Term Deposit Analysis and Prediction using Machine Learning

Submitted as a partial fulfillment of Bachelor of Technology in Computer Science & Engineering
of
Maulana Abul Kalam Azad University of Technology
(Formerly known as West Bengal University of Technology)

Project Report

Submitted by

Name of Students	University Roll No.
UTKARSH MISHRA	11600119031
TIYASHA PARUI	11600119028
ROSHNI DEY	11600119033
TANISHA GHOSH	11600119050
PRATYUSH MAJUMDER	11600119046

Under the supervision of

Ms. Rachita Ghoshhajra

Assistant Professor, Dept. of Computer Science and Engineering

Department of Computer Science & Engineering,
MCKV Institute of Engineering
243, G.T. Road(N)
Liluah, Howrah - 711204

Department of Computer Science & Engineering,
MCKV Institute of Engineering
243, G. T. Road (N), Liluah, Howrah-711204

CERTIFICATE OF RECOMMENDATION

I hereby recommend that the thesis prepared under my supervision by **Utkarsh Mishra, Roshni Dey, Tiyasha Parui, Tanisha Ghosh, Pratyush Majumder** entitled **Term Deposit Analysis and Prediction using Machine Learning** be accepted in partial fulfilment of the requirements for the degree of Bachelor of Technology in Computer Science & Engineering Department.

Mr. Avijit Bose
Professor & Head of the Department,
Computer Science & Engineering Department.
Department MCKV Institute of Engineering, Howrah

Ms. Rachita Ghoshhajra
Assistant Professor.
Computer Science & Engineering
MCKV Institute of Engineering, Howrah

Department of Computer Science & Engineering,
MCKV Institute of Engineering
243, G. T. Road (N), Liluah,
Howrah-711204

Affiliated to

Maulana Abul Kalam Azad University of Technology
(Formerly known as West Bengal University of Technology

CERTIFICATE

This is to certify that the project entitled **"Term Deposit Analysis and Prediction using Machine Learning"** and submitted by

Name of students	University Roll No.
UTKARSH MISHRA	11600119031
ROSHNI DEY	11600119033
TIYASHA PARUI	11600119028
TANISHA GHOSH	11600119050
PRATYUSH MAJUMDER	11600119046

has been carried out under the guidance of myself following the rules and regulations of the degree of Bachelor of Technology in Computer Science & Engineering of **Maulana Abul Kalam Azad University of Technology (**Formerly West Bengal University of Technology).

(Signature of the project guide)
Ms. Rachita Ghoshhajra
Assistant Professor
Department of Computer Science and Engineering

Utkarsh Mishra
Pratyush Majumder
Tanisha Ghosh
Tiyasha Parui
Roshni Dey

CERTIFICATE OF APPROVAL
(B. Tech Degree in Computer Science & Engineering)

This project report is hereby approved as a creditable study of an engineering subject carried out and presented in a manner satisfactory to warrant its acceptance as a prerequisite to the degree for which it has been submitted. It is to be understood that by this approval, the undersigned do not necessarily endorse or approve any statement made, opinion expressed and conclusion drawn therein but approve the project report only for the purpose for which it has been submitted

COMMITTEE ON FINAL
EXAMINATION FOR
EVALUATION OF
PROJECT REPORT

1. ______________________
2. ______________________
3. ______________________
4. ______________________
5. ______________________

ACKNOWLEDGEMENT

We would like to express our sincere gratitude to MCKV Institute of Engineering for supporting us throughout our B. Tech curriculum.

We would take the opportunity to express our deep sense of gratitude to **Mr. Avijit Bose,** Assistant Professor and Head of the Department, Computer Science and Engineering, MCKV Institute of Engineering for supporting us to make our project worth it.

We would like to express our sincere gratitude to **Ms. Rachita Ghoshhajra**, our project advisor, for providing us with the wonderful opportunity to complete this admirable project on the subject of **"Term Deposit Analysis and Prediction using Machine Learning."** This project has assisted us in conducting extensive research and has taught us a lot of new information.'

The project coordinator**, Mr. Sumit Majumdar,** is also to be thanked for providing and delineating the administrative procedures pertaining to project processes.

We also like to express our gratitude to the MCKV Institute of Engineering's Computer Science and Engineering Department for all of their help, suggestions, and insightful advice in completing this project report successfully. Their advice has been a huge assistance to us at every stage of this project's research and writing. Their vast knowledge, extensive experience, and professional expertise allowed us to effectively accomplish this job. It would not have been able to complete this job without their help and direction. It enabled us to get more information and expertise.

We thank all of the MCKV Institute of Engineering's other professors and technical assistants for contributing significantly to the project's development. Last but not least, we would want to express our gratitude to all of our friends for their support.

1. Utkarsh Mishra
2. Tiyasha Parui
3. Roshni Dey
4. Tanisha Ghosh
5. Pratyush Majumder

TABLE OF CONTENTS

ABSTRACT

Majority of the revenue from the banking sector is usually generated from long term deposits by customers. It’s very important for banks to understand customer characteristics to increase product sales. To aid this, marketing strategies are employed to target potential customers and let them interact with the banks directly, generating a big amount of data on customer characteristics and demographics. In recent years, it has been discovered using various data analysis, feature selection, and machine learning techniques can be employed to analyse customer characteristics as well as variables that can impact customer decision significantly. These methods can be used to identify consumers in different categories to predict whether a customer would subscribe to a long-term deposit, allowing the marketing strategy to be more successful. In this study, we used Python programming to analyse financial transaction data in order to obtain insight into how business processes might be improved, to identify intriguing trends, and to make better data-driven decisions. In the given data set, we used statistical analysis such as exploratory data analysis (EDA), data pre-processing, and correlations. Furthermore, the study's purpose is to apply at least three common classification algorithms, such as Logistic Regression, Random Forest, Support Vector Machine, and Random Forest, and then build predictive models based on clients signing up for long-term deposits. We obtained the highest accuracy via Decision Tree, which is 90.63% for the test dataset. The accuracy, sensitivity, and specificity scores of these algorithms were used to analyse the results.

1. INTRODUCTION

Bank marketing efforts are typically carried out in one of two ways. The first is through mass marketing campaigns intended at the general public, while the second is through targeted marketing initiatives focused at a specific set of people. Term deposits are typically short-term investments with maturities ranging from one month to many years. A customer will deposit or invest in one of these accounts, agreeing not to withdraw their funds for a fixed period in return for a higher rate of interest paid on the account. The interest earned on a term deposit account is slightly higher than that paid on standard savings or interest-bearing checking accounts. The increased rate is because access to the money is limited for the timeframe of the term deposit.

Term deposits are an extremely safe investment and are therefore very appealing to conservative, low-risk investors. The financial instruments are sold by banks, thrift institutions, and credit unions. Term deposits sold by banks are insured by the Federal Deposit Insurance Corporation (FDIC). The National Credit Union Administration (NCUA) provides coverage for those sold by credit unions. How a Bank Uses a Term Deposit If a customer places money in a term deposit, the bank can invest the money in other financial products that pay a higher rate of return (RoR) than what the bank is paying the customer for the use of their funds. The bank can also lend the money out to its other clients, thereby receiving a higher interest rate from the borrowers as compared to what the bank is paying in interest for the term deposit.

It’s imperative for marketing managers to use their scarce resources to make fewer calls to clients while selling more, i.e., raise the positive response rate. With a view to achieving that, bank marketing managers may use multivariate data classification methods to identify potential customers as they already possess data from previous campaigns to analyse

1.1 Literature Review

Healthcare, banking, retail, intelligence, and telecommunications are just a few of the industries that employ machine learning and data mining tools and strategies to grow their businesses [1]. In the banking industry, for example, they used to construct models for risk analysis, Customer Relationship Management (CRM), direct marketing, and credit card fraud prevention. Machine Learning has gained popularity in the healthcare industry as a means of preventing medical insurance fraud and violence, as well as predicting patient behaviour and well-being patterns. The same applies for the banking industry.

According to M. Sergio, L. Raul and C. Paulo,", when compared to focused marketing, mass marketing initiatives had a lower positive reaction rate to purchase a product or subscribe to a service [2]. As a result, despite the fact that mass marketing help sell the goods, a lot of money is wasted on them. Finding potential clients from a certain set of people, on the other hand, is challenging. Because of the emergence of data-driven decisions in recent years, marketing managers have begun to employ statistical strategies to identify potential buyers for a product [3]. This helps them identify which customers are more likely to invest in the bank, avoiding potential impasses in the process.

But, due to the recent economic turmoil in many countries, banks need to sell more long-term deposits to raise their financial reserves. As a result, marketing managers are under pressure to sell long-term stakes to the general public [4]. Our review aims to present a thorough picture of how machine learning techniques have been applied in risk assessment so far from the perspective of a central bank.

It has recently been established that various data analysis, feature selection, and machine learning techniques can be used to analyse client traits as well as variables that might greatly influence customer decisions. These strategies can be used to identify consumers in various categories in order to anticipate if a customer will subscribe to a long-term deposit, making the marketing plan more successful [5].

Our target is to develop a system that prioritises the preservation of banking resources, optimises, and analyses how model-based baking prediction might improve the outcome output. This method aims to reduce the burden on bank personnel and raise their productivity by providing them with access to technology advancements.

1.2 Major Applications of Machine Learning in Banking

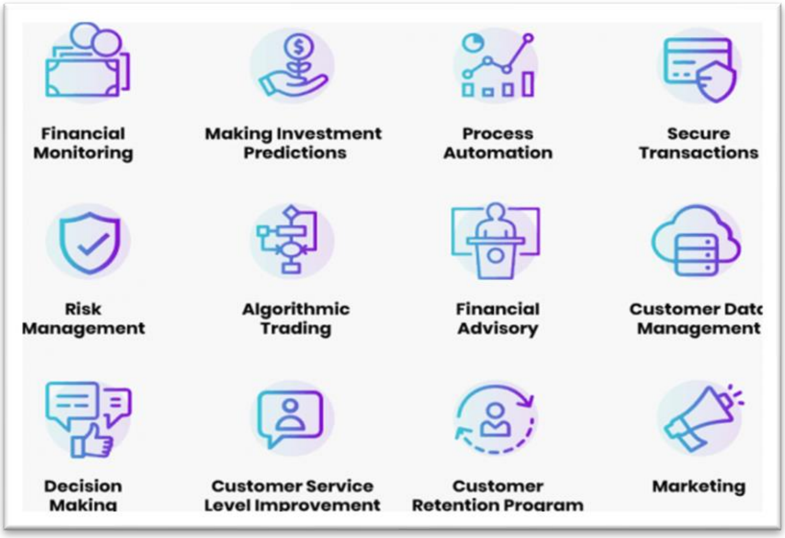


Fig 1.1.(a) Applications of ML in Banking

In recent years, machine learning and, to a lesser extent, deep learning technologies have been employed to assess credit risk and, more broadly, anticipate bank failures. Traditional statistical approaches are still widely utilized for this purpose today. Nonetheless, machine learning techniques are outperforming traditional ways by allowing practitioners to take previous decisions, use them in different contexts, and forecast future chaotic phenomena. This review aims to present a thorough picture of how machine learning techniques have been applied in risk assessment so far from the perspective of a central bank [4].

Using the vast databases that banks amass, machine learning can be applied in the banking industry to produce insights that can be put into practise. Machine learning models can assist banks in processing and analysing this data to have a better understanding of their customers and internal procedures, whether it be a history of transactions, chat logs with bank staff, or corporate documents, or customer's details. Since the advent of machine learning, reaching out to particular demographics has undergone a radical change thanks to the use of data analytics to offer banks with precise information on which clients are most likely to sign up for a financial product

1.3 Benefits of Banking using Machine Learning

- **Improved personalization**

  Machine learning can help banks to identify patterns in customer behaviour, enable a deeper understanding of customer needs and wants, and help create highly personalized service offerings.

- **Reduced costs**

  With the help of machine learning and NLP, banks can automate back-office operations, speed up document processing workflows, and minimize operational costs.

- **Accelerated decision-making**
  Banks can use ML-generated insights to make important decisions more frequently and with fewer risks.
- **Enhanced risk management**
  Machine learning's ability to model how a bank will react to certain economic conditions allows decision-makers to create more informed strategies.
- **Streamlined customer support**
  With the help of machine learning-enabled virtual assistants, banks can handle significantly more customer requests without compromising service quality.
- **Refined document processing**
  Machine learning allows banks to make sense of unstructured data, automatically index and label documents, and improve document management overall.

1.4 Shortfalls of Banking using Machine Learning

The machine learning method heavily relies on data. Lack of high-quality data is one of the major problems that machine learning expert's encounter. It can be exceedingly taxing to process noisy and erratic data. We don't want our system to produce predictions that are unreliable or flawed. Therefore, improving the result depends on the quality of the data.

The quantity of data financial institutions gathers, keep, systematise, analyse, and use to their profit is growing, which poses ethical problems. Although some users dislike this tendency, it is currently difficult to behave without leaving a trail of personal data.

1.5 Objective

- Our banking-based research aims to establish a decision-support framework for bank management.
- To suggest clients who are likely to submit a term deposit application.
- To create a system that focuses on the preservation of banking resources, optimizes and investigates how model-based baking prediction might help the outcome output.
- This strategy strives to lessen the onerous strain of bank employees and increase their productivity by giving them access to technological improvements.

2. WORKFLOW MODEL AND LIBRARIES USED FOR THE PROPOSED PRODUCT

2.1 Libraries used:

- **Matplotlib:**

Python scripts can be used to create 2D graphs and plots using the Matplotlib module. With features to control line styles, font attributes, formatting axes, and other features, it offers a module called pyplot that makes things simple for plotting. It offers a huge range of graphs and plots, including error charts, bar charts, power spectra, and histograms. It is combined with NumPy to create an environment that serves as a strong open-source replacement for MatLab.

- **NumPy:**

NumPy is a Python package. It stands for 'Numerical Python'. It is a library consisting of multidimensional array objects and a collection of routines for processing of array. NumPy is often used along with packages like SciPy (Scientific Python) and Mat−plotlib (plotting library). This combination is widely used as a replacement for MatLab, a popular platform for technical computing.

- **Pandas:**

Pandas is defined as an open-source library that provides high-performance data manipulation in Python. The name of Pandas is derived from the word Panel Data, which means an Econometrics from Multidimensional data. Pandas is built on top of the NumPy package, meaning a lot of the structure of NumPy is used or replicated in Pandas. Data in pandas is often used to feed statistical analysis in SciPy, plotting functions from Matplotlib, and machine learning algorithms in Scikit-learn.

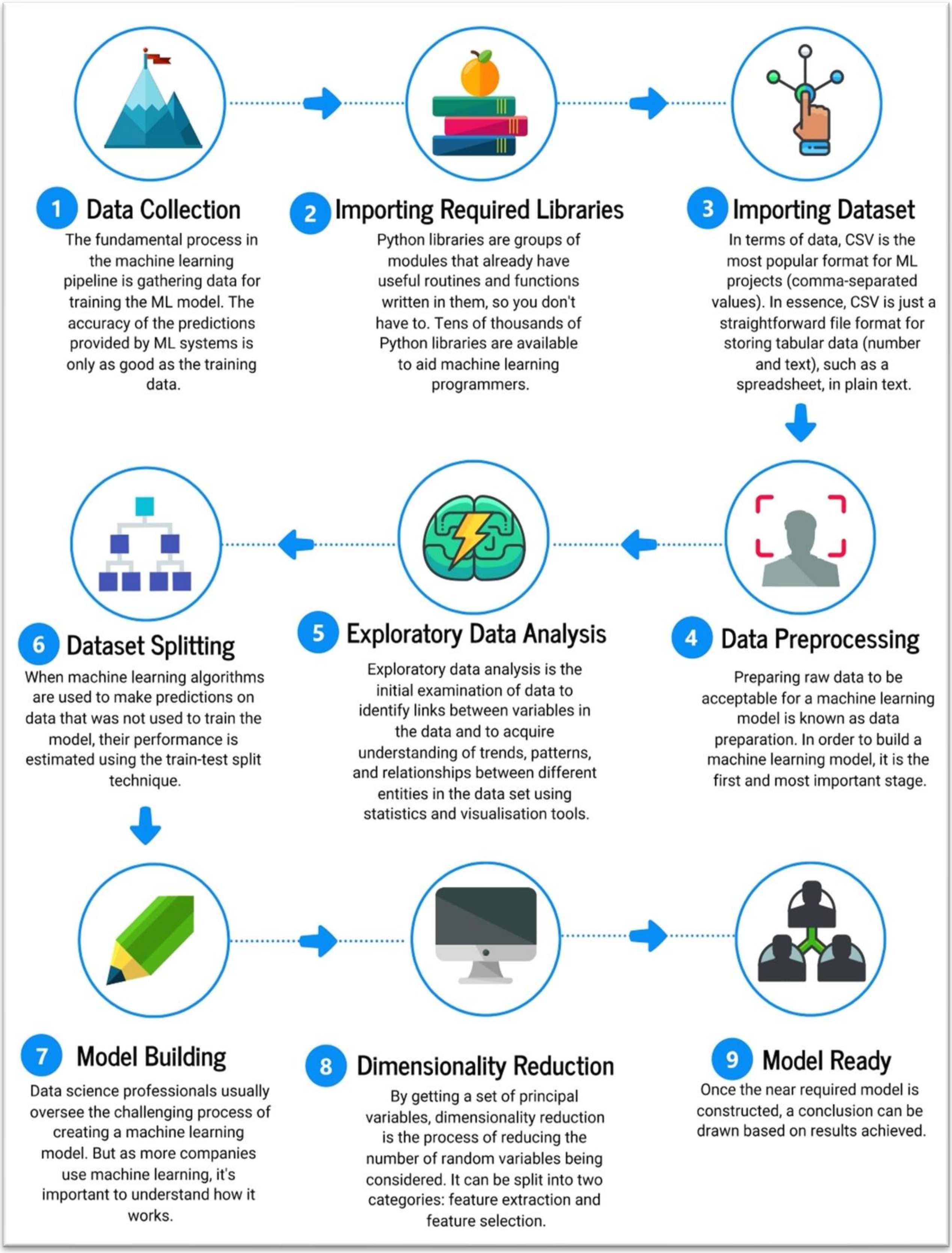
- **Scikit-learn:**

Scikit-learn (Sklearn) is the most useful and robust library for machine learning in Python. It provides a selection of efficient tools for machine learning and statistical modelling including classification, regression, clustering and dimensionality reduction via a consistence interface in Python. This library, which is largely written in Python, is built upon NumPy, SciPy and Matplotlib.

- **Seaborn:**

Seaborn is a library mostly used for statistical plotting in Python. It is built on top of Matplotlib and provides beautiful default styles and colour palettes to make statistical plots more attractive.

2.2 Workflow Model for Backend Model Building:



1 Data Collection

The fundamental process in the machine learning pipeline is gathering data for training the ML model. The accuracy of the predictions provided by ML systems is only as good as the training data.

2 Importing Required Libraries

Python libraries are groups of modules that already have useful routines and functions written in them, so you don't have to. Tens of thousands of Python libraries are available to aid machine learning programmers.

3 Importing Dataset

In terms of data, CSV is the most popular format for ML projects (comma-separated values). In essence, CSV is just a straightforward file format for storing tabular data (number and text), such as a spreadsheet, in plain text.

4 Data Preprocessing

Preparing raw data to be acceptable for a machine learning model is known as data preparation. In order to build a machine learning model, it is the first and most important stage.

5 Exploratory Data Analysis

Exploratory data analysis is the initial examination of data to identify links between variables in the data and to acquire understanding of trends, patterns, and relationships between different entities in the data set using statistics and visualisation tools.

6 Dataset Splitting

When machine learning algorithms are used to make predictions on data that was not used to train the model, their performance is estimated using the train-test split technique.

7 Model Building

Data science professionals usually oversee the challenging process of creating a machine learning model. But as more companies use machine learning, it's important to understand how it works.

8 Dimensionality Reduction

By getting a set of principal variables, dimensionality reduction is the process of reducing the number of random variables being considered. It can be split into two categories: feature extraction and feature selection.

9 Model Ready

Once the near required model is constructed, a conclusion can be drawn based on results achieved.

Fig 2.2.(a) Workflow Model of the project

3. METHODOLOGY

- In order to divide a client list into those who want to subscribe to a long-term deposit and those who do not, this study tries to discover the key features and construct a machine learning model that can do so.
- Sampling strategies and algorithm-based approaches might not be able to correct the class imbalance due to the difficulties of the high dimensional in the dataset that often goes along with an unbalanced dataset. A feature selection strategy is needed to choose a subset of characteristics that will contribute to the model's optimum efficiency in order to solve this challenge. The feature selection procedure is crucial when working with a dataset that has a high degree of dimension.
- Three ML classifier models are used to predict the output class.
- Additionally, the goal of this project is to develop a trustworthy and practical recommendation algorithm for forecasting customer uptake depending on client type. Regarding the clients' term deposit subscription, the target value is a binary class yes or no.
- Logistic regression, Random Forest, Decision Tree might all be used to categorise the data in order to complete the task. The optimal way for creating a classifier with higher predictive capabilities that could be used to evaluate whether a consumer will sign up for a term deposit will be determined by using various algorithms and comparing their accuracy.
- In order to develop precise guidelines that banks might use to identify clients who were likely to sign up for a term deposit, correlation analysis was applied. The found techniques may help identify which class subgroups are more likely to subscribe. Their age, marital status, gender, level of education, or other factors could be to blame for this.
- Determining whether the personal efforts made by banks during their campaign process have an impact on how well customers would subscribe to this service is necessary. For instance, the quantity of calls each customer has received, the outcomes of earlier campaign activities, etc. These traits were examined to determine if any trends could be found that could help with marketing efforts and services.

3.1. Dataset Overview:

Obtained from Kaggle, our target variable is **“subscribed**”, and there are 17 different independent variables included in the dataset that we used to build our model and to arrive at a conclusive outcome. In the dataset, out of 31647 records, 3715 number of customers actually subscribed to the term deposit which roughly estimates to around 11.73% of the total number of records.

3.1.1 Graphical representation of target variable:

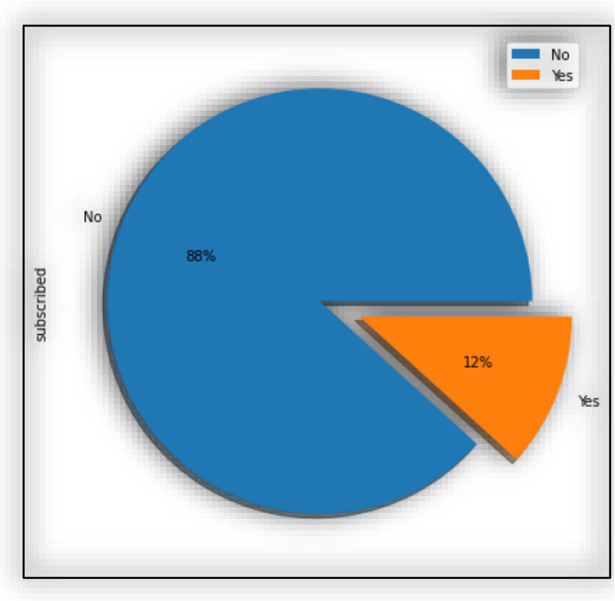


Fig 3.1.1.(a) Original dataset target variable overview

From figure 3.1.1.(a), it can be said that:

- The dataset serves as an illustration of a bivariate classification model.
- The various modelling strategies that we will be using are as follows:

  Logistic Regression

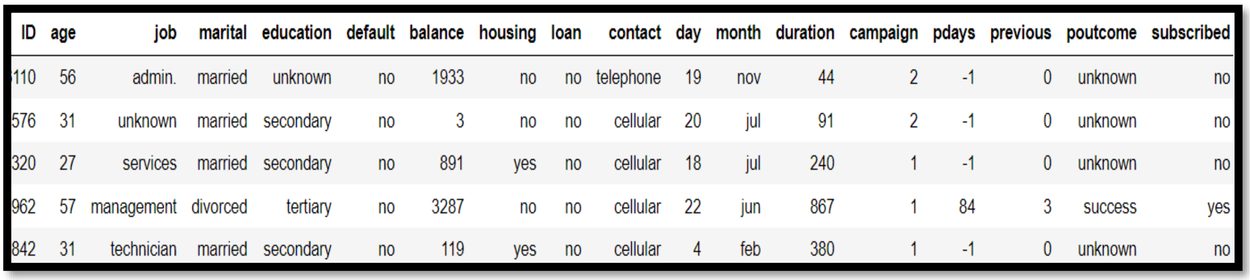
  Decision Tree

  Random Forest

3.1.2 A small overview to the dataset is given below:

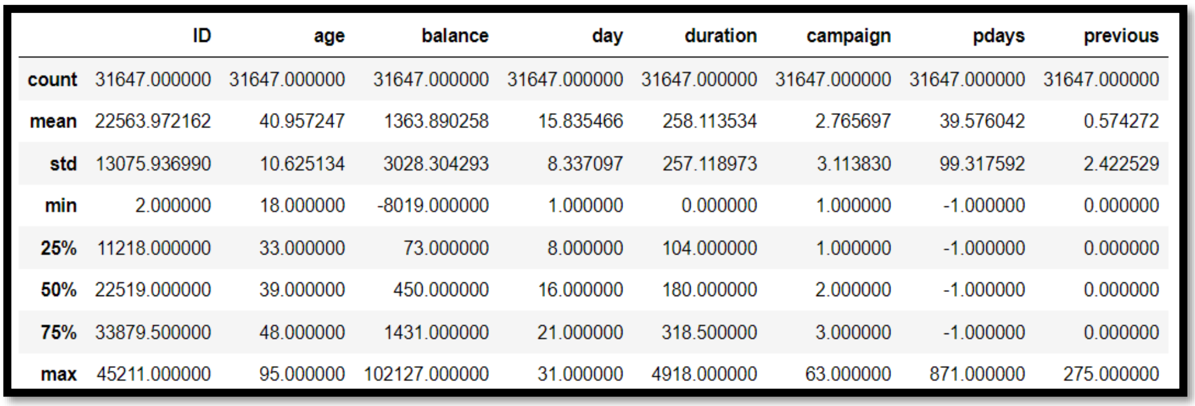
Variable	Definition
ID	Unique client ID
age	Age of the client
job	Type of job
marital	Marital status of the client
education	Education level
default	Credit in default
balance	average yearly balance
housing	Housing loan
loan	Personal loan
contact	Type of communication
month	Contact month
day	Day of week of contact
duration	Contact duration
campaign	number of contacts performed during this campaign to the client
pdays	number of days that passed by after the client was last contacted
previous	number of contacts performed before this campaign
poutcome	outcome of the previous marketing campaign
Subscribed (target)	has the client subscribed a term deposit?

Dataset overview:



ID	age	job	marital	education	default	balance	housing	loan	contact	day	month	duration	campaign	pdays	previous	poutcome	subscribed
110	56	admin.	married	unknown	no	1933	no	no	telephone	19	nov	44	2	-1	0	unknown	no
576	31	unknown	married	secondary	no	3	no	no	cellular	20	jul	91	2	-1	0	unknown	no
320	27	services	married	secondary	no	891	yes	no	cellular	18	jul	240	1	-1	0	unknown	no
962	57	management	divorced	tertiary	no	3287	no	no	cellular	22	jun	867	1	84	3	success	yes
842	31	technician	married	secondary	no	119	yes	no	cellular	4	feb	380	1	-1	0	unknown	no

3.1.3 Description of the dataset



	ID	age	balance	day	duration	campaign	pdays	previous
count	31647.000000	31647.000000	31647.000000	31647.000000	31647.000000	31647.000000	31647.000000	31647.000000
mean	22563.972162	40.957247	1363.890258	15.835466	258.113534	2.765697	39.576042	0.574272
std	13075.936990	10.625134	3028.304293	8.337097	257.118973	3.113830	99.317592	2.422529
min	2.000000	18.000000	-8019.000000	1.000000	0.000000	1.000000	-1.000000	0.000000
25%	11218.000000	33.000000	73.000000	8.000000	104.000000	1.000000	-1.000000	0.000000
50%	22519.000000	39.000000	450.000000	16.000000	180.000000	2.000000	-1.000000	0.000000
75%	33879.500000	48.000000	1431.000000	21.000000	318.500000	3.000000	-1.000000	0.000000
max	45211.000000	95.000000	102127.000000	31.000000	4918.000000	63.000000	871.000000	275.000000

It is clear that a variety of variables, including balance, duration, campaign, and previous, may contain outliers due to absurd max value and mean and median difference. The skewness of the distribution of the cited series is explained by the difference between mean and median.

3.1.4 Correlation Heatmap of the dataset:

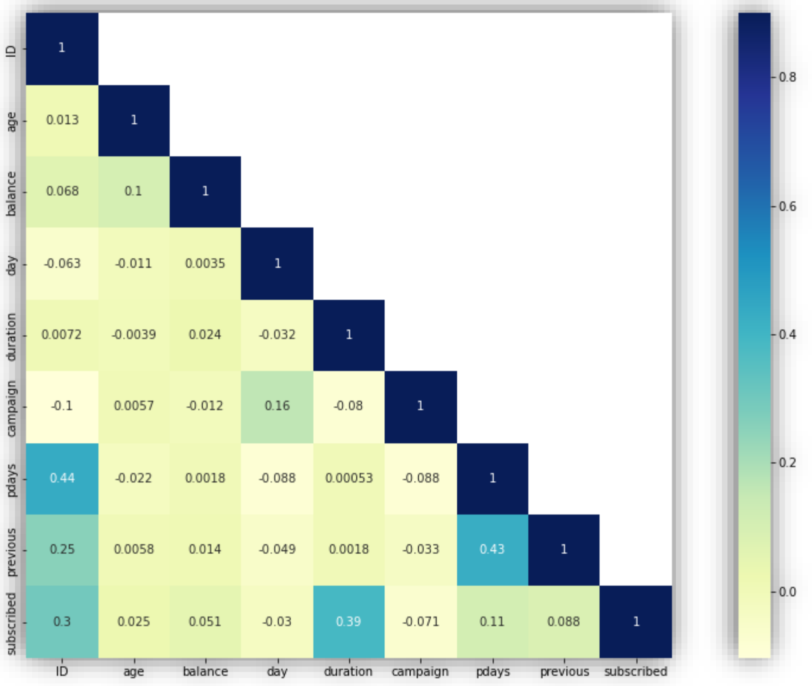
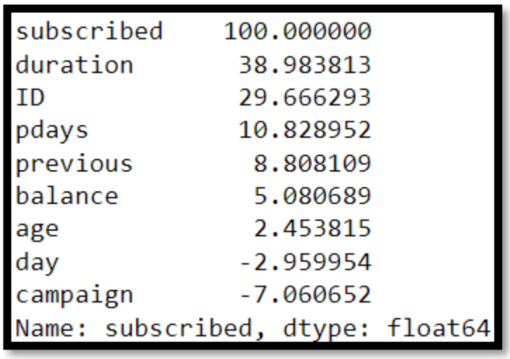


Fig. 3.1.4.(a) Correlation heatmap of the dataset

A statistical indicator of the strength of a linear link between two variables is the correlation coefficient. When there is no known response component, correlation is often used [6]. Its values may be between -1 and 1. Values in one series rise as those in the other drop, and vice versa, according to a correlation coefficient of 1, which denotes a complete negative or inverse connection. A value of 1 indicates a direct and flawlessly positive link. No linear relationship exists when the correlation coefficient is 0.



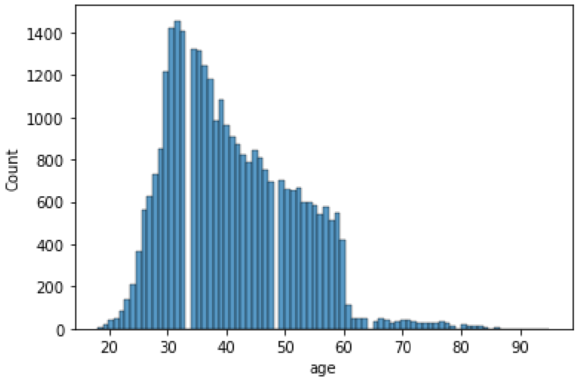
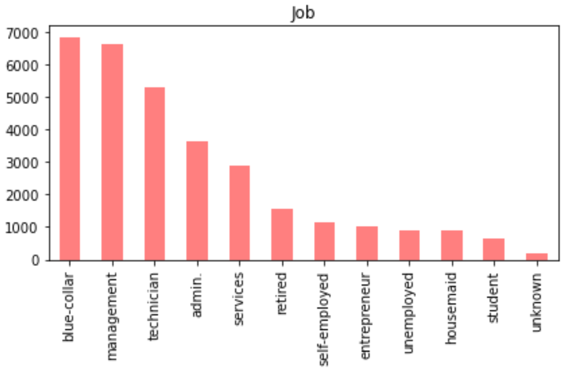
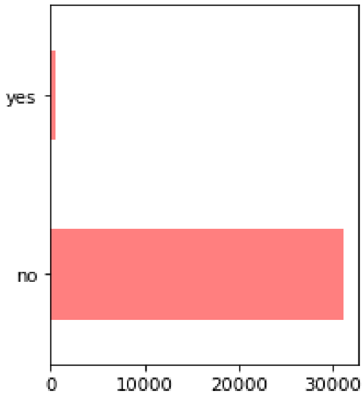
```
subscribed    100.000000
duration       38.983813
ID             29.666293
pdays          10.828952
previous        8.808109
balance         5.080689
age             2.453815
day            -2.959954
campaign       -7.060652
Name: subscribed, dtype: float64
```

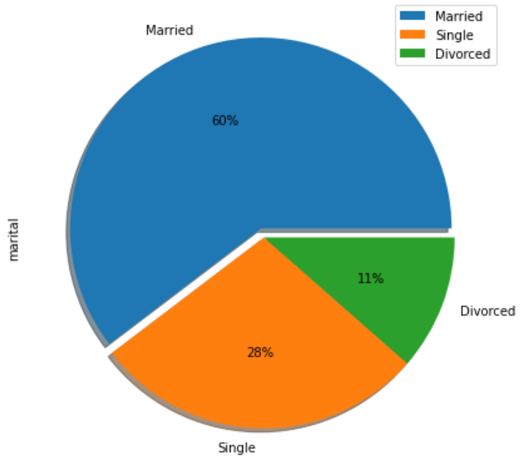
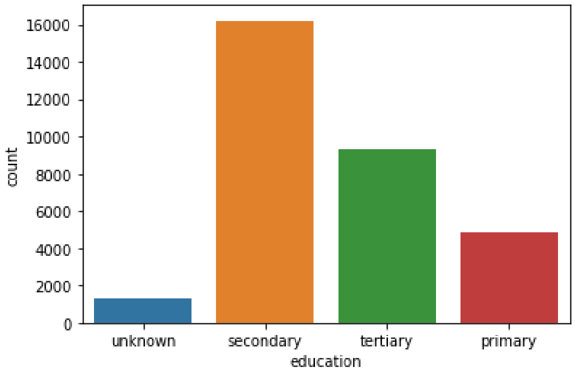
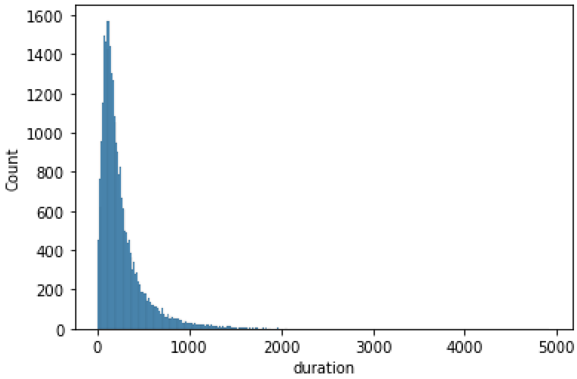
Fig. 3.1.4.(b) Correlation percentage table

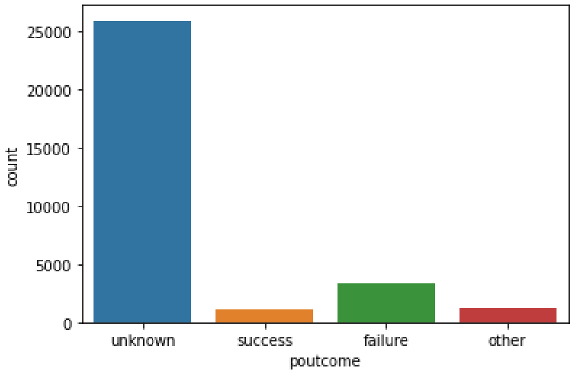
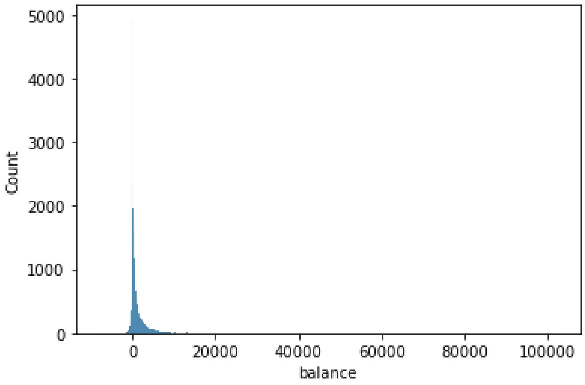
3.2. EDA (Exploratory Data Analysis)

- Exploratory data analysis is the crucial process of doing preliminary analyses on data in order to find patterns, identify anomalies, test hypotheses, and double-check assumptions with the aid of summary statistics and graphical representations. Understanding the data first and attempting to glean as many insights from it as possible is a smart strategy. Before using the data in question, EDA focuses on making sense of it [5].
- Analysing data in the form of graphs or maps makes it much simpler to grasp the trends or patterns in the data. This process is known as data visualization. Different sorts of visualizations exist:
    - Univariate Analysis. Since there is only one variable that varies, univariate data analysis is the most straightforward type of analysis. The analysis's primary goal is to explain the data and identify any patterns in it; it does not deal with causes or correlations.
    - Bi-variate Analysis: Two distinct variables are present in this type of data. The analysis of this kind of data focuses on linkages and causes, and it seeks to understand the causal connection between the two variables.
    - Multi-Variate Analysis: Data that includes three or more variables is referred to as multivariate when analysed.
- Handling outliers: An outlier is a data item or object that dramatically differs from the other (so-called normal) objects. Errors in measurement or execution may be the reason for them. Outlier mining is the analysis used for outlier discovery. There are numerous methods for finding outliers, and removing them works in much the same manner as removing a piece of data from a Panda data frame.

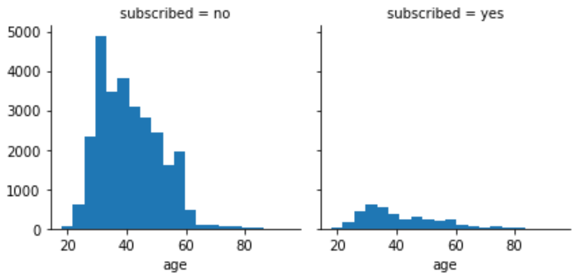
3.2.1 Univariate Analysis:

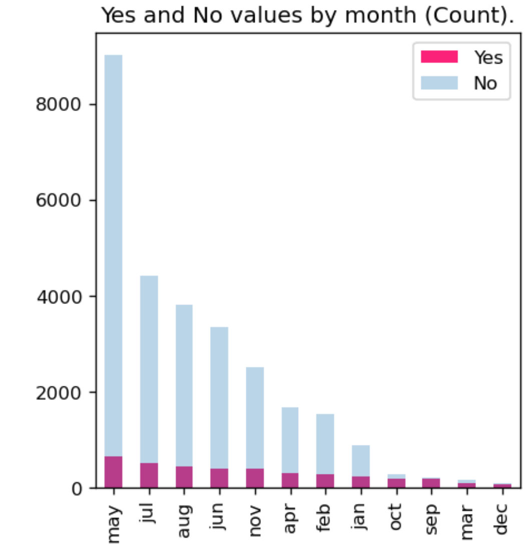
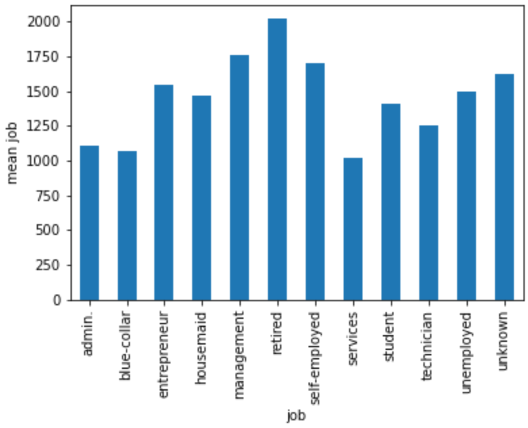
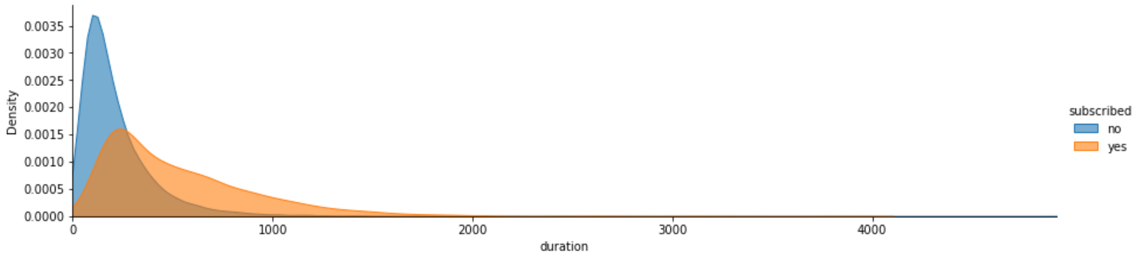
SL	GRAPH	OBSERVATION
1	 **Fig 3.2.1.(a) Age vs Count Histplot**	• From figure 3.2.1.(a), we can see that the clients called by the bank have an extensive age range, from 18 to 95 years old. • However, a majority of customers called is in the range of 30s to 40s. • This particular age group has been targeted since they are in general, working individuals and more likely to subscribe to term deposit to ensure low-risk, stable and safe return on their investments.
2	 **Fig 3.2.1.(b) Job vs Count Barplot**	• From figure 3.2.1.(b), we can see highest count of 'blue-colour' job clients in the underline dataset with the count being as high as 7000. 'Management' and 'technician' clients count are also very high in comparison to other job holders, thereby indicating that high salaried customers are being targeted. • Although term deposits are expected to be subscribed by salaried individuals, yet noticeable number of non-salaried/ less-salaried individuals like students, housemaids and unemployed people have been approached as seen in the graph.
3	 **Fig 3.2.1.(c) Default History vs Count Barplot**	• Approximately 90% of the clients have no default history. • Having no default history explains that most of the clients are eligible to subscribe for a term deposit. • A detailed analysis can be performed for default history since most of the clients have no default history.

SL	GRAPH	OBSERVATION
4	 Fig 3.2.1.(d) Marital status Pieplot	• It may be deduced that the marital variable contains around 60% of clients with the status "married." Married people are more inclined to subscribe to term deposits since they can be utilized for family costs where both partners contribute money. • The remaining 40% is classified as "single" or "divorced." • A thorough correlation graph can assist in determining whether or not marital data need any type of preprocessing.
5	 Fig 3.2.1.(e) Education vs Count Barplot	• In comparison to other education levels, the number of clients with a "secondary" level of education is relatively high. • Additionally, clients with secondary and tertiary education levels should be more inclined to sign up for term deposits compared to clients with primary education levels. By doing a bivariate analysis of clients in relation to their educational achievement, this can be checked. • Here, it is clear that over half of the clients are in the secondary education bracket.
6	 Fig 3.2.1.(f) Duration vs Count Histplot	• The occurrence of big values on the extended axis of duration beyond 2000 in this case can be used to demonstrate the existence of outliers, necessitating further analysis. • The graph's right skewness indicates that the curve has a peak, and the majority of the clients were contacted at that time.

SL	GRAPH	OBSERVATION
7	 **Fig 3.2.1.(g) Poutcome vs Count Barplot**	- We can see that the majority of the poutcome results are unknown out of this. Success, failure, and other outcomes are extremely rare. - To ascertain whether or not this variable may be taken into account while creating our model, a dependency check is necessary. A chi square test will assist us in determining that since the variable is categorical in nature.
8	 **Fig 3.2.1.(h) Balance vs Count Histplot**	- The occurrence of big values on the extended axis of duration beyond 10000 in this case can be used to demonstrate the existence of outliers, necessitating further analysis. - When the distribution is skewed to the right, the mean is often greater than the median.

3.2.2 Bivariate Analysis:

SL	GRAPH	OBSERVATION
1	 **Fig 3.2.2.(a) Age vs Count Histplot wrt Subscribed**	- It is implied that the distribution of customers across all age ranges who subscribe to term deposit is essentially similar to those who didn't. - In contrast, the age group under 60 makes up the majority of customers. - Thus, dividing age with respect to subscribed and non-subscribed customers failed to make much of a difference.

2	 **Fig 3.2.2.(b) Month vs Count Histplot wrt Subscribed**	• The largest number of active involvements is shown in the month of May, according to a properly constructed stacked bar graph demonstrating subscribed status relative to monthly data. • To simplify the dataset, the last two or three columns can be combined. After that, we can determine the proportion by which our predictive model has changed.
3	 **Fig 3.2.2.(c) Job vs Mean Balance bar plot**	• It can be inferred that the average balance of retirees is the highest, followed by that of management and self-employed. • One noticeable thing is that students have a fairly higher amount of balance with that compared to most of other professions.
4	 **Fig 3.2.2.(d) Duration vs Density KDE plot wrt Subscribed**	
	• It can be stated that the density of clients who have not subscribed to the term deposit is significantly more than that of those who have; yet, the value of duration observed was much bigger among those who subscribed when compared to those who did not. • Excessively high duration values along the x axis indicate the presence of outliers in the dataset that will require additional processing before model development.	

Pair plot Analysis:

A pairs plot shows the distribution of a single variable as well as the relationships between two variables. Pair plots are an excellent tool for identifying tendencies for further investigation. The scatter figure in figure 3.2.2.(e), depicts the variation of various numerical variables in our dataset in a scatter plot format, with the 'subscribed' variable functioning as a differentiator. The diagonal plot depicts the range of values that a specific variable has, once again distinguishing subscribers from non-subscribers.

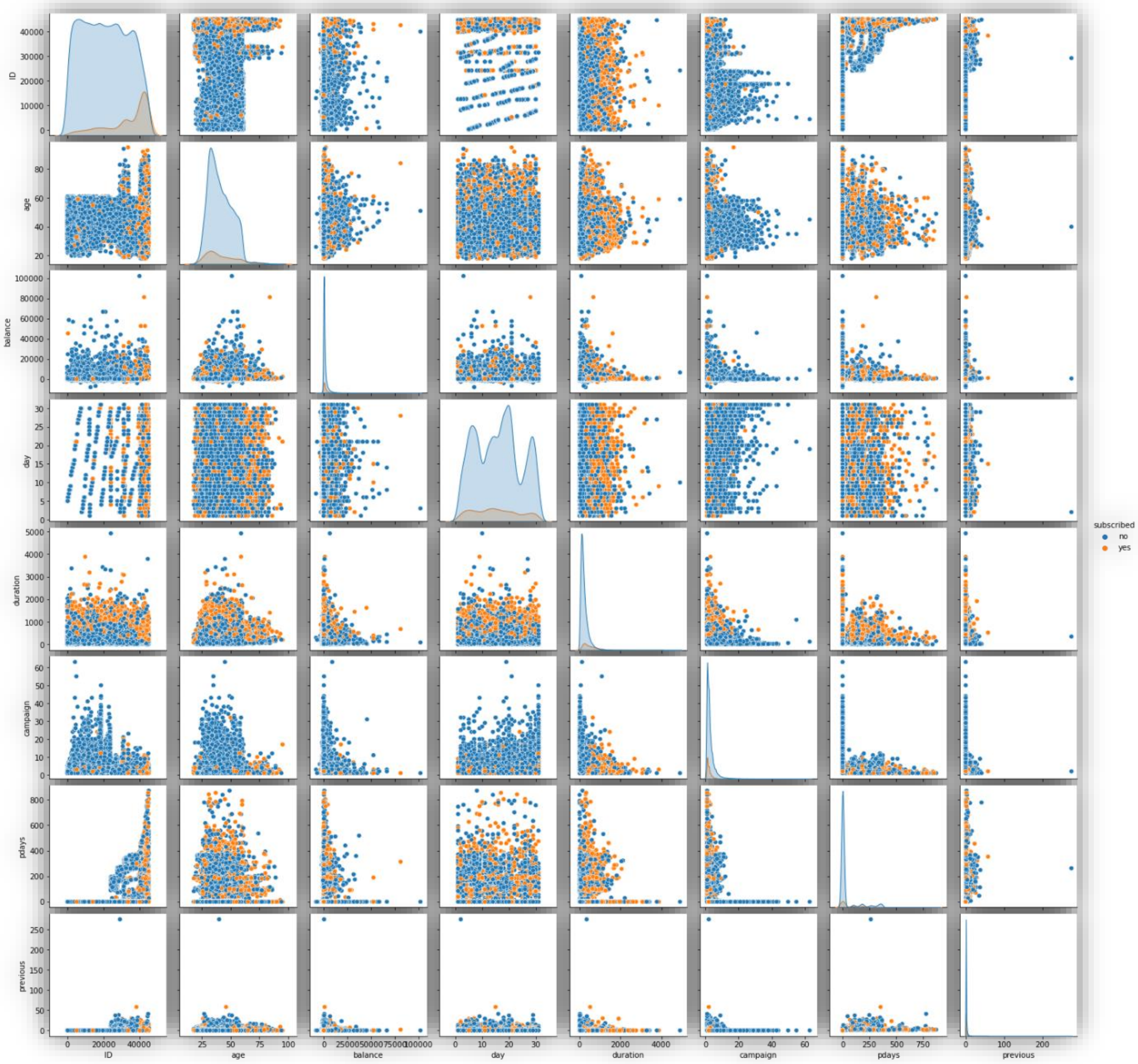


Fig 3.2.2.(e) Pair plot of all numerical variables in the dataset

- According to the diagonal plots in figure x, practically all subscribers have lower values for id (20000), age (30), balance (20000), last contacted day (5), duration (1000), and so on.
- A significant number of plots demonstrate the presence of outliers with regard to our goal variable, which may necessitate additional processing depending on the degree of correlation they have with the target variable.

3.3. Data Pre-processing

The adjustments made to our data prior to feeding it to the algorithm are referred to as pre-processing. Data preprocessing is a method for transforming unclean data into clean data sets. In other words, anytime data is acquired from various sources, it is done so in a raw manner that makes analysis impossible.

Need of Data Preprocessing:

- For achieving better results from the applied model in Machine Learning projects the format of the data has to be in a proper manner. Some specified Machine Learning model needs information in a specified format, for example, Random Forest algorithm does not support null values, therefore to execute random forest algorithm null values have to be managed from the original raw data set.
- Another aspect is that the data set should be formatted in such a way that more than one Machine Learning and Deep Learning algorithm are executed in one data set, and best out of them is chosen.

No columns in our dataset have null values, so we can only govern the outlier segment and conduct outlier treatment as necessary. The dataset's correlation heatmap demonstrates how well characteristics like duration, campaign, and pdays contribute to our model development; as a result, they must be properly examined for the presence of outliers and addressed before we initiate the model building phase.

Outlier treatment:

Although it may be alluring, it is not always the best course of action to simply discard the records where the data collection contains outliers. Before deciding on the method, the business should be consulted. The outlier treatment method can differ from case to case. There are various methods, such as replacing the outlier with the mean value, median value, or in certain cases eliminating the observation containing the suspected outlier in order to remove any bias. If the outliers are the result of human or computer error during data entry or processing, we have a tendency to eliminate them.

Steps to detect outliers using IQR (Inter Quantile Range):

- Calculate the first and third quartiles by sorting the dataset in ascending order (Q1, Q3)
- Compute IQR=Q3-Q1
- Determine the upper bound (Q3+1.5*IQR) and lower bound (Q1-1.5*IQR).
- The dataset's values are iterated over, and any values that deviate from the higher or lower bounds are flagged as outliers and are replaced with lower and upper bounds respectively.

3.3.1 Duration

It is evident that duration plays a significant influence in the creation of our model, with a correlation value of roughly 0.39. Given that the distribution's mean, 258, is higher than its median, 180, and that the distribution's right tail is longer than its left tail, the dataset can be considered to be right skewed since most values are concentrated near the distribution's left tail. Furthermore, it is obvious that the dataset contains outliers and requires outlier treatment given that the greatest value is 4918. The figure 3.3.1.(a), can be used to illustrate the same.

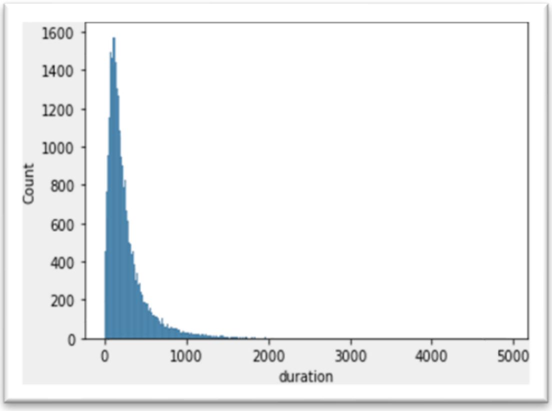


Fig 3.3.1.(a) Duration vs Count before Pre Processing

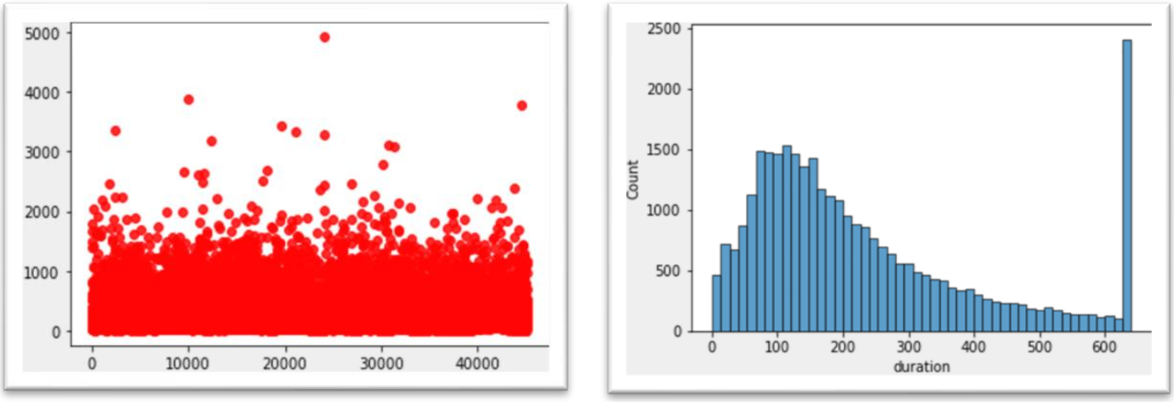


Fig 3.3.1.(b) Duration Scatter Plot showing outliers Fig 3.3.1.(c) Duration vs Count after Pre Processing

The outliers in figure 3.2.1.(b), are treated using the IQR method, and the distribution is now better than it was before as seen in figure 3.3.1.(b), with the outliers being removed and complexity being decreased. The upper and lower limits come out to be 640.25 and -217.75

3.3.2 Campaign

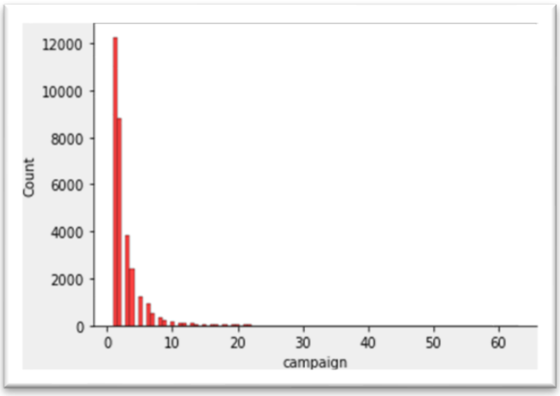
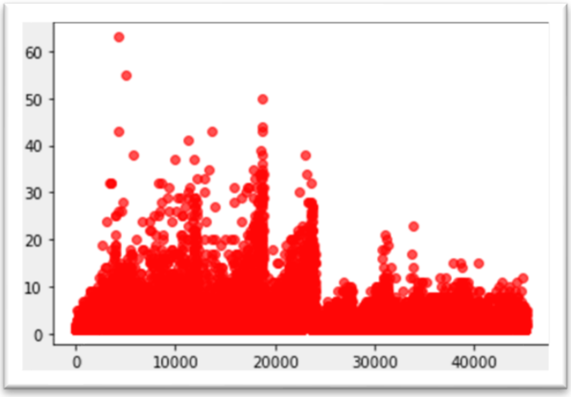


Fig 3.3.2.(a) Campaign vs Count before Pre Processing

Campaign provides information on the total number of contacts made for a customer throughout the campaign. It can be said that it plays a noticeable function for our model with a correlation of 7.1%. Since majority of the clients were contacted one to two times, the median and mean results are both same i.e. two. The histogram in figure 3.3.2.(a) shows that the same conclusion can be drawn. However, given that the maximum value is 63, it can be asserted that it is extremely uncommon to contact a person that many times within a single campaign. Similarly, a negative sign represents absurdness too. The absurdity of the maximum value indicates the existence of outliers.



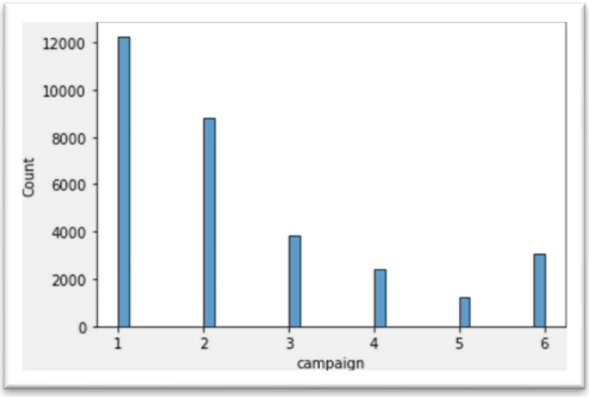


Fig 3.3.2.(b) Campaign Scatter Plot showing outliers Fig 3.3.2.(c) Campaign vs Count after Pre Processing

The distribution is now better than it was before when the outliers in figure 3.3.2.(b) are removed and complexity is reduced, as seen in figure 3.3.2.(c). This is because the outliers in 3.3.2.(b) were treated using the IQR approach. The calculated top and lower limits are 6.0 and 1.0, respectively. The correlation value increased by 1%, going from 7% to 8%, with a negative sign.

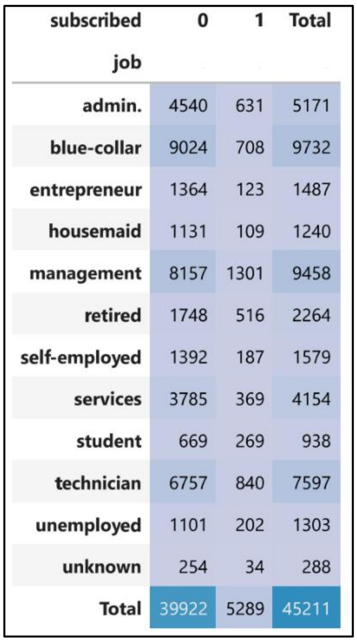
3.4. Extensive Data Processing

Processing a lot of data entails handling, maintaining, and doing analysis on a lot of data. To extract patterns and insights from the data, a variety of methods and technologies are used.

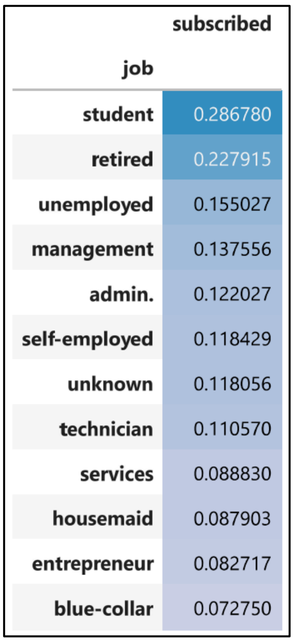
The processes of data cleaning, transformation, integration, and analysis may be a part of extensive data processing. It is frequently used in fields where a lot of data is produced and needs to be analysed to get useful information, such banking, healthcare, marketing, and scientific research.

Extensive data processing can be a difficult and time-consuming operation that calls for specialised knowledge and software solutions. However, the understanding that can be attained from delving into vast amounts of data can result in appreciable improvements in organisational performance, effectiveness, and decision-making.

3.4.1 EDP : Job



subscribed	0	1	Total
job			
admin.	4540	631	5171
blue-collar	9024	708	9732
entrepreneur	1364	123	1487
housemaid	1131	109	1240
management	8157	1301	9458
retired	1748	516	2264
self-employed	1392	187	1579
services	3785	369	4154
student	669	269	938
technician	6757	840	7597
unemployed	1101	202	1303
unknown	254	34	288
Total	39922	5289	45211



	subscribed
job	
student	0.286780
retired	0.227915
unemployed	0.155027
management	0.137556
admin.	0.122027
self-employed	0.118429
unknown	0.118056
technician	0.110570
services	0.088830
housemaid	0.087903
entrepreneur	0.082717
blue-collar	0.072750

Fig 3.4.1. Tables showing distribution of different classes of variable ‘Job’

By looking to the actual number of each contacted class we can clearly see the huge gap between Student and Blue-collar classes. But we also can make the following decision, that the Blue-collar class are not interested joining the deposit. while other classes like Student, retried and manager are more interested to subscribe.

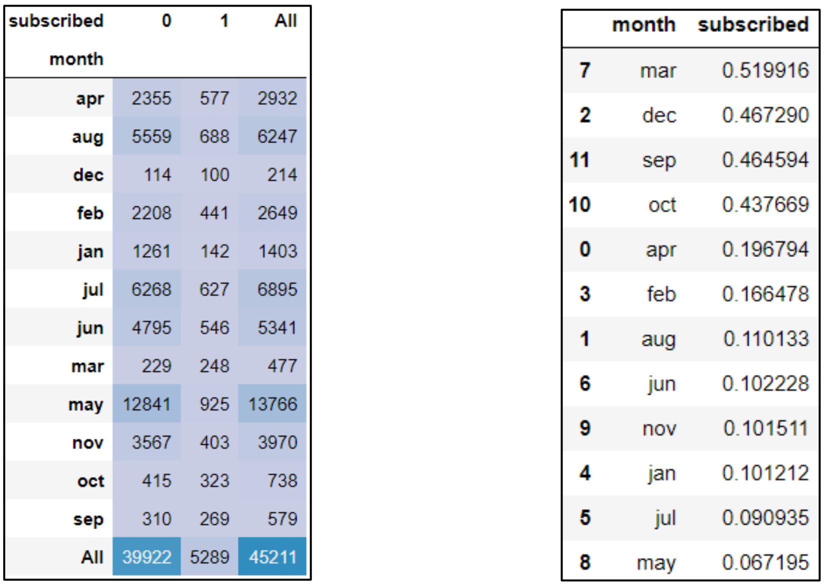
It can also be observed that certain jobs like entrepreneur, blue-collar, services and housemaid share same percentage of subscribed individuals and it is way more less when compared with jobs of other categories. Hence, they can be grouped together while converting them into category based numeric form for them to be categorized as a single entity of job. The correlation after grouping them together and numerical transformation turned out to be 0.12148 that is near to 12%.

3.4.2 EDP : Age

Age Value (in Yrs.)	Group
Age <= 17	0
17 < Age <= 60	1
60 < Age <= 75	2
Age > 75	3

Age is a numerical variable and can be used directly in building models, but its complexity can be minimized by filtering out the outliers or grouping ages into bands. Originally, age had a correlation of 0.02183 (or 2%) with the target variable. However, after removing outliers and grouping the data into groups of four, the correlation increased significantly to 0.14843 (or close to 15%).

3.4.3 EDP : Month



subscribed	0	1	All
month			
apr	2355	577	2932
aug	5559	688	6247
dec	114	100	214
feb	2208	441	2649
jan	1261	142	1403
jul	6268	627	6895
jun	4795	546	5341
mar	229	248	477
may	12841	925	13766
nov	3567	403	3970
oct	415	323	738
sep	310	269	579
All	39922	5289	45211

	month	subscribed
7	mar	0.519916
2	dec	0.467290
11	sep	0.464594
10	oct	0.437669
0	apr	0.196794
3	feb	0.166478
1	aug	0.110133
6	jun	0.102228
9	nov	0.101511
4	jan	0.101212
5	jul	0.090935
8	may	0.067195

Fig 3.4.3 Tables showing distribution of different classes of variable 'Month'

Upon analyzing the number of contacts made with each class, it is apparent that there is a significant difference between the December and May classes. Based on this, it can be concluded that the December and March classes are highly interested in subscribing to the deposit, while the May and July classes are less interested.

Additionally, it can be observed that certain months, including May, June, July, January, and November, share a similar percentage of individuals who have subscribed, and this percentage is significantly lower than that of other job categories. Therefore, these months can be grouped together and converted into a category-based numeric form to be categorized as a single job entity. The correlation before and after grouping and numerical transformation was 2% and 13.5%, respectively.

3.4.4 EDP : Balance

The variable "balance" is a numerical variable that can be used for modeling purposes, but its complexity can be reduced by removing outliers or grouping the data into bands. At first, "balance" had a correlation value of 0.05283 (or 5%) with the target variable. However, after applying outlier removal and grouping the data into groups of four, the correlation value increased considerably to 0.09617 (or approximately 10%).

Balance Value (in Yrs.)	Group
Age <= 0	0
0 < Age <= 750	1
750 < Age <= 1650	2
Age > 1650	3

3.5 Upgraded HeatMap

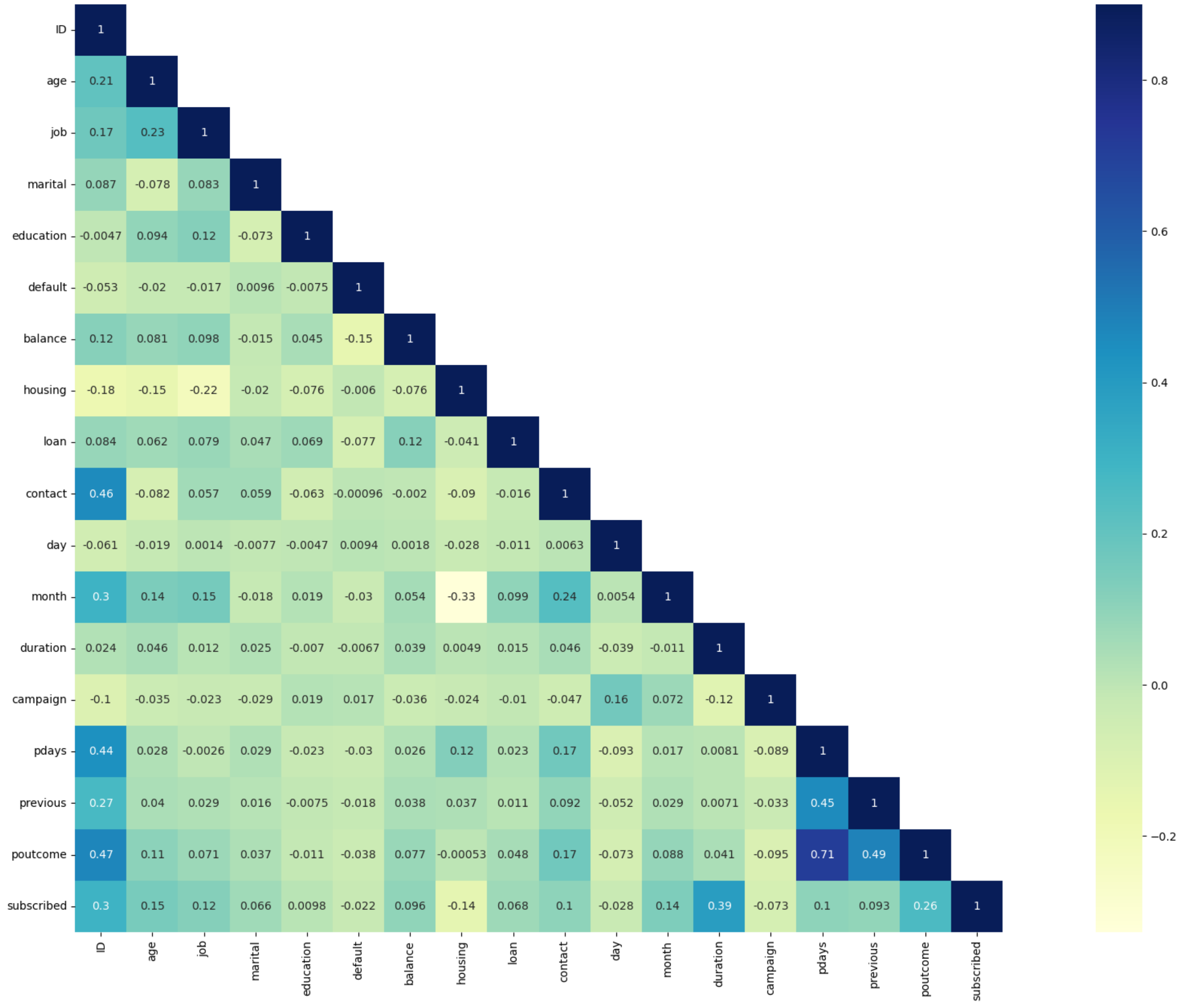
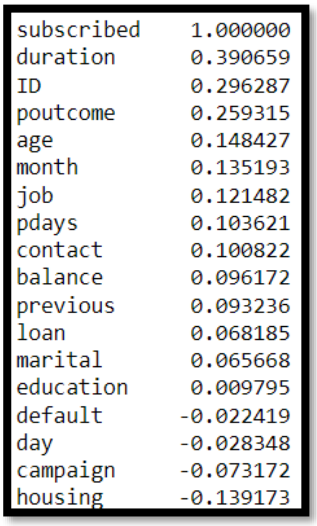


Fig 3.5.1 Upgraded correlation HeatMap

A correlation heatmap is a diagram that shows the correlation matrix, which is a summary of the pairwise correlation coefficients between variables in a dataset. It represents the strength and direction of the correlation between variable pairs. The correlation coefficient has a range of -1 to 1, where 1 means a perfect positive correlation, -1 means a perfect negative correlation, and 0 means no correlation.



```
subscribed     1.000000
duration       0.390659
ID             0.296287
poutcome       0.259315
age            0.148427
month          0.135193
job            0.121482
pdays          0.103621
contact        0.100822
balance        0.096172
previous       0.093236
loan           0.068185
marital        0.065668
education      0.009795
default       -0.022419
day           -0.028348
campaign      -0.073172
housing       -0.139173
```

The correlation heatmap is a valuable tool for identifying relationships and patterns among variables in a dataset, especially in exploratory data analysis, where it can reveal hidden relationships that may not be evident in the raw data.

Color coding is often used in correlation heatmaps to indicate the strength of the correlation, with darker colors representing stronger correlations. This makes it easier to spot strong or weak correlations.

4. ALGORITHM USED

Machine learning is a form of data analysis that allows for analytical models to be automated. It's a branch of artificial intelligence based on the idea that computers can learn from data, analyse data, and make decisions with minimal human intervention [7]. To determine how to achieve the highest degree of accuracy, it's imperative to first understand the capabilities of the intended algorithm, as follows:

4.1. Logistic Regression

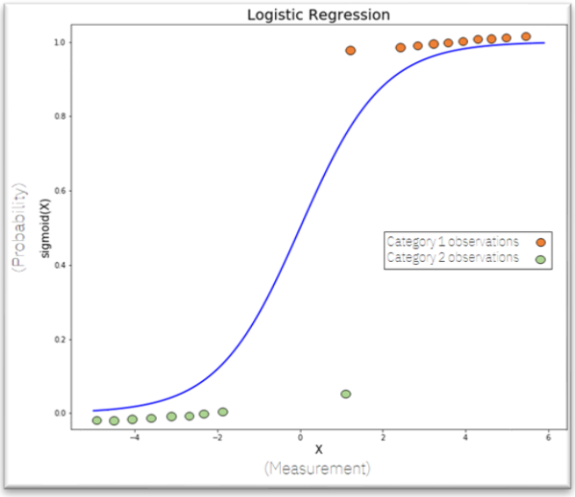


Fig 4.1.(a) Logistic Regression Sigmoid Function

- One of the most well-known machine learning algorithms that falls within supervised learning techniques is logistic regression.
- It can be applied to Classification and Regression issues, but is primarily utilized for Classification issues.
- With the aid of independent factors, categorical dependent variables are predicted using logistic regression.
- The solution to a Logistic Regression problem is only valid for values between 0 and 1.
- Where the probabilities between two classes are needed, logistic regression can be employed. such as true or false, 0 or 1, whether it will rain today, etc.
- Maximum Likelihood estimate serves as the foundation for logistic regression. This estimation suggests
- For logistic regression, the weighted sum of the inputs is passed via an activation function that can transfer values between 0 and 1. Such an activation function is referred to as a sigmoid function, and the resulting curve is known as an S-curve or sigmoid curve.

4.1.1 What is the purpose of Logistic Regression?

- The simplest non-linear transformation of linear regression is logistic regression. Because the output of the sigmoid function is always between 0 and 1, we utilise logistic regression. If we apply linear to the identical situation, we are unable to limit its range to 0 and 1.

- The mathematics for determining how many factors have an effect on a certain result are simplified by the use of logistic regression. The generated models can be used to disentangle the relative efficacy of various interventions for distinct outcome categories [8].
- Additionally, logistic models can modify raw data streams to produce characteristics for various AI and machine learning methods. In fact, one of the often-employed machine learning techniques for binary classification applications is logistic regression.

4.1.2 Why do we need Logistic Regression?

- Logistic regression is significant because it reduces complicated probability calculations to simple arithmetic problems. Although the computation itself is admittedly somewhat complicated, most of the tedious work is automated by contemporary statistical software. This helps to significantly reduce the impact of confusing factors and considerably simplifies studying the impact of various variables.
- As a result, statisticians can study and predict many elements' contributions to a certain outcome fast.

4.1.3 Why Is Logistic Regression Called Regression?

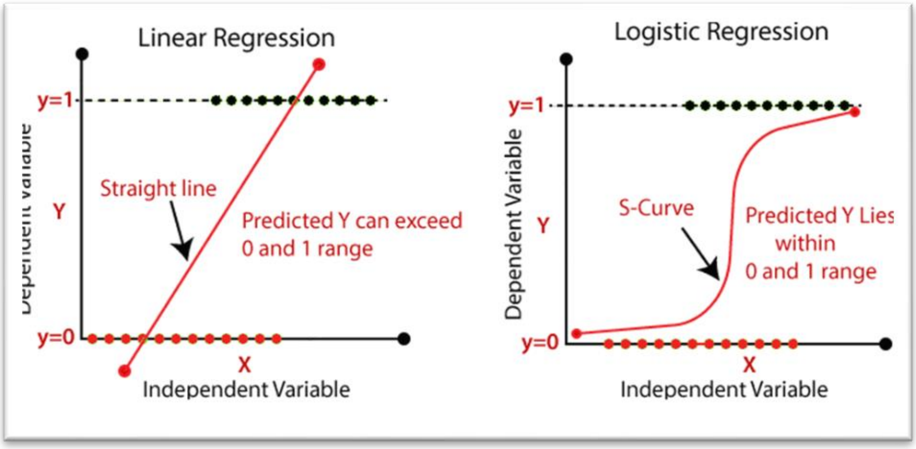


Fig 4.1(b). Comparative observation of Linear and Logistic Regression

- The fundamental formula for logistic regression is identical to that for linear regression, but it is used to calculate the likelihood of a categorical outcome.
- For a given input X, linear regression provides a continuous value of the output y. The output of logistic regression, in contrast, is a continuous value of P(Y=1) for a given input X that is later converted to Y=0 or Y=1 depending on a threshold value.
- The weighted sum of the input variables is used in linear regression to predict the output var Y.

The equation reads as follows:

$$\mathbf{Y = b_0 + b_1X_1 + b_2X_2 + \ldots + b_nX_n}$$

where,

- Y = Dependent Variable (DV)
- $X_1, X_2, \ldots, X_n$ = Independent Variable (IV)
- b_0 = Y-Intercept
- $b_1, b_2, \ldots, b_n$ = Coefficient of slope

- Our primary goals in linear regression are to forecast the output variable Y, minimize the cost function, and estimate the values of the Y-intercept and weights. We carry out the same identical action in logistic regression, with one minor modification. To forecast the output Y, we put the result via a unique function called the Sigmoid Function.

$$\mathbf{Y = Sigmoid\ (b_0 + b_1X_1 + b_2X_2 + \ldots + b_nX_n)}$$

where,

- Sigmoid = f(x) = $1/\ (1+e^{-Y})$

Therefore,

$$\mathbf{Y = 1/\ [1+e\text{-}^{(b_0 + b_1X_1 + b_2X_2 + \ldots + b_nX_n)}]}$$

4.1.4. How does Logistic Regression work?

The logistic regression equation and the linear regression model are very similar.

Consider a model with one predictor "x" and one Bernoulli response variable " ŷ" where p is the probability that ŷ=1. The linear equation is written as follows:

$$\mathbf{p = b_0+b_1x} \quad \text{-------- eq 1}$$

The right-hand side of the equation (b0+b1x) is a linear equation that can hold values that are greater than the range (0,1). However, we know that probability will always be in the range of (0,1).

To circumvent this, we anticipate odds rather than probability where odds is the ratio of the probability of an event occurring to the probability of an event not occurring.

$$\mathbf{Odds = p/(1-p)}$$

Equation 1 can be rewritten as follows:

$$\mathbf{p/(1-p) = b_0+b_1x} \quad \text{-------- eq 2}$$

Odds can only be positive; thus, we anticipate the logarithm of odds to deal with negative quantities.

Odds log = ln(p/(1-p))

The equation 2 can be re-written as:

$$\mathbf{ln(p/(1-p)) = b_0+b_1x \quad \text{-------- eq 3}}$$

We use exponential on both sides to recover p from equation 3.

$$\mathbf{exp(ln(p/(1-p))) = exp(b_0+b_1x)}$$

$$\mathbf{e^{ln(p/(1-p))} = e^{(b0+b1x)}}$$

From the inverse rule of logarithms,

$$\mathbf{p/(1-p) = e^{(b0+b1x)}}$$

Simple algebraic manipulations

$$\mathbf{p = (1-p) * e^{(b0+b1x)}}$$

$$\mathbf{p = e^{(b0+b1x)}- p * e^{(b0+b1x)}}$$

Taking p as common on the right-hand side

$$\mathbf{p = p * ((e^{(b0+b1x)})/p - e^{(b0+b1x)})}$$

$$\mathbf{p = e^{(b0+b1x)} / (1 + e^{(b0+b1x)})}$$

Dividing numerator and denominator by $e^{(b0+b1x)}$ on the right-hand side

$$\mathbf{p = 1 / (1 + e^{-(b0+b1x)})}$$

Similarly, the equation for a logistic model with ‘n’ predictors is as below:

$$\mathbf{p = 1/ (1 + e^{-(b0+b1x1+b2x2+b3x3+----+bnXn)})}$$

The right-side is the **sigmoid function**. It helps to squeeze the output to be in the range between 0 and 1.

4.1.5. Advantages of Logistic Regression

- Logistic regression is easier to implement, interpret, and very efficient to train.
- It can easily extend to multiple classes and a natural probabilistic view of class predictions.
- It is very fast at classifying unknown records.
- Logistic regression is less inclined to over-fitting but it can overfit in high dimensional datasets

4.1.6 Disadvantages of Logistic Regression

- If the number of observations is lesser than the number of features, Logistic Regression should not be used, otherwise, it may lead to overfitting.
- Logistic Regression requires average or no multicollinearity between independent variables.

4.2 Decision Tree

- A supervised learning method called a decision tree can be used to solve classification and regression problems, but it is typically favored for doing so. It is a tree-structured classifier, where internal nodes stand in for a dataset's features, branches for the decision-making process, and each leaf node for the classification result [9].

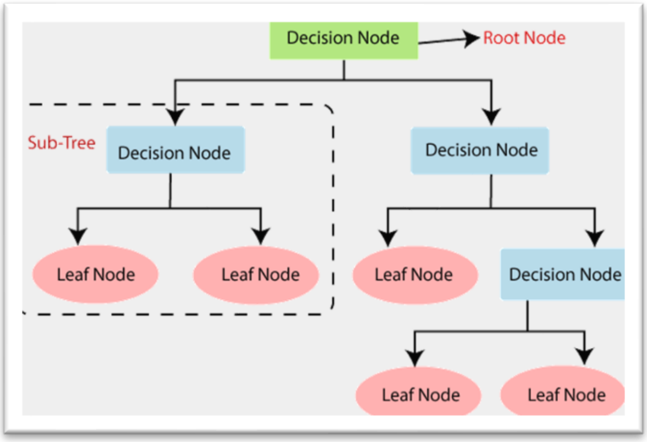


Fig 4.2.(a) Decision Tree Overview

- The Decision Node and Leaf Node are the two nodes of a decision tree. While Leaf nodes are the results of decisions and do not have any more branches, Decision nodes are used to create decisions and have numerous branches.
- The given dataset's features are used to execute the test or make the decisions.
- It is a graphical depiction for obtaining all feasible answers to a choice or problem based on predetermined conditions.
- It is known as a decision tree because, like a tree, it begins with the root node and grows on subsequent branches to form a structure resembling a tree.
- The CART algorithm, which stands for Classification and Regression Tree algorithm, is used to construct a tree.
- A decision tree only poses a question and divides the tree into subtrees according to the response (Yes/No).

4.2.1 Assumptions for decision tree

- At first, the entire training set is regarded as the root.
- Categorical feature values are desired. If the values are continuous, they must first be discretized before the model can be constructed.
- On the basis of attribute values, records are dispersed recursively.
- Using various statistical methods, the order to place characteristics as the root or internal node of the tree is determined.

4.2.2 Why use decision trees?

- Decision trees assist you in weighing your options.
- Decision trees are great tools for assisting you in selecting one course of action over others.
- They offer a very useful framework within which you can present options and research the potential results of those options. They also assist you in developing a balanced understanding of the benefits and drawbacks of each potential course of action.

4.2.3 How does a decision tree works?

- A decision tree looks like a tree, naturally. The root node is where the tree starts. A string of decision nodes that represent decisions to be made flow from the root node. Leaf nodes that indicate the decisions' implications emerge from the decision nodes.
- The leaf nodes that emanate from a decision node indicate the potential replies, and each decision node acts as a question or split point.
- Node splitting, or simply splitting, is the process of dividing a node into multiple sub-nodes to create relatively pure nodes. There are multiple ways of doing this, which can be broadly divided into two categories based on the type of target variable:
  - Continuous Target Variable: Reduction in Variance
  - Categorical Target Variable: Gini Impurity, Information Gain, Chi-Square
- We’ve performed our splitting and analysis using Gini Impurity.

Gini Impurity

When the target variable is categorical, the nodes can be divided using the Gini Impurity technique. The most common and straightforward method of splitting a decision tree is this one. The value of the Gini impurity is

$Gini\ Impurity = 1 - Gini$

And Gini Impurity is:

$Gini = \sum_{1}^{n} p_i^2$

And Gini Impurity is:

$Gini\ Impurity = 1 - \sum_{1}^{n} p_i^2$

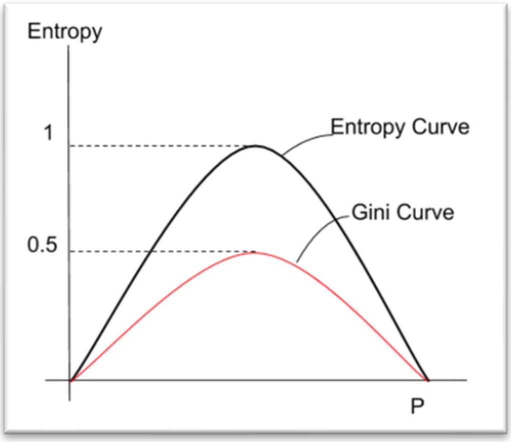


Fig 4.2.(b) Gini Impurity and Entropy Curve

The homogeneity of the node is higher when the Gini Impurity is lower. An entirely pure node has zero Gini impurities.

Steps followed:

1. In the same manner as how, we obtained information, determine the Gini Impurity of each child node for each split.
2. Calculate the weighted average of the Gini Impurity for each split. Gini Child Node Impureness
3. Choose the split with the lowest Gini Impurity value.
4. Repeated steps 1-3 until homogenous nodes are achieved.

4.2.4 Advantages of using Decision Tree:

- Decision trees take less work to prepare the data during pre-processing than other methods do.
- Data normalization is not necessary for a decision tree.
- Scaling of data is not necessary when using a decision tree.
- Additionally, the construction of a decision tree is not significantly impacted by missing values in the data.
- Technical teams and stakeholders can understand a decision tree model very quickly.

4.2.4 Disadvantages of using Decision Tree:

- A slight change in the data can result in a big change in the decision tree's structure, which can lead to instability.
- When compared to other algorithms, a decision tree's calculations can occasionally become significantly more complex.
- The model training process for decision trees typically takes longer.

- Because of its intricacy and lengthier training period, decision tree training is relatively expensive.
- One of the most significant problems is overfitting on train data.

4.3. Random Forest

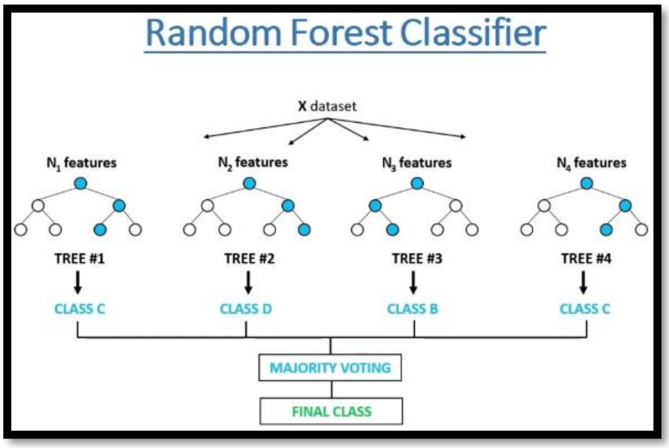


Fig 4.3.(a) Random Forest Overview

- An effective supervised learning method is Random Forest, a well-known machine learning algorithm. It can be applied to ML issues involving both classification and regression. It is built on the idea of ensemble learning, which is a method of integrating various classifiers to address difficult issues and enhance model performance.
- Random Forest, as the name implies, is a classifier that uses a number of decision trees on different subsets of the provided dataset and averages them to increase the dataset's predictive accuracy. Instead, then depending on a single decision tree, the random forest uses forecasts from each tree and predicts the result based on the votes of the majority of predictions.
- The more trees there are in the forest, the higher the accuracy and less chance of overfitting.

4.3.1 Assumptions for Random Forest

- Some decision trees may predict the correct output, while others may not, because the random forest combines numerous trees to forecast the class of the dataset. But when all the trees are combined, they forecast the right result. As a result, the following are two hypotheses for an improved Random Forest classifier:
- The feature variable in the dataset should have some actual values so that the classifier can forecast accurate outcomes rather than a guessed result.
- There must be extremely little correlation between each tree's predictions.

4.3.2 Why use Random Forest?

The Random Forest method should be used for the reasons listed below:

- In comparison to other algorithms, it requires less training time.
- Even with the enormous dataset, it operates effectively and predicts the outcome with a high degree of accuracy.
- When a significant amount of the data is absent, accuracy can still be maintained.

4.3.3 How does Random Forest algorithm work?

First, N decision trees are combined to generate the random forest, and then predictions are made for each tree that was produced in the first phase. The following steps can be used to explain the working process:

1. Pick K data points at random from the training set.
2. Create the decision trees linked to the chosen data points (Subsets).
3. For any decision trees you intend to construct, select N.
4. Replicate steps 1 and 2.
5. Find each decision tree's forecasts for any new data points, then place them in the category that receives the most votes.

4.3.4 Applications of Random Forest

Random Forest is mostly utilised in four industries:

1. Banking: This algorithm is mostly used in the banking sector to identify loan risk.
2. Medicine: This technique can be used to identify disease patterns and disease risks.
3. Marketing: This algorithm can be used to spot marketing trends.

4.3.5 Advantages of Random Forest

- Both classification and regression tasks can be handled by Random Forest.
- It is able to handle big datasets with lots of dimensions.
- It improves the model's accuracy and avoids the overfitting problem.

4.3.6 Disadvantages of Random Forest

- Random forest can be used for both classification and regression tasks, but regression tasks are not better suited for it.

4.4 Naïve Bayes

Naive Bayes is a simple yet powerful probabilistic algorithm used for classification tasks. It's based on the Bayes theorem, which states that the probability of an event (such as a classification) occurring given some evidence (such as the features of a sample) is proportional to the probability of that evidence given the event.

Naive Bayes is called "naive" because it assumes that all the features are independent of each other, even though that might not be true in practice. Despite this oversimplification, Naive Bayes often performs well and is widely used in various applications, including text classification, spam filtering, and sentiment analysis.

There are three main types of Naive Bayes classifiers:

- Bernoulli Naive Bayes: used for binary data (i.e., data that can take on only two values, such as 0 and 1).
- Multinomial Naive Bayes: used for discrete data (i.e., data that can take on a countable number of values, such as word frequencies in a text document).
- Gaussian Naive Bayes: used for continuous data (i.e., data that can take on any value in a continuous range, such as the height of a person).

To train a Naive Bayes classifier, we use a labelled dataset where the features and their corresponding classifications are known. The algorithm estimates the prior probabilities of each class and the likelihood of each feature given each class. Once the model is trained, it can be used to classify new samples by calculating the probabilities of each class given the features and selecting the class with the highest probability.

Naive Bayes is known for its simplicity, speed, and ability to handle high-dimensional data. However, it may not perform well if the independence assumption is strongly violated, or if there is insufficient data to estimate the probabilities accurately.

4.4.1 Assumptions of Naïve Bayes Classifier

The Naive Bayes classifier is based on some assumptions that must hold for the classifier to work well. The main assumption is that the features are conditionally independent given the class label. This means that the presence or absence of one feature should not affect the probability of another feature occurring, given the class label. In other words, all the features are assumed to contribute independently and equally to the classification decision. This assumption is often referred to as the "naive" assumption, as it is considered oversimplified.

Other assumptions of the Naive Bayes classifier include:

- The data is randomly sampled from the population of interest.
- The features are mutually exclusive and exhaustive, i.e., each sample can belong to only one class, and all possible outcomes are covered by the features.
- The feature distribution follows a specific probability distribution, depending on the type of Naive Bayes classifier used (e.g., Bernoulli, multinomial, or Gaussian).
- The training set is representative of the population of interest, and the class labels are accurately assigned to each sample.
- The size of the training set is sufficient to estimate the probability distributions accurately.

If these assumptions are violated, the Naive Bayes classifier's performance may degrade, and its accuracy may be compromised. Therefore, it is essential to carefully consider the data and the problem at hand before applying Naive Bayes and to validate the assumptions by testing the model's performance on a separate test set.

4.4.2 Why do we use Naïve Bayes Classifier?

There are several reasons why the Naive Bayes classifier is commonly used in machine learning:

- Simplicity: Naive Bayes is a simple and easy-to-implement algorithm that requires minimal training data compared to other classification algorithms. It is computationally efficient and can handle large datasets with high-dimensional feature spaces.
- Speed: Naive Bayes is a fast algorithm, making it suitable for real-time prediction tasks or applications that require quick responses.
- Robustness: Naive Bayes can handle missing data or incomplete information by ignoring irrelevant features. It is also less prone to overfitting, as it is less sensitive to noise in the data.
- Good performance: Despite its simplicity, Naive Bayes can perform well on many classification problems, especially when the assumption of feature independence holds

approximately.

- Interpretable: Naive Bayes is a probabilistic algorithm, which allows for interpreting the results in terms of probabilities and conditional probabilities. It provides a measure of uncertainty in the predictions, which can be useful in some applications.

Naive Bayes is commonly used in various applications, including text classification, spam filtering, sentiment analysis, recommendation systems, and medical diagnosis. Its simplicity and speed make it a popular choice for many real-world problems, especially when there is limited training data or when the feature space is high-dimensional. However, it is important to validate the assumptions of the algorithm and carefully evaluate its performance before applying it to a specific problem.

4.4.3 How does Naïve Bayes classifier works?

1. Data Preparation: The first step is to collect and pre-process the data. This includes selecting relevant features, cleaning the data, and splitting it into training and testing sets.
2. Calculate Prior Probabilities: The next step is to calculate the prior probabilities of each class, which are the probabilities of each class occurring in the training data without considering any features. This can be calculated by dividing the number of samples in each class by the total number of samples.
3. Calculate Likelihood Probabilities: For each feature and class combination, the likelihood probability is calculated. This is the probability of observing a specific feature given a specific class. Depending on the type of Naive Bayes classifier used, this probability can be calculated differently.
4. Calculate Posterior Probabilities: The posterior probabilities are the probabilities of each class given the observed features. To calculate the posterior probability of each class, the prior probability of the class is multiplied by the likelihood probabilities of each feature in the class. This calculation is performed for each class, and the class with the highest posterior probability is selected as the predicted class.
5. Model Evaluation: The final step is to evaluate the model's performance on the testing data. This includes calculating various performance metrics such as accuracy, precision, recall, and F1-score.

4.4.4 Applications of Naïve Bayes Classifier:

1. Text classification: Naive Bayes is widely used in natural language processing tasks such as sentiment analysis, spam filtering, and document classification.
2. Recommendation systems: Naive Bayes can be used in recommendation systems to predict user preferences based on their past behavior or feedback.
3. Medical diagnosis: Naive Bayes can be used to classify medical data, such as patient symptoms and test results, to aid in the diagnosis of diseases.
4. Customer segmentation: Naive Bayes can be used in marketing to segment customers based on their demographics, behavior, and preferences.
5. Quality control: Naive Bayes can be used in manufacturing to classify defective products based on their features or attributes.
6. Financial analysis: Naive Bayes can be used in finance to predict market trends, classify investments, or detect anomalies.

4.4.5 Advantages of Naïve Bayes Classifier:

- Simple and easy to implement: Naive Bayes is a simple and straightforward algorithm that is easy to understand and implement. It is a suitable choice for beginners in machine learning.
- Fast and efficient: Naive Bayes is a fast algorithm that can handle large datasets with high-dimensional feature spaces. It requires minimal training time and can make predictions in real-time.
- Robust to noise and irrelevant features: Naive Bayes can handle missing data or irrelevant features by ignoring them. It is less prone to overfitting and can handle noisy or incomplete data.
- Can handle categorical and continuous data: Naive Bayes can handle both categorical and continuous data, making it a versatile algorithm that can be applied to various types of data.
- Performs well on small datasets: Naive Bayes can perform well on small datasets and can handle sparse data. It requires fewer training samples compared to other machine learning algorithms.

4.4.5 Disadvantages of Naïve Bayes Classifier:

- Strong assumption of independence: The Naive Bayes classifier assumes that all features are independent of each other given the class, which may not hold true in real-world data. This can lead to inaccuracies in the model's predictions.
- Limited expressiveness: The Naive Bayes classifier is a linear classifier, meaning it cannot capture complex relationships between features or non-linear decision boundaries. It may not perform well on data with complex interactions between features.
- Sensitivity to irrelevant features: Although Naive Bayes can handle irrelevant features, it can still be sensitive to them. If irrelevant features are included in the model, they can negatively impact the model's performance.
- Limited ability to handle continuous data: Naive Bayes assumes that continuous data follows a Gaussian distribution, which may not hold true in all cases. If the data does not follow a Gaussian distribution, the model's performance may suffer.
- Requires large amounts of training data for accurate probability estimation: The accuracy of the Naive Bayes classifier depends on the quality and quantity of the training data. It may require a large amount of training data to accurately estimate the probabilities and avoid overfitting.

4.5 KNN Algorithm:

K-Nearest Neighbours (KNN) is a simple, non-parametric algorithm used for classification and regression. It is a supervised learning algorithm that uses a database of labelled instances to make predictions on new, unlabelled instances.

In the KNN algorithm, each instance is represented as a point in a multi-dimensional space, where each dimension corresponds to a feature or attribute. To classify a new instance, KNN finds the K nearest labelled instances to the new instance based on the distance metric, where K is a user-defined parameter. The majority class of these nearest neighbours is then assigned as the predicted class for the new instance.

The distance metric used in KNN can be Euclidean distance, Manhattan distance, or other distance measures. The choice of distance metric can affect the performance of the algorithm, and it is important to select an appropriate metric based on the nature of the data.

KNN can be used for both classification and regression tasks. In classification, KNN assigns a class label to a new instance based on the majority class of the K nearest neighbours. In regression, KNN estimates the output value of a new instance based on the average or weighted average of the K nearest neighbours.

4.5.1 Assumptions of KNN

- The data should be numerical: KNN works best with numerical data, as it uses distance metrics to calculate the similarity between the data points.
- The data should be normalized: KNN is sensitive to the scale of the features, and therefore, it is recommended to normalize the data before applying the algorithm. This is important to ensure that no single feature dominates the distance calculation.
- The distance metric should be carefully chosen: The choice of distance metric can have a significant impact on the performance of KNN. The most commonly used distance metrics are Euclidean distance and Manhattan distance.
- The value of K should be chosen carefully: The performance of KNN is highly dependent on the value of K. If K is too small, the algorithm may overfit the data, while if K is too large, the algorithm may underfit the data. Therefore, it is important to choose the optimal value of K based on the specific dataset and problem being solved.
- The dataset should be representative: KNN works best when the dataset is representative of the problem being solved. If the dataset is biased or incomplete, the algorithm may not be able to make accurate predictions.

4.5.2 Why do we use KNN?

KNN is particularly useful in cases where the relationship between the input features and the output variable is nonlinear or when the data is not well represented by a simple parametric model. KNN can also be used when there is no prior knowledge about the underlying distribution of the data.

We use K-Nearest Neighbours (KNN) algorithm in machine learning for two main reasons:

- Classification: KNN can be used for classification tasks, where the goal is to predict the class of a given input based on its features. For example, KNN can be used to classify whether an email is spam or not, based on the words in the email.

- Regression: KNN can also be used for regression tasks, where the goal is to predict a continuous output variable based on the input features. For example, KNN can be used to predict the price of a house based on its features such as number of bedrooms, bathrooms, and square footage.

4.5.3. How does KNN works?

1. Select the value of K: K is a hyperparameter that represents the number of neighboring points that will be considered when making a prediction. It is important to choose an appropriate value of K, as a larger value of K will result in a smoother decision boundary, while a smaller value of K will result in a more complex decision boundary.
2. Calculate the distance: Calculate the distance between the input point and all the points in the dataset. The most commonly used distance metric is the Euclidean distance, which measures the straight-line distance between two points.
3. Select the K nearest neighbors: Select the K points in the dataset that are closest to the input point based on the distance metric. These are the K nearest neighbors.
4. Make a prediction: For classification, the predicted class is the mode of the K nearest neighbors. For regression, the predicted value is the mean or median of the K nearest neighbors.
5. Evaluate the performance: Finally, evaluate the performance of the model using a performance metric such as accuracy (for classification) or mean squared error (for regression).

4.5.4. Applications of KNN

- Image recognition: KNN can be used in image recognition to classify images based on their features such as color, texture, and shape.
- Text classification: KNN can be used in text classification to classify documents based on their content, such as sentiment analysis or spam detection.
- Bioinformatics: KNN can be used in bioinformatics to classify proteins or DNA sequences based on their similarity.
- Finance: KNN can be used in finance to predict stock prices or detect fraud based on historical data.

- Healthcare: KNN can be used in healthcare to predict patient outcomes or classify medical images such as MRI scans.
- Marketing: KNN can be used in marketing to predict customer behavior or segment customers based on their preferences.

4.5.4. Advantages of KNN:

- Simple and easy to implement: KNN is a simple algorithm that is easy to understand and implement. It does not require any assumptions about the underlying data distribution or the functional form of the model.
- No training required: Unlike other machine learning algorithms such as neural networks or support vector machines, KNN does not require any training. This means that the model can be used immediately after the data is prepared.
- Non-parametric: KNN is a non-parametric algorithm, which means that it does not assume any particular distribution of the data. This makes it more flexible and able to handle a wider range of data types.
- Robust to noisy data: KNN can handle noisy and missing data effectively by taking into account the neighboring points when making predictions. This makes it more robust to outliers and errors in the data.
- Can handle multi-class problems: KNN can be used to solve multi-class problems, where there are more than two classes to be predicted.
- Can be used for both classification and regression: KNN can be used for both classification and regression tasks, making it a versatile algorithm that can be used in a wide range of applications.

4.5.4. Disadvantages of KNN:

- Computationally expensive: KNN is computationally expensive, especially when working with large datasets. This is because the algorithm has to calculate the distance between the input point and all the points in the dataset.
- Sensitive to feature scaling: KNN is sensitive to the scale of the features. If the features are not scaled properly, the algorithm may give too much importance to some features and too little to others.

- Not suitable for high-dimensional data: KNN is not suitable for high-dimensional data, as the distance between the points becomes less meaningful in high-dimensional space. This is known as the "curse of dimensionality."
- Can be biased towards the majority class: In imbalanced datasets where one class is significantly more frequent than the others, KNN can be biased towards the majority class.
- Not suitable for noisy data: KNN can be sensitive to noisy data, as noisy data can affect the distance calculation and lead to incorrect predictions.

5. EXPERIMENTAL RESULTS

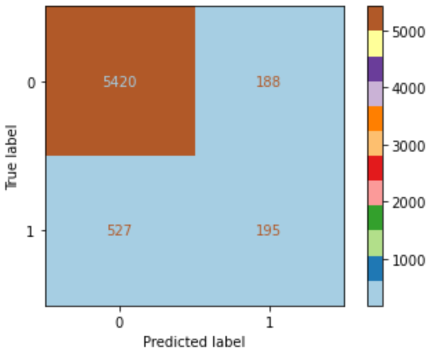
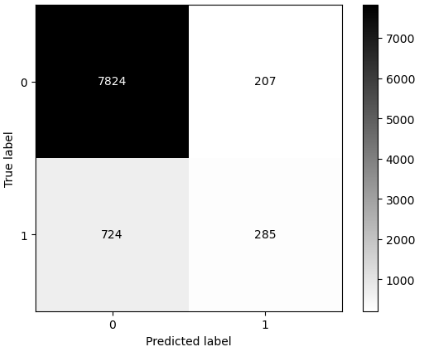
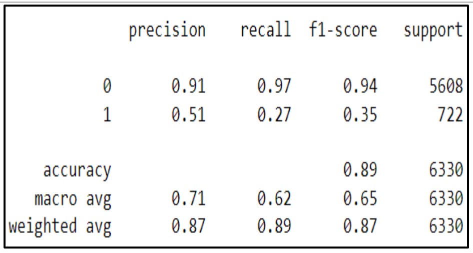
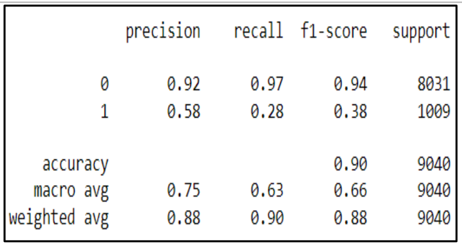
Our original model was created using a dataset that had over 30,000 observations and all 17 independent variables were included. However, we have since used backward feature selection on a new dataset with over 45,000 observations to reduce the number of independent variables considered from 17 to 8. This was accomplished by excluding variables that displayed high collinearity with other independent variables, as well as variables that contributed little to the model building phase.

The basic idea of backward feature selection is to start with a model that includes all the available features, and then iteratively remove the least important features until a desirable level of performance is achieved. The process begins by fitting a model with all the available features, and then determining the importance of each feature using a specified criterion, such as p-values or coefficients from a regression model. The feature with the lowest importance score is then removed from the model, and the process is repeated with the remaining features until a stopping criterion is met.

Below table shows results for difference Machine Learning models along with evaluation metrics [10].

5.1. Logistic Regression:

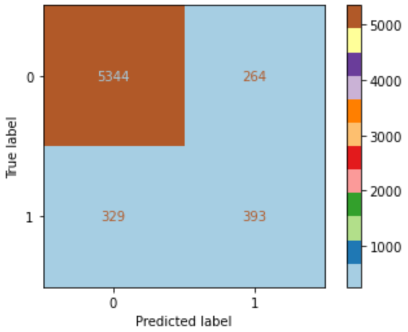
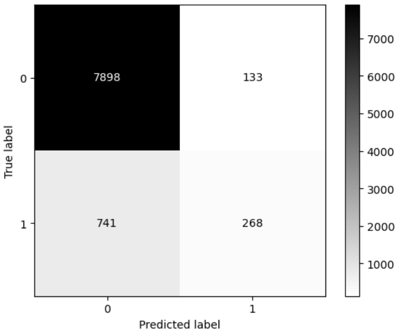
SL	**Observed Parameter**	**Result on Dataset 1 (All 17 columns considered with 30k+ rows)**	**Result on Dataset 2 (8 columns Considered with 45k+ rows)**
01	Accuracy	88.70%	89.70%
02	Cross Validation Score	89.27% 88.56% 88.81% 89.20% 88.84%	89.38% 92.63% 90.73% 86.98% 88.08%
03	Train-Test Accuracy	Train Set: 88.70% Test Set: 88.61%	Train Set: 89.70% Test Set: 89.20%

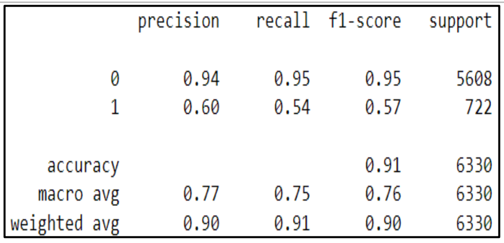
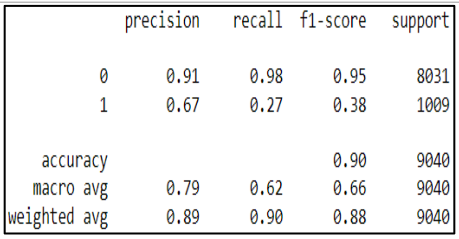
04	Confusion Matrix		
05	Report		

```
              precision    recall  f1-score   support

           0       0.91      0.97      0.94      5608
           1       0.51      0.27      0.35       722

    accuracy                           0.89      6330
   macro avg       0.71      0.62      0.65      6330
weighted avg       0.87      0.89      0.87      6330
```

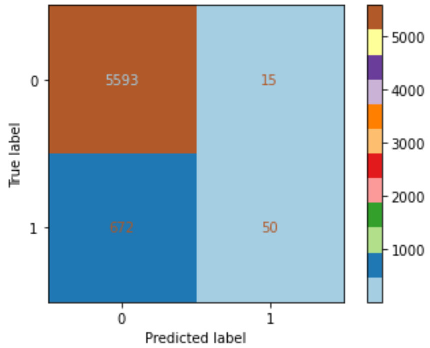
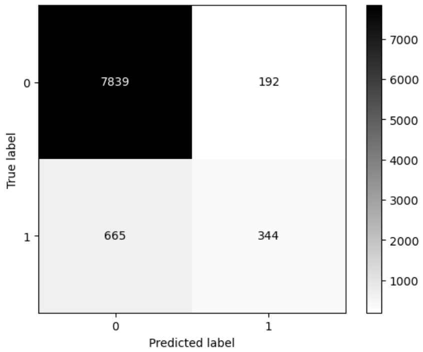
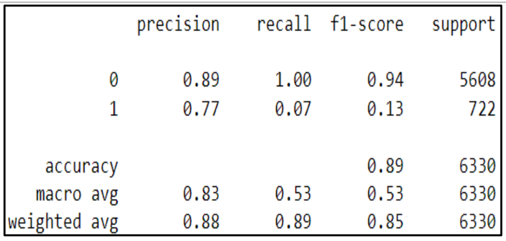
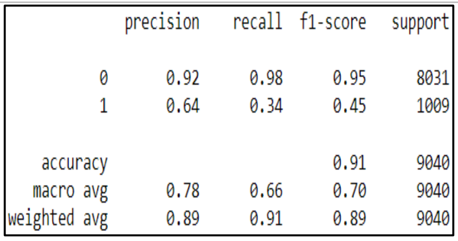
```
              precision    recall  f1-score   support

           0       0.92      0.97      0.94      8031
           1       0.58      0.28      0.38      1009

    accuracy                           0.90      9040
   macro avg       0.75      0.63      0.66      9040
weighted avg       0.88      0.90      0.88      9040
```

5.2. Decision Tree:

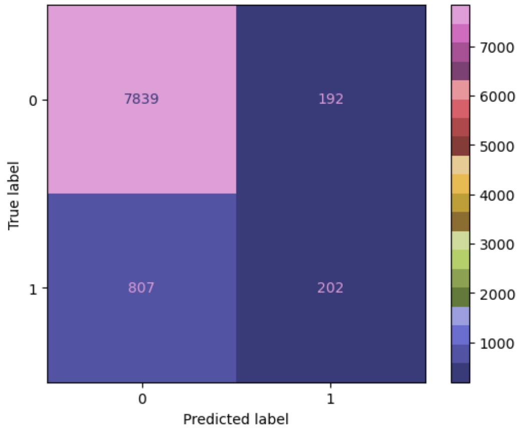
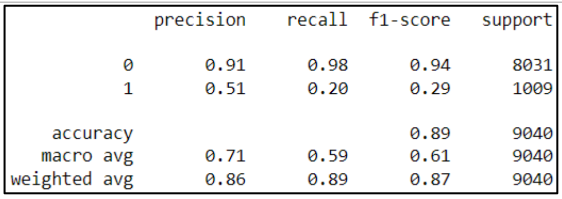
SL	Observed Parameter	Result on Dataset 1 (All 17 columns considered with 30k+ rows)	Result on Dataset 2 (8 columns Considered with 30k+ rows)
01	Accuracy	90.63%	90.33%
02	Cross Validation Score	91.05% 90.44% 90.42% 91.16% 90.55%	88.27% 88.66% 89.91% 87.92% 85.81%
03	Train-Test Accuracy	Train Set: 90.63% Test Set: 91.40%	Train Set: 90.33% Test Set: 90.04%
04	Confusion Matrix		

05	Report	 precision recall f1-score support 0 0.94 0.95 0.95 5608 1 0.60 0.54 0.57 722 accuracy 0.91 6330 macro avg 0.77 0.75 0.76 6330 weighted avg 0.90 0.91 0.90 6330	 precision recall f1-score support 0 0.91 0.98 0.95 8031 1 0.67 0.27 0.38 1009 accuracy 0.90 9040 macro avg 0.79 0.62 0.66 9040 weighted avg 0.89 0.90 0.88 9040

5.3. Random Forest:

SL	Observed Parameter	Result on Dataset 1 (All 17 columns considered with 30k+ rows)	Result on Dataset 2 (8 columns Considered with 30k+ rows)
01	Accuracy	89.15%	90.51%
02	Cross Validation Score	88.84% 88.59% 88.79% 88.78% 88.87%	88.70% 88.10% 89.29% 84.93% 86.86%
03	Train-Test Accuracy	Train Set: 89.15% Test Set: 89.17%	Train Set: 90.51% Test Set: 90.11%
04	Confusion Matrix		
05	Report	 precision recall f1-score support 0 0.89 1.00 0.94 5608 1 0.77 0.07 0.13 722 accuracy 0.89 6330 macro avg 0.83 0.53 0.53 6330 weighted avg 0.88 0.89 0.85 6330	 precision recall f1-score support 0 0.92 0.98 0.95 8031 1 0.64 0.34 0.45 1009 accuracy 0.91 9040 macro avg 0.78 0.66 0.70 9040 weighted avg 0.89 0.91 0.89 9040

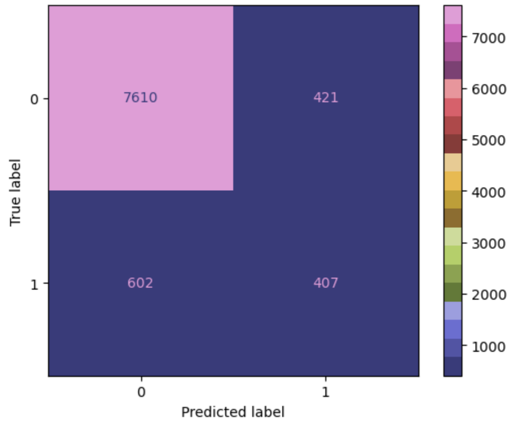
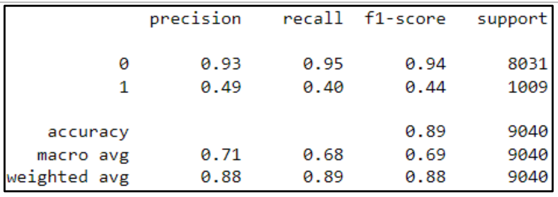
5.3. KNN on new Dataset:

SL	Observed Parameter	Result
01	Accuracy	88.94%
02	Cross Validation Score	88.30% 88.50% 87.36% 87.82% 86.20%
03	Train-Test Accuracy	Train Set: 88.94% Test Set: 89.98%
04	Confusion Matrix	
05	Report	

```
              precision    recall  f1-score   support

           0       0.91      0.98      0.94      8031
           1       0.51      0.20      0.29      1009

    accuracy                           0.89      9040
   macro avg       0.71      0.59      0.61      9040
weighted avg       0.86      0.89      0.87      9040
```

5.3. Naïve Bayes on new Dataset:

SL	Observed Parameter	Result
01	Accuracy	88.68%

02	Cross Validation Score	91.68% 90.37% 90.73% 88.56% 89.69%
03	Train-Test Accuracy	Train Set: 88.68% Test Set: 88.14%
04	Confusion Matrix	
05	Report	

```
              precision    recall  f1-score   support

           0       0.93      0.95      0.94      8031
           1       0.49      0.40      0.44      1009

    accuracy                           0.89      9040
   macro avg       0.71      0.68      0.69      9040
weighted avg       0.88      0.89      0.88      9040
```

5.4. Accuracy Graph for different Algorithms used:

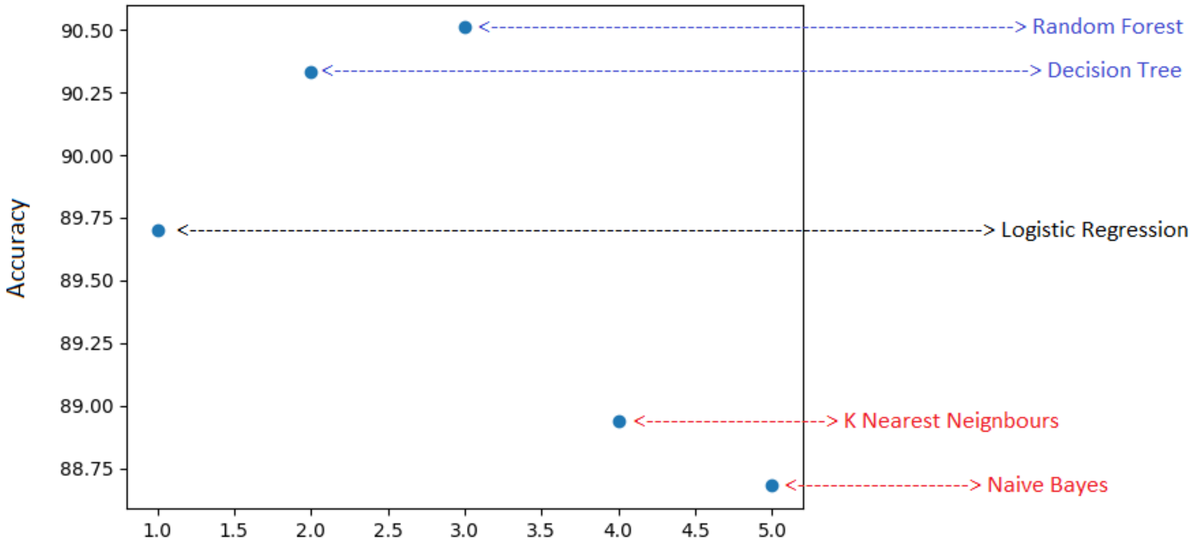


Fig 5.4.(a) Accuracy study of different ML Models

5.5 Test vs Train Accuracy:

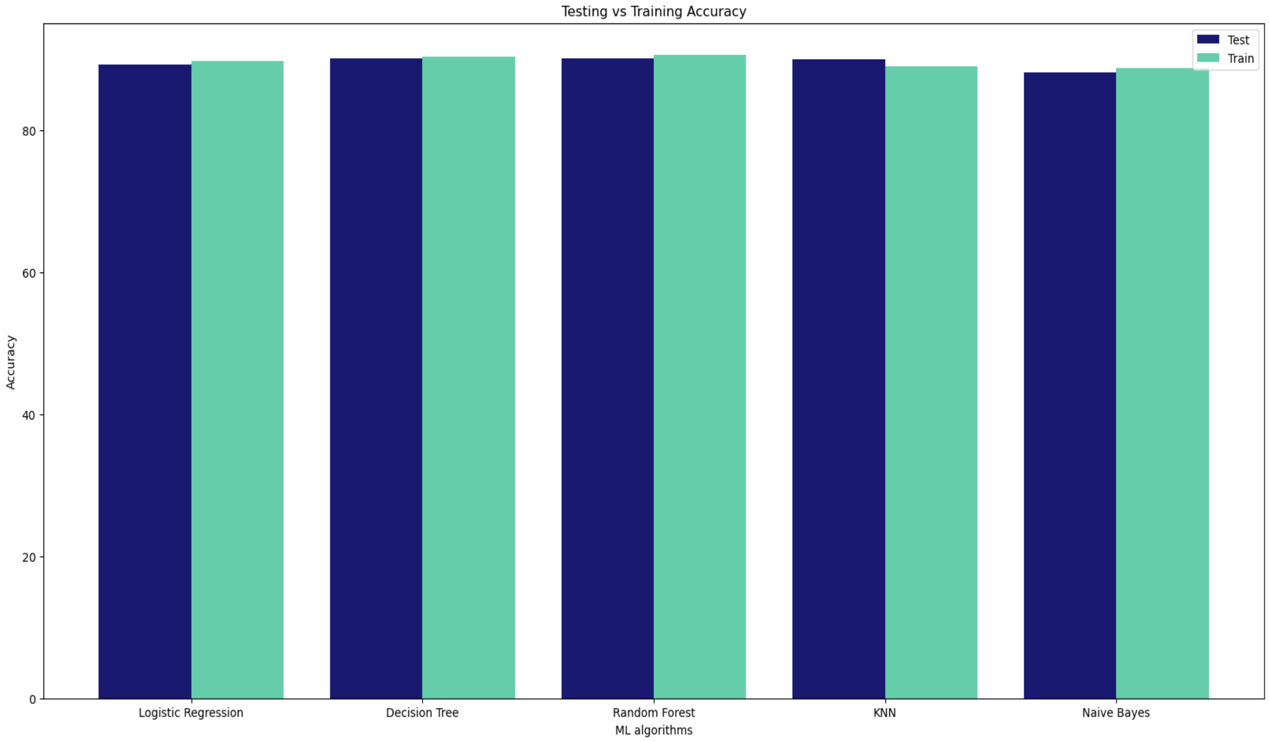


Fig 5.5.(a) Accuracy study of different ML Models wrt test and train dataset

5.6 Frontend Integration using Python Flask

Python Flask is a lightweight and flexible web framework that allows developers to build web applications quickly and efficiently. One of its key features is its seamless integration with frontend technologies, making it a popular choice for developing full-stack web applications.

Flask provides a robust set of tools and features for creating RESTful APIs and serving dynamic web pages. It follows the Model-View-Controller (MVC) architectural pattern, allowing developers to separate concerns and maintain a clean codebase. With Flask, developers can easily integrate the frontend and backend components of their application, enabling a smooth user experience.

In a typical Flask application, the frontend files such as HTML, CSS, and JavaScript are organized within a static or templates folder. The static folder is used to store static assets like CSS and JavaScript files, while the templates folder is used to store the templates. Flask's built-in templating engine allows developers to write reusable templates with placeholders that can be populated with data from the backend.

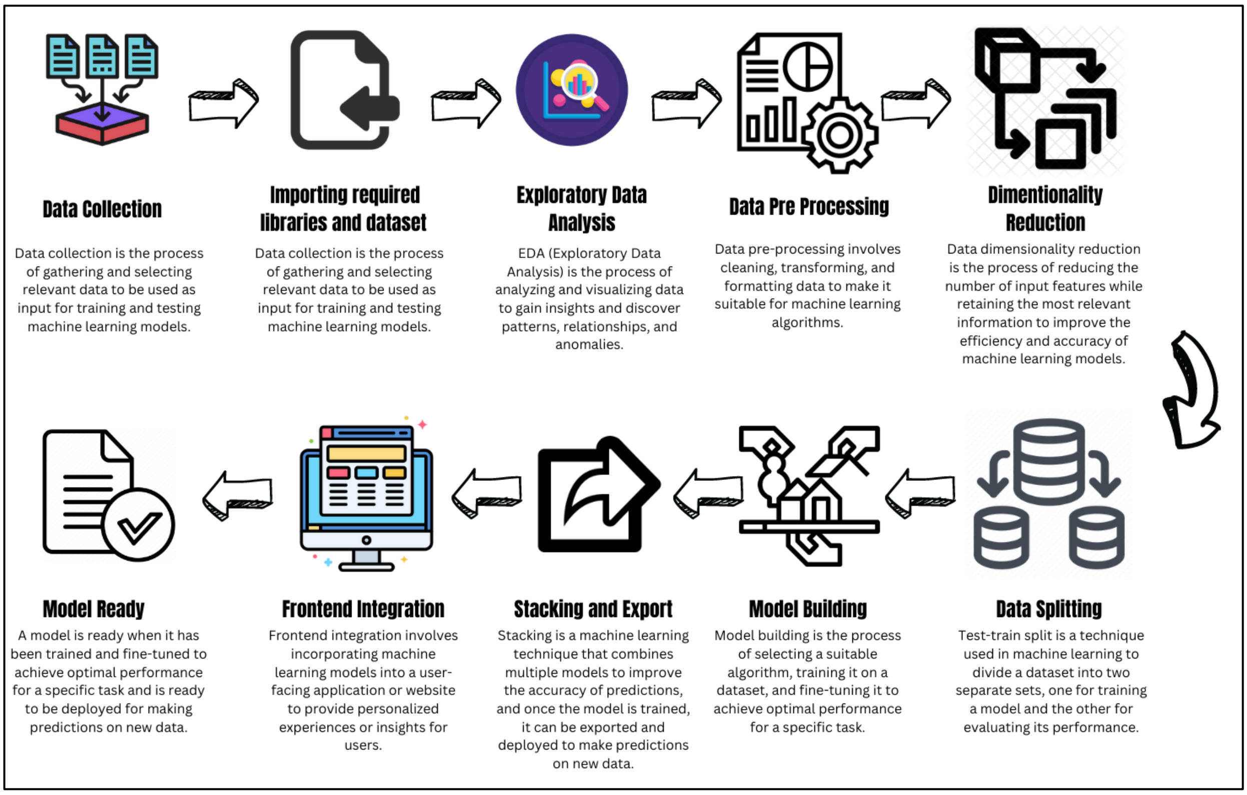
When a user makes a request to a Flask application, the server-side code processes the request and fetches any required data from a database or external APIs. This data is then passed to the appropriate template, which renders the HTML dynamically. The rendered HTML is then sent back to the user's browser, where it is displayed as a fully rendered webpage.

A number of frontend frameworks can communicate with the Flask backend using RESTful APIs provided by Flask. This decoupled architecture enables developers to create powerful and interactive user interfaces while leveraging the capabilities of Flask for backend processing.

Flask also provides extensions and plugins that further simplify the integration with frontend technologies. These extensions enhance Flask's capabilities and facilitate seamless communication between the frontend and backend components.

In conclusion, Python Flask is an excellent choice for developing web applications with a strong frontend-backend integration. Its flexibility, simplicity, and support for templating engines make it easy to create dynamic web pages. By leveraging Flask's routing mechanism and integrating with frontend frameworks, developers can build robust and feature-rich applications that provide a delightful user experience.

5.7 Updated New Workflow Model:



Data Collection

Data collection is the process of gathering and selecting relevant data to be used as input for training and testing machine learning models.

Importing required libraries and dataset

Data collection is the process of gathering and selecting relevant data to be used as input for training and testing machine learning models.

Exploratory Data Analysis

EDA (Exploratory Data Analysis) is the process of analyzing and visualizing data to gain insights and discover patterns, relationships, and anomalies.

Data Pre Processing

Data pre-processing involves cleaning, transforming, and formatting data to make it suitable for machine learning algorithms.

Dimentionality Reduction

Data dimensionality reduction is the process of reducing the number of input features while retaining the most relevant information to improve the efficiency and accuracy of machine learning models.

Data Splitting

Test-train split is a technique used in machine learning to divide a dataset into two separate sets, one for training a model and the other for evaluating its performance.

Model Building

Model building is the process of selecting a suitable algorithm, training it on a dataset, and fine-tuning it to achieve optimal performance for a specific task.

Stacking and Export

Stacking is a machine learning technique that combines multiple models to improve the accuracy of predictions, and once the model is trained, it can be exported and deployed to make predictions on new data.

Frontend Integration

Frontend integration involves incorporating machine learning models into a user-facing application or website to provide personalized experiences or insights for users.

Model Ready

A model is ready when it has been trained and fine-tuned to achieve optimal performance for a specific task and is ready to be deployed for making predictions on new data.

Fig 5.7.(a) Updated Workflow Model.

5.8 Stacking:

Stacking, also known as stacked generalization, is a powerful ensemble learning technique in machine learning. It involves combining the predictions of multiple models, called base models or learners, to create a meta-model that provides a more accurate and robust prediction.

The idea behind stacking is to train several diverse base models on the same dataset. Each base model learns different patterns and captures various aspects of the data. These base models can be of different types, such as decision trees, support vector machines, or neural networks, allowing for a diverse set of predictions.

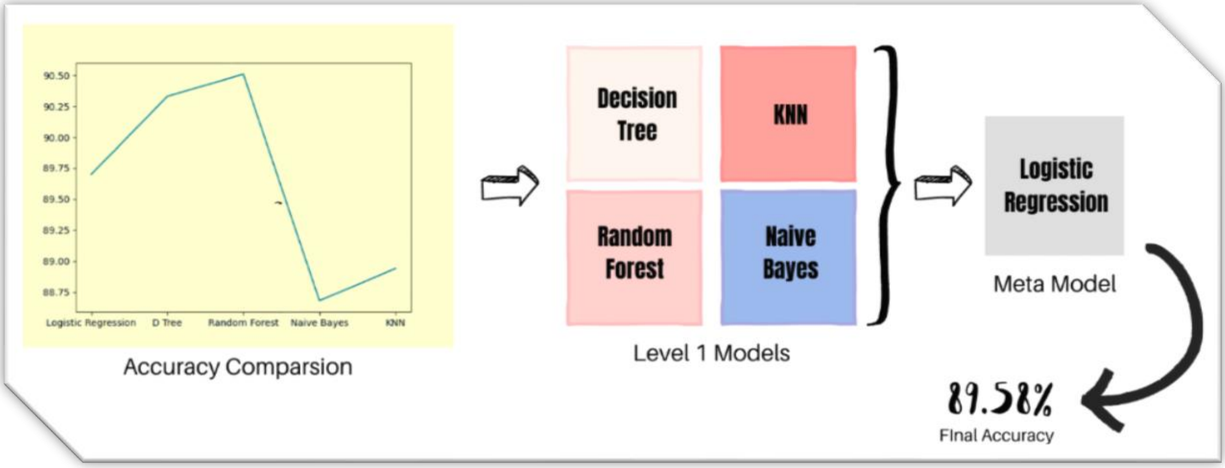


Fig 5.8.(a) Stacking Models used with combined Accuracy Score

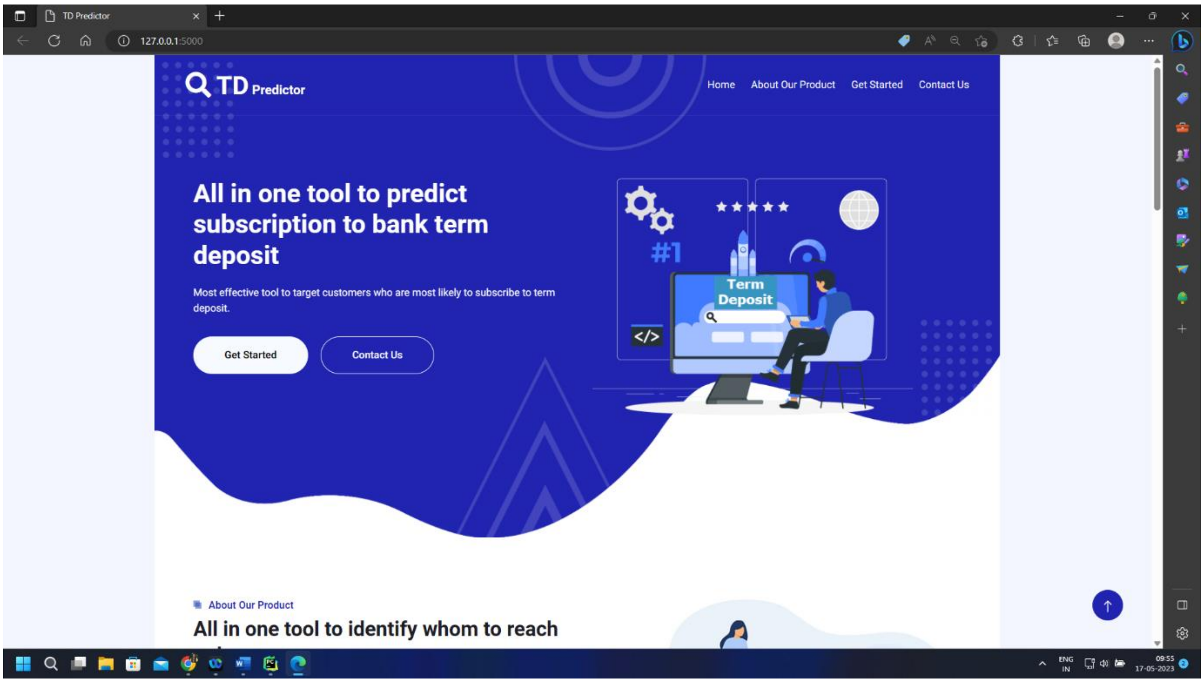
In the stacking process, the dataset is divided into two or more parts. One part is used to train the base models, and the other part, known as the holdout set, is used to create predictions from the base models. The holdout set predictions are then used as input features for a meta-model, which is trained to make the final prediction. The meta-model learns to combine the predictions of the base models effectively.

The key advantage of stacking is its ability to leverage the strengths of different models. By combining the predictions of multiple models, stacking can overcome individual model biases and improve prediction accuracy. It can also provide better generalization and robustness by reducing overfitting.

5.9 Frontend working of the project:

A frontend using HTML, CSS, Bootstrap, and JavaScript is prepared by combining these technologies to create a user interface that is visually appealing, responsive, and interactive. HTML provides the structure of the webpage, defining the elements and their hierarchy. CSS is used to style and layout the elements, controlling their appearance and positioning. Bootstrap provides pre-built components and responsive design features. JavaScript adds interactivity to the frontend. Together, these technologies enable the creation of engaging and functional web interfaces.

The working of our website is as follows:



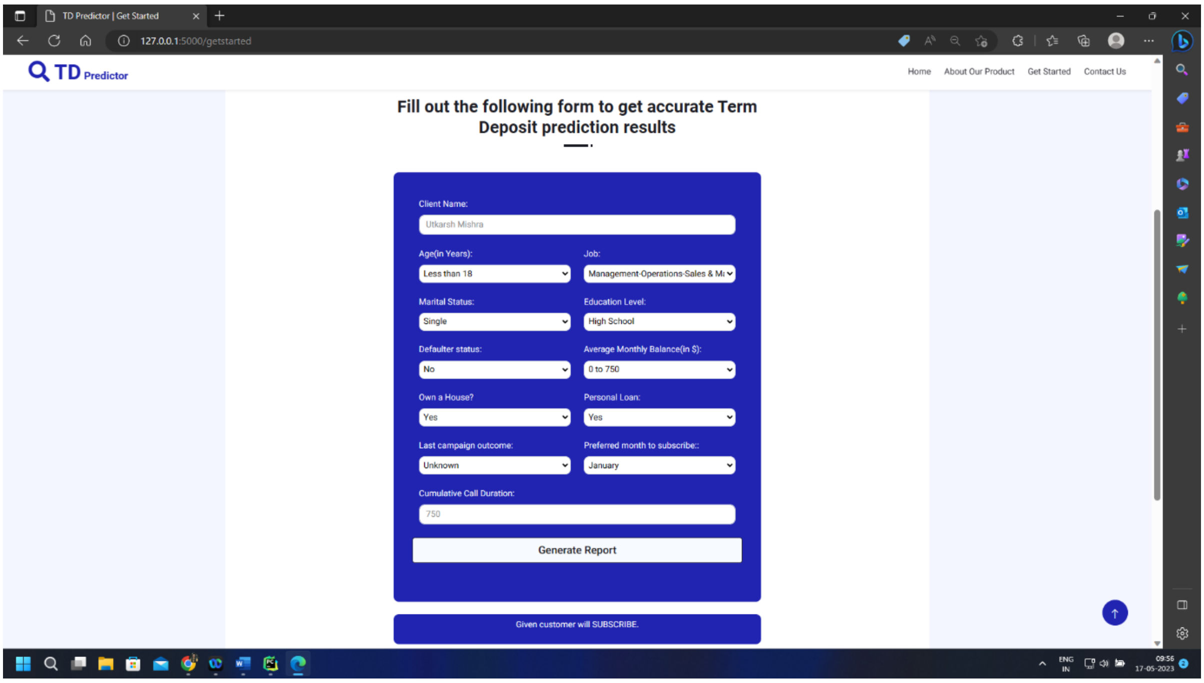
All in one tool to predict subscription to bank term deposit

Most effective tool to target customers who are most likely to subscribe to term deposit.

About Our Product

All in one tool to identify whom to reach

Fig 5.9.(a) Landing page of our website



Fill out the following form to get accurate Term Deposit prediction results

Client Name: Utkarsh Mishra

Age(in Years): Less than 18

Job: Management-Operations-Sales & M

Marital Status: Single

Education Level: High School

Defaulter status: No

Average Monthly Balance(in $): 0 to 750

Own a House? Yes

Personal Loan: Yes

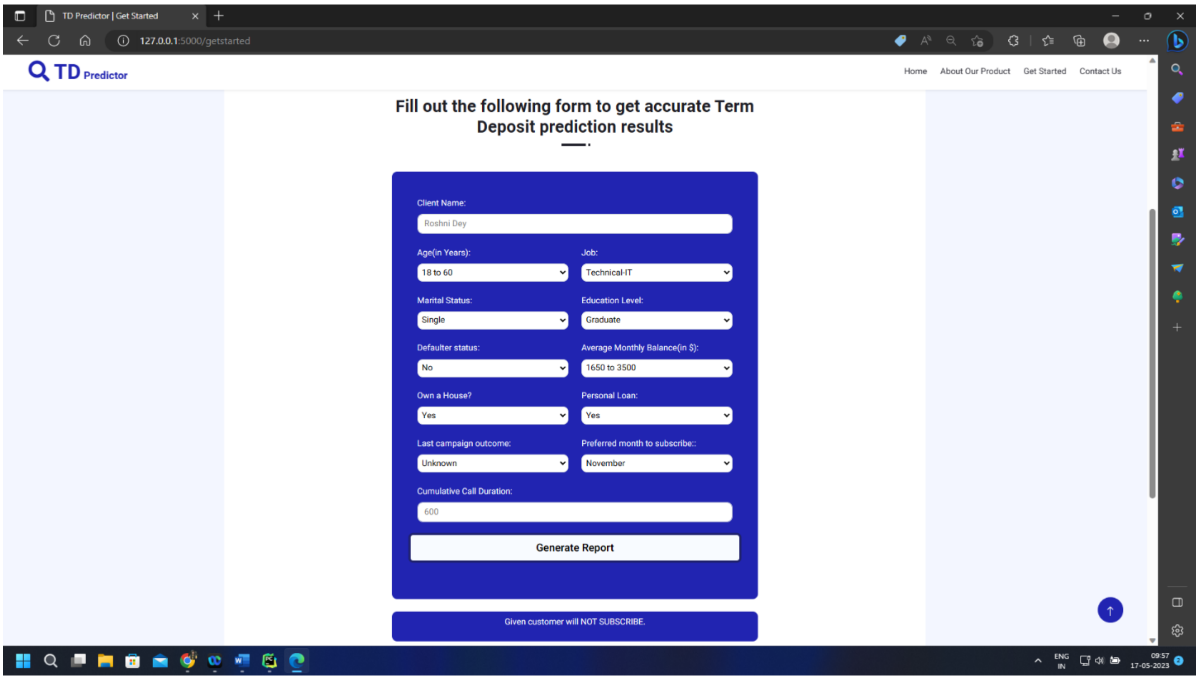
Last campaign outcome: Unknown

Preferred month to subscribe:: January

Cumulative Call Duration: 750

Given customer will SUBSCRIBE.

Fig 5.9.(b) Test case 1 to check working of our model



Fill out the following form to get accurate Term Deposit prediction results

Client Name: Roshni Dey

Age(in Years): 18 to 60

Job: Technical-IT

Marital Status: Single

Education Level: Graduate

Defaulter status: No

Average Monthly Balance(in $): 1650 to 3500

Own a House? Yes

Personal Loan: Yes

Last campaign outcome: Unknown

Preferred month to subscribe:: November

Cumulative Call Duration: 600

Generate Report

Given customer will NOT SUBSCRIBE.

Fig 5.9.(c) Test case 2 to check working of our model

Python Pickle is commonly used to export and serialize machine learning models in Python. With Pickle, trained models can be saved to disk and easily loaded back into memory for later use. It allows models to retain their state, including parameters, weights, and other attributes, enabling seamless deployment and sharing of machine learning models. Pickle supports various types of models, such as scikit-learn classifiers or TensorFlow models.

5.10 Testing on Real-Time Dataset

Testing machine learning algorithms based on real-time datasets offers several advantages that contribute to the improvement and effectiveness of the models. Here are some key advantages:

1. Real-world Performance: Real-time datasets reflect the dynamic and evolving nature of the problem domain. Testing machine learning algorithms on such datasets allows for evaluating their performance under realistic conditions. It helps identify how well the algorithms handle variations, outliers, and new patterns that may arise over time.

2. Timeliness: Real-time datasets enable the evaluation of machine learning algorithms in the context of current and up-to-date information. This is particularly valuable in domains where timely decision-making is crucial.

3. Model Generalization: Real-time datasets provide a broader and more diverse range of instances, allowing for better model generalization. Testing on real-time data helps assess how well the algorithms can generalize their learnings from historical data to new, unseen instances

4. Continuous Model Evaluation: Real-time datasets facilitate continuous evaluation and monitoring of machine learning models. By regularly testing models on real-time data, it becomes possible to identify any degradation in performance or concept drift.

6. Feedback Loop: Testing on real-time datasets establishes a feedback loop between the performance of the machine learning models and the quality of the data being collected. By analysing model performance on real-time data, insights can be gained into data quality issues, potential biases, or missing features.

We have gathered over 450 real-time data entries from various individuals. Subsequently, we applied exploratory data analysis (EDA) and data pre-processing techniques to transform the dataset. Finally, we utilized the processed dataset to evaluate our model, leading to the following observations:

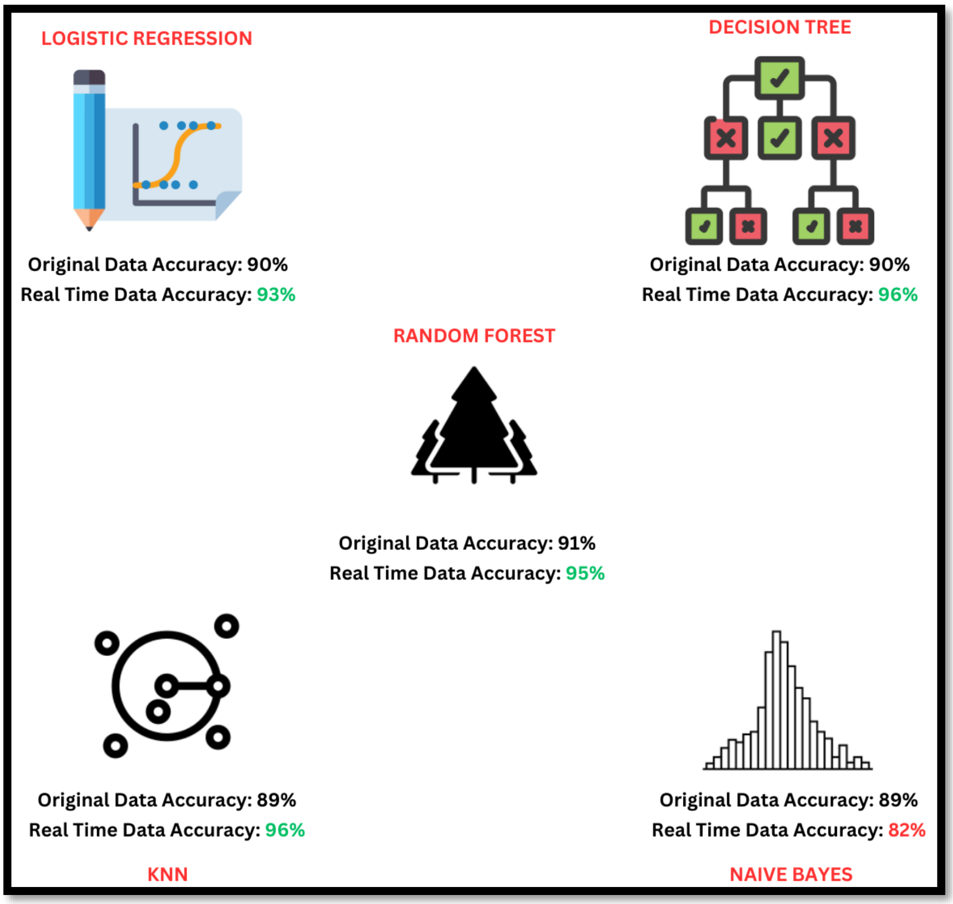

LOGISTIC REGRESSION
Original Data Accuracy: 90%
Real Time Data Accuracy: 93%
DECISION TREE
Original Data Accuracy: 90%
Real Time Data Accuracy: 96%
RANDOM FOREST
Original Data Accuracy: 91%
Real Time Data Accuracy: 95%
Original Data Accuracy: 89%
Real Time Data Accuracy: 96%
KNN
Original Data Accuracy: 89%
Real Time Data Accuracy: 82%
NAIVE BAYES

Fig 5.10.(c) Real Time Dataset Result

6. CONCLUSION AND FUTURE SCOPE

Machine learning can be used in banking to generate actionable insights from massive databases collected by banks. Machine learning models may assist banks handle and analyze data, whether it's a history of transactions, chat logs with bank employees, or corporate documentation, to gain a better knowledge of their customers and internal procedures. Banks have two business lines: borrowing and lending. A bank is a safe place for individuals and businesses to save their money. In exchange for depositing funds with banks, banks offer interest based on the account. Banks encourage individuals to invest in Fixed Deposits and Recurring Deposits in addition to Savings and Current Accounts by offering a greater rate of interest. This generates funds for the bank. The bank technically 'borrows' money from you. The bank undertakes lending operations using cash accumulated in various accounts. Most banks provide a diverse range of loans to their customers, including Home Loans, Business Loans, Personal Loans, Car Loans, and so on. They charge interest to those who take out such loans.

In this entire money exchange process and banks undergoing digitalization, our work will assist banks in determining the clients who are most likely to opt for a term deposit plan, allowing banking institutions to save a significant amount of spending. Along with saving money on investments, it will also save them a lot of time hunting for people who are likely to choose a term deposit.

The future scope of our project include:

1. Model Improvement: There is always room for enhancing the performance and accuracy of machine learning models. This can be achieved by exploring advanced algorithms, optimizing hyperparameters, or incorporating new features and data sources. Continuous model improvement is essential to stay ahead in the rapidly evolving field of machine learning.

2. Scalability and Deployment: Scaling up a machine learning project to handle larger datasets or higher user loads is crucial. Future work may involve implementing distributed computing techniques, such as parallel processing or cloud-based solutions, to improve scalability. Additionally, deploying the model to different platforms, including mobile devices or edge computing devices, can expand the project's reach and impact.

3. Continuous Learning and Adaptation: Machine learning models can benefit from continuous learning and adaptation to changing data distributions or evolving user preferences. Implementing

techniques like online learning or reinforcement learning can enable models to adapt and improve over time, making them more robust and accurate.

4. Integration with Other Technologies: Machine learning can be integrated with other emerging technologies to create powerful and innovative solutions. Future work may involve combining machine learning with natural language processing, computer vision, robotics, or Internet of Things (IoT) to build advanced applications that have a broader impact.

7. REFERENCES

[1] S. Manlangit, S. Azam, B. Shanmugam and A. karim,” Novel Machine Learning Approach for Analyzing Anonymous Credit Card Fraud Patterns”, International Journal of Electronic Commerce Studies 10.2 (2019): 175-202

[2] M. Sergio, L. Raul and C. Paulo,” Using data mining for bank direct marketing: an application of the CRISP-DM methodology”, RepositoriUM, 2011

[3] C. S. T. Koumetio, W. Cherif and S. Hassan,” Optimizing the prediction of telemarketing target calls by a classification technique,” 2018 6th International Conference on Wireless Networks and Mobile Communications (WINCOM), 2018

[4] J. Asare-Frempong and M. Jayabalan,” Predicting customer response to bank direct telemarketing campaign,” 2017 International Conference on Engineering Technology and Technopreneurship (ICE2T), 2017

[5] Rony MA, Hassan MM, Ahmed E, Karim A, Azam S, Reza DA. Identifying Long-Term Deposit Customers: A Machine Learning Approach. In2021 2nd International Informatics and Software Engineering Conference (IISEC) 2021 Dec 16 (pp. 1-6). IEEE.

[6] Bisong E. Matplotlib and seaborn. InBuilding machine learning and deep learning models on google cloud platform 2019 (pp. 151-165). Apress, Berkeley, CA.

[7] Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V, Vanderplas J. Scikit-learn: Machine learning in Python. the Journal of machine Learning research. 2011 Nov 1;12:2825-30.

[8] A. Gupta, A. Raghav and S. Srivastava,” Comparative Study of Machine Learning Algorithms for Portuguese Bank Data,” 2021 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS), 2021.

[9] G. Sell and D. Garcia-Romero,” Diarization resegmentation in the factor analysis subspace,” 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2015, pp. 4794-4798

[10] A. Sarmento, K. Yeo, S. Azam, A. Karim, A. Al Mamun and B. Shanmugam,” Applying Big Data Analytics in DDos Forensics: Challenges and Opportunities”, Cybersecurity, Privacy and Freedom Protection in the Connected World, pp. 235-252, 2021. 42

[11] M. M. Hassan, M. A. Mamun Billah, M. M. Rahman, S. Zaman, M. M. Hasan Shakil and J. H. Angon," Early Predictive Analytics in Healthcare for Diabetes Prediction Using Machine Learning Approach," 2021 12th International Conference on Computing Communication and Networking Technologies (ICCCNT), 2021.

[12] Song YY, Ying LU. Decision tree methods: applications for classification and prediction. Shanghai archives of psychiatry. 2015 Apr 4;27(2):130.

[13] Hossin M, Sulaiman MN. A review on evaluation metrics for data classification evaluations. International journal of data mining & knowledge management process. 2015 Mar 1;5(2):1