Bank Marketing Campaign

Aditi Dadariya

July 30, 2023

Declaration

I, Aditi Dadariya, Data Science Intern with Data Glacier, confirm that this is my own work and figures, artworks, and illustrations in this report are original and have not been taken from any other person's work, except where the works of others have been explicitly acknowledged, quoted, and referenced. I understand that if failing to do so will be considered a case of plagiarism. Plagiarism is a form of academic misconduct and will be penalised accordingly.

Aditi Dadariya
July 30, 2023

Abstract

The bank marketing campaign dataset presents a supervised classification problem aimed at predicting whether customers will subscribe to a bank's term deposit product. The dataset contains diverse attributes, including age, job, marital status, education, and previous interactions with the bank. Preprocessing techniques such as missing value handling, encoding of categorical variables, and outlier removal were applied to prepare the data for machine learning models.

Various machine learning algorithms, including linear models, ensemble models, and boosting models, were explored and evaluated to identify the most effective solution. The dataset was split into training and testing sets using cross-validation techniques to ensure reliable performance evaluation.

Results showed that the XGBoost model performed exceptionally well, achieving accuracy levels above 90% through out the research. Furthermore, the model with the best-tuned hyperparameters demonstrated outstanding accuracy for predicting customer responses.

The final model has been effectively deployed to optimize the bank's marketing efforts, allowing the bank to focus on potential customers most likely to subscribe to the term deposit product. This research opens avenues for further exploration of deep learning and hybrid models to enhance predictive accuracy and streamline marketing strategies in the banking industry.

Acknowledgements

I am truly thankful to Data Glacier for providing me with the opportunity to present my research on the Bank Marketing Campaign. I am grateful for this platform that allowed me to showcase my work and share valuable insights with a wider audience.

Contents

List of Figures

List of Abbreviations

LR	Linear Regression Model
LDA	Linear Discriminant Analysis Model
RFC	Random Forest Classifier Model
ADAB	ADABoost Model
XGB	XGBoost Model
HGBC	Hist Gradient Boosting Classifier Model
IQR	Interquantile Range
SMOTE	Synthetic Minority Oversampling Technique

Chapter 1

Introduction

1.1 Background and motivation

In the realm of decision-making, direct marketing and telemarketing play a significant role as customers rely on reviews to assess long-term benefits. While technology influences response behavior, customers seek advantageous options for loans and term deposits. Simultaneously, banks aim to enhance customer interactions for profitable outcomes. This study evaluates multiple machine learning models to predict the term deposit product for ABC Bank.

1.2 Research Problem

ABC Bank aims to introduce its term deposit product to customers. To ensure a successful launch, they seek to develop a model that can predict whether a customer is likely to purchase the product based on their past interactions with the bank or other financial institutions.

1.3 Aims and objectives

The dataset pertains to direct marketing campaigns conducted by a Portuguese banking institution. The bank aims to leverage a machine learning model to identify and prioritize customers with a higher likelihood of purchasing the product. This will enable the marketing channels, such as tele marketing, SMS/email marketing, to concentrate their efforts on these targeted customers.

1.4 Solution approach

This research paper focuses on a binary classification problem. It evaluates the performance and results of three different model methods: Linear Models, Ensemble Models, and Boosting Models. The Linear Models examined in this study include Logistics Regression and Linear Discriminant Analysis. The Ensemble Models evaluated are Random Forest Classifier and AdaBoost. Lastly, the Boosting Models analyzed are XGBoost and HistGradientBoosting. These various models are compared to determine their effectiveness in solving the binary classification task.

The approach to the solution for predicting the term deposit is divided into different steps as described below.

- Two files have been created as libraries to manage variable and function declarations. The "config.yaml" file handles variable declarations, while the "utilities.py" file handles function declarations. Data modeling is performed in the "main.py" file. The required dependencies are listed in the "requirements.txt" file. Model deployment is accomplished using the "app.py", "model.pkl", "request.py", "style.css", and "index.html" files.
- Various techniques, including Label encoding, OneHot Encoding, Interquantile Range (IQR), ZScore, Simple Imputer, and Quantile Transformer, have been applied to analyze the bank data and transform it into a machine learning-readable format.
- To address the issue of imbalanced data, both SMOTE (Synthetic Minority Oversampling Technique) and RandomUnderSampler techniques were employed. These methods help in handling the oversampling and undersampling of data, ensuring a balanced dataset for better model performance.
- The data was divided into training and testing datasets through two methods: Hold-Out and Cross-Validation. For hyperparameter tuning, StratifiedKFold and RepeatedStratifiedKFold techniques were employed to ensure balanced class distributions in the folds. This approach enhances the model's generalization and robustness during the evaluation process.
- The dataset was initially evaluated using the baseline model of Support Vector Machine Classifier.

1.5 Summary of contributions and achievements

- The first and foremost important challenge with the dataset was to null values as the data showed there was no null value but it contained "unknown" values, for which the unknown value were converted to null value and then substituted using Simple Imputer technique.
- Secondly, as the code was growing, it was essential to maintain the coding standards, for which the code was divided into multiple files sequentially.

1.6 Organization of the report

- Chapter 2 describes the past work around the term deposit predictions. Also focuses on the various approaches carried out to predict term deposit.
- Chapter 3 outlines the methodology to achieve the aim of finding the best solution for term deposit prediction.
- Chapter 4 discusses the results produced by different models.
- Chapter 5 walks through the analysis, findings and limitations.
- Chapter 6 outlines the conclusions and future work.
- Chapter 7 shows the reflection of my work for this research paper.

Chapter 2

Literature Review

2.1 Literature and Review

Direct marketers are increasingly turning to more sophisticated techniques to model response behavior, as the costs of campaigns have risen and response rates have declined. However, the data used for this modeling is often imbalanced, with far more non-respondents than respondents. The paper in [3] compares different ensemble models based on bagging and boosting using bank direct marketing data. The results showed that the bagged neural network model performed the best, followed by the gradient boosting classifier and the logistic regression model. Although gradient boosting did not outperform the bagged classification models in this study, it is still an efficient approach to handle imbalanced classes. The performance of different ensemble classifiers depends on several factors, including the sample size of the training data, the degree of class imbalance, and the type of classification problem.

A deep learning model was developed to predict customer preferences for loans or deposits in the banking industry [5]. The model analyzed various parameters from a dataset obtained from Kaggle. Preprocessing techniques were applied, and three machine learning algorithms were evaluated: linear regression, K-nearest neighbor, and random forest. The random forest algorithm achieved the highest accuracy of over 95%, making it a promising choice for call-list filtering. The RF algorithm outperformed others in accuracy, precision, TPR, TNR, and F1 score. The author suggested that this algorithm could be integrated into websites or applications in the future.

In today's competitive banking industry, cutting-edge data mining and machine learning techniques have replaced traditional methods [4]. Telemarketing has emerged as a cost-effective strategy, but banks are constantly seeking efficient ways to target interested customers and boost profits. This research presents a data mining solution using an ensemble classifier based on hybrid machine learning models to predict customer responses in telemarketing campaigns. Through a comprehensive Exploratory Data Analysis (EDA), valuable insights are gained about customer attributes that lead to higher subscription rates. The proposed ensemble classifier effectively addresses the class imbalance in the dataset, achieving an impressive 95.6% accuracy in predicting customer responses.

2.2 Summary

In this chapter, firstly the customer perspective of direct marketing has been discussed where gradient boosting was an efficient approach to handle imbalanced classes [3]. Hereinafter, the customer preferences was discussed for which various models has been developed [5],

out of which random forest achieved an outstanding accuracy. Lastly, Telemarketing was discussed to increase the bank customers and profits [4], where ensemble classifier addressing the imbalance class has showed better performance.

Chapter 3

Methodology

This paper aims to find the optimal solution for predicting term deposit subscription among bank customers. The approach involves dividing the implementation into several stages: importing required libraries and reading the data, data pre-processing, data visualization, feature selection, building baseline machine learning models with train-test split, constructing basic models without parameters, finding the best hyperparameters, tuning the models with the selected hyperparameters, and finally discussing the results.

All the coding to develop these models has been carried out in Python using the Spyder tool. All files related to this project are available on GitHub [1]. Additionally, a web application has been developed using Flask to demonstrate the prediction results interactively.

The project consists of three Python files: "config.yaml", "utility.py", and "main.py". In the "config.yaml" file, you can find the configurations, including variable declarations and their corresponding values. The "utility.py" file contains function declarations for data pre-processing, data visualization, and data modeling of the dataset. Lastly, the "main.py" file encompasses all the data modeling processes.

Several other files have been developed to handle model deployment. The "app.py" file contains the Flask code to create the API. The "model.pkl" is a pickle file containing details of the best-performing model. The "request.py" file consists of a request JSON. The "requirements.txt" file contains the prerequisites of the required packages. The "textfile.txt" serves as the log file. The "style.css" file defines the styling and format for the web application. Lastly, the "index.html" file contains the labels and text required for the web application.

3.1 Dataset Description

Bank Marketing Campaign dataset is focused on a supervised classification problem, aiming to predict whether a customer will purchase the bank's term deposit product.

The dataset consists of 41188 records and 21 variables. The variables are 'age', 'job', 'marital', 'education', 'default', 'housing', 'loan', 'contact', 'month', 'day_of_week', 'duration', 'campaign', 'pdays', 'previous', 'poutcome', 'emp.var.rate', 'cons.price.idx', 'cons.conf.idx', 'euribor3m', 'nr.employed', 'y'. The variable 'y' is the target variable, while the rest of the variables are considered as features. The variables 'age', ''duration', 'campaign', 'pdays' and 'previous' are of int64 data type. The variables 'emp.var.rate', 'cons.price.idx', 'cons.conf.idx', 'euribor3m' and 'nr.employed' are of float64 data type. The variables 'job', 'marital', 'education', 'default', 'housing', 'loan', 'contact', 'month', 'day_of_week', 'poutcome' and 'y' are of object data type. Please refer Figure 3.1 below

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 41188 entries, 0 to 41187
Data columns (total 21 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   age             41188 non-null  int64
 1   job             41188 non-null  object
 2   marital         41188 non-null  object
 3   education       41188 non-null  object
 4   default         41188 non-null  object
 5   housing         41188 non-null  object
 6   loan            41188 non-null  object
 7   contact         41188 non-null  object
 8   month           41188 non-null  object
 9   day_of_week     41188 non-null  object
 10  duration        41188 non-null  int64
 11  campaign        41188 non-null  int64
 12  pdays           41188 non-null  int64
 13  previous        41188 non-null  int64
 14  poutcome        41188 non-null  object
 15  emp.var.rate    41188 non-null  float64
 16  cons.price.idx  41188 non-null  float64
 17  cons.conf.idx   41188 non-null  float64
 18  euribor3m       41188 non-null  float64
 19  nr.employed     41188 non-null  float64
 20  y               41188 non-null  object
dtypes: float64(5), int64(5), object(11)
memory usage: 6.6+ MB
None
```

Figure 3.1: Dataset Info

Age varies between 17 to 98. Duration varies between 0 to 4918. Campaign varies between 1 to 56. Pdays varies between 0 to 999. Previous varies between 1 to 7. Emp.var.rate varies between -3.4 to 1.4. Cons.price.idx varies between 92.2 to 94.77. Cons.conf.idx varies to -50.8 to -26.9. Euribor3m varies between 0.63 to 5.04. Nr.employed varies between 4963.6 to 5228.1. Please refer Figure 3.2 below for these details.

	age	duration	campaign	pdays	previous	\
count	41188.00000	41188.000000	41188.000000	41188.000000	41188.000000	
mean	40.02406	258.285010	2.567593	962.475454	0.172963	
std	10.42125	259.279249	2.770014	186.910907	0.494901	
min	17.00000	0.000000	1.000000	0.000000	0.000000	
25%	32.00000	102.000000	1.000000	999.000000	0.000000	
50%	38.00000	180.000000	2.000000	999.000000	0.000000	
75%	47.00000	319.000000	3.000000	999.000000	0.000000	
max	98.00000	4918.000000	56.000000	999.000000	7.000000	

	emp.var.rate	cons.price.idx	cons.conf.idx	euribor3m	nr.employed
count	41188.000000	41188.000000	41188.000000	41188.000000	41188.000000
mean	0.081886	93.575664	-40.502600	3.621291	5167.035911
std	1.570960	0.578840	4.628198	1.734447	72.251528
min	-3.400000	92.201000	-50.800000	0.634000	4963.600000
25%	-1.800000	93.075000	-42.700000	1.344000	5099.100000
50%	1.100000	93.749000	-41.800000	4.857000	5191.000000
75%	1.400000	93.994000	-36.400000	4.961000	5228.100000
max	1.400000	94.767000	-26.900000	5.045000	5228.100000

Figure 3.2: Dataset description

3.2 Data Preprocessing

Dataset has been analysed in various aspects and transformed into machine learning format. Various plots are created to visualise and understand the data to perform necessary techniques

for correcting the data. Below are the steps followed, after which the cleaned data has been saved:

- Identification of missing values: The dataset does not contain any null (NaN) values. However, there are 'unknown' values present in 'job', 'marital', 'education', 'default', 'housing', 'loan' columns, which can be treated as missing values. The variable 'job' has 330 "unknown" values. The variable 'marital' has 80 "unknown" values. The variable 'education' has 1731 "unknown" values. The variable 'default' has 8597 "unknown" values. The variable 'housing' has 990 "unknown" values. The variable 'loan' has 990 "unknown" values. Please refer Figure 3.3. The "unknown" values in the dataset have been replaced with NaN values. Subsequently, the NaN values have been replaced with its most frequently used value by SimpleImputer method. The decision was made not to drop the missing values due to their substantial presence in the dataset. Removing these values could potentially impact the integrity of the training data.

```
Missing values in the dataset:
age               0
job               0
marital           0
education         0
default           0
housing           0
loan              0
contact           0
month             0
day_of_week       0
duration          0
campaign          0
pdays             0
previous          0
poutcome          0
emp.var.rate      0
cons.price.idx    0
cons.conf.idx     0
euribor3m         0
nr.employed       0
y                 0
dtype: int64
```

(a) Missing Values

```
Variables having "unknown" value ['job' 'marital' 'education' 'default' 'housing' 'loan']


Variables with "unknown" counts:
age                0
job              330
marital           80
education       1731
default         8597
housing          990
loan             990
contact            0
month              0
day_of_week        0
duration           0
campaign           0
pdays              0
previous           0
poutcome           0
emp.var.rate       0
cons.price.idx     0
cons.conf.idx      0
euribor3m          0
nr.employed        0
y                  0
dtype: int64
```

(b) Unknown Values

Figure 3.3: Missing values details

- Inconsistent datatypes: Once the "unknown" values were transformed into NaN values, the categorical variables ('job', 'marital', 'education', 'default', 'housing', 'loan') were

examined for potential mixed datatypes. However, it was confirmed that these variables do not exhibit multiple datatypes, as they are consistently of the same datatype throughout the dataset.

- Categorical distribution: The categorical variables 'job', 'marital', 'education', 'default', 'housing', 'loan', 'contact', 'month', 'day_of_week', 'poutcome' and 'y' were transformed into numeric representations to ensure data consistency. Binary categorical variables were encoded using Label Encoder, while Multi-categorical variables were encoded using One-Hot Encoder.

- Dimensionality Reduction of features: No dimensionality reduction technique was employed since the dataset comprises only a few features.

- Detecting and Correcting Outliers: A box plot was generated to visually examine the presence of outliers in the dataset. Please refer to Figures 3.4 for the plot. Upon analysis, it is evident that the variables contain outliers, as indicated by the data points beyond the whiskers of the box plot.

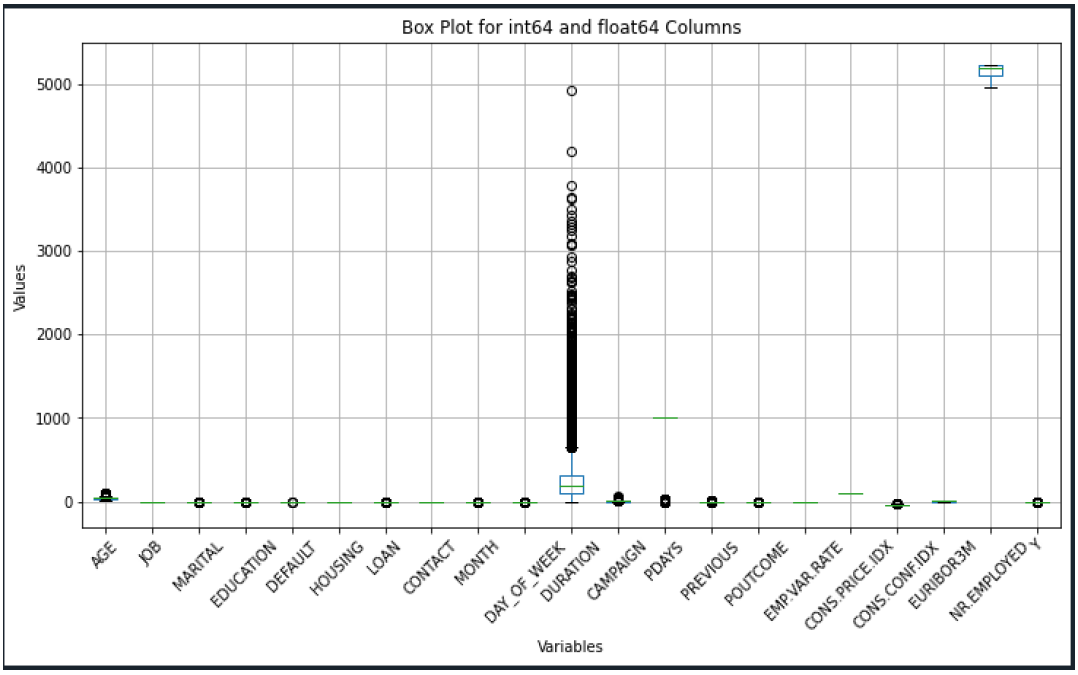


Figure 3.4: Boxplot for outliers

To identify outliers in the dataset, the Interquartile Range (IQR) method was used. The analysis revealed the presence of outliers in multiple columns, including 'age', 'marital', 'education', 'default', 'loan', 'month', 'day_of_week', 'duration', 'campaign', 'pdays', 'previous', 'poutcome', 'cons.conf.idx' and 'y'. The IQR method identified a significant number of outliers. After detecting outliers using the IQR method, the dataset was further examined using the Z-score method to validate the findings. The analysis revealed the presence of outliers in the columns 'month', 'duration', 'campaign', 'pdays', and 'previous'. To address this issue, the QuantileTransformer technique was employed to impute the outliers. The outliers were replaced with the corresponding quantile value, resulting in a more robust representation of the data. As the 'age' variable is assumed to be a critical factor and should not be imputed with any value, it was excluded from

the outlier detection evaluation. Please refer to Figures 3.5 for a representation of the number of outliers in each column.

```
Outlier detection by Interquantile Range method:
469 number of row are Outliers in column AGE
4612 number of row are Outliers in column MARITAL
4176 number of row are Outliers in column EDUCATION
3 number of row are Outliers in column DEFAULT
6248 number of row are Outliers in column LOAN
2632 number of row are Outliers in column MONTH
7827 number of row are Outliers in column DAY_OF_WEEK
2963 number of row are Outliers in column DURATION
2406 number of row are Outliers in column CAMPAIGN
1515 number of row are Outliers in column PDAYS
5625 number of row are Outliers in column PREVIOUS
4252 number of row are Outliers in column POUTCOME
447 number of row are Outliers in column CONS.CONF.IDX
4640 number of row are Outliers in column Y
```

(a) Outliers By IQR

```
Outlier detection by ZScore method:
2632 number of row are Outliers in column MONTH
861 number of row are Outliers in column DURATION
869 number of row are Outliers in column CAMPAIGN
1515 number of row are Outliers in column PDAYS
1064 number of row are Outliers in column PREVIOUS
```

(b) Outliers By ZScore

Figure 3.5: Outliers Details by IQR and ZScore

- Imbalance data: The target variable 'y' in the dataset consists of 36,548 values for the category 'no' and 4,640 values for the category 'yes', which means that 88% of data is for the category 'no' and only 11% is for 'yes'. This reveals an imbalance in the dataset. To address this issue, the SMOTE (Synthetic Minority Over-sampling Technique) and RandomUnderSampler methods were employed to balance the dataset, ensuring a more equitable distribution for subsequent analysis. Please refer Figure 3.6.

```
The count of each category in the target variable is:
0    36548
1     4640
Name: Y, dtype: int64
The percentage distribution of target class is:
0    0.887346
1    0.112654
Name: Y, dtype: float64
```

Figure 3.6: Imbalance Data

- The cleaned dataset has been saved for further use.

3.3 Data Visualisation

The dataset was analysed in three stages to gain insights and understand the relationships between variables.

In the first stage, a heat map was created to visualize the correlation between variables, and the Pearson Correlation coefficient was calculated. This helped identify the variables with the highest correlation. Additionally, a Pairplot was generated to provide a comprehensive view of the relationships between these highly correlated variables.

The second stage involved examining the unique values of each categorical variable and plotting them for better understanding. For numerical variables, statistical measures such as mean and standard deviation were computed to gain insights into their distribution and variability.

In the final stage, the relationship between the categorical variables and the target variable was examined using a bar plot. This visualization helped to identify any significant patterns or trends between the categorical variables and the target.

Overall, these three stages of analysis provided valuable insights into the dataset, uncovering important relationships and patterns that can inform further decision-making and analysis.

- Correlation between variables: A heat map was generated to gain insights into the relationships between variables. As depicted in Figure 3.7, variables with a correlation coefficient greater than 0.5 are presented in Figure 3.8 below along with their respective correlation values. Notably, the highest correlations are observed between EMP.VAR.RATE and EURIBOR3M, EMP.VAR.RATE and NR.EMPLOYED, and EURIBOR3M and NR.EMPLOYED. These findings suggest a potential relationship among these three variables. To further illustrate this relationship, a Pair plot is showcased in Figure 3.9

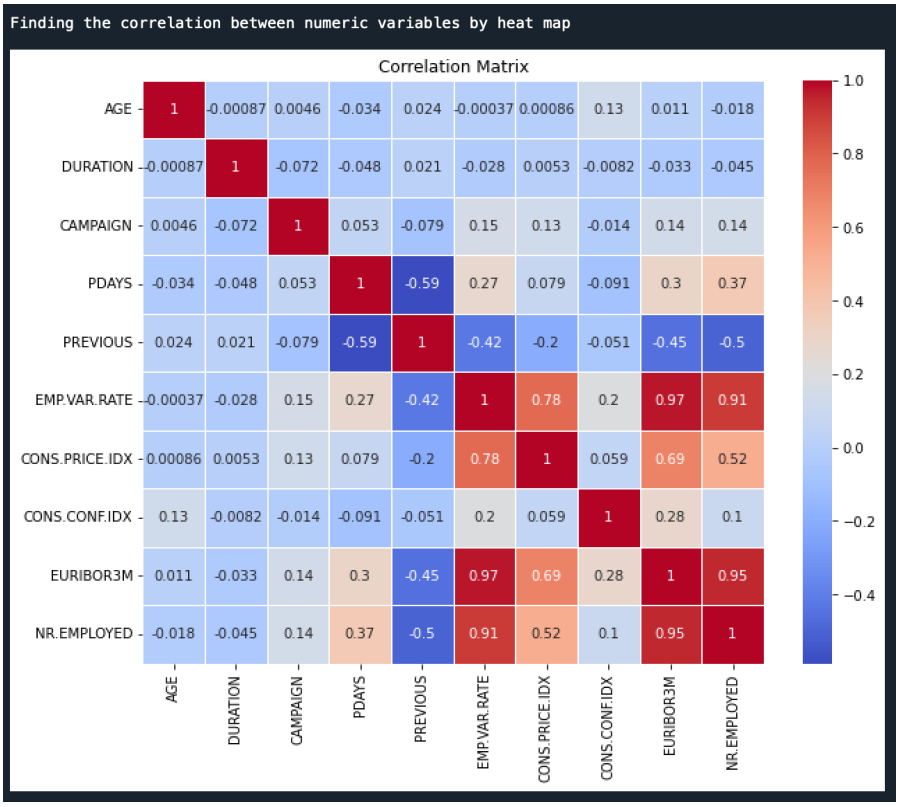


Figure 3.7: Heat map

```
Correlation between EMP.VAR.RATE and CONS.PRICE.IDX: 0.7753
Correlation between EMP.VAR.RATE and EURIBOR3M: 0.9722
Correlation between EMP.VAR.RATE and NR.EMPLOYED: 0.9070
Correlation between CONS.PRICE.IDX and EURIBOR3M: 0.6882
Correlation between CONS.PRICE.IDX and NR.EMPLOYED: 0.5220
Correlation between EURIBOR3M and NR.EMPLOYED: 0.9452
```

Figure 3.8: Correlation coefficients

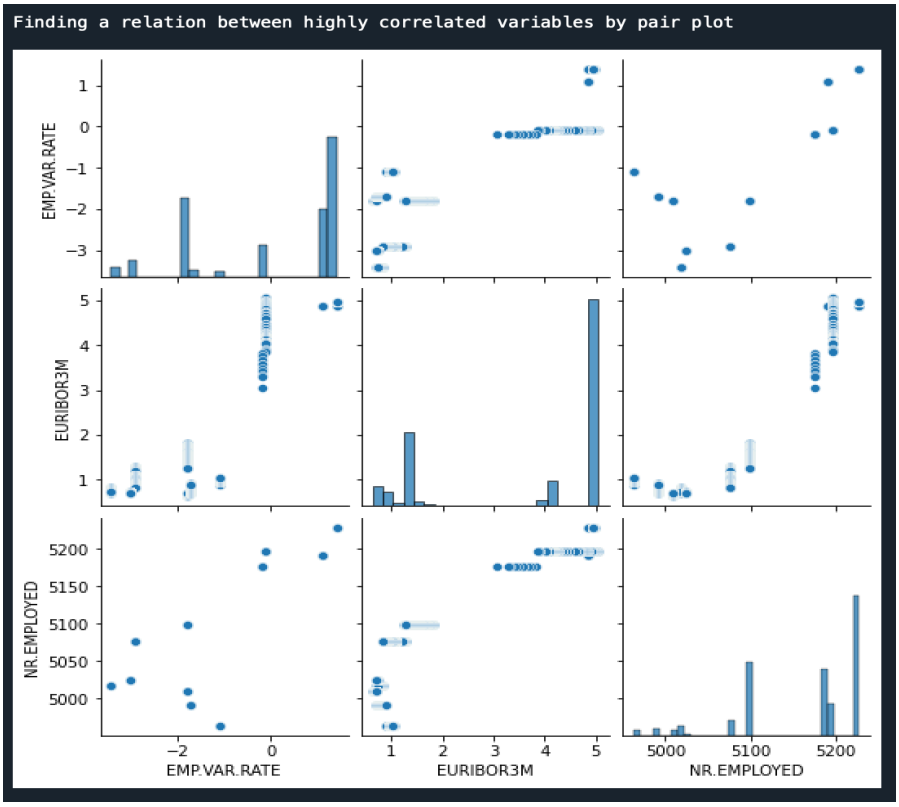
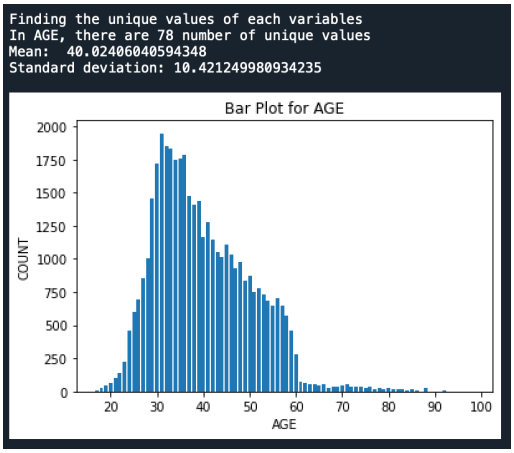


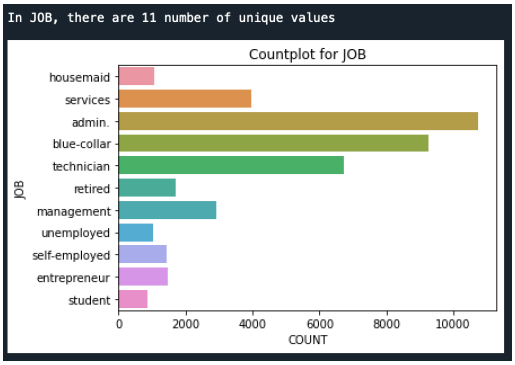
Figure 3.9: Pair plot

- Understanding each variable individually: The unique values of categorical variables were explored and visualized using bar plots. The numerical variables were analyzed by calculating their mean and standard deviation, providing insights into their distribution. Please refer to Figure 3.11 for the detailed visualizations. Figure 3.10(a) shows that the age range of customers who are likely to be eligible for a term deposit is between 20 and 60 years. Among the various job categories, administrative jobs have the highest representation, as depicted in Figure 3.10(b). Figure 3.10(c) indicates that the number of married customers surpasses that of single and divorced customers, suggesting their eligibility for term deposit offers. Figure 3.10(d) showcases the presence of university degree holders who are potentially targeted for term deposit contact. Figure 3.10(e) reveals that the majority of customers do not have any credit default. Figure 3.10(f) illustrates that the number of customers with housing loans is greater than the number of customers without housing loans. As shown in Figure 3.10(g), the count of customers with personal loans is significantly lower than the count of customers without personal loans. As depicted in Figure 3.10(h), the count of customers with cellular phones is higher compared to customers with telephone connections. As shown in Figure 3.10(i), the majority of customers were contacted during the month of May in the past. As depicted in Figure 3.10(j), the number of customers contacted during the weekdays shows a slight increase on Mondays and Thursdays, although the overall impact is not substantial. As shown in Figure 3.10(k), the duration of calls ranged from 0 to 1000 seconds. On average,the calls lasted for approximately 258.28 seconds with the customers. As shown in Figure 3.10(l), the number of contacts made during the campaign ranged from 1 to 10, with a significant proportion of customers being contacted only once. As depicted in Figure 3.10(m), the average number of times

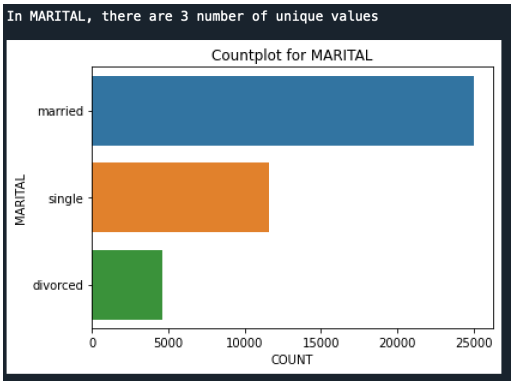
customers were contacted before the campaign was relatively low. As shown in Figure 3.10(n), the number of calls made to customers prior to the campaign is relatively low. Only a small number of customers have been contacted once or twice before the campaign. Based on Figure 3.10(o), the majority of the previous marketing campaign outcomes are non-existent, with a smaller number of successes and failures. Based on Figure 3.10(p), the employment variation rate is highest between 1 and 2, with an average of 0.0818. According to Figure 3.10(q), the average customer price index is 93.57, with a range between 92.5 and 94.5. Figure 3.10(r) indicates that the customer confidence index ranges from -35 to -38, with an average value of -40.50. Figure 3.10(s) shows that the maximum Euribor 3 month rate falls within the range of 4 to 5. Figure 3.10(t) illustrates that the average number of employees is 5167, with a maximum exceeding 5200. Figure 3.10(u) reveals that a small number of customers already have a term deposit.



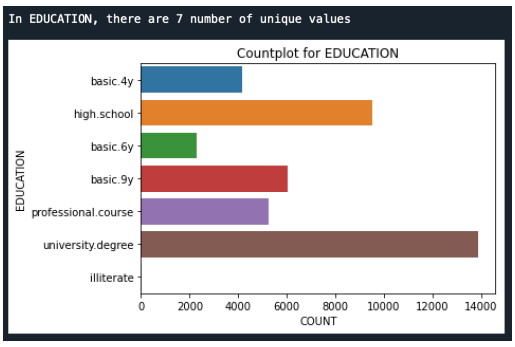
(a) Plot for Age



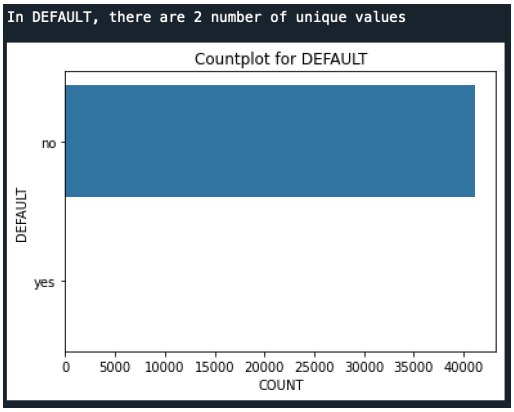
(b) Plot for Job



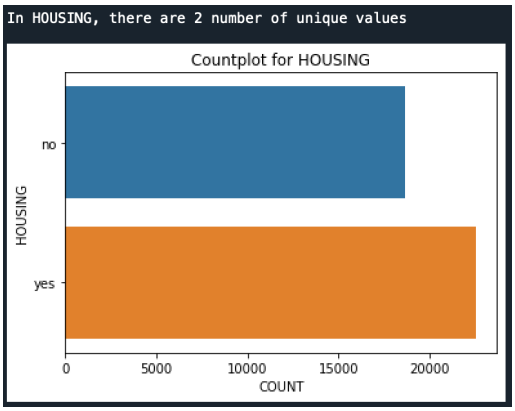
(c) Plot for Marital



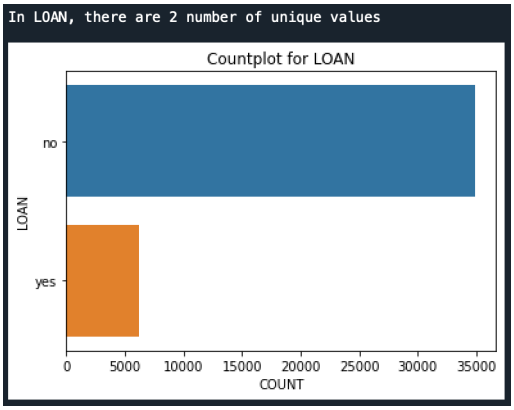
(d) Plot for Education



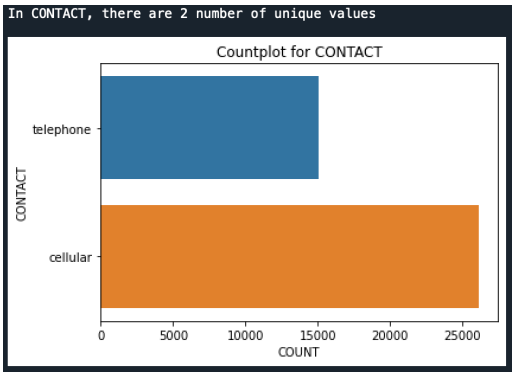
(e) Plot for Default



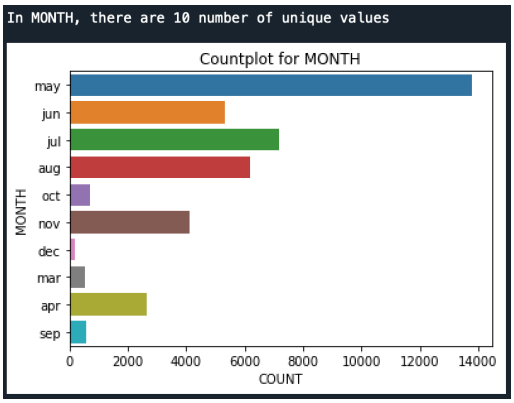
(f) Plot for Housing



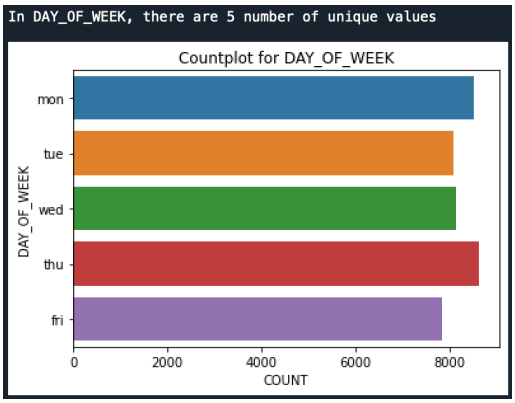
(g) Plot for Loan



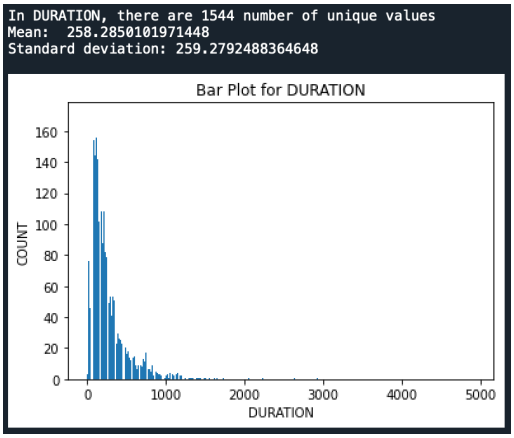
(h) Plot for Contact



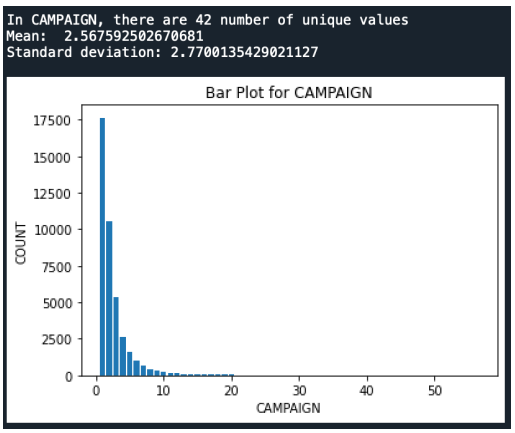
(i) Plot for Month



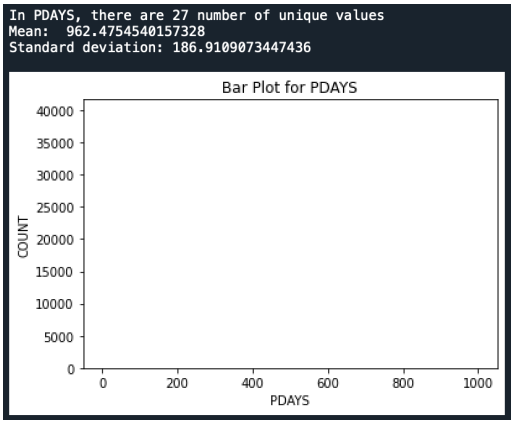
(j) Plot for Day_of_Week



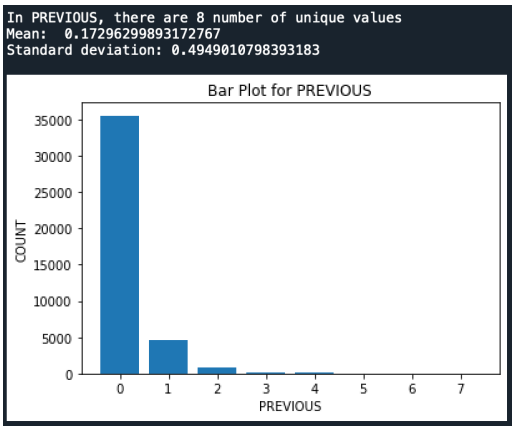
(k) Plot for Duration



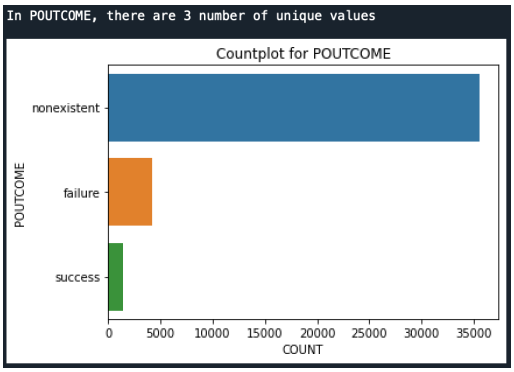
(l) Plot for Campaign



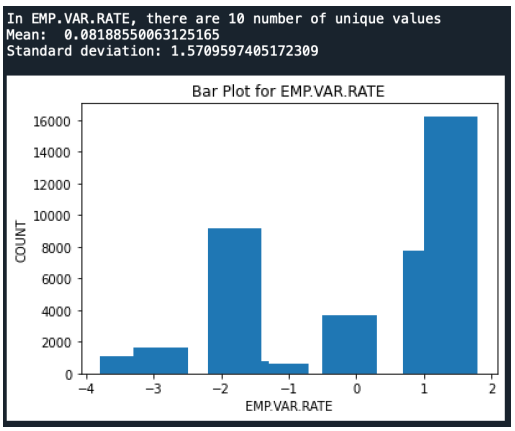
(m) Plot for PDays



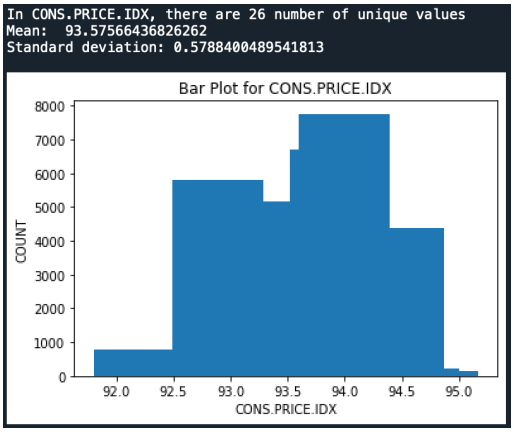
(n) Plot for Previous



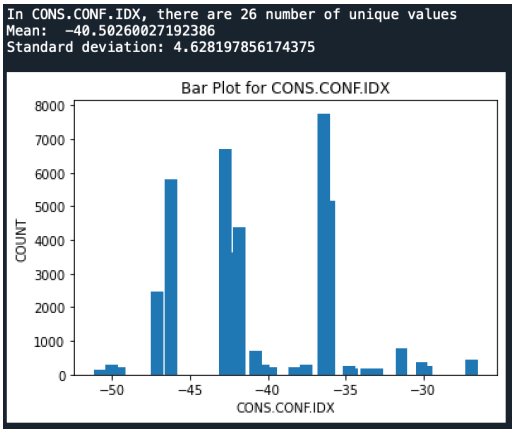
(o) Plot for POutcome



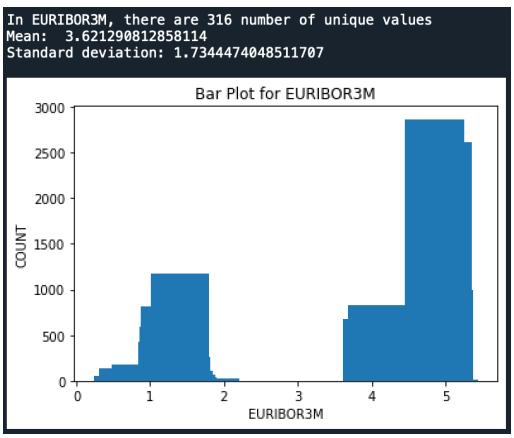
(p) Plot for EMP.VAR.RATE



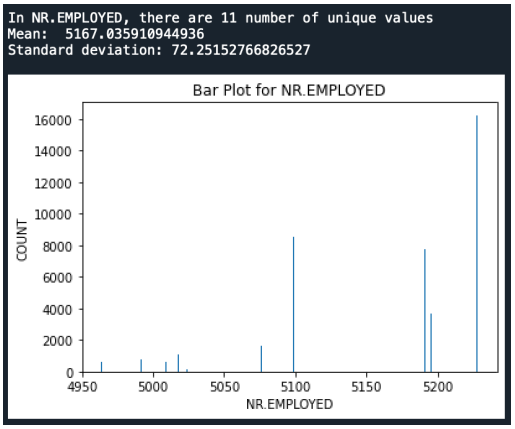
(q) Plot for CONS.PRICE.IDX



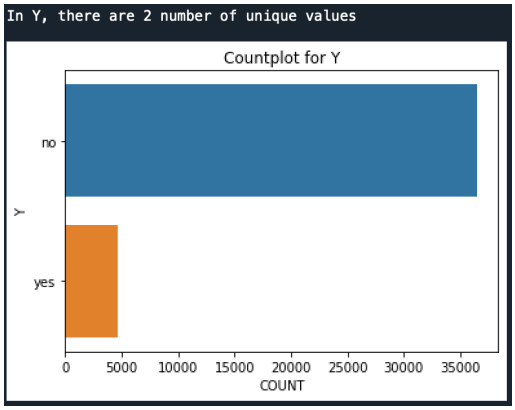
(r) Plot for CONS.CONF.IDX



(s) Plot for EURIBOR3M



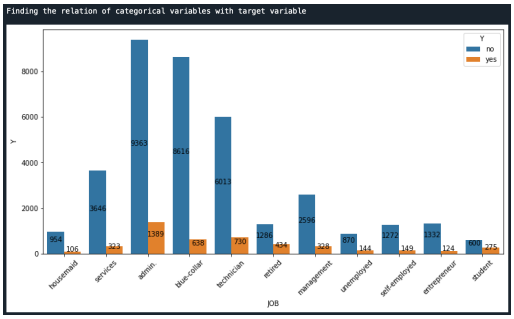
(t) Plot for NR.Employed



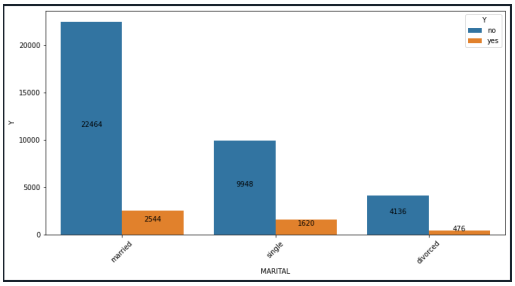
(u) Plot for Y

Figure 3.10: Plots of all variables

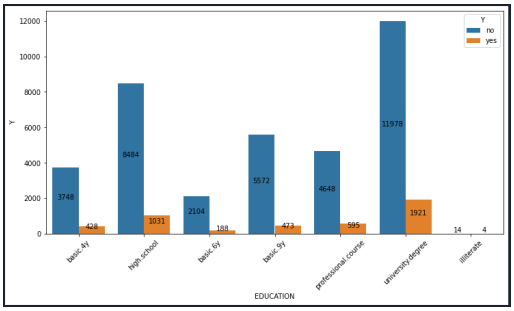
- Relation of categorical variable with target variable: Below are the plots that illustrate the relationship between categorical variable and the target variable. Figure 3.11(a) demonstrates that the highest number of customers with a term deposit are employed in administrative roles. Figure 3.11(b) illustrates that the highest number of customers with a term deposit are married individuals. As shown in Figure 3.11(c), the majority of customers with a term deposit are those who hold a university degree. According to Figure 3.11(d), there is a total of 4640 customers who have had a term deposit in the past. Based on the information shown in Figure 3.11(e), there is a relatively similar proportion between housing loan holders who have a term deposit and those who do not have a term deposit. Based on the data presented in Figure 3.11(f), it can be observed that the number of term deposits for customers without a personal loan is significantly higher compared to those with a personal loan. Based on the information depicted in Figure 3.11(g), it can be observed that the majority of customers who have opted for a term deposit have a cellular phone. Based on the data visualized in Figure 3.11(h), it is evident that the majority of customers were contacted during the month of May. Furthermore, this period also witnessed the highest number of conversions to a term deposit. Based on the information presented in Figure 3.11(i), it can be observed that the majority of customers were contacted on Monday and Thursday. Additionally, there are only 1,045 customers who have a term deposit. According to the data presented in Figure 3.11(j), there are 3,141 non-existent customers but have a term deposit.



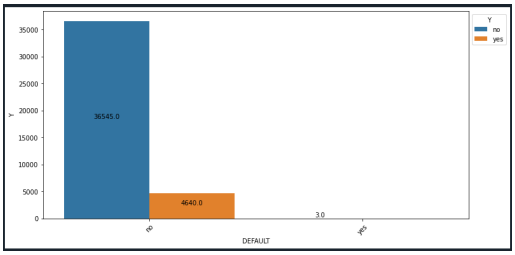
(a) Plot for Y versus Job



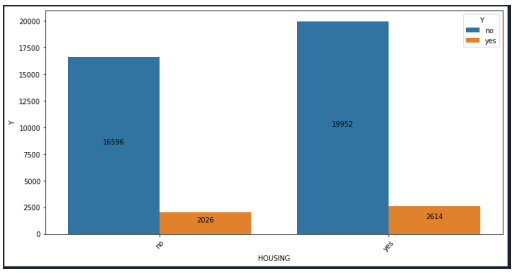
(b) Plot for Y versus Marital



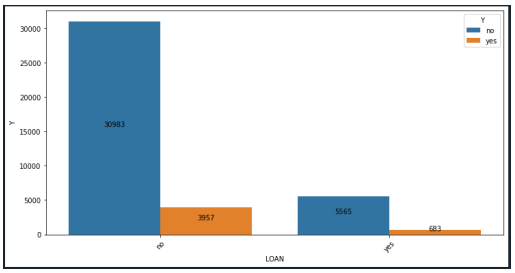
(c) Plot for Y versus Education



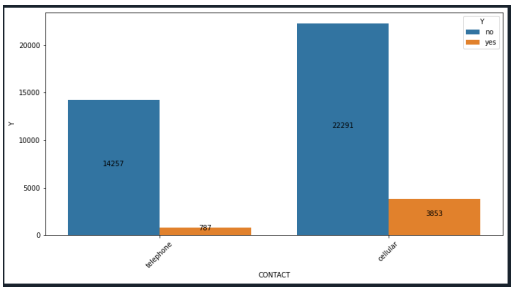
(d) Plot for Y versus Defualt



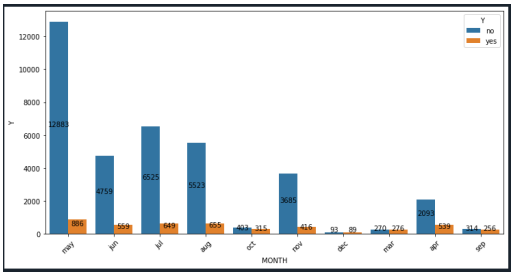
(e) Plot for Y versus Housing



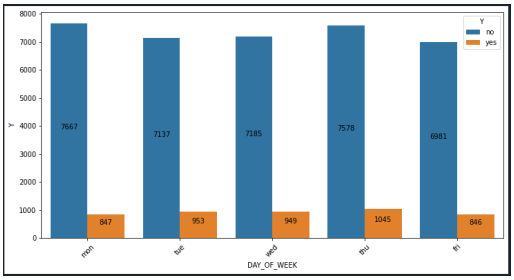
(f) Plot for Y versus lOan



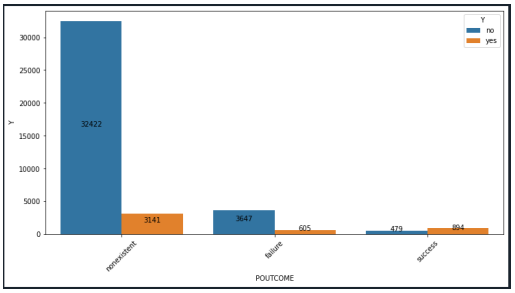
(g) Plot for Y versus Contact



(h) Plot for Y versus Month



(i) Plot for Y versus Day of Week



(j) Plot for Y versus POutcome

Figure 3.11: Plots of all variables

3.4 Data Modelling

As mentioned earlier, the implementation is divided into various steps. Firstly the baseline models are created to draw the accuracy. After which linear, ensemble and boosting models are developed to predict the term deposit.

3.4.1 Feature Selection

Before implementing any model on the data, feature selection is necessary. Thus, the dataset is divided into two parts: features (all attributes except the target attribute) and target (class attribute). The target dataset includes only the "Y" column, as it represents the class attribute. The remaining attributes are placed in the features dataset. Feature selection methods were not employed in this case due to the dataset's limited number of attributes. Consequently, dimensionality reduction was not performed on this dataset.

3.4.2 Train and Test Split

The data is divided into training and test sets using the HoldOut method, with 70% of the data allocated to the training set and 30% to the test set. The training dataset contains 28,831 records and 20 attributes, while the test dataset comprises 12,357 records.

3.4.3 Baseline models

The research focuses on a binary classification problem, aiming to predict whether bank customers will subscribe to a term deposit based on the marketing campaign. As a baseline model, we employed the Support Vector Machine Classifier algorithm, following the steps mentioned in [2]. The SVC model is built to maintain the coding standards as the same steps are followed for implementing other models. The SVC model is trained using 70% of the data as the training set and then tested on the remaining 30% as the test set to assess accuracy. The evaluation results, depicted in Figure 3.12, show an accuracy of 89.54%. In Chapter 4, we will compare these SVC results with other models.

```
On SVC Accuracy is: 89.544388
Confusion matrix:
[[10779   160]
 [ 1132   286]]
Classification report:
              precision    recall  f1-score   support

           0       0.90      0.99      0.94     10939
           1       0.64      0.20      0.31      1418

    accuracy                           0.90     12357
   macro avg       0.77      0.59      0.63     12357
weighted avg       0.87      0.90      0.87     12357

Confusion matrix plot:
```

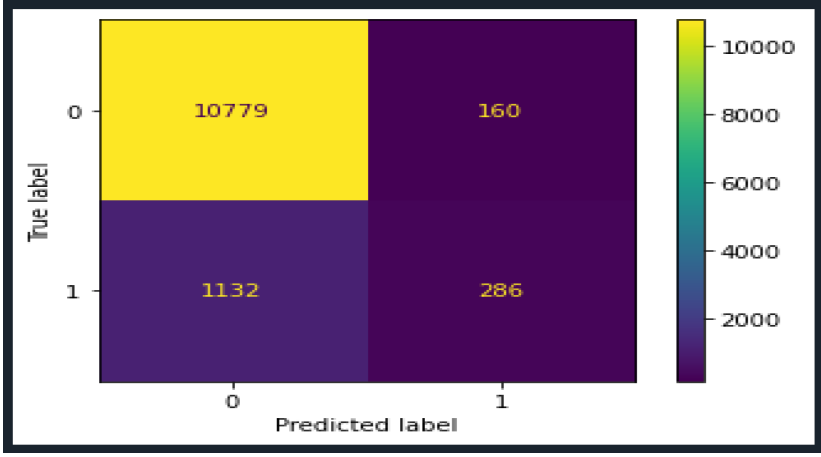


Figure 3.12: Baseline Support Vector Classifier model

3.4.4 Basic Model Evaluation

Linear models accessed are Linear Regression and Linear Discriminant Analysis. Ensemble models accessed are Random Forest Classifier and ADABoost. Boosting models accessed are XGBoost and HistGradientBoosting.

Linear Models

Linear Regression and Linear Discriminant Analysis models has been created without any parameters. The models are trained using 70% of the data as the training set and then tested on the remaining 30% as the test set to assess accuracy. Figure 3.13 shows accuracy of Linear Regression is 90.28%. Figure 3.14 shows that the accuracy of Linear Discriminant Analysis model is 90.24%.

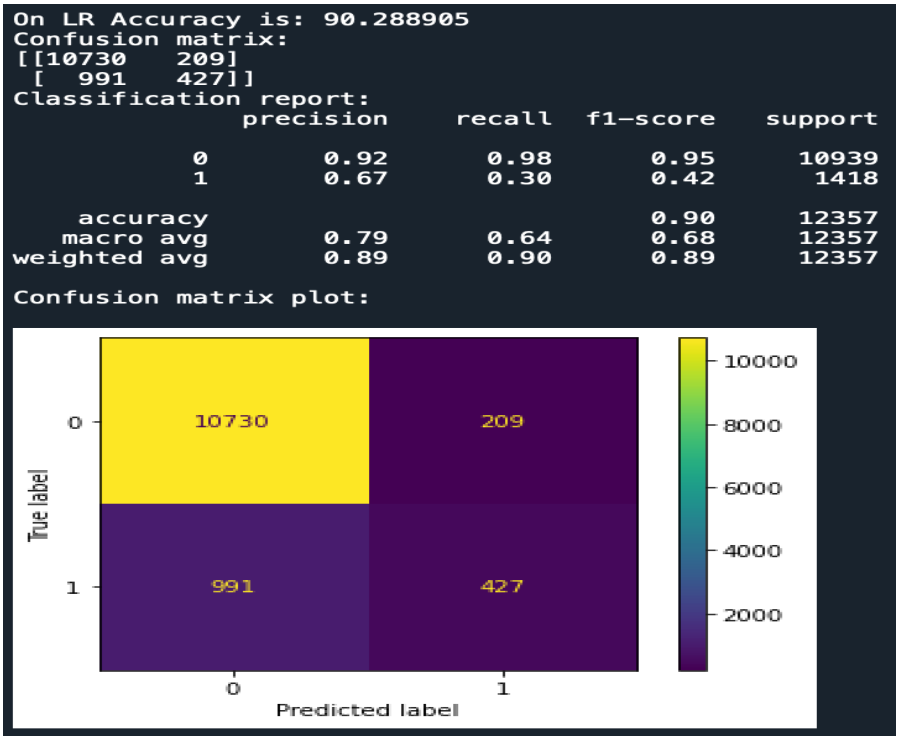


Figure 3.13: Basic Linear Regression model

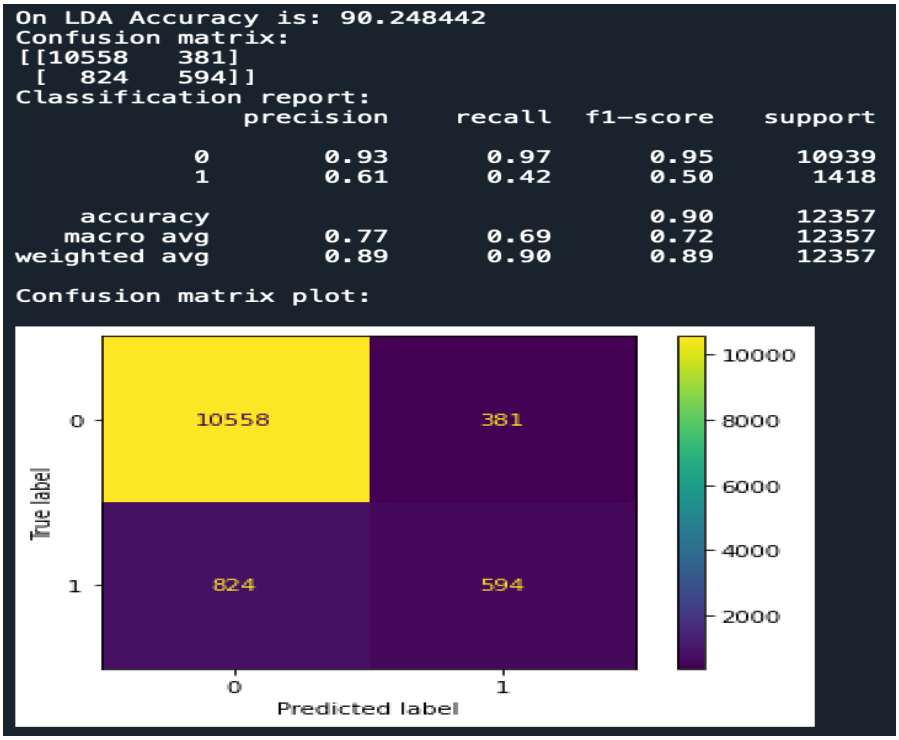


Figure 3.14: Basic Linear Discriminant Analysis model

Ensemble Models

Random Forest Classifier model has been created without any parameters, whereas ADABoost model has been created with DecisionTreeClassifier having max_depth as 1. The models are trained using 70% of the data as the training set and then tested on the remaining 30% as the test set to assess accuracy. Figure 3.15 shows accuracy of Random Forest Classifier model is 91.01%. Figure 3.14 shows that the accuracy of ADABoost model is 90.87%.

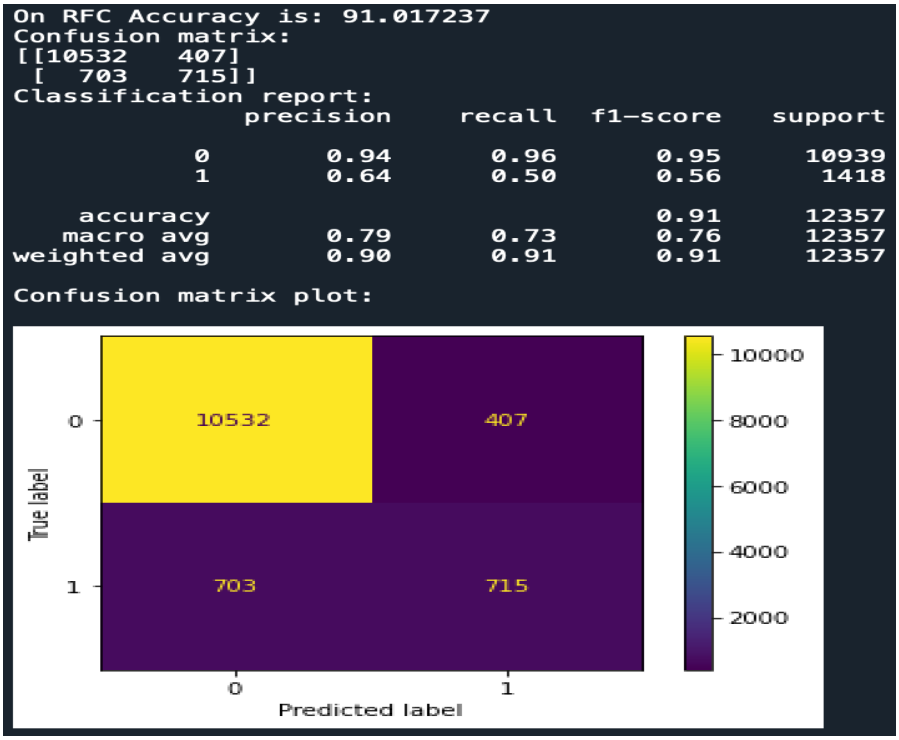


Figure 3.15: Basic Random Forest Classifier model

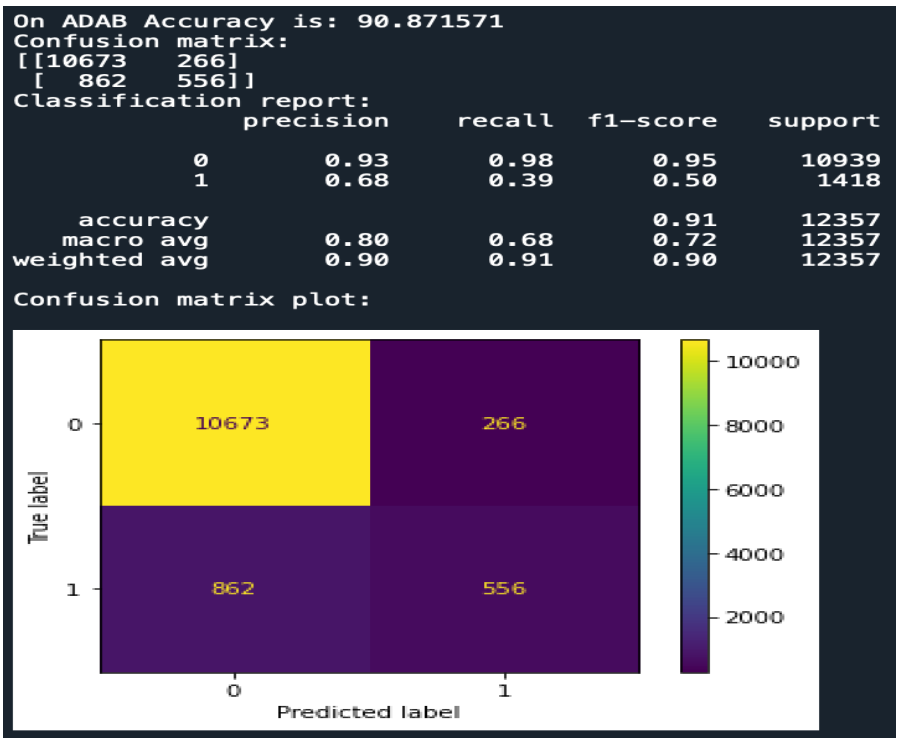


Figure 3.16: Basic ADABoost model

Boosting Models

XGBoost and HistGradientBoosting models has been created without any parameters. The models are trained using 70% of the data as the training set and then tested on the remaining 30% as the test set to assess accuracy. Figure 3.17 shows accuracy of XGBoost model is 91.19%. Figure 3.18 shows that the accuracy of HistGradientBoosting model is 91.50%.

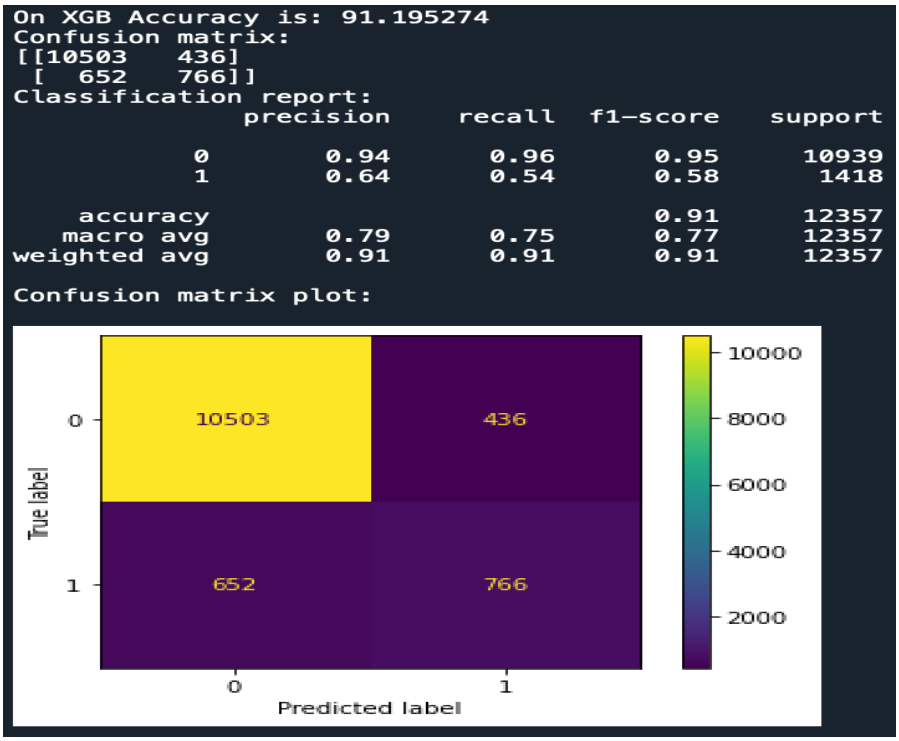


Figure 3.17: Basic XGBoost model

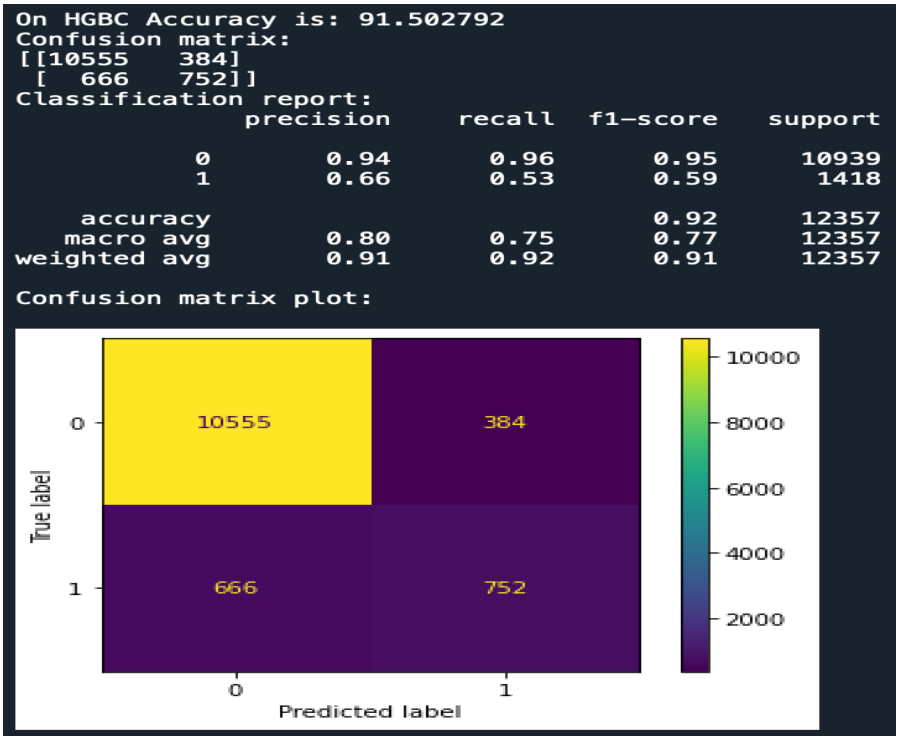


Figure 3.18: Basic HistGradientBoosting model

3.4.5 Model Evaluation with Random State

Each model is trained using different random states, specifically 10, 20, 30, 40, and 50. As depicted in Figure 3.19, all the models have been evaluated on these random states, and it appears that there is minimal variation in the results across different states. For a clearer view, Figure 3.20 displays the models alongside their respective best-performing random state.

Accuracy of each model after optimizing model with random states:

	Model Name	Random State	Accuracy
0	LR	10	90.232257
1	LR	20	90.644979
2	LR	30	90.547868
3	LR	40	90.240350
4	LR	50	90.402201
5	LDA	10	89.932832
6	LDA	20	90.491220
7	LDA	30	90.442664
8	LDA	40	90.321275
9	LDA	50	90.167516
10	RFC	10	90.798738
11	RFC	20	91.251922
12	RFC	30	91.421866
13	RFC	40	90.895849
14	RFC	50	90.952497
15	ADAB	10	90.466942
16	ADAB	20	91.162904
17	ADAB	30	90.839200
18	ADAB	40	90.531682
19	ADAB	50	90.750182
20	XGB	10	91.268107
21	XGB	20	91.421866
22	XGB	30	91.349033
23	XGB	40	91.130533
24	XGB	50	91.049608
25	HGBC	10	91.251922
26	HGBC	20	91.794125
27	HGBC	30	91.640366
28	HGBC	40	91.308570
29	HGBC	50	91.251922

Figure 3.19: Random State Results

The best performed random state for each model:

	Model Name	Random State	Accuracy
0	ADAB	20	91.162904
1	HGBC	20	91.794125
2	LDA	20	90.491220
3	LR	20	90.644979
4	RFC	30	91.421866
5	XGB	20	91.421866

Figure 3.20: Best Random State Results

3.4.6 Model Evaluation with Cross Validation

StratifiedKFold and RepeatedStratifiedKFold Cross Validation has been utilized to improve the performance as it splits the data approximately in the same percentage. With multiple classes in the dataset, an appropriate split of the train and test sets can be achieved using these techniques. These methods ensure that the distribution of classes remains balanced across the train and test sets, thereby enhancing the reliability of the evaluation process for classification models. Each model undergoes training using either the RepeatedStratifiedKFold or StratifiedKFold technique for Cross Validation. Specifically, ADABoost and XGBoost models are trained using RepeatedStratifiedKFold, while all other models are trained using StratifiedKFold. StratifiedKFold and RepeatedStratifiedKFold are used with 10 splits and the best random_state as obtained in above step in Section 3.4.5. The outcomes of cross validation with the mentioned techniques are depicted in Figure 3.21, displaying the performance of each model. Accuracy levels in cross validation vary across all models, ranging from 80% to 96%. For a more representative view of the mean accuracy, please refer to Figure 3.22.

```
Mean and Standard deviation of each model on balanced data with cross validation:
  Model Name       Mean       STD
0         LR  80.847115  0.007591
1        LDA  83.131764  0.003205
2        RFC  95.467601  0.002491
3       ADAB  92.685421  0.002787
4        XGB  95.094585  0.002332
5       HGBC  94.765799  0.002836
```

Figure 3.21: Cross Validation Result

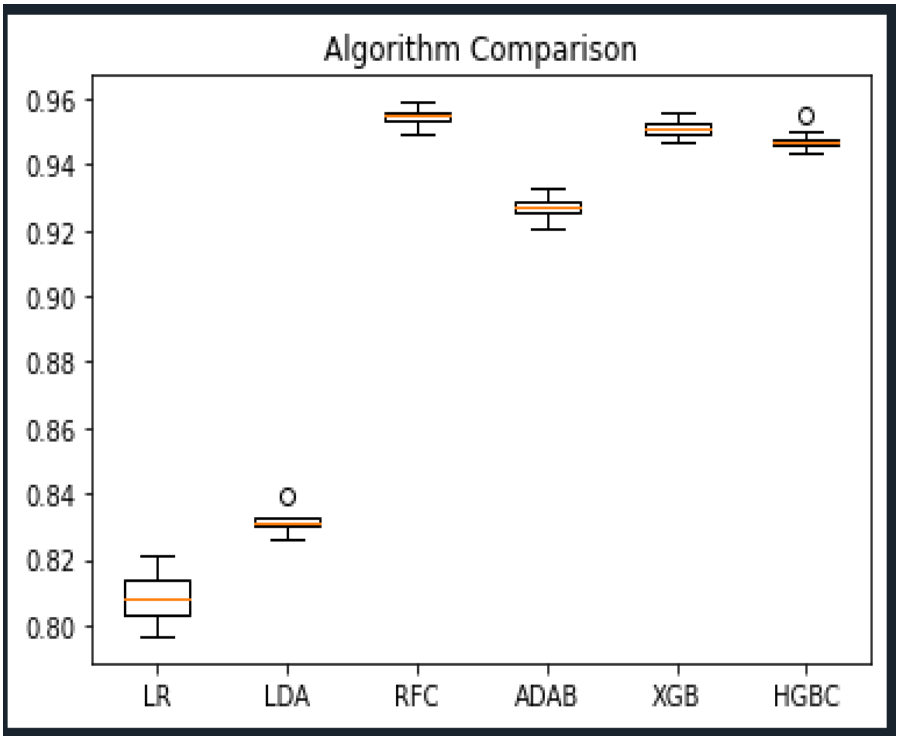


Figure 3.22: Cross Validation Algorithm Plot

3.4.7 Hyperparameter Tuning

All the models are tuned with various parameters as mentioned below.

Linear Regression model is tuned with 'solver' as ['sag', 'saga'], 'penalty' as 'l2'. Linear Discriminant Analysis model is tuned with 'solver' as ['svd', 'lsqr', 'eigen'].

Random Forest Classifier model is tuned with 'max_depth' as [1, 3, 5, 7], 'n_estimators' as [10, 50, 100, 200] and 'max_features' as ['sqrt', 'log2']. ADABoost model is tuned with 'n_estimators' as [10, 50, 100] and 'learning_rate' as [0.001, 0.01, 0.1, 1.0].

XGBoost is tuned with 'max_depth' as [1, 3, 5, 7] and 'min_child_weight' as [1,2]. HistGradientBoosting model is tuned with 'max_bins' as [10, 30, 50, 70], 'max_iter' as [100,200].

The models were evaluated based on their accuracy, and the hyperparameters that yielded the best performance were recorded along with its accuracy, as shown in Figure 3.23.

```
LR Mean Accuracy: 0.810879
Config: {'penalty': 'l2', 'solver': 'sag'}
LDA Mean Accuracy: 0.831194
Config: {'solver': 'svd'}
RFC Mean Accuracy: 0.908600
Config: {'max_depth': 7, 'max_features': 'sqrt', 'n_estimators': 200}
ADAB Mean Accuracy: 0.934141
Config: {'learning_rate': 1.0, 'n_estimators': 100}
XGB Mean Accuracy: 0.951064
Config: {'max_depth': 7, 'min_child_weight': 1}
HGBC Mean Accuracy: 0.947494
Config: {'max_bins': 70, 'max_iter': 200}
```

Figure 3.23: Best Hyperparameters

3.4.8 Tuned Model Evaluation

The hyperparameter values retrieved by the hyperparameter tuning in previous Section 3.4.7 was used here to assess the final models. Summary of all models is displayed in Figure 3.24 and Figure 4.1. The accuracy has been evaluated as shown in Figure 3.24 and plotted in Figure 4.1.

	Model Name	Accuracy - Mean	Accuracy - STD
0	LR	0.811016	0.004172
1	LDA	0.831331	0.003949
2	RFC	0.909188	0.003539
3	ADAB	0.934894	0.002585
4	XGB	0.951639	0.002045
5	HGBC	0.947754	0.001736

Figure 3.24: Models performance with parameters

3.5 Model Deployment

After evaluating all the models, the best-performing model was used to create the "model.pkl" pickle file. Model deployment was achieved using Flask, with the Predict API created in the "app.py" file. The web page style was defined in the "style.css" file, and the page layout was specified in the "index.html" file. When executing the "app.py" file, the URL "http://127.0.0.1:5000/" is generated. The Figure 3.25 shows the web application produced. When entered the values in all the field to test the prediction as in Figure 3.26, the result produced in displayed below the "predict" button as displayed in Figure 3.27.

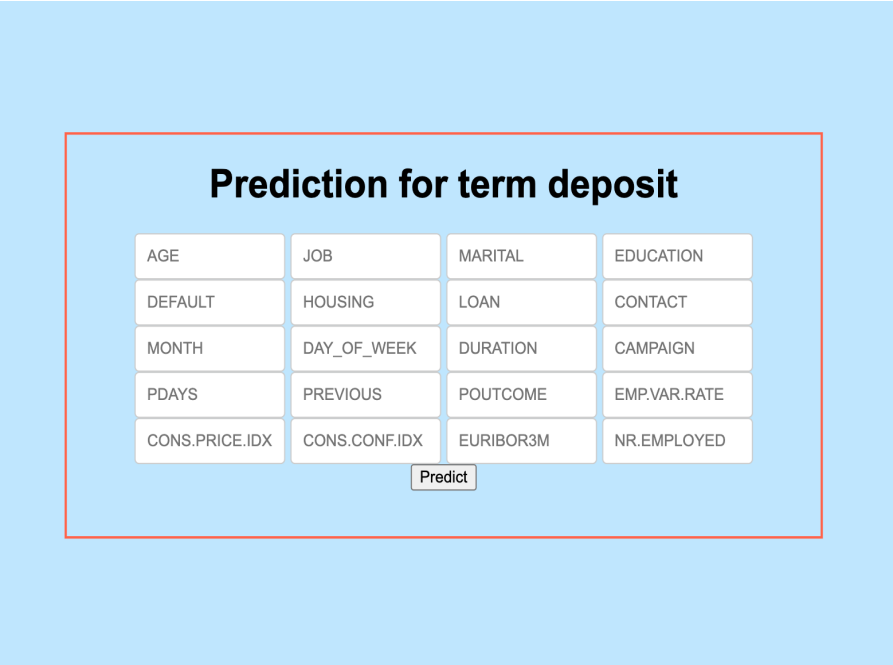


Figure 3.25: Web application

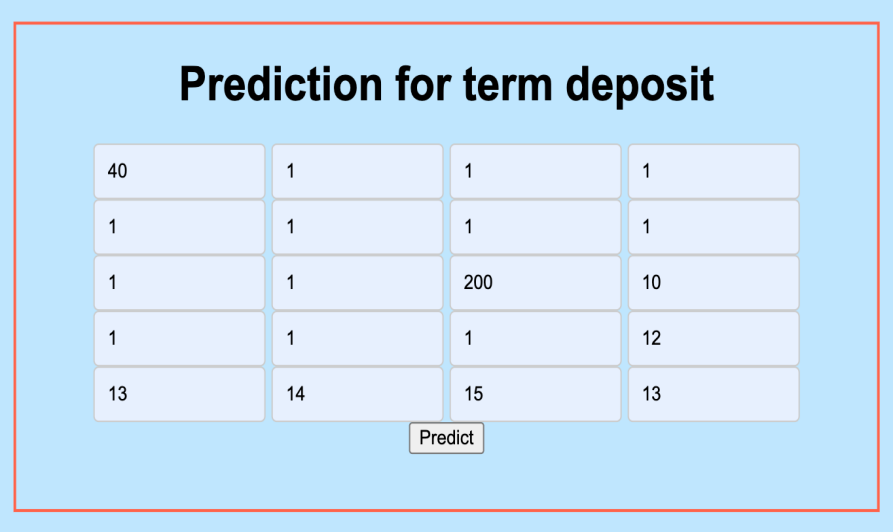


Figure 3.26: Values entered in web page

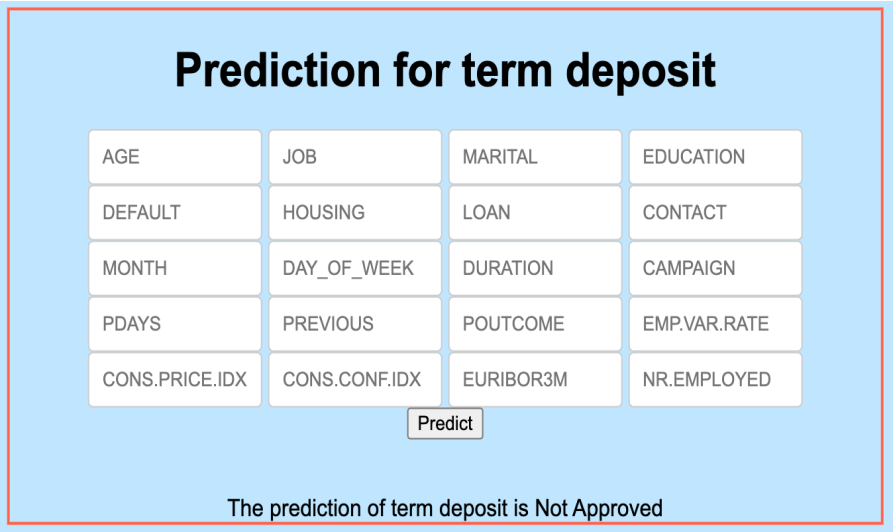


Figure 3.27: Prediction of term deposit

3.6 Summary

In this chapter, all the experiments carried out on several machine learning models have been clearly explained with their accuracy including its demonstration on how the basic and final network of each model is constructed along with the parameters. Also, the model deployment carried out to predict the term deposit. The results of each model will be discussed in Chapter 4.

Chapter 4

Results

Initially, the results presented pertain to the baseline model. Subsequently, a comprehensive comparison among the other models is provided in the summary.

4.1 Baseline model - Support Vector Machine Classifier

The baseline models "Support Vector Machine Classifier" has produced an accuracy of 89.54% as depicted in Figure 3.12

4.2 Linear Models

The basic Linear Regression model achieved an accuracy of 90.28% as depicted in Figure 3.13, while the Linear Discriminant Analysis model achieved an accuracy of 90.24% as shown in Figure 3.14. There is minimal difference in accuracy between these models without any hyperparameters. When considering the best random state, as illustrated in Figure 3.20, the Linear Regression model obtained an accuracy of 90.64%, and the Linear Discriminant Analysis model achieved 90.49%. Moving on to the model evaluation with cross-validation and random state, as displayed in Figure 3.21 and 4.1, the Linear Regression model scored an accuracy of 80.84%, and the Linear Discriminant Analysis model achieved 83.13% accuracy.

For the tuned model evaluation as shown in Figure 3.24, which involved random state, cross-validation, and the best hyperparameters, the Linear Regression model achieved an accuracy of 81.10%, while the Linear Discriminant Analysis model maintained an accuracy of 83.13%. Comparing the results of the basic models, models with random state, cross-validation along with random state, and the tuned models, it can be observed that the models' performance exhibited a slight decrease.

4.3 Ensemble Models

The basic Random Forest Classifier model achieved an accuracy of 91.01%, as shown in Figure 3.15, while the ADABoost model achieved an accuracy of 90.87% as depicted in Figure 3.16. There is minimal difference in accuracy between these models without any hyperparameters. When considering the best random state, as illustrated in Figure 3.20, the Random Forest Classifier model obtained an accuracy of 91.42%, and the ADABoost model achieved 91.16%.

Moving on to the model evaluation with cross-validation and random state, as displayed in Figure 3.21, the Random Forest Classifier model scored an accuracy of 95.46%, and the ADABoost model achieved 92.68% accuracy.

For the tuned model evaluation as shown in Figure 3.24 and 4.1, which involved random state, cross-validation, and the best hyperparameters, the Random Forest Classifier model achieved an accuracy of 90.91%, while the ADABoost model maintained an accuracy of 93.48%. Comparing the results of the basic models, models with random state, cross-validation along with random state, and the tuned models, it can be observed that the performance of the Random Forest Classifier model has slightly decreased, whereas the performance of ADABoost has improved.

4.4 Boosting Models

The basic XGBoost model achieved an accuracy of 91.19%, as shown in Figure 3.17, while the HistGradientBoosting model achieved an accuracy of 91.50% as depicted in Figure 3.18. There is minimal difference in accuracy between these models without any hyperparameters. When considering the best random state, as illustrated in Figure 3.20, the XGBoost model obtained an accuracy of 91.42%, and the HistGradientBoosting model achieved 91.79%.

Moving on to the model evaluation with cross-validation and random state, as displayed in Figure 3.21, the XGBoost model scored an accuracy of 95.09%, and the HistGradientBoosting model achieved 94.76% accuracy.

For the tuned model evaluation as shown in Figure 3.24 and 4.1, which involved random state, cross-validation, and the best hyperparameters, the XGBoost model achieved an accuracy of 95.16%, while the HistGradientBoosting model maintained an accuracy of 94.77%. Comparing the results of the basic models, models with random state, cross-validation along with random state, and the tuned models, it can be observed that the performance of the XGBoost model has shown a significant improvement, whereas the performance of the HistGradientBoosting model has slightly decreased.

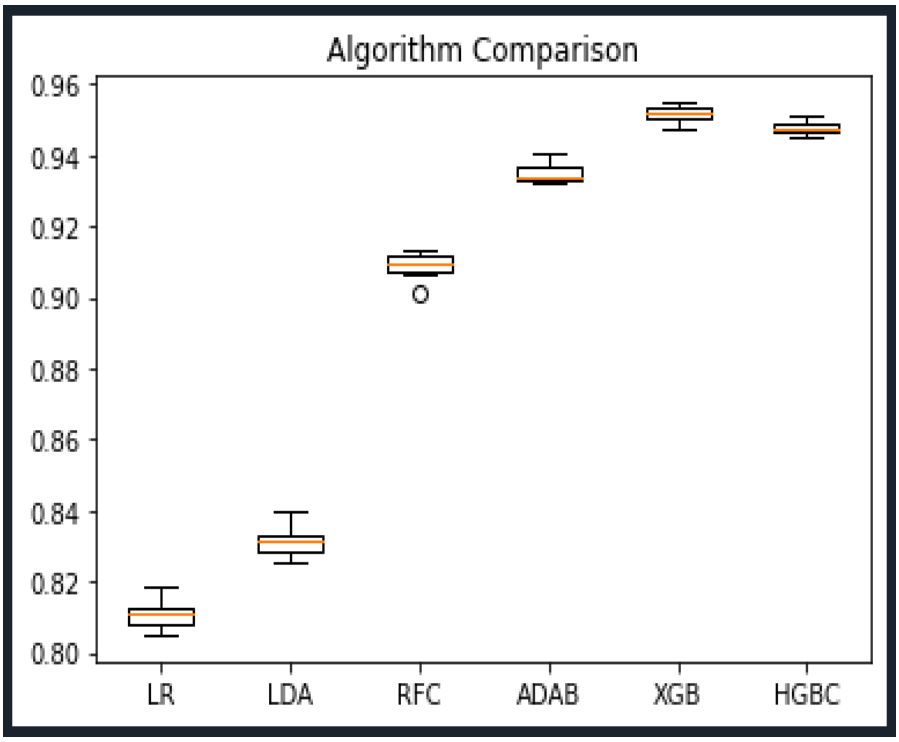


Figure 4.1: Plot of all models algorithm

4.4.1 Summary

Upon comparing the results of the baseline model with other models, it was evident that the performance of XGBoost was notably good. Evaluating all models without any hyper-parameters revealed that the performance of HistGradientBoosting was the best. However, after tuning the models, the performance of XGBoost surpassed all others and emerged as the best-performing model.

Chapter 5

Discussion and Analysis

5.1 Analysis

The analysis of the bank marketing campaign dataset was carried out with the goal of predicting whether customers would subscribe to the bank's term deposit product. The dataset consisted of various attributes, such as customer demographics, contact details, campaign-related data and previous interactions with the bank. The initial analysis involved exploring distribution of categorical and numerical variables. Data visualization was performed to gain insights into the relationships between different variables and the target variable, 'y'.

To build prediction, the models were trained and evaluated using various machine learning algorithms, including linear models (such as Linear Regressiona and Linear Discremininant Analysis), ensemble models (such as Random Forest Classifier and ADABoost), and boosting models (such as XGBoost and HistGradientBoosting). Performance metrics such as accuracy, precision, recall, F1-score, and ROC-AUC were used to evaluate model performance. Boosting model XGBoost consistently achieved high accuracy, above 90%, indicating their effectiveness in predicting customer subscriptions.

5.2 Significance of findings

The findings of the bank marketing campaign analysis hold significant implications for the banking industry and marketing strategies. Here are some key points highlighting the significance of the findings:

- Class imbalance is a common challenge in marketing datasets, where non-respondents outnumber respondents. The analysis successfully addressed this issue by employing techniques like SMOTE and RandomUnderSampler. This ensures that the model does not favor the majority class and provides more balanced and reliable predictions.
- Data-Driven Decision Making: The analysis promotes a data-driven approach to decision-making in the banking industry. Instead of relying solely on intuition or assumptions, banks can leverage the insights from the machine learning models to make informed decisions regarding their marketing campaigns, product offerings, and customer engagement strategies.

The significance of the findings from the bank marketing campaign analysis lies in its ability to provide valuable insights that empower banks to optimize their marketing efforts, improve customer targeting, and make data-driven decisions. These findings pave the way for more

efficient and effective marketing strategies, ultimately benefiting both the bank and its customers.

5.3 Limitation

The bank marketing campaign analysis has provided valuable insights, but it is essential to acknowledge its limitations to ensure a comprehensive understanding of the study. Limited dataset, imbalanced data, missing data comprises of "unknown" values are some limitation encountered in this research. By acknowledging these limitations, future research can focus on mitigating them and enhancing the analysis's accuracy and applicability to real-world scenarios.

Chapter 6

Conclusions and Future Work

6.1 Conclusions

This research paper explores various approaches to find the best machine learning solution for predicting term deposits based on bank marketing campaign data. The baseline model considered was the Support Vector Machine model, as referenced in the data [1]. Among the Linear models, we evaluated Linear Regression and Linear Discriminant Analysis. In the Ensemble models, we considered Random Forest Classifier and ADABoost models. For boosting models, we evaluated XGBoost and HistGradientBoosting models. Chapter 4 provides detailed results of all the models, and XGBoost stood out as the best-performing model. Consequently, we utilized XGBoost to create the web application using Flask.

6.2 Future work

Despite the significant contributions made in handling the numerous categorical and numeric variables in the bank marketing campaign dataset, there are still a few limitations in this research paper. As discussed in Chapter 4, the paper primarily focuses on machine learning models. In the future, it would be worthwhile to explore other avenues, such as cross-platform and deep learning models, to further evaluate and predict future term deposits. These additional approaches could potentially enhance the accuracy and insights derived from the dataset.

Chapter 7

Reflection

During the bank marketing campaign project, I had the opportunity to delve into the world of data analysis and machine learning, and it has been an enlightening journey. Working on this project allowed me to apply various data preprocessing techniques and explore different machine learning algorithms to predict the term deposit utilizing the customer responses.

The process of data cleaning and preprocessing was crucial in ensuring the dataset's quality and reliability. Handling missing values, encoding categorical variables, and balancing class distribution helped in preparing the data for modeling. The comparison between linear models, ensemble methods, and boosting models showcased their varying performances, with some algorithms standing out in accuracy and predictive power.

Overall, this project has been a valuable learning experience, and I have gained confidence in applying data science techniques to real-world problems. It has sparked my interest in further exploring machine learning and its applications in the financial industry. As I reflect on this project, I am excited to continue honing my skills and contributing to data-driven decision-making processes in the future.

References

[1] Aditi, D. [2023], 'aditidadariya / bankmarketingcampaign', `https://github.com/aditidadariya/BankMarketingCampaign`. (accessed regularly).

[2] Moro, S., R. P. and Cortez, P. [2012], 'Bank Marketing', UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C5K306.

[3] Pan, Y. and Tang, Z. [2014], Ensemble methods in bank direct marketing, *in* '2014 11th International Conference on Service Systems and Service Management (ICSSSM)', pp. 1–5.

[4] Saeed, S. E., Hammad, M. and Alqaddoumi, A. [2022], Predicting customer's subscription response to bank telemarketing campaign based on machine learning algorithms, *in* '2022 International Conference on Decision Aid Sciences and Applications (DASA)', pp. 1474–1478.

[5] Subramanian, M., Bhukya, S. N., Vijaya Prakash, R., Raju, K. N., Ray, S. and Pandian, M. [2023], Deploy machine learning model for effective bank telemarketing campaign, *in* '2023 International Conference on Distributed Computing and Electrical Circuits and Electronics (ICDCECE)', pp. 1–6.