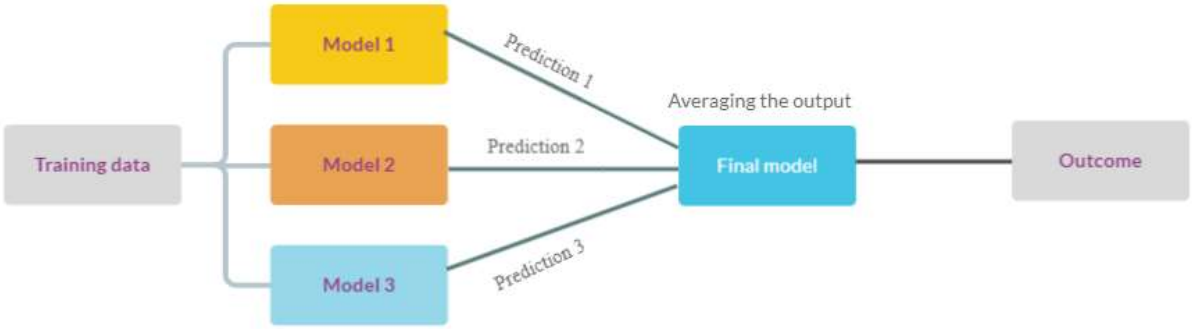
The Random Forest statistical technique uses an ensemble method to predict the outcome. There are two types of random forest methods:

Random forest regression: For predicting continuous outcome variables. (Salary, Income, Enrollments, etc.)
Random forest classifier: For predicting a class label (Yes or No, Active Vs. Lapsed, etc.)
In this module, let us first understand some terms we will use later in upcoming modules.

Ensemble learning: The term ensemble means combining multiple model outcomes to predict the outcome in machine learning. Combining multiple models is a more efficient method than predicting the outcome using a single model. Some popular ensemble methods include Stacking, bagging, blending etc.

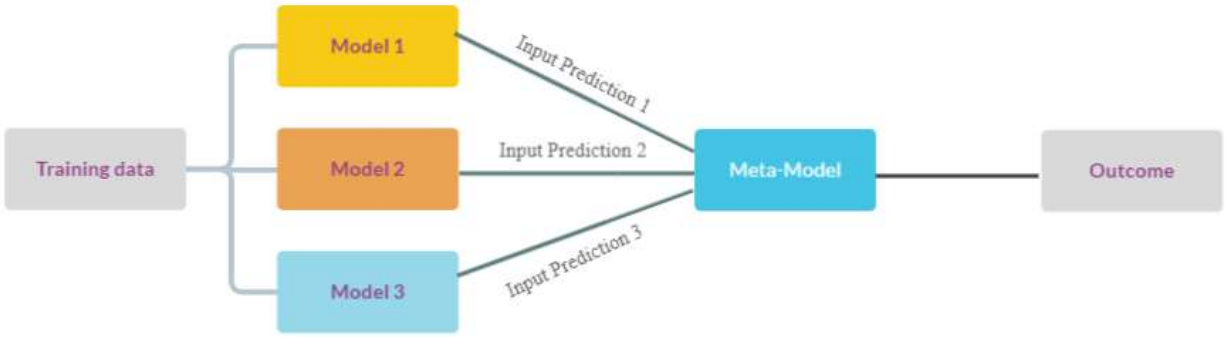
Stacking: Stacking is a machine learning process that uses training data, runs multiple models, and generalizes predictions to get the final output. The generalization of the predictions can be of two types.

Averaging: In this method, the output from the predictive models is averaged out to get the final prediction.



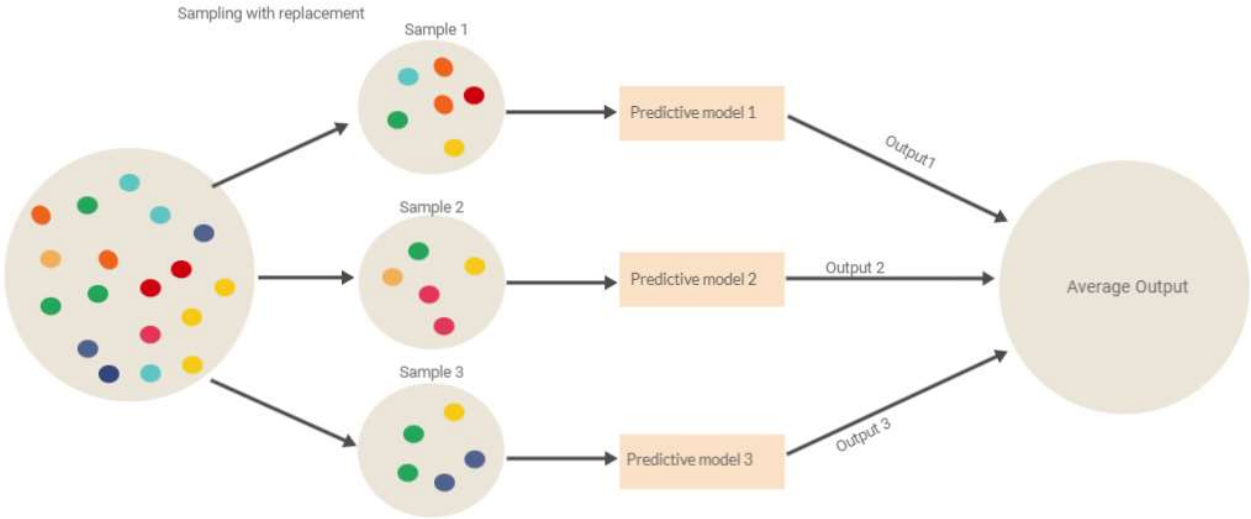
(fig 4.3.2.a)

Meta-model: In this method, the output from the predictive models is used as input features for training a Meta-model.



(fig 4.3.2.b)

Bagging: Bagging, also known as bootstrapping, is an ensemble method in which the train data is run on multiple versions of predictive models and combines the final prediction using averaging process.



(fig 4.3.2.c)

Blending: Blending is an ensemble machine learning technique that learns how to best combine the predictions that we derive from multiple models. Blending and stacking are usually interchangeable. The only difference is that in stacking, we use two or more base models and a meta-model that either uses the average of all the predictions derived from the model or uses the predictions as input for the meta-model (see stacking above). But in blending, the meta-model is usually a linear regression model (for continuous outcome) or logistic regression (for categorical outcome). Blending uses the weighted sum of the predictions and hence the term blending.

Regression tree: A random forest regression/classification tree is an ensemble machine learning

algorithm. It starts with running the regression/classification tree algorithm on several subsets of the training dataset. This addresses the overfitting problem. The algorithm uses the bootstrapping method to pick sample datasets (by replacement) from the original train dataset and runs the data thru the decision trees to get the predictions. The final predictions are either the majority predictions or the average of all the predictions made by decision trees.

4.3.3 XGBoost

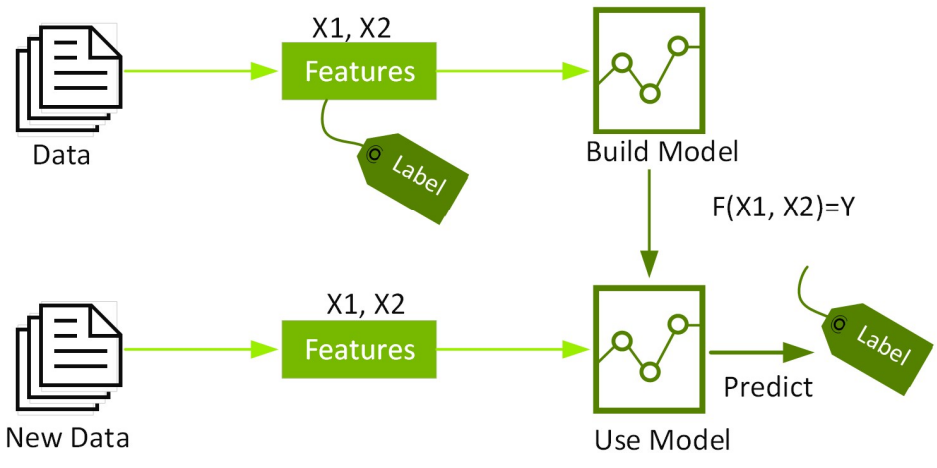
XGBoost is an open-source software library that implements optimized distributed gradient boosting machine learning algorithms under the Gradient Boosting framework.

What is XGBoost?

XGBoost, which stands for Extreme Gradient Boosting, is a scalable, distributed gradient-boosted decision tree (GBDT) machine learning library. It provides parallel tree boosting and is the leading machine learning library for regression, classification, and ranking problems.

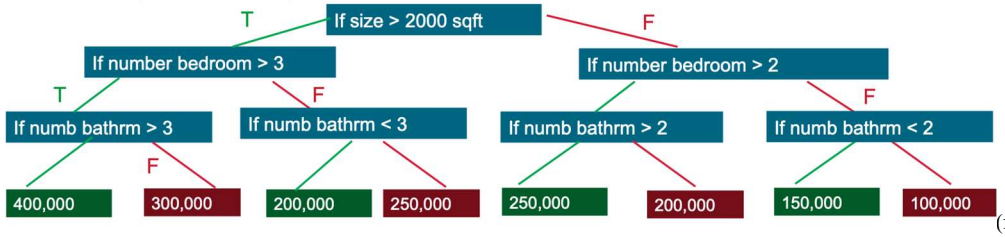
It's vital to an understanding of XGBoost to first grasp the machine learning concepts and algorithms that XGBoost builds upon: supervised machine learning, decision trees, ensemble learning, and gradient boosting.

Supervised machine learning uses algorithms to train a model to find patterns in a dataset with labels and features and then uses the trained model to predict the labels on a new dataset's features.



(fig 4.3.3.a)

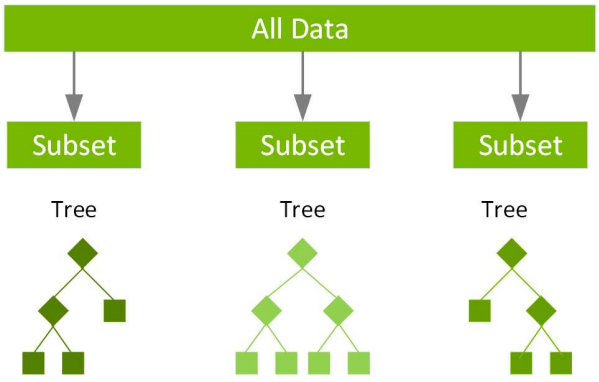
Decision trees create a model that predicts the label by evaluating a tree of if-then-else true/false feature questions and estimating the minimum number of questions needed to assess the probability of making a correct decision. Decision trees can be used for classification to predict a category, or regression to predict a continuous numeric value. In the simple example below, a decision tree is used to estimate a house price (the label) based on the size and number of bedrooms (the features).



(fig 4.3.3.b)

A Gradient Boosting Decision Trees (GBDT) is a decision tree ensemble learning algorithm similar to random forest, for classification and regression. Ensemble learning algorithms combine multiple machine learning algorithms to obtain a better model.

Both random forest and GBDT build a model consisting of multiple decision trees. The difference is in how the trees are built and combined.



(fig 4.3.3.c)

Random forest uses a technique called bagging to build full decision trees in parallel from random bootstrap samples of the data set. The final prediction is an average of all of the decision tree predictions.

The term “gradient boosting” comes from the idea of “boosting” or improving a single weak model by combining it with a number of other weak models in order to generate a collectively strong model. Gradient boosting is an extension of boosting where the process of additively generating weak models is formalized as a gradient descent algorithm over an objective function. Gradient boosting sets targeted outcomes for the next model in an effort to minimize errors. Targeted outcomes for each case are based on the gradient of the error (hence the name gradient boosting) with respect to the prediction.

GBDTs iteratively train an ensemble of shallow decision trees, with each iteration using the error residuals of the previous model to fit the next model. The final prediction is a weighted sum of all of the tree predictions. Random forest “bagging” minimizes the variance and overfitting, while GBDT “boosting” minimizes the bias and underfitting.

XGBoost is a scalable and highly accurate implementation of gradient boosting that pushes the limits of computing power for boosted tree algorithms, being built largely for energizing machine learning model performance and computational speed. With XGBoost, trees are built in parallel, instead of sequentially like GBDT. It follows a level-wise strategy, scanning across gradient values and using these partial sums to evaluate the quality of splits at every possible split in the training set.

Chapter 5

IMPLEMENTATION

Importing basic python modules to extract and visualize dataset.

```
import os
import pandas as pd
import numpy as np
import warnings
```

Reading the dataset using pandas.

```
bank = pd.read_csv('bank.csv',sep = ',')
```

Gathering information about Dataset by inspecing it.

```
bank.head()
```

	age	job	marital	education	default	balance	housing	loan	contact	day
0	59	admin.	married	secondary	no	2343	yes	no	unknown	5
1	56	admin.	married	secondary	no	45	no	no	unknown	5
2	41	technician	married	secondary	no	1270	yes	no	unknown	5
3	55	services	married	secondary	no	2476	yes	no	unknown	5
4	54	admin.	married	tertiary	no	184	no	no	unknown	5

```
bank.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 11162 entries, 0 to 11161
Data columns (total 17 columns):
 #   Column     Non-Null Count  Dtype 
---  ------     --------------  ----- 
 0   age        11162 non-null  int64 
 1   job        11162 non-null  object
 2   marital    11162 non-null  object
 3   education  11162 non-null  object
 4   default    11162 non-null  object
 5   balance    11162 non-null  int64 
 6   housing    11162 non-null  object
 7   loan       11162 non-null  object
 8   contact    11162 non-null  object
 9   day        11162 non-null  int64 
 10  month      11162 non-null  object
 11  duration   11162 non-null  int64 
 12  campaign   11162 non-null  int64 
 13  pdays      11162 non-null  int64 
 14  previous   11162 non-null  int64 
 15  poutcome   11162 non-null  object
 16  deposit    11162 non-null  object
```

We observe that in the dataset there are **17 attributes** in total out of which **deposit** is the dependent Y variable on all rest 16 X variables. The values of deposit is either Yes or No. so later it would needed to be converted to 0 for No and 1 for Yes.

Shape of Dataset is:

```
bank.shape
```

```
(11162, 17)
```

overall description of dataset:

```
bank.describe()
```

	age	balance	day	duration	campaign	pday
count	11162.000000	11162.000000	11162.000000	11162.000000	11162.000000	11162.0000(
mean	41.231948	1528.538524	15.658036	371.993818	2.508421	51.3304(
std	11.913369	3225.413326	8.420740	347.128386	2.722077	108.7582(
min	18.000000	-6847.000000	1.000000	2.000000	1.000000	-1.0000(
25%	32.000000	122.000000	8.000000	138.000000	1.000000	-1.0000(
50%	39.000000	550.000000	15.000000	255.000000	2.000000	-1.0000(
75%	49.000000	1708.000000	22.000000	496.000000	3.000000	20.7500(
max	95.000000	81204.000000	31.000000	3881.000000	63.000000	854.0000(

Let's figure out the number of unique values assigned to all attributs in dataset of 'object' type.

```
for col in bank.select_dtypes(include='object').columns:
  print(col)
  print(bank[col].unique())
  print('\n')
```

```
job
['admin.' 'technician' 'services' 'management' 'retired' 'blue-collar'
 'unemployed' 'entrepreneur' 'housemaid' 'unknown' 'self-employed'
 'student']


marital
['married' 'single' 'divorced']

education
['secondary' 'tertiary' 'primary' 'unknown']


default
['no' 'yes']


housing
['yes' 'no']


loan
['no' 'yes']


contact
['unknown' 'cellular' 'telephone']


month
['may' 'jun' 'jul' 'aug' 'oct' 'nov' 'dec' 'jan' 'feb' 'mar' 'apr' 'sep']


poutcome
['unknown' 'other' 'failure' 'success']


deposit
['yes' 'no']
```

Exploratory Data Analysis

- Find Unwanted Columns
- Find Missing Values
- Find Features with one value
- Explore the Categorical Features
- Find Categorical Feature Distribution
- Relationship between Categorical Features and Label
- Explore the Numerical Features
- Find Discrete Numerical Features
- Relation between Discrete numerical Features and Labels
- Find Continous Numerical Features
- Distribution of Continous Numerical Features
- Relation between Continous numerical Features and Labels
- Find Outliers in numerical features
- Explore the Correlation between numerical features
- Find Pair Plot
- Check the Data set is balanced or not based on target values in classification

```
df=bank
```

1. Find Unwanted Columns

Take-away:

- these is no unwanted column present in given dataset to remove

2. Find Missing Values

```
features_na = [features for features in bank.columns if bank[features].isnull().sum() > 0]
for feature in features_na:
    print(feature, np.round(bank[feature].isnull().mean(), 4),  ' % missing values')
else:
    print("No missing value found")
```

```
No missing value found
```

Take-away:

- No missing value found

3. Find Features with One Value

```
for column in df.columns:
    print(column,df[column].nunique())
```

```
 age 76
 job 12
 marital 3
 education 4
 default 2
 balance 3805
 housing 2
 loan 2
 contact 3
 day 31
 month 12
 duration 1428
 campaign 36
 pdays 472
 previous 34
 poutcome 4
 deposit 2
```

Take-away:

- No feature with only one value

4. Explore the Categorical Features

```
categorical_features=[features for features in df.columns if ((df[features].dtype=='O') &
categorical_features
```

```
 ['job',
  'marital',
  'education',
  'default',
  'housing',
  'loan',
  'contact',
  'month',
  'poutcome']
```

```
for feature in categorical_features:
    print('The feature is {} and number of categories are {}'.format(feature,df[feature].n
```

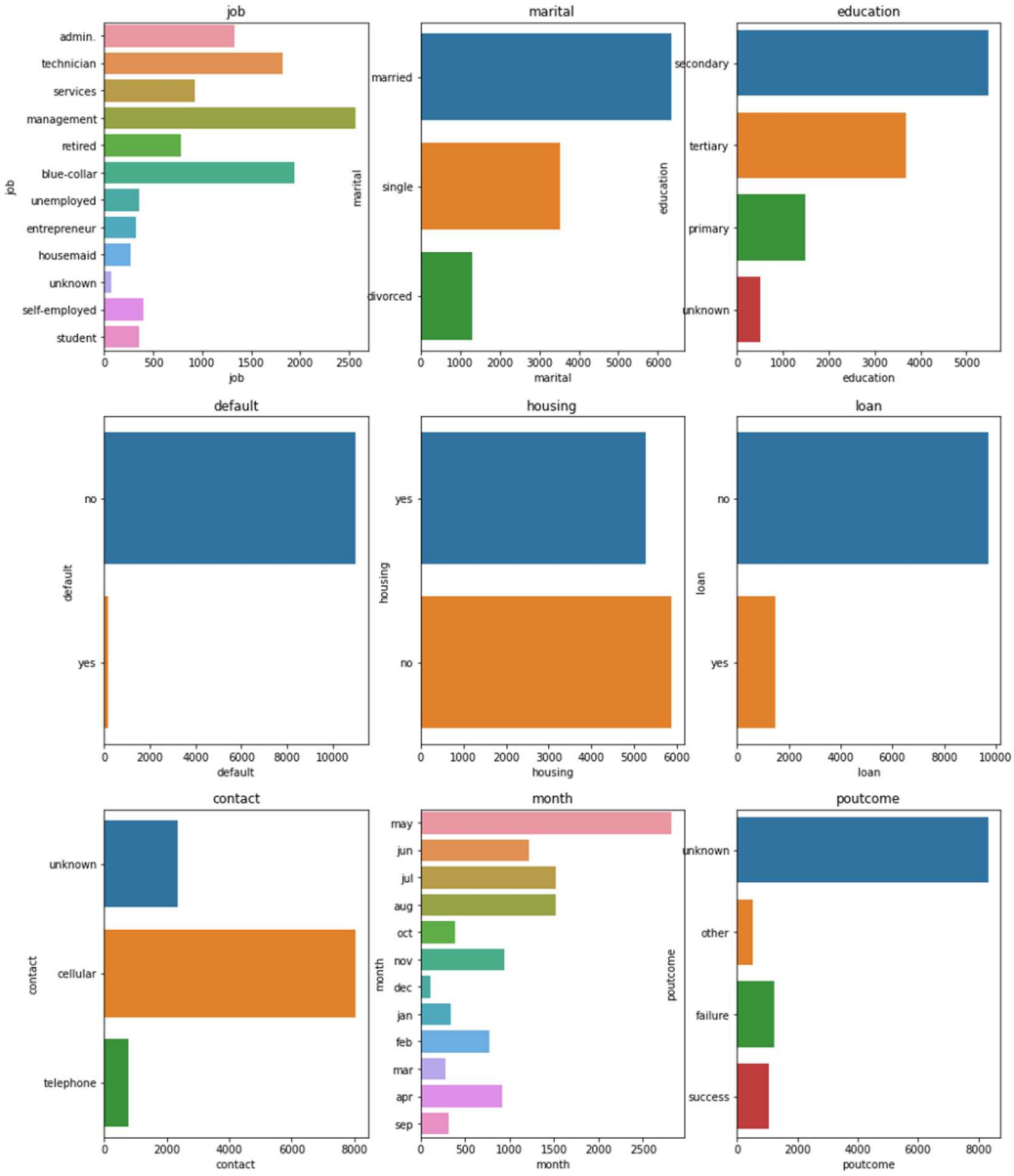
```
 The feature is job and number of categories are 12
 The feature is marital and number of categories are 3
 The feature is education and number of categories are 4
 The feature is default and number of categories are 2
 The feature is housing and number of categories are 2
 The feature is loan and number of categories are 2
 The feature is contact and number of categories are 3
 The feature is month and number of categories are 12
 The feature is poutcome and number of categories are 4
```

Take-away:

- there are 9 categorical features
- feature job and month has highest number of categorical values

5. Find Categorical Feature Distribution

```
import matplotlib.pyplot as plt
import seaborn as sns
plt.figure(figsize=(15,80),facecolor='white')
plotnumber=1
for categorical_feature in categorical_features:
    ax = plt.subplot(12,3,plotnumber)
    sns.countplot(y=categorical_feature,data=df)
    plt.xlabel(categorical_feature)
    plt.title(categorical_feature)
    plotnumber+=1
plt.show()
```



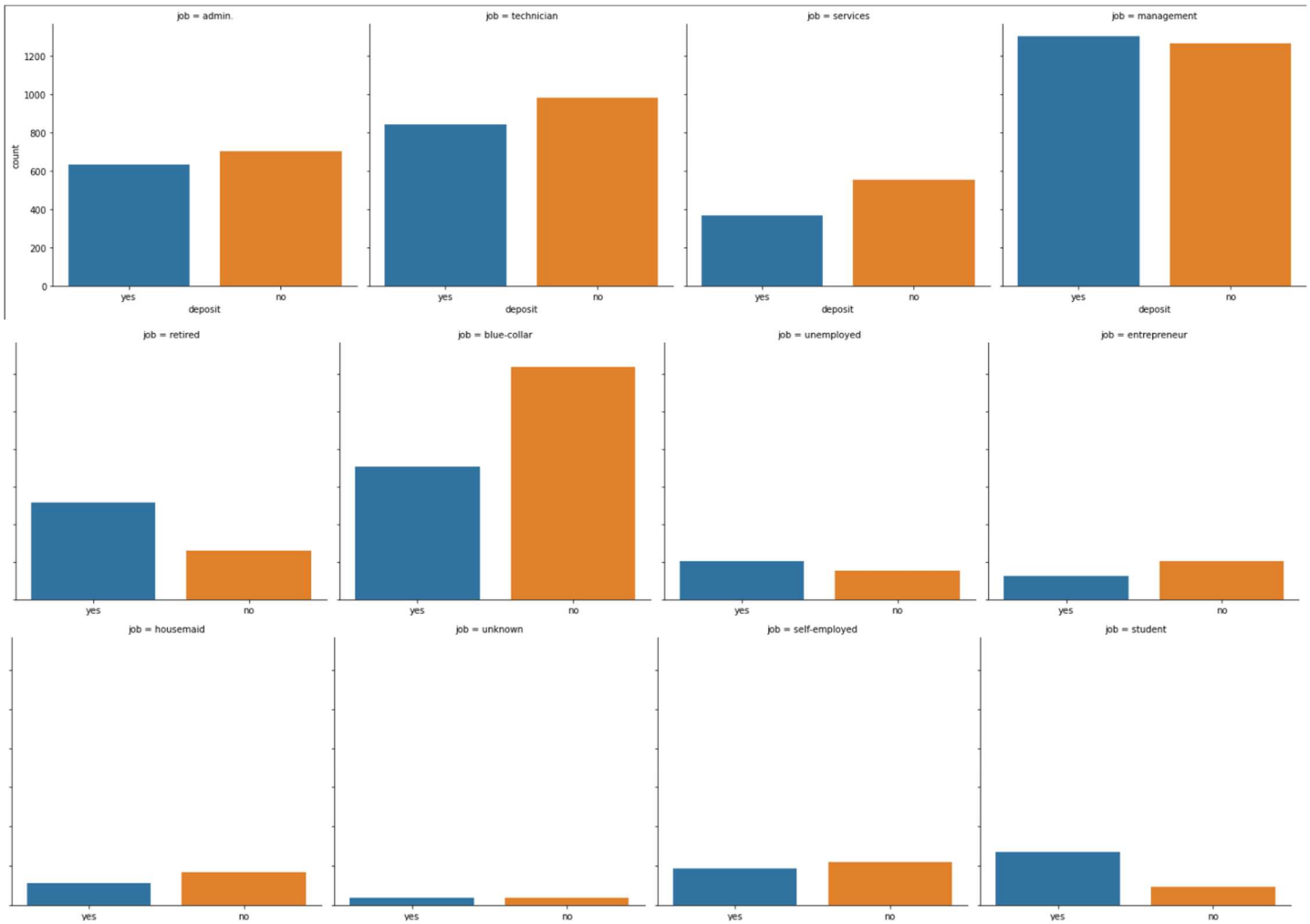
(fig 5.a)

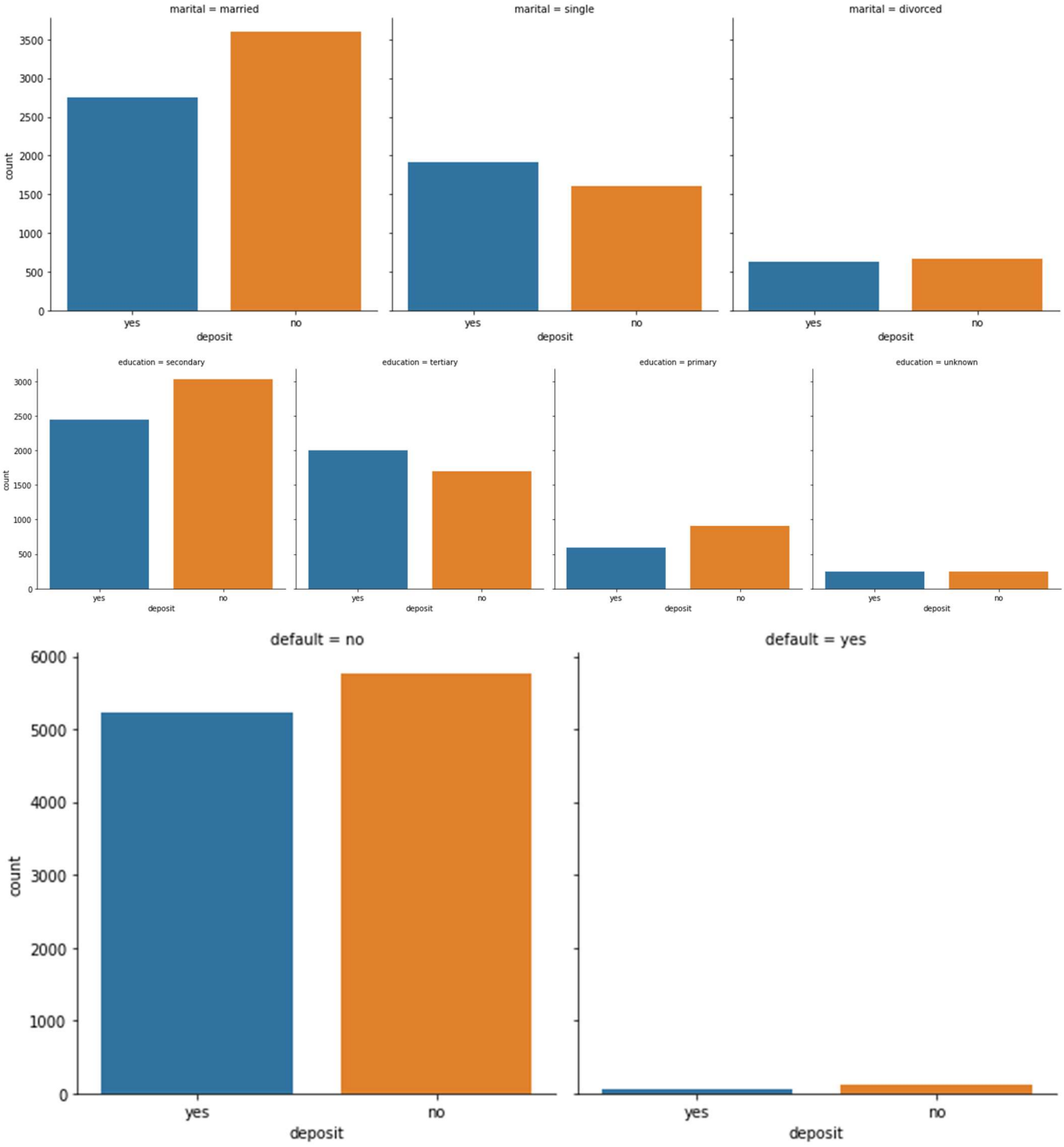
Take-away:

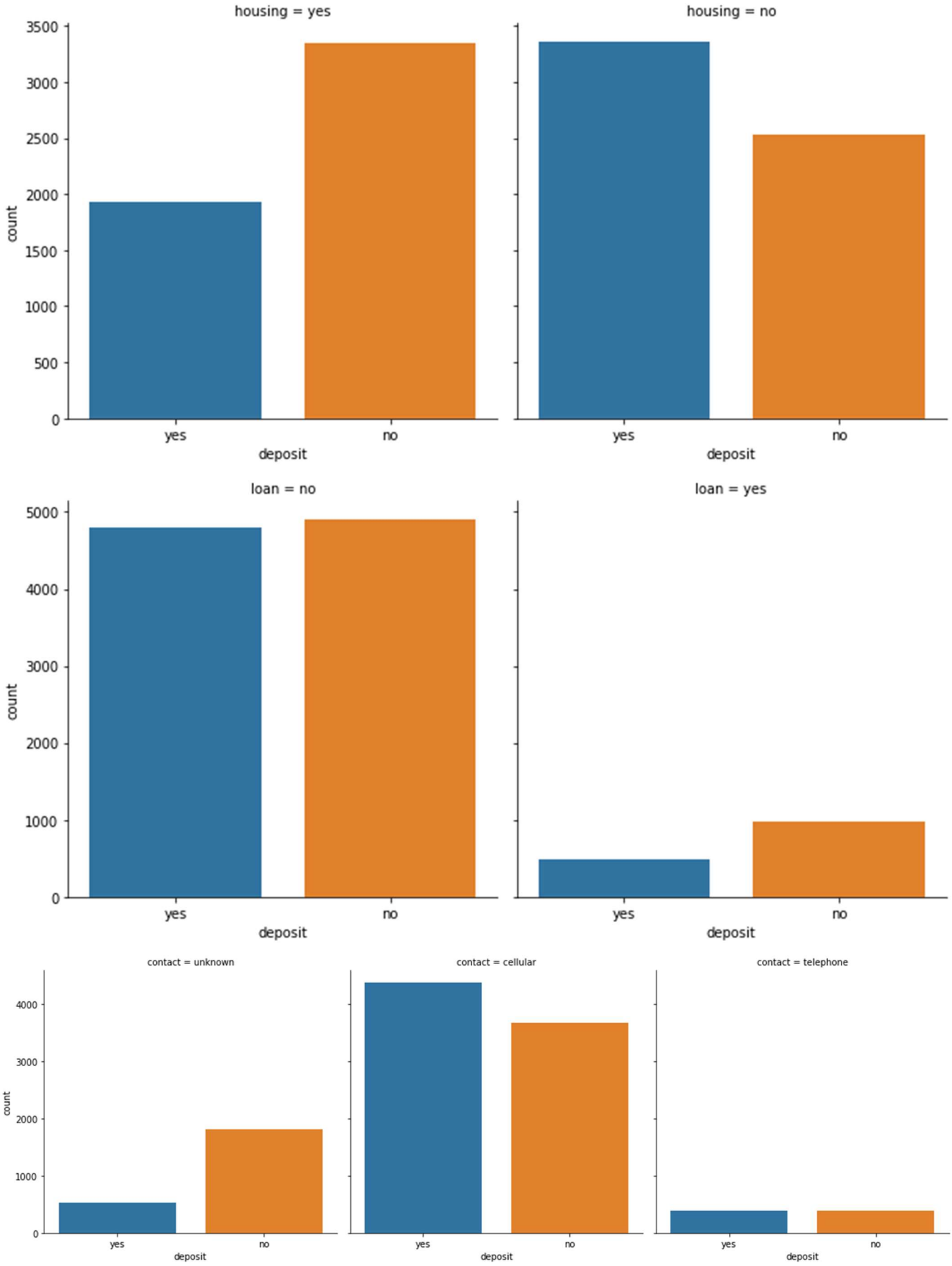
- client with job type as management records are high in given dataset and housemaid are very less
- client who married are high in records in given dataset and divorced are less
- client whoes education background is secondary are in high numbers in given dataset
- defualt feature seems to be does not play importand role as it has value of no at high ratio to value yes which can drop
- data in month of may is high and less in dec

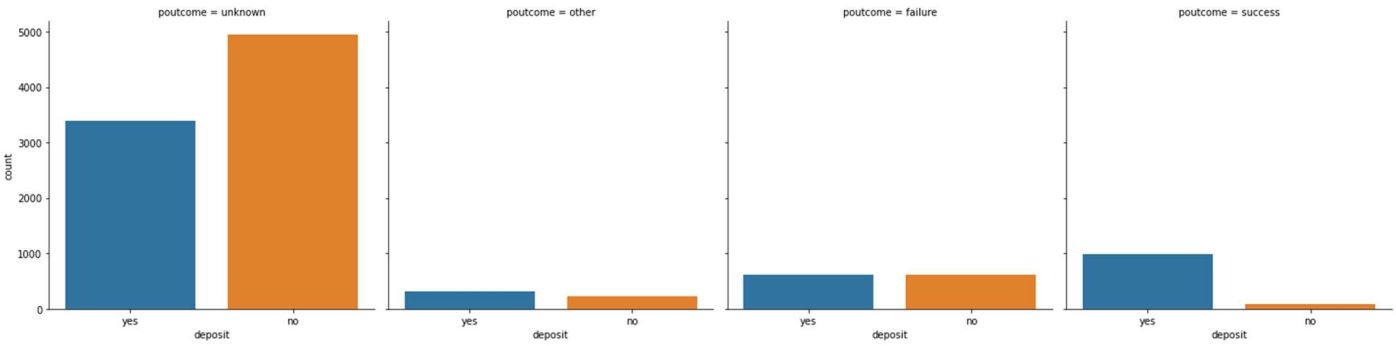
6. Relationship between Categorical Features and Label

```
for categorical_feature in categorical_features:
    sns.catplot(x='deposit', col=categorical_feature, kind='count', data= df)
plt.show()
```




marital = married
marital = single
marital = divorced
education = secondary
education = tertiary
education = primary
education = unknown
default = no
default = yes
count
deposit
yes
no


housing = yes
housing = no
loan = no
loan = yes
contact = unknown
contact = cellular
contact = telephone
count
deposit
yes
no
0
500
1000
1500
2000
2500
3000
3500
4000
5000



(fig 5.b)

Take-away:

- retired client has high interest on deposit
- client who has housing loan seems to be not interested much on deposit
- if pre campagin outcome that is poutcome=success then, there is high chance of client to show interest on deposit
- in month of March, September, October and December, client show high interest to deposit
- in month of may, records are high but client interst ratio is very less

7. Explore the Numerical Features

```
numerical_features = [feature for feature in df.columns if ((df[feature].dtypes != 'O') &
print('Number of numerical variables: ', len(numerical_features))

df[numerical_features].head()
```

```
Number of numerical variables:  7
```

	age	balance	day	duration	campaign	pdays	previous
0	59	2343	5	1042	1	-1	0
1	56	45	5	1467	1	-1	0
2	41	1270	5	1389	1	-1	0
3	55	2476	5	579	1	-1	0
4	54	184	5	673	2	-1	0

Take-away:

- there are 7 numerical features

8. Find Discrete Numerical Features

```
discrete_feature=[feature for feature in numerical_features if len(df[feature].unique())<2
print("Discrete Variables Count: {}".format(len(discrete_feature)))
```

```
Discrete Variables Count: 0
```

Take-away:

- there is no Discrete Variables in give dataset

9. Relation between Discrete numerical Features and Labels

- NA

10. Find Continous Numerical Features

```
continuous_features=[feature for feature in numerical_features if feature not in discrete_
print("Continuous feature Count {}".format(len(continuous_features)))
```

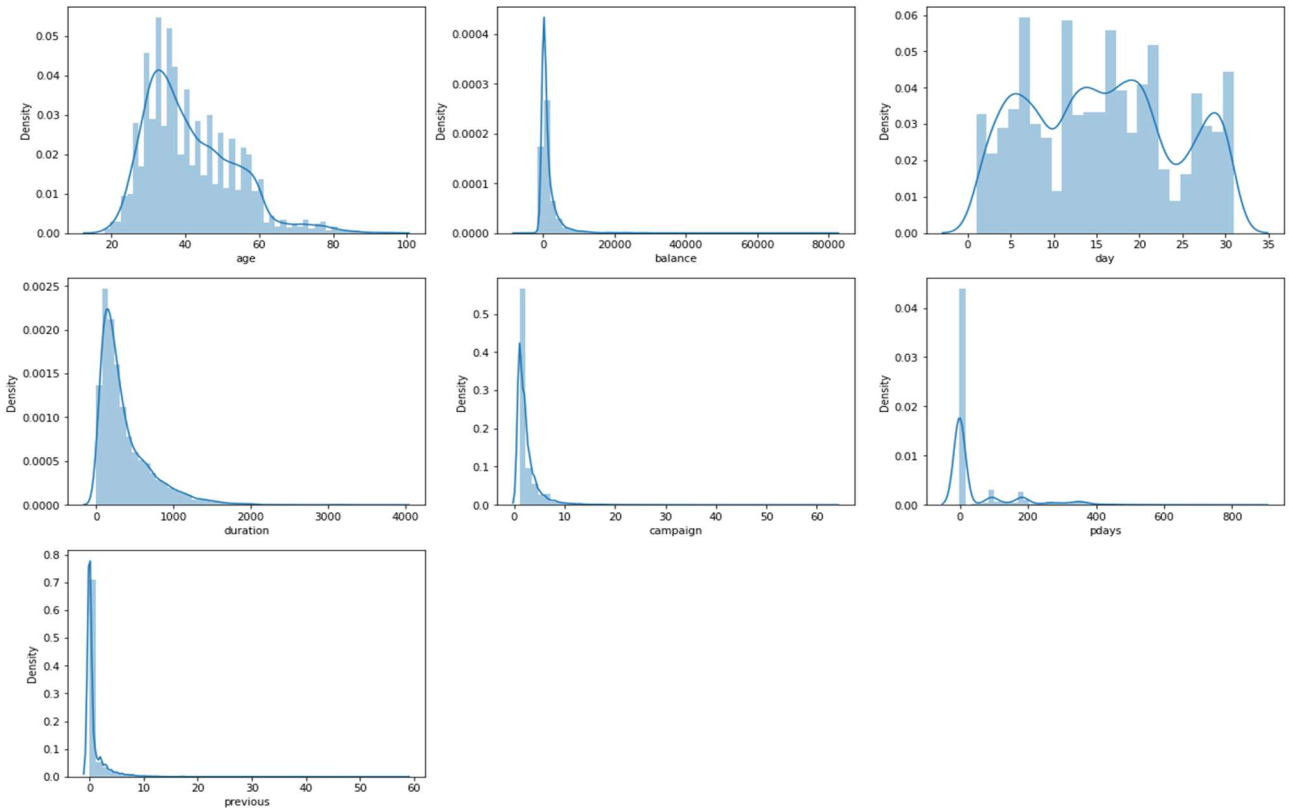
```
Continuous feature Count 7
```

Take-away:

- there are 7 continuous numerical features

11. Distribution of Continous Numerical Features

```
plt.figure(figsize=(20,60), facecolor='white')
plotnumber =1
for continuous_feature in continuous_features:
    ax = plt.subplot(12,3,plotnumber)
    sns.distplot(df[continuous_feature])
    plt.xlabel(continuous_feature)
    plotnumber+=1
plt.show()
```



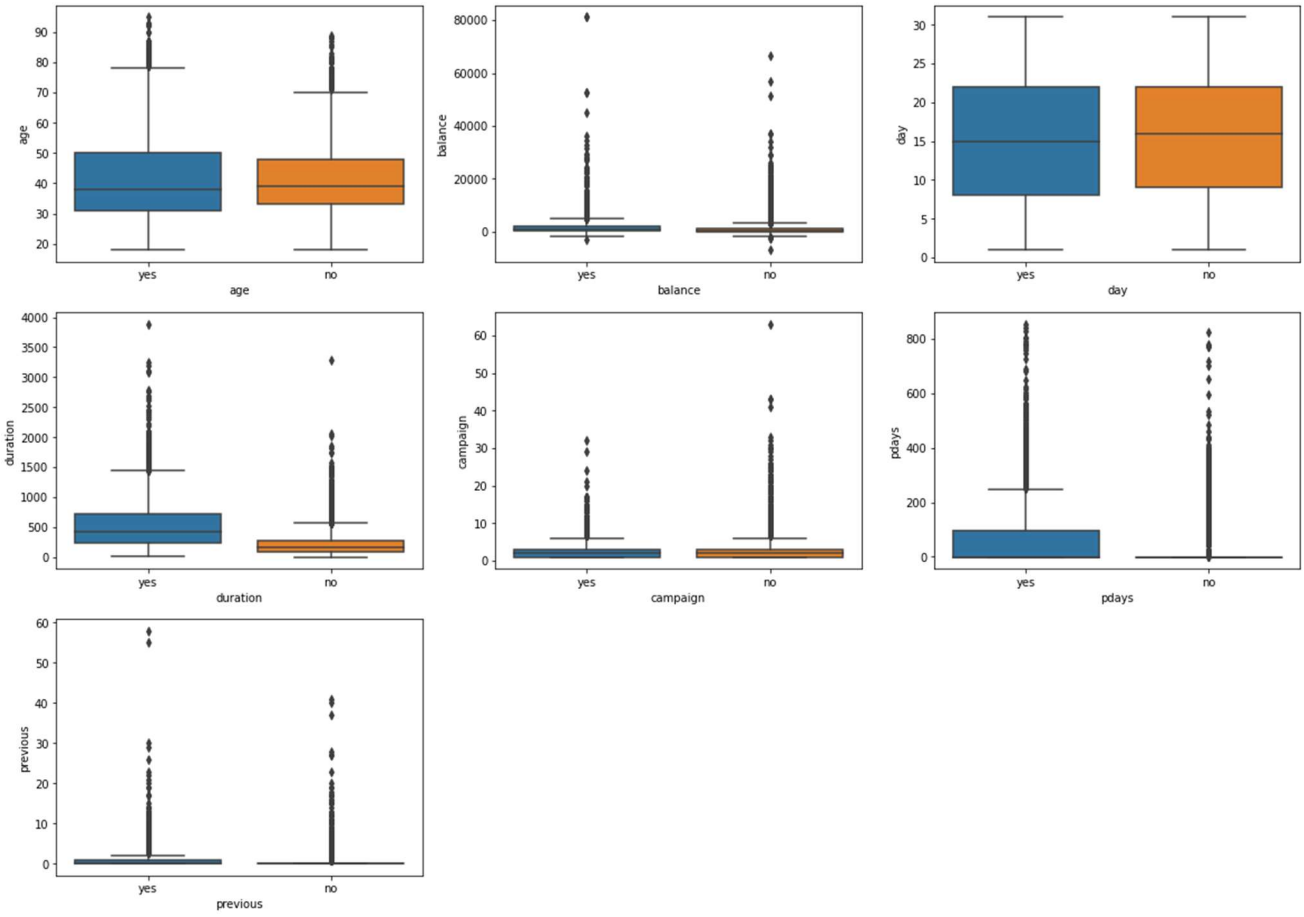
(fig 5.c)

Take-away:

- it seems age, days distributed normally
- balance, duration, compaign, pdays and previous heavely skewed towards left and seems to be have some outliers.

12. Relation between Continous numerical Features and Labels

```
plt.figure(figsize=(20,60), facecolor='white')
plotnumber =1
for feature in continuous_features:
    ax = plt.subplot(12,3,plotnumber)
    sns.boxplot(x="deposit", y= df[feature], data=df)
    plt.xlabel(feature)
    plotnumber+=1
plt.show()
```



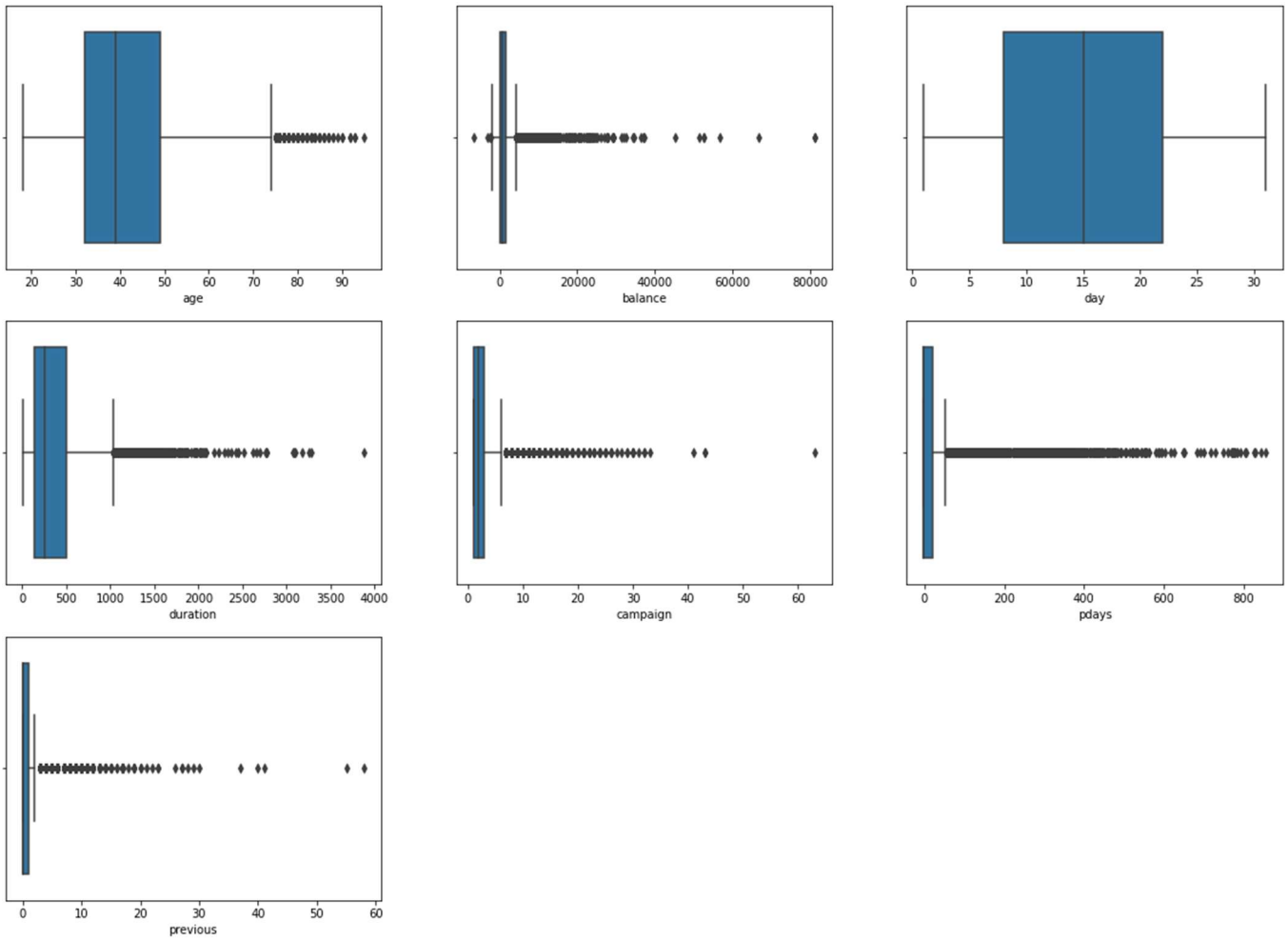
(fig 5.d)

Take-away:

- client shows interest on deposit who had discussion for longer duration

13. Find Outliers in numerical features

```
#boxplot on numerical features to find outliers
plt.figure(figsize=(20,60), facecolor='white')
plotnumber =1
for numerical_feature in numerical_features:
    ax = plt.subplot(12,3,plotnumber)
    sns.boxplot(df[numerical_feature])
    plt.xlabel(numerical_feature)
    plotnumber+=1
plt.show()
```



(fig 5.e)

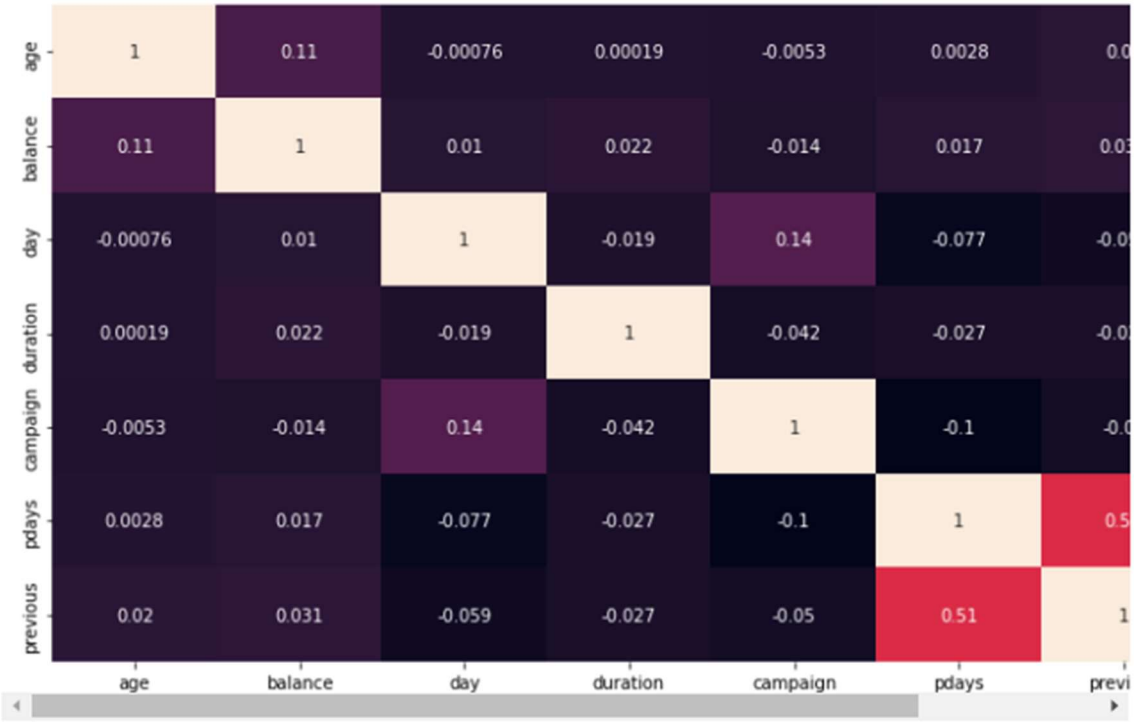
Take-away:

- age, balance, duration, compaign, pdays and previous has some outliers

14. Explore the Correlation between numerical features

```
cor_mat=df.corr()
fig = plt.figure(figsize=(15,7))
sns.heatmap(cor_mat,annot=True)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fcc21241490>
```



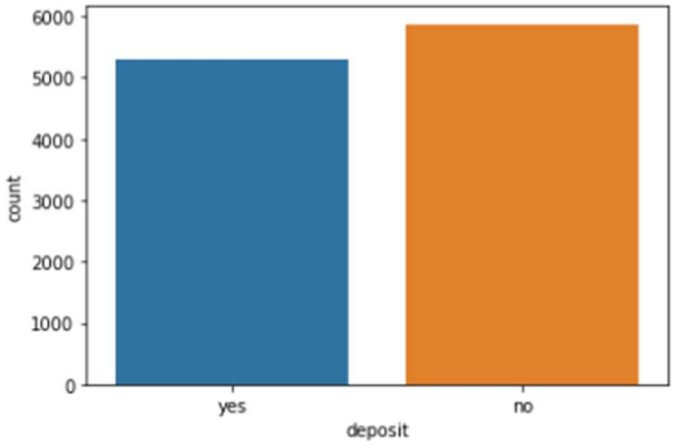
(fig 5.f)

Take-away:

- it seems no feature is heavily correlated with other features

15. Check the Data set is balanced or not based on target values in classification

```
sns.countplot(x='deposit',data=df)
plt.show()
```



(fig 5.g)

```
df['deposit'].groupby(df['deposit']).count()
```

```
deposit
no     5873
yes    5289
Name: deposit, dtype: int64
```

Take-away:

- given dataset seems to be balanced.

Feature Engineering

Removing Outliers in age, balance, duration, compaign, pdays as well as dropping unwanted attributes to improve accuracy of training.

```
df2=df.copy()
```

defaut features does not play important role

```
df2.groupby(['deposit','default']).size()
```

```
deposit  default
no       no         5757
         yes         116
yes      no         5237
         yes          52
dtype: int64
```

```
df2.drop(['default'],axis=1, inplace=True)
```

```
df2.groupby(['deposit','pdays']).size()
```

```
deposit  pdays
no       -1       4940
          1          2
          2          6
          5          2
          6          2
                   ...
yes       804        1
          805        1
          828        1
          842        1
          854        1
Length: 732, dtype: int64
```

drop pdays as it has -1 value for around 40%+

```
df2.drop(['pdays'],axis=1, inplace=True)
```

remove outliers in feature age.

```
df2.groupby('age',sort=True)['age'].count()
```

```
age
18     8
19    13
20    20
21    30
22    48
      ..
89     1
90     2
92     2
93     2
95     1
Name: age, Length: 76, dtype: int64
```

these can be ignored and values lies in between 18 to 95

remove outliers in feature balance.

```
df2.groupby(['deposit','balance'],sort=True)['balance'].count()
```

```
deposit  balance
no       -6847      1
         -2712      1
         -2282      1
         -2049      1
         -1965      1
                    ..
yes       34646     1
          36252     1
          45248     1
          52587     2
          81204     2
Name: balance, Length: 5082, dtype: int64
```

these outlier should not be remove as balance goes high, client show interest on deposit

remove outliers in feature duration.

```
df2.groupby(['deposit','duration'],sort=True)['duration'].count()
```

```
deposit  duration
no       2           1
         3           1
         4           2
         5           4
         6           6
                    ..
yes      3094        1
         3102        1
         3183        1
         3253        1
         3881        1
Name: duration, Length: 2157, dtype: int64
```

these outlier should not be remove as duration goes high, client show interest on deposit

remove outliers in feature campaign.

```
df2.groupby(['deposit','campaign'],sort=True)['campaign'].count()
```

```
df3 = df2[df2['campaign'] < 33]
```

```
df3.groupby(['deposit','campaign'],sort=True)['campaign'].count()
```

remove outliers in feature previous.

```
df3.groupby(['deposit','previous'],sort=True)['previous'].count()
```

```
df4 = df3[df3['previous'] < 31]
```

creating dummy categorical features for all features with multiple categorical values

```
cat_columns = ['job', 'marital', 'education', 'contact', 'month', 'poutcome']
for col in  cat_columns:
    df4 = pd.concat([df4.drop(col, axis=1),pd.get_dummies(df4[col], prefix=col, prefix_sep

bool_columns = ['housing', 'loan', 'deposit']
for col in  bool_columns:
    df4[col+'_new']=df4[col].apply(lambda x : 1 if x == 'yes' else 0)
    df4.drop(col, axis=1, inplace=True)


df4.head()
```

	age	balance	day	duration	campaign	previous	job_blue-collar	job_entrepreneur	job_
0	59	2343	5	1042	1	0	0	0	
1	56	45	5	1467	1	0	0	0	
2	41	1270	5	1389	1	0	0	0	
3	55	2476	5	579	1	0	0	0	
4	54	184	5	673	2	0	0	0	

5 rows × 41 columns

Split Dataset into Training set and Test set

```
X = df4.drop(['deposit_new'],axis=1)
y = df4['deposit_new']
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X,y,test_size=0.2, random_state=0)
```

```
len(X_train)
```

```
8921
```

```
len(X_test)
```

```
2231
```

Machine Learning Models and evaluation.

Logistic Regression

```
from sklearn.linear_model import LogisticRegression

logistic_model = LogisticRegression(max_iter=10000)
result = logistic_model.fit(X_train, y_train)
result.score(X_train, y_train)
```

```
0.8253559018047304
```

```
predict = result.predict(X_test)
```

```
from sklearn.metrics import accuracy_score
print(accuracy_score(y_test, predict))
```

```
0.8265351860152398
```

```
from sklearn.metrics import roc_auc_score
roc_auc_score(y_test, predict)
```

```
0.8250241069153792
```

Random Forest Classifier

```
from sklearn.ensemble import RandomForestClassifier
rfc = RandomForestClassifier()
rfc.fit(X_train,y_train)
rfc.score(X_test,y_test)
```

```
0.8561183325862842
```

```
predict_rfc=rfc.predict(X_test)
```

```
from sklearn.metrics import accuracy_score
print(accuracy_score(y_test, predict_rfc))
```

```
0.8561183325862842
```

```
from sklearn.metrics import roc_auc_score
roc_auc_score(y_test, predict_rfc)
```

```
0.8570072917372136
```

XGBoost Classifier

```
from xgboost import XGBClassifier
model_xgb = XGBClassifier(objective='binary:logistic',learning_rate=0.1,max_depth=10,n_est
model_xgb.fit(X_train,y_train)
model_xgb.score(X_test,y_test)
```

```
0.8583594800537876
```

```
predict_xgb=model_xgb.predict(X_test)
```

```
from sklearn.metrics import accuracy_score
print(accuracy_score(y_test, predict_xgb))
```

```
0.8583594800537876
```

```
from sklearn.metrics import roc_auc_score
roc_auc_score(y_test, predict_xgb)
```

```
0.8595885054357465
```

Observatons and conclusions

let's see the Confusion Matrix of all the 3 models.

Logistic Regression

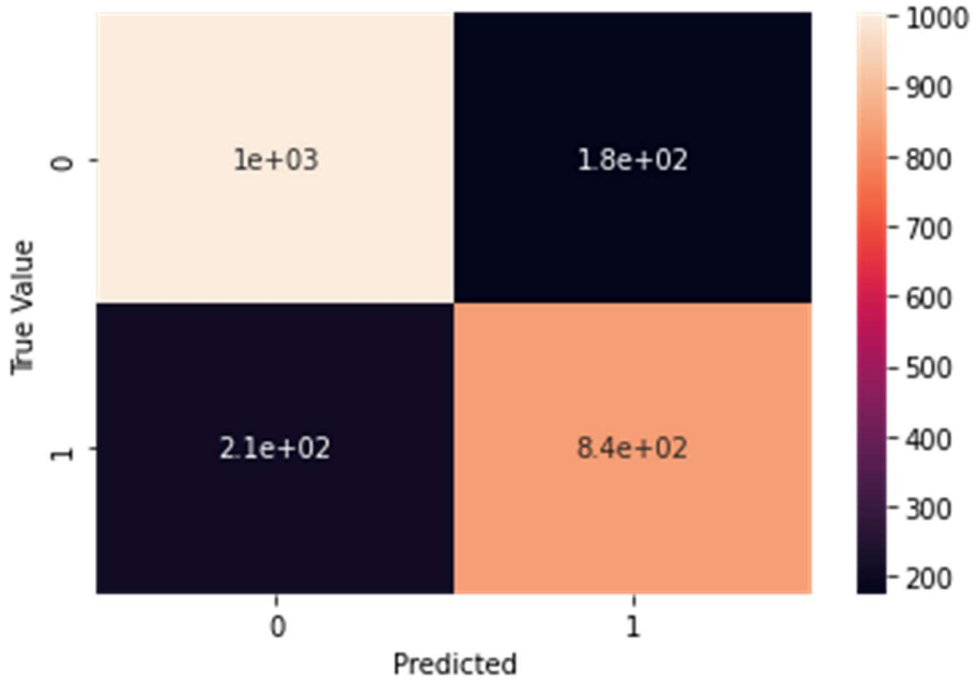
```
from sklearn.metrics import roc_auc_score
roc_auc_score(y_test, predict)
```

```
0.8250241069153792
```

```
from sklearn.metrics import confusion_matrix
lr = confusion_matrix(y_test,predict)
lr
```

```
array([[1004,  175],
       [ 212,  840]])
```

```
from matplotlib import pyplot as plt
import seaborn as sn
sn.heatmap(lr, annot=True)
plt.xlabel('Predicted')
plt.ylabel('True Value')
plt.show()
```



(fig 5.h)

Random Forest Classifier

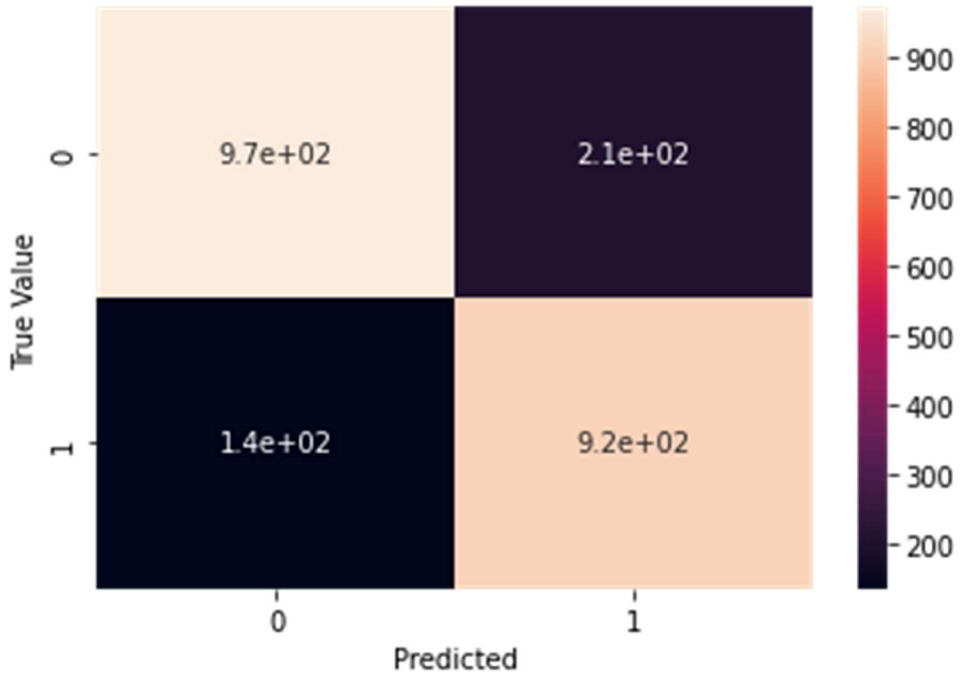
```
from sklearn.metrics import roc_auc_score
roc_auc_score(y_test, predict_rfc)
```

```
0.8570072917372136
```

```
from sklearn.metrics import confusion_matrix
cx = confusion_matrix(y_test,rfc.predict( X=X_test))
cx
```

```
array([[992, 187],
       [134, 918]])
```

```
from matplotlib import pyplot as plt
import seaborn as sn
sn.heatmap(cx, annot=True)
plt.xlabel('Predicted')
plt.ylabel('True Value')
plt.show()
```



(fig 5.i)

XGBoost Classifier

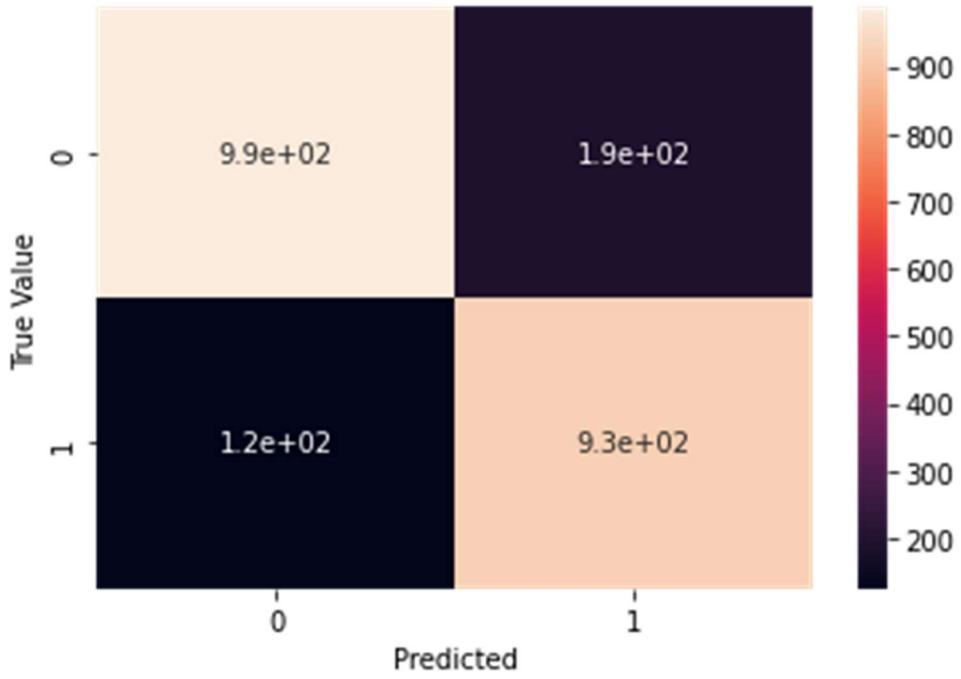
```
from sklearn.metrics import roc_auc_score
roc_auc_score(y_test, predict_xgb)
```

```
0.8595885054357465
```

```
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test,model_xgb.predict(X_test))
cm
```

```
array([[988, 191],
       [125, 927]])
```

```
from matplotlib import pyplot as plt
import seaborn as sn
sn.heatmap(cm, annot=True)
plt.xlabel('Predicted')
plt.ylabel('True Value')
plt.show()
```



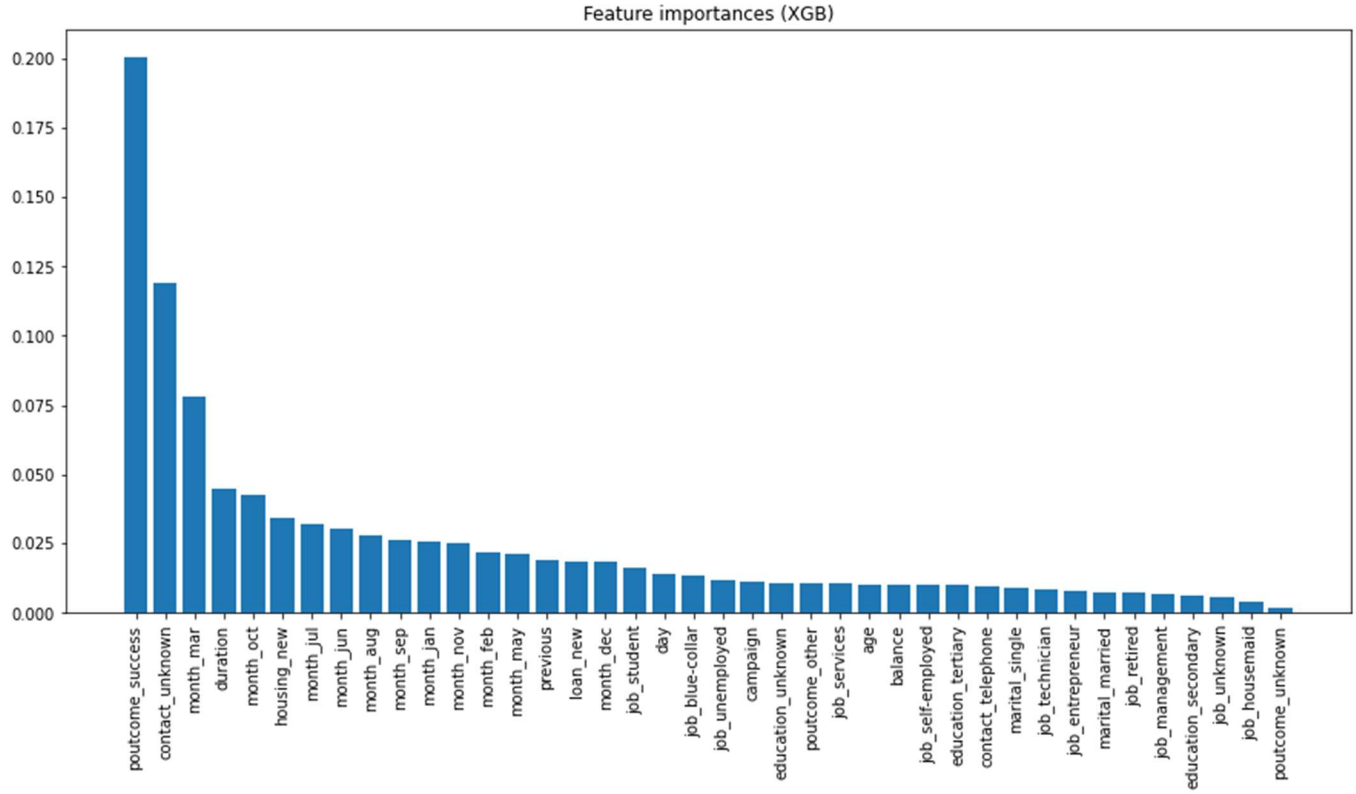
(fig 5.j)

We observe that we get the best possible score from XGBoost Classifier model.

so we will try to figure out the important features which this model used for prediction

```
headers = ["name", "score"]
values = sorted(zip(X_train.columns, model_xgb.feature_importances_), key=lambda x: x[1] *
xgb_feature_importances = pd.DataFrame(values, columns = headers)

#plot feature importances
fig = plt.figure(figsize=(15,7))
x_pos = np.arange(0, len(xgb_feature_importances))
plt.bar(x_pos, xgb_feature_importances['score'])
```


Feature importances (XGB)
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
poutcome_success
contact_unknown
month_mar
duration
month_oct
housing_new
month_jul
month_jun
month_aug
month_sep
month_jan
month_nov
month_feb
month_may
previous
loan_new
month_dec
job_student
day
job_blue-collar
job_unemployed
campaign
education_unknown
poutcome_other
job_services
age
balance
job_self-employed
education_tertiary
contact_telephone
marital_single
job_technician
job_entrepreneur
marital_married
job_retired
job_management
education_secondary
job_unknown
job_housemaid
poutcome_unknown

(fig 5.k)

Chapter 6

RESULT AND DISCUSSION

6.1 Evaluation

Following the implementation of the data mining methodology used for this analysis, as stated earlier, three algorithms are extensively used to help recognize patterns from the collected data. These methods used in this study utilize multiple forms to deliver performance or reports. The results obtained from the use of the algorithms will then be presented and analyzed at this point in the analysis but not clearly explained as will be discussed in the following paragraphs. Classification models are described in a variety of ways. This implies that it is possible to use multiple matrices to test findings resulting from the use of models of classification. For this research, the findings for this analysis are calculated based on the confusion matrix for calculating the errors that were misclassified for each algorithm or method from which each value was measured for accuracy. In this analysis, accuracy is used because its outcome is on average (equal value) with the appropriate quality. It is pertinent to realize that this model was originally divided into training and testing. Therefore, the analysis and outcomes of the three models will be listed below.

6.1.1 Logistic regression

The confusion matrix for this algorithm's test dataset is shown in table 1 below.
Prediction using the test dataset:

N=2231	Predicted:0	Predicted:1
Actual:0	1004	175
Actual:1	212	840

(tab 6.1.1.a)

According to the model, the AUC accuracy is 82.5 whereas the misclassification rate is 17.5%. From the total observation of 2231, the model predicted 1004 No and 840 as Yes. also, it incorrectly predicted 175 negatives as Yes and 212 positives as No.

6.1.2 Random Forest Classifier

The confusion matrix for this algorithm's test dataset is shown in table 1 below.
Prediction using the test dataset:

N=2231	Predicted:0	Predicted:1
Actual:0	992	187
Actual:1	134	918

(tab 6.1.1.b)

According to the model, the AUC accuracy is 85.7 whereas the misclassification rate is 14.3%. From the total observation of 2231, the model predicted 992 No and 918 as Yes. Also, it incorrectly predicted 187 negatives as Yes and 134 positives as No.

6.1.3 XGBoost Classifier

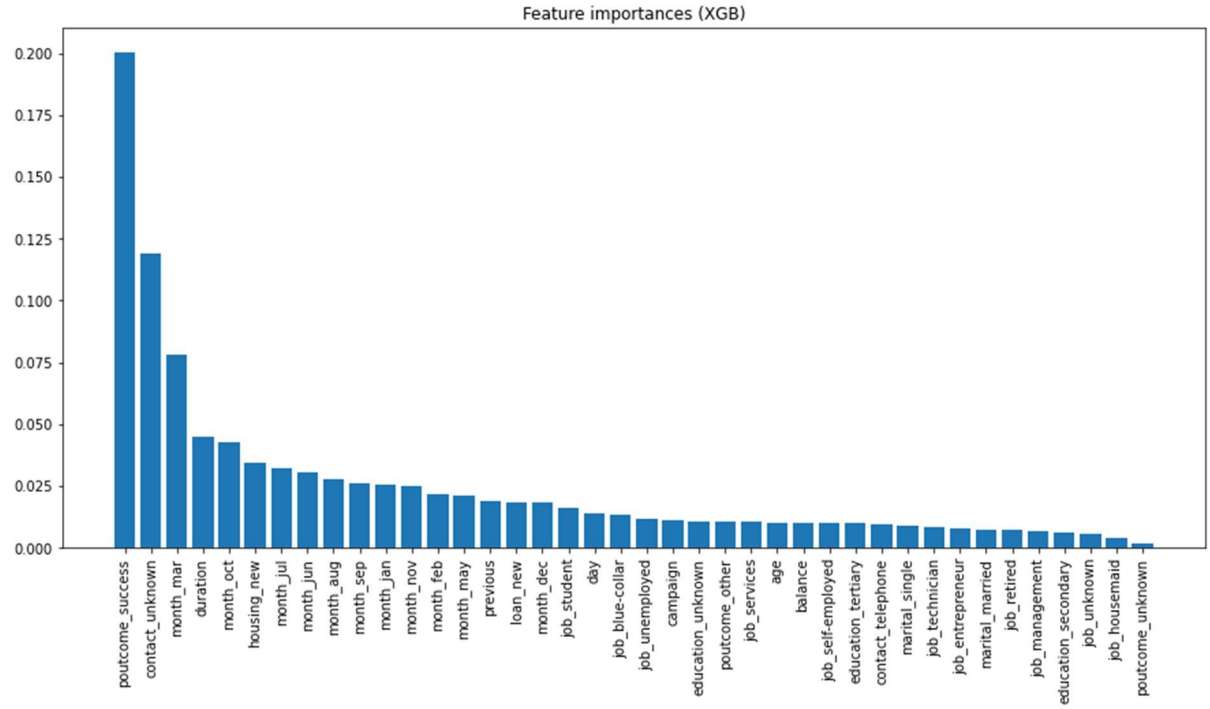
The confusion matrix for this algorithm's test dataset is shown in table 1 below. Prediction using the test dataset:

N=2231	Predicted:0	Predicted:1
Actual:0	988	191
Actual:1	125	927

(tab 6.1.1.c)

According to the model, the AUC accuracy is 85.95 whereas the misclassification rate is 14.05%. From the total observation of 2231, the model predicted 988 No and 927 as Yes. Also, it incorrectly predicted 191 negatives as Yes and 125 positives as No.

6.2 Analysis



(fig 6.2.a)

When we use XGBoost classifier as a training method we observe that the model is affected by various attributes like, previous success outcome, contact in month of march, which may be related to release of new interest rates of the financial year, duration of the call to a potential subscriber, if they have housing loan or not and how often was the person contacted etc.

Chapter 7

CONCLUSION AND FUTURE SCOPE

Many banks use direct marketing strategies to enable customers to access adequate information about the products. Researchers suggest that the applicability of data mining techniques depends on the availability of customers' information. Also, studies reveal that machine learning techniques determine customer response to bank products. The ability of the customers to subscribe to term deposits depends on the marketing campaign by the bank. In this research work, the resampling method was used in dealing with the problem of imbalanced data, and three machine learning algorithms (Logistic regression, Random Forest Classifier, and XGBoost Classifier) were deployed and the best one was used to find out the main factor that influences customers decision to subscribe to a term deposit in the bank.

When categorical features were checked the major factors that can be influencing the customer's decision were 'poutcome_success', 'contact', 'month_mar', 'duration' were few with the highest score with dependent variable. Which means that the longer the bank continue to advertise their product and service, the more customers could subscribe to a term deposit also if bank contacted the person who has previously subscribed to some product in the bank, then he might be inclined to do it again. Banks should focus on direct marketing techniques when applying statistical and mathematical approaches to determine customer response. This project can further be enhanced by using other techniques like univariate selection and feature importance on the dataset to identify more factors that can influence a customer's decision to subscribe to a term deposit in the bank.

REFERENCES

- https://www.kaggle.com/datasets/janiobachmann/bank-marketing-dataset
- https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html
- https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.RFE.html
- https://www.nvidia.com/en-us/glossary/data-science/xgboost/
- https://learn.theprogrammingfoundation.org/getting_started/intro_data_science/module5/
- https://learn.theprogrammingfoundation.org/getting_started/intro_data_science/module4
- https://www.sciencedirect.com/topics/engineering/confusion-matrix#:~:text=A%20confusion%20matrix%20is%20a,performance%20of%20a%20classification%20algorithm.
- https://towardsdatascience.com/exploratory-data-analysis-8fc1cb20fd15