BANK MARKETING CAMPAIGN

Team 2: Project Report

ABHINAYA KUMAR SINGAMPALLI

ATHARVA BHANAGAY

YASVANTH PAMIDI

DEVA SAI HRITVIK BOLLEPALLI

VINAY REDDY POREDDY

Table of Contents

1. Introduction--- 2

2. Data Description--- 2

3. Literature Review-- 4

4. Data Pre-Processing-- 5

5. Data Exploration--- 7

6. Predictive Modeling--- 9

7. Business Recommendations-- 11

8. References-- 12

9. Appendix-- 13

Introduction:

Our project involves working with the 'Bank Marketing Dataset' from Kaggle. The dataset pertains to a Portuguese banking institution's recent campaign to encourage their customers to enroll in a term deposit program. However, this campaign involved reaching out to customers via phone calls, which required significant resources, time, and costs for the bank.

In order to optimize their resources and save time and costs, the bank is interested in identifying which customers are most likely to enroll in the term deposit program. Our project aims to analyze the dataset and develop a predictive model that can help the bank to identify the most promising customers for the campaign.

By leveraging the power of machine learning algorithms, we hope to create a model that can accurately predict which customers are most likely to enroll in the term deposit program. This will help the bank to optimize their marketing strategy and maximize the effectiveness of their outreach efforts.

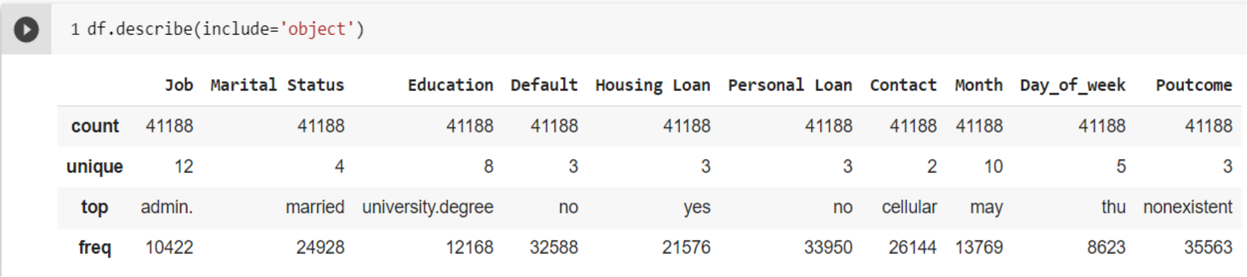
Data Description:

The dataset has 20 columns and 41188 rows out of which 10 variables are categorical and 10 variables are continuous. The target variable here is labeled as 'Result' which says whether the customer has enrolled for the term deposit or not.

Variable	Type	Description
Age	Continuous	Age of the customer
Job	Nominal	The job of the customer
Marital Status	Nominal	Marital status of the customer
Education	Nominal	Educational background of the customer
Default	Nominal	Has the customer had credit in default?
Housing Loan	Nominal	Has the customer had a housing loan?
Personal Loan	Nominal	Has the customer had a personal loan?
Contact	Nominal	Medium of communication with the customer
Month	Nominal	The month in which the customer was last contacted

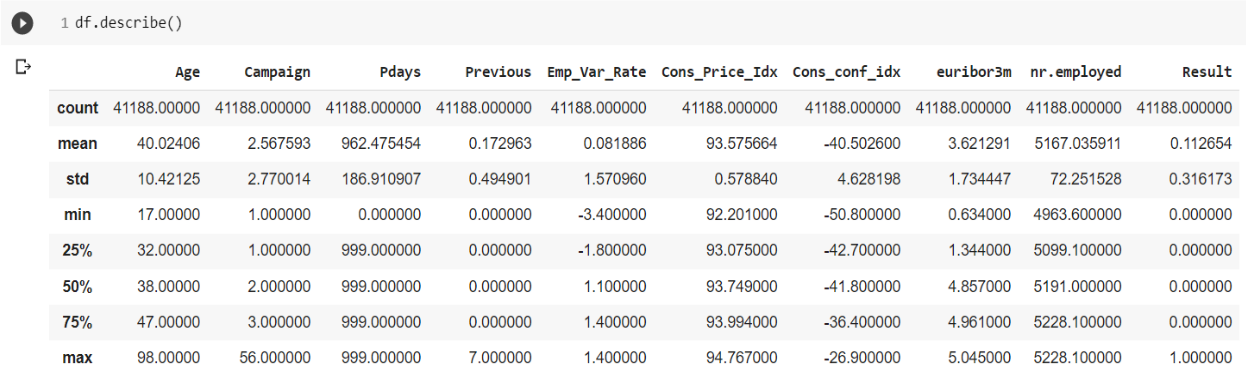
Day_of_Week	Nominal	Last contacted day of the week
Campaign	Continuous	No. of contacts done with the customer in this campaign
Pdays	Continuous	No. of days passed from the last day of contact in this campaign
Previous	Continuous	No. of contacts performed before this campaign for this month
Poutcome	Nominal	The outcome of the previous marketing campaign
Emp.Var.Rate	Continuous	Employment variation rate - quarterly indicator
Cons.Price.Idx	Continuous	Consumer price index - monthly indicator
Cons.Conf.Idx	Continuous	Consumer confidence index - monthly indicator
Euribor3M	Continuous	Euribor 3-month rate - daily indicator
Nr.Employed	Continuous	No of employees - quarterly indicator
Result	Nominal	Has the client subscribed to the term deposit?

The data description of all the variables is as below:



```
df.describe(include='object')
```

	Job	Marital Status	Education	Default	Housing Loan	Personal Loan	Contact	Month	Day_of_week	Poutcome
count	41188	41188	41188	41188	41188	41188	41188	41188	41188	41188
unique	12	4	8	3	3	3	2	10	5	3
top	admin.	married	university.degree	no	yes	no	cellular	may	thu	nonexistent
freq	10422	24928	12168	32588	21576	33950	26144	13769	8623	35563



```
df.describe()
```

	Age	Campaign	Pdays	Previous	Emp_Var_Rate	Cons_Price_Idx	Cons_conf_idx	euribor3m	nr.employed	Result
count	41188.00000	41188.000000	41188.000000	41188.000000	41188.000000	41188.000000	41188.000000	41188.000000	41188.000000	41188.000000
mean	40.02406	2.567593	962.475454	0.172963	0.081886	93.575664	-40.502600	3.621291	5167.035911	0.112654
std	10.42125	2.770014	186.910907	0.494901	1.570960	0.578840	4.628198	1.734447	72.251528	0.316173
min	17.00000	1.000000	0.000000	0.000000	-3.400000	92.201000	-50.800000	0.634000	4963.600000	0.000000
25%	32.00000	1.000000	999.000000	0.000000	-1.800000	93.075000	-42.700000	1.344000	5099.100000	0.000000
50%	38.00000	2.000000	999.000000	0.000000	1.100000	93.749000	-41.800000	4.857000	5191.000000	0.000000
75%	47.00000	3.000000	999.000000	0.000000	1.400000	93.994000	-36.400000	4.961000	5228.100000	0.000000
max	98.00000	56.000000	999.000000	7.000000	1.400000	94.767000	-26.900000	5.045000	5228.100000	1.000000

Most of the banking customers are married and don’t have credit default in the past and were contacted through cellular. Most of the calls to customers were made in May. The average age of the customers is 40 and 75% of the customers are below the age of 47. Our target variable is a binary variable with 0 & 1. There are 11% of 1’s which is our class of interest and 89% of 0’s available in the target column ‘Result’.

Literature Review:

After analyzing and understanding the data set using the data dictionary, we have gone through the tasks that are done by other people in the Kaggle. We found a few tasks and techniques that they followed during the preprocessing and modeling steps. We chose to explore and implement other new techniques and tried to build a machine-learning model with better accuracy. The tasks performed by others and we were compared below.

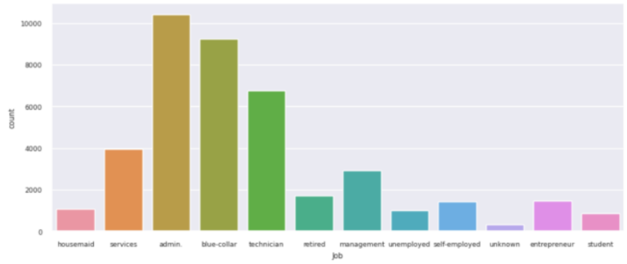
Other Did	We Did
Used One Hot Encoder to convert categorical variables to continuous.	Used Label Encoder to reduce the model complexity.
Used shorter quantile ranges and intervals for outlier analysis.	Used wider quantile range and intervals for outlier analysis.
Tried only SVM and Logistic Regression models.	Fitted many other models and achieved better accuracy.
The age column is taken as continuous.	The age column was transformed to categorical by data binning.

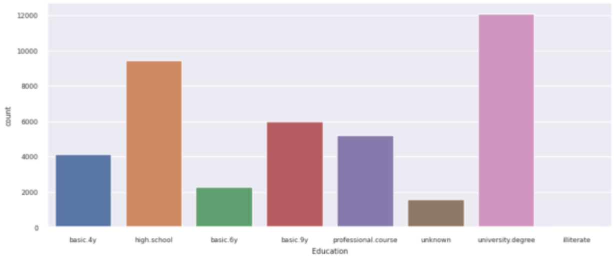
- We used Label Encoder to avoid the generation of dummy variables and further reduce the model complexity.
- We used a wider quantile range and intervals for outlier analysis so that we will be able to retain more data.
- We tried many models and evaluated the performance of each model and achieved better accuracy.
- We created a new column by binning the customer's Age to explore if there is any trend with the result column.

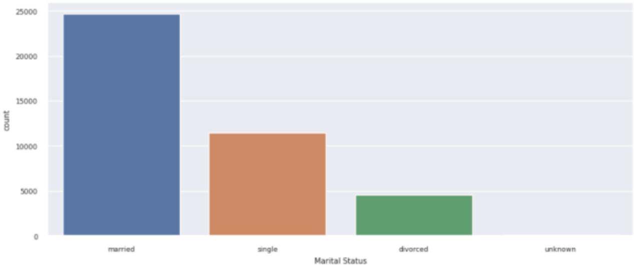
Data Pre-processing:

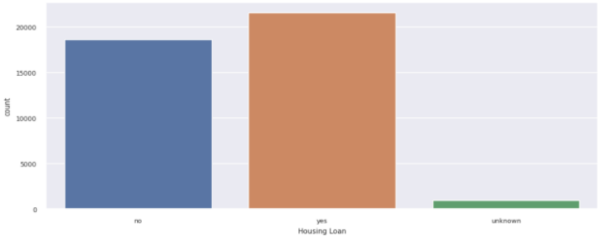
Missing Values:

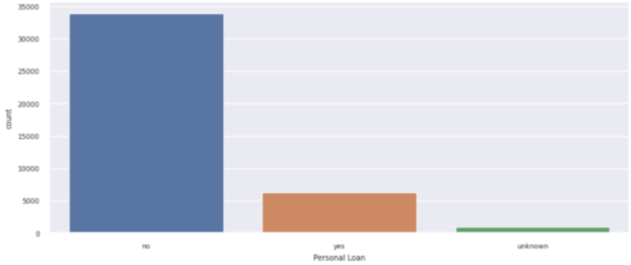
The dataset didn’t have any blank columns. But in a few columns, there is a value labeled ‘unknown’ in the dataset. The variables Job, Education, Marital Status, Housing Loan, and Personal Loan are having the value ‘unknown’ in the column. Considering this value as a missing value, the rows containing the value ‘unknown’ are eliminated from the dataset. Under this process of treating the missing values, 2961 rows were eliminated.





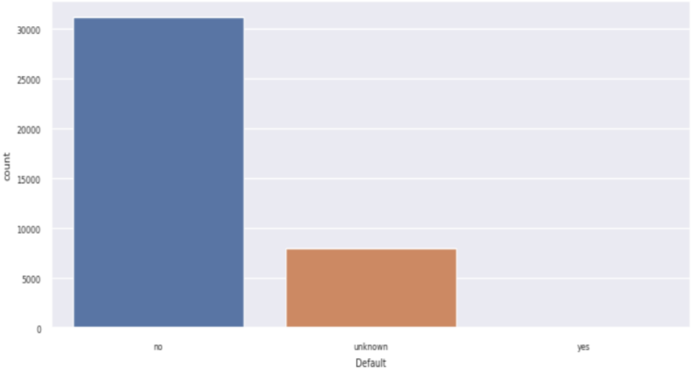




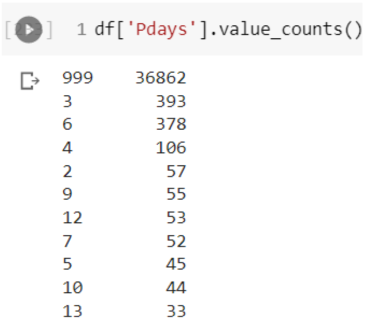


Column Elimination:

In the default column, there are 21% of total rows with value ‘unknown’ and only 3 rows with value ‘yes’, and all the others are having the value ‘no’. But if we remove both ‘unknown’ and ‘yes’ rows, there will be a constant value left in the column which is ‘no’ and that doesn’t have any impact on the target variable. So, we have eliminated the ‘default’ column.



The column Pdays has the 96% of rows with the value ‘999’ which we assumed can be a default value written in place of a missing value. So, as the column has 96% of rows missing, we decided to remove the column Pdays.

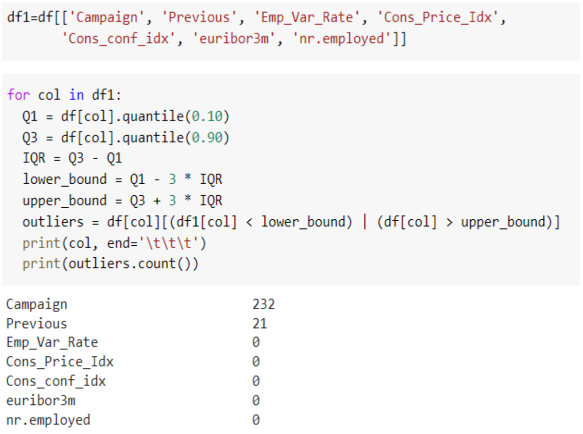


```
df['Pdays'].value_counts()
```

```
999     36862
3         393
6         378
4         106
2          57
9          55
12         53
7          52
5          45
10         44
13         33
```

Outlier Analysis:

We have performed outlier analysis for all the numerical variables using the IQR (Inter Quantile Range) method with a lower quantile range of 10%, an upper quantile range is 90%, and 3 tolerable intervals. By performing the outlier analysis, we found a total of 253 outliers with 232 outliers in the campaign column and 21 outliers in the previous column. All these 253 rows with outliers were removed from the dataset.



```
df1=df[['Campaign', 'Previous', 'Emp_Var_Rate', 'Cons_Price_Idx',
       'Cons_conf_idx', 'euribor3m', 'nr.employed']]
```

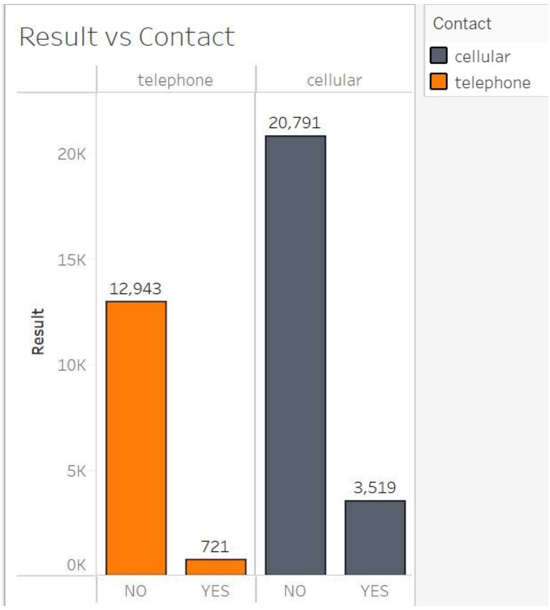
```
for col in df1:
  Q1 = df[col].quantile(0.10)
  Q3 = df[col].quantile(0.90)
  IQR = Q3 - Q1
  lower_bound = Q1 - 3 * IQR
  upper_bound = Q3 + 3 * IQR
  outliers = df[col][(df1[col] < lower_bound) | (df[col] > upper_bound)]
  print(col, end='\t\t\t')
  print(outliers.count())
```

```
Campaign                    232
Previous                    21
Emp_Var_Rate                0
Cons_Price_Idx              0
Cons_conf_idx               0
euribor3m                   0
nr.employed                 0
```

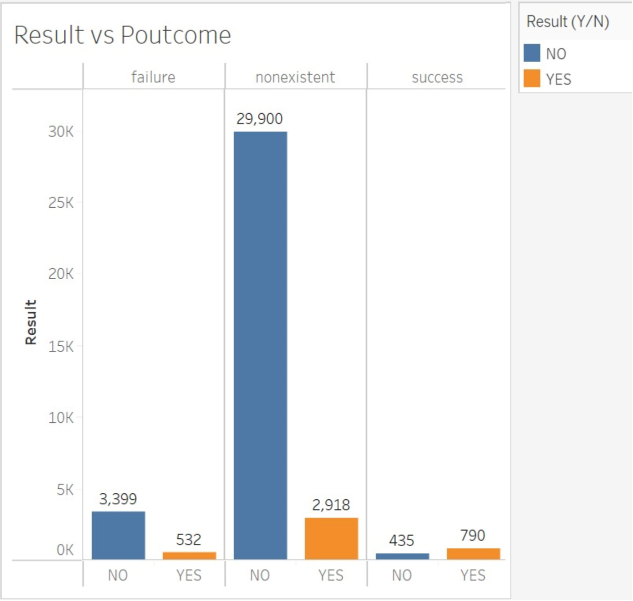
After completing the data preprocessing and binning the Age column in a column named ‘Age Binned’, there are a total of 37974 rows and 18 columns left in the dataset.

Data Exploration:

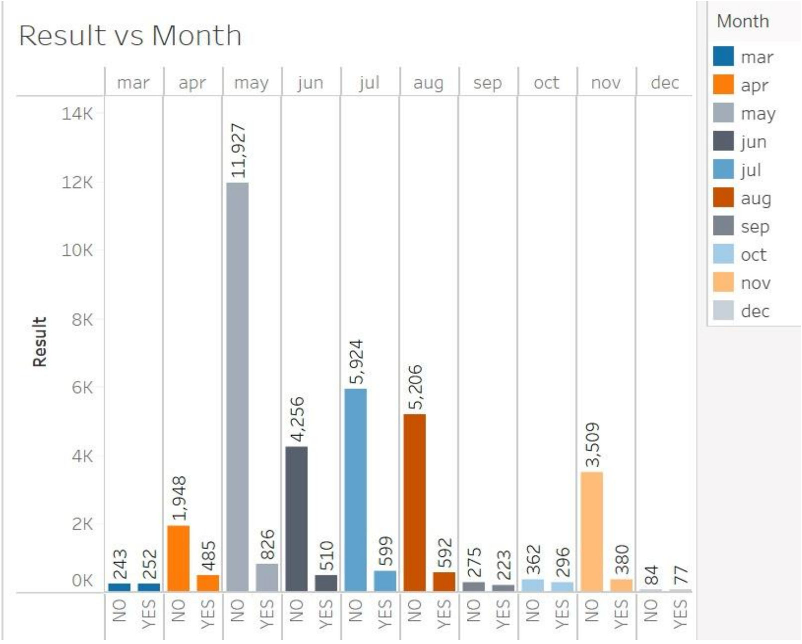
- When the type of contact is plotted concerning the result column, we can see that the probability of customers enrolling for the term deposit is ~3 times higher when contacted via cellular than when contacted through telephone.



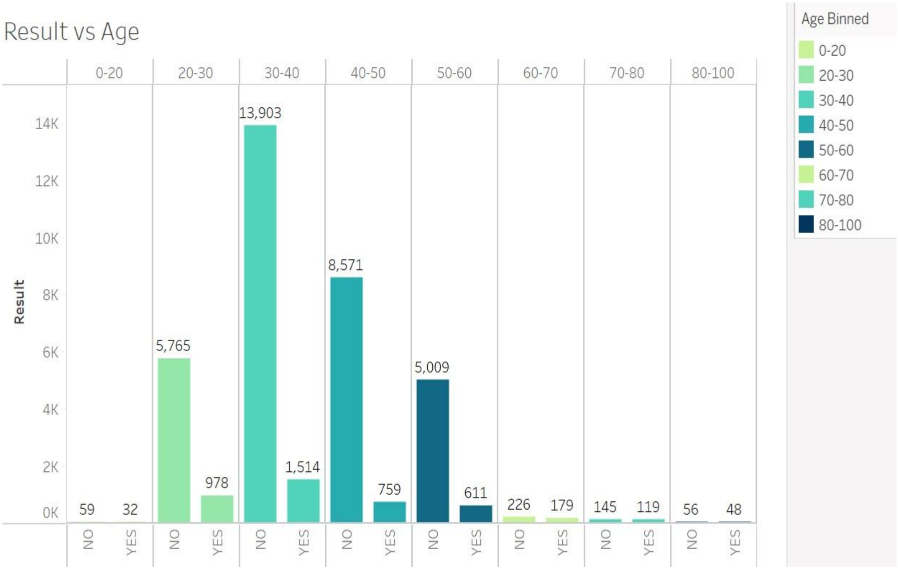
- When Poutcome is plotted concerning the result column, There are high chances of customers enrolling in a term deposit in this campaign when the customers have enrolled for it in the previous campaign.



- When Month is plotted concerning the result column, The probability of customers enrolling to the term deposit when contacted in months March, September, October, and December is ~50%.



- When the Age Binned column is plotted concerning the result column, The chance of people enrolling in term deposits with an age above 50 is higher when compared to people with an age below 50.



Predictive Modeling:

As our target variable is a categorical type, we chose 4 models to fit and evaluate their performance and finalize the best-performing model. The four models that we chose to fit and evaluate are as below:

1. Logistic Regression Model
2. Decision Tree Model
3. Random Forest Model
4. Gradient Boosting Model

Data Transformation:

As there are 10 categorical variables and these categorical variables are to be converted to numerical variables, we chose to convert them using the ‘Label Encoder’ instead of using ‘get_dummies’ or ‘One Hot Encoder’ functions. When we use ‘Label Encoder’, it will not create dummy columns as the other functions do. It will recode the columns by assigning a unique integer to each unique value. This will help in reducing the model complexity by not adding new dummy columns.

Data Split:

From the final dataset, the predictors and the target variables are separated and those were split into training and test data sets with 80% of the data as training to train the models.

Data Standardization:

The data standardization technique was applied to the data using the standard scalar function. Both the training and tests of the predictors are standardized with this technique.

Modeling:

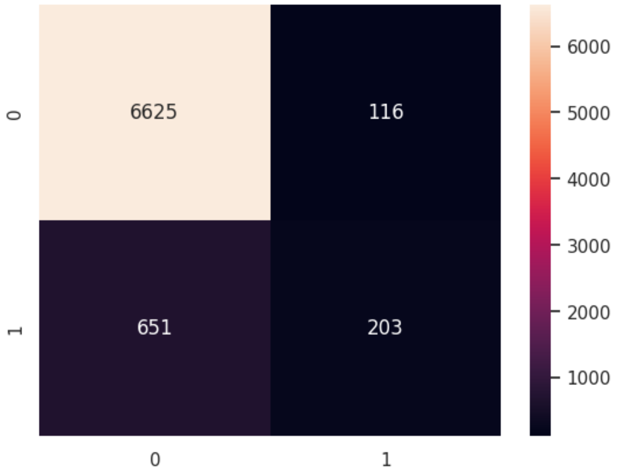
All the models that are finalized above are built and their performance metrics were compared to evaluate and select the best-performing model. The model performance metrics comparison of all the models is as below:

Model	Total Accuracy	Precision	Lift	AUC
Logistic Regression	90%	64%	5.7	0.7720
Decision Tree	85%	31%	2.76	0.6214
Random Forest	89%	53%	4.72	0.7704
Gradient Boosting	90%	64%	5.7	0.7877

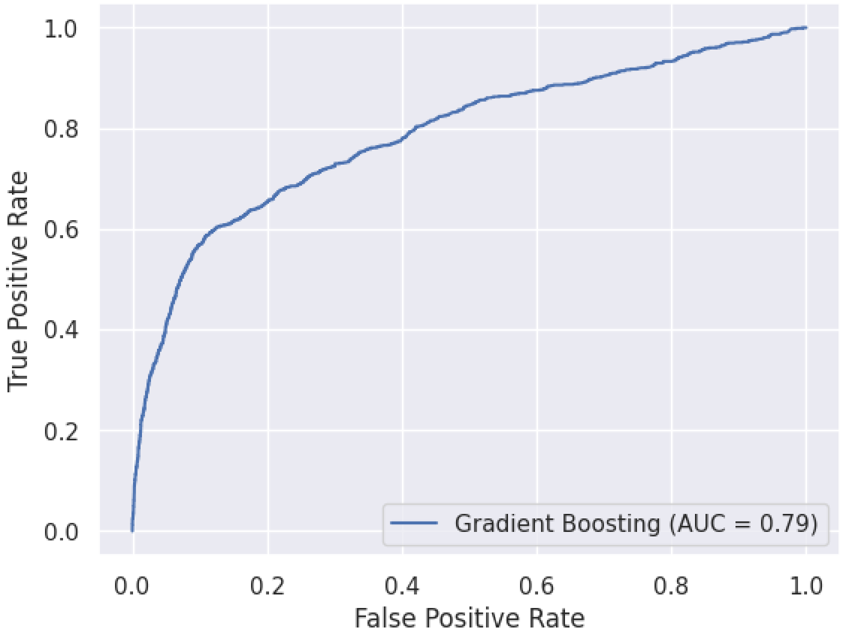
From the above metrics, it is evident that the 'Gradient Boosting' model outperforms all the other models with higher Accuracy, Precision, Lift & AUC values. For the Gradient Boosting model, the accuracy we got is 90%, the precision we got is 64%, the lift we got is 5.7 and the AUC is 0.7877. Though there is not much difference in the performance metrics of the Logistic Regression and The Gradient Boosting model, the latter has a higher AUC which says that it is predicting more number of 1's in the target variable.

Also, to cross-check the consistency of the model on the whole training set, the cross-validation technique is applied to the training set for the Gradient Boosting model to verify the accuracy of the model. We used the number of splits as 10 and the mean of accuracy values of all those 10 models that are built is 0.902 with a standard deviation of 0.03. This says that our best model, the Gradient Boosting model is consistent over the training set.

The confusion matrix of our best model Gradient Boosting is as below:



The ROC curve of our best model Gradient Boosting is as below:



Business Recommendations:

From the above data exploration and predictive modeling, there are a few findings that better serve as business recommendations for this banking organization, and those recommendations are as below:

- We can see that the customers who have enrolled for term deposits in the previous campaign are more likely to be enrolled for the term deposit again. So, the banks can prioritize those customers.
- As the probability for customers to enroll for the term deposit is high when the customers were contacted in the month's March, September, October, and December, banks can plan to arrange more calls to customers during these months.
- The banks can contact the customer on cellular instead of telephone to contact the right person to whom the call was intended and convince them to enroll.

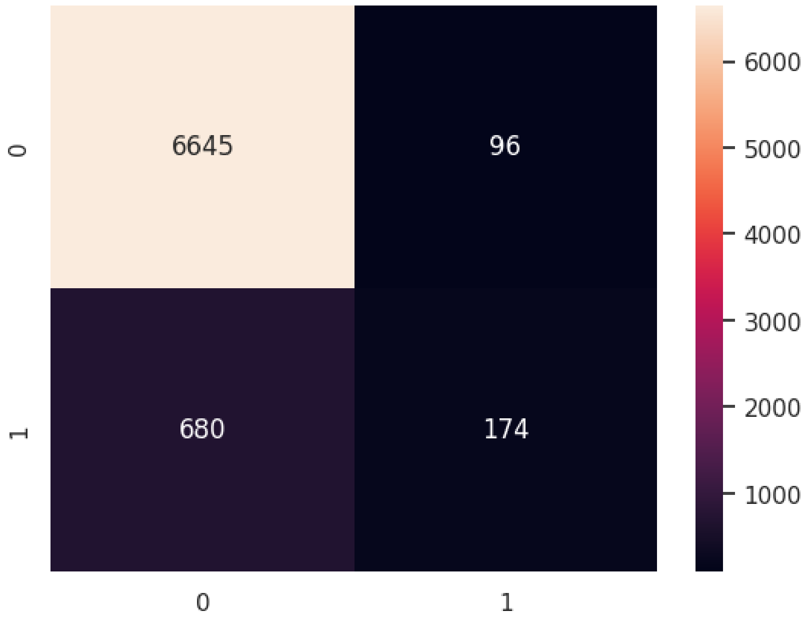
References:

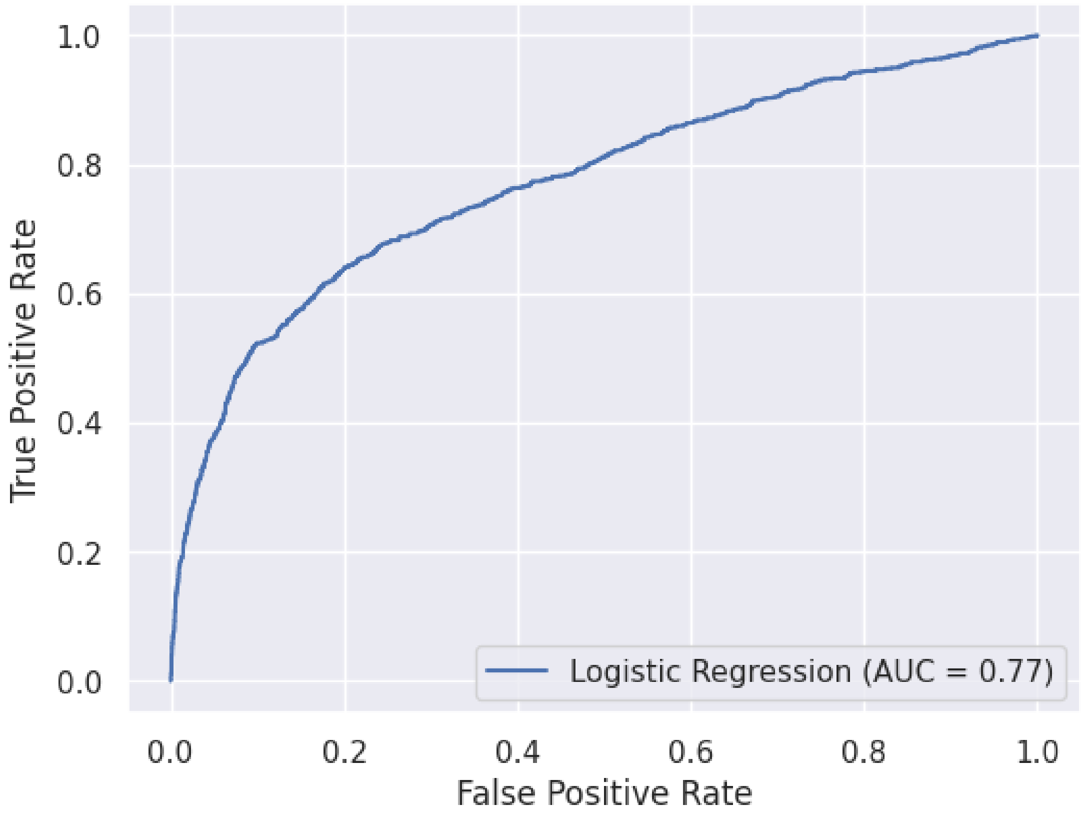
1. Kaggle: https://www.kaggle.com/datasets/ruthgn/bank-marketing-data-set/code
2. Scikit – learn: https://scikit-learn.org/stable/index.html
3. Data Science with Python, In-class colab files.
4. Predictive Modeling lectures.

Appendix:

Logistic Regression:

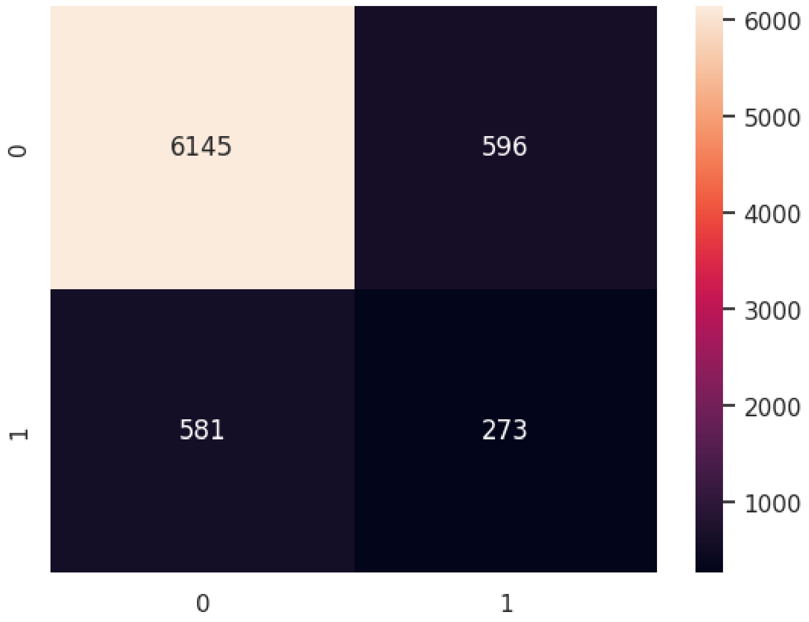
Accuracy: 81.78%

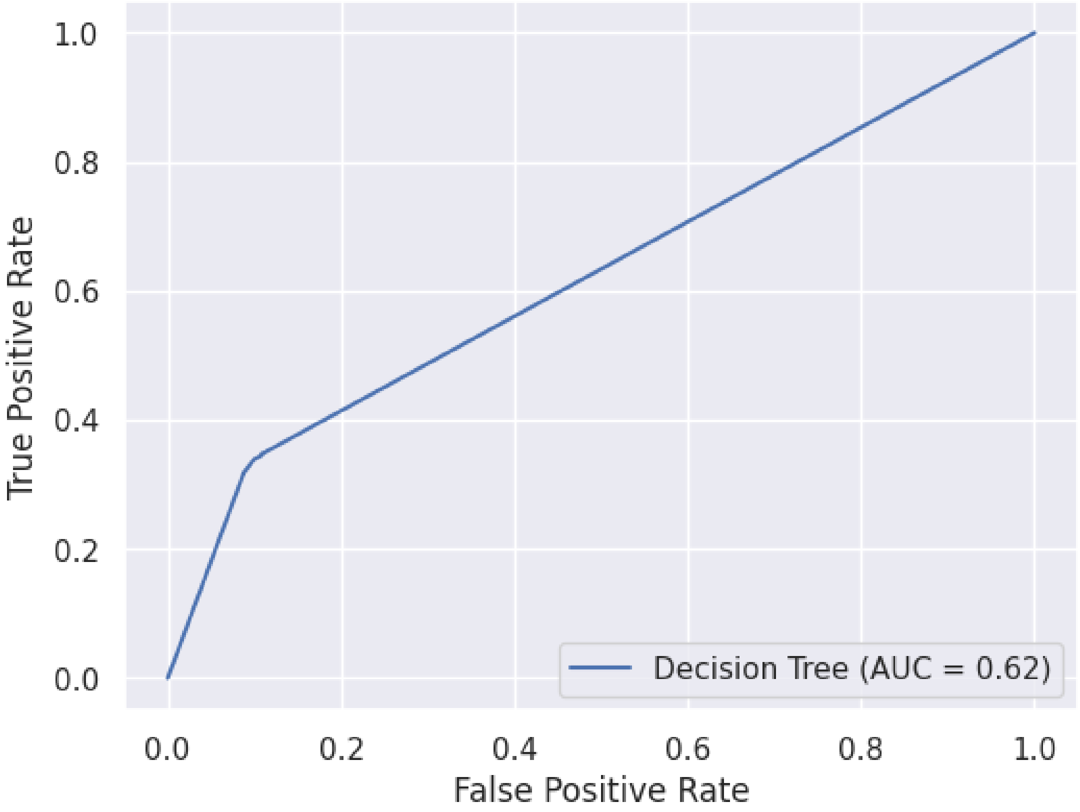




Decision Tree:

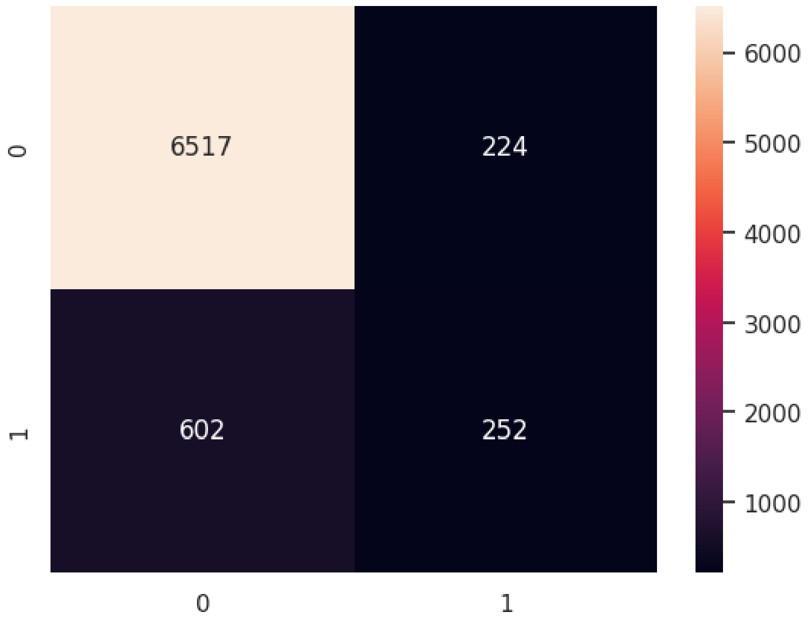
Accuracy: 84.5%

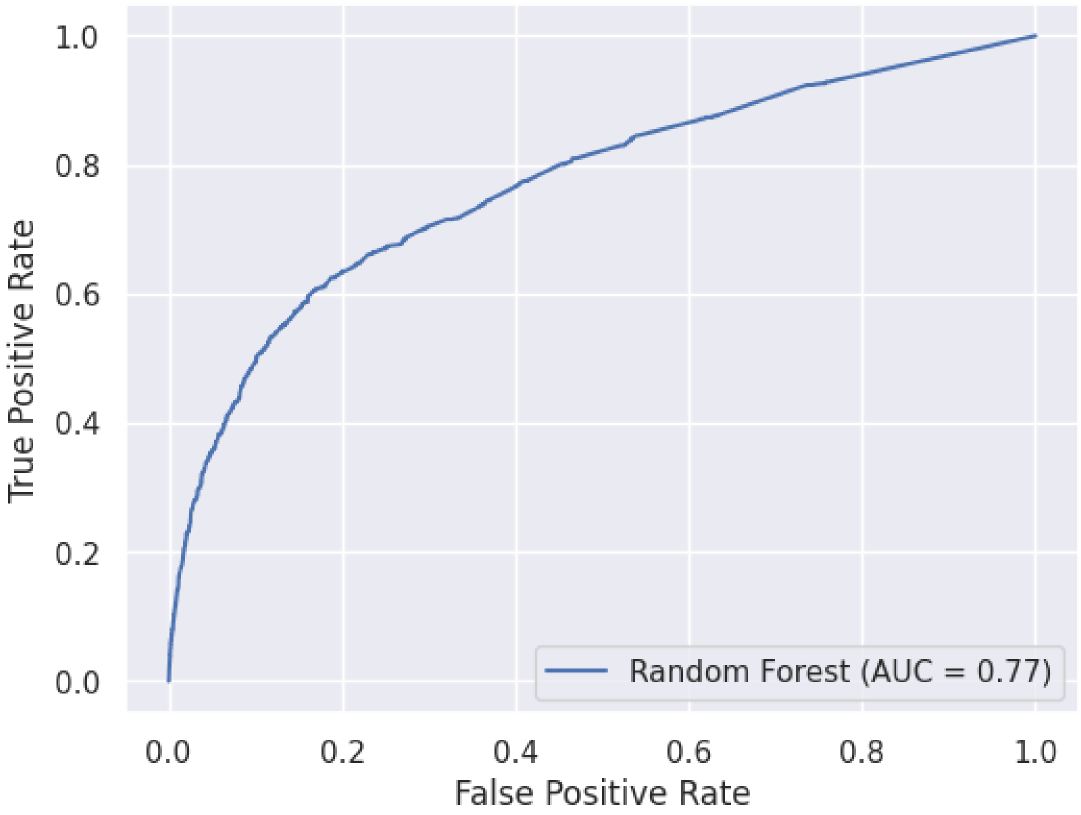




Random Forest:

Accuracy: 89.12%


6517
224
602
252
0
1
0
1
6000
5000
4000
3000
2000
1000


True Positive Rate
Random Forest (AUC = 0.77)
False Positive Rate