1. Business Problem:

- There has been a revenue decline for the Portuguese bank and they would like to know what actions to take. After investigation, they found out that the root cause is that their clients are not depositing as frequently as before. Knowing that term deposits allow banks to hold onto a deposit for a specific amount of time, so banks can invest in higher gain financial products to make a profit. In addition, banks also hold better chance to persuade term deposit clients into buying other products such as funds or insurance to further increase their revenues. As a result, the Portuguese bank would like to identify existing clients that have higher chance to subscribe for a term deposit and focus marketing effort on such clients.
- To resolve the problem, we suggest a classification approach to predict which clients are more likely to subscribe for term deposits.

2. Objective:

- The basic objective of the project is to analyse Tele-Marketing data collected from a Bank and predict whether the customer will subscribe the term deposit or not, also identify the driving factors behind this. This would inform the Bank's decisions on which Customers to target for their Marketing Campaign, which would ultimately increase their Product Sales. Understanding their customers is critical for effectively executing their Marketing campaign.

3. Dataset Information:

- This dataset contains 45211 rows and 17 columns, where each observation corresponds to a customer response. Among the total 45211 observations, 5289 observations (11.7%) are those who actually bought the product.

4. Data Dictionary:

- We are provided with Customer details such as Age, Job, nuptial, etc. also the campaign details such as Number of calls performed, No of _days passed after and before the campaign to better understand the Problem.
- Data can be seen from the following table:

Sl. No	Column Name	Description
1	age	Age of the targeted Customers
2	job	what does the customer do? (Type of jobs)
3	nuptial	married or not?
4	education	level of education of Customers

5	Defaulter_presence	if the customer in default list or not (has credit in default?)
6	balance_amount	remaining balance in their accounts
7	house_loan	has house loan ?
8	Personal_loan	has personal loan?
9	contact_via	contact communication type
10	last_contact_day	last contact day of the week
11	last_contact_month	last contact month of year
12	duration	last contact duration, in seconds (numeric). Important note: this attribute highly affects the output target (e.g., if duration=0 then y="no"). Yet, the duration is not known before a call is performed. Also, after the end of the call y is obviously known. Thus, this input should only be included for benchmark purposes and should be discarded if the intention is to have a realistic predictive model.
13	campaign	number of contacts performed during this campaign and for this client
14	Passed_days	number of days that passed by after the client was last contacted from a previous campaign
15	Previous_contact	number of contacts performed before this campaign and for this client
16	poutcome	outcome of the previous marketing campaign
17	y	Response - Target column

5. <u>Findings and Implications in Outset Data:</u>

	age	balance_amount	last_contact_day	duration	campaign	passed_days	previous_contact
count	45211.000000	45211.000000	45211.000000	45211.000000	45211.000000	45211.000000	45211.000000
mean	40.936210	1362.272058	15.806419	258.163080	2.763841	40.197828	0.580323
std	10.618762	3044.765829	8.322476	257.527812	3.098021	100.128746	2.303441
min	18.000000	-8019.000000	1.000000	0.000000	1.000000	-1.000000	0.000000
25%	33.000000	72.000000	8.000000	103.000000	1.000000	-1.000000	0.000000
50%	39.000000	448.000000	16.000000	180.000000	2.000000	-1.000000	0.000000
75%	48.000000	1428.000000	21.000000	319.000000	3.000000	-1.000000	0.000000
max	95.000000	102127.000000	31.000000	4918.000000	63.000000	871.000000	275.000000

	count	unique	top	freq
job	45211	12	blue-collar	9732
nuptial	45211	3	married	27214
education	45211	4	secondary	23202
defaulter_presence	45211	2	no	44396
house_loan	45211	2	yes	25130
personal_loan	45211	2	no	37967
contact_via	45211	3	cellular	29285
last_contact_month	45211	12	may	13766
poutcome	45211	4	unknown	36959
y	45211	2	no	39922

- There are 45211 distinct id's in the dataset which means 45211 Account holders and the average Bank Balance amount is 1362.2 with std. 3044.7 where the highest amount is 102127.0
- Account holders age ranges from 18 to 95 and the average age is 40.93 with standard deviation 10.61
- Target column which includes two distinct values Yes and No
- There are total of 17 columns and 45211 records in the dataset and none of the columns has null values.
- There are 7 continuous columns and 10 categorical columns in the dataset.
- Though there are no null values in the dataset there were some unknown classified records which were are being pre-processed before building the base model.
- The Target in the data has a large imbalance with respect to class 'Yes' (Minority Class)
- Resampling methods needs to be applied to increase the percent of Defaulter_presence in order to achieve better model training results.
- We created new column Age group On the basis of Age

- Age group:
- 10 - 19 = 1
- 20 - 29 = 2
- 30 - 39 = 3
- 40 - 49 = 4
- 50 - 59 = 5
- 60 - 69 = 6
- 70 - 79 = 7
- 80 - 89 = 8

- 90 - 99 = 9

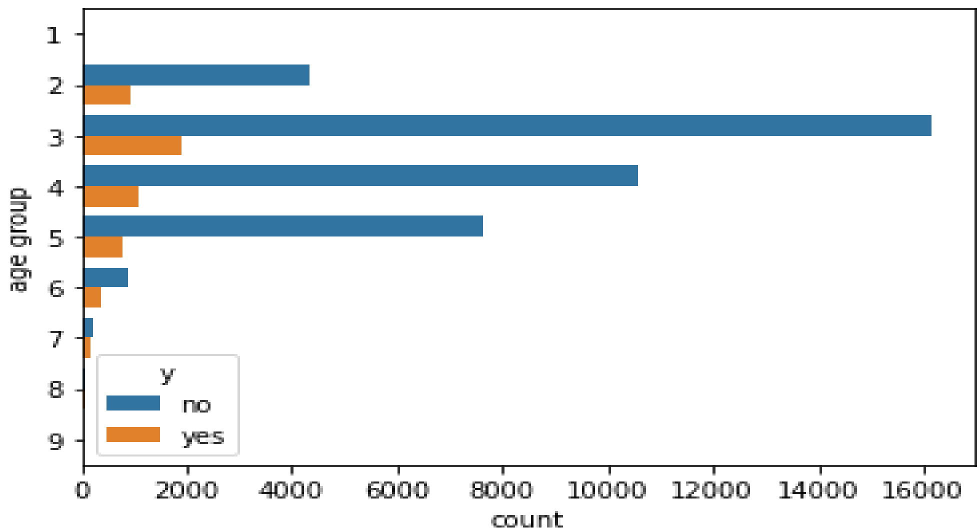
- There are no null values in the dataset.

```
age                   0
job                   0
nuptial               0
education             0
defaulter_presence    0
balance_amount        0
house_loan            0
personal_loan         0
contact_via           0
last_contact_day      0
last_contact_month    0
duration              0
campaign              0
passed_days           0
previous_contact      0
poutcome              0
y                     0
dtype: int64
```

6. Data Visualizations:

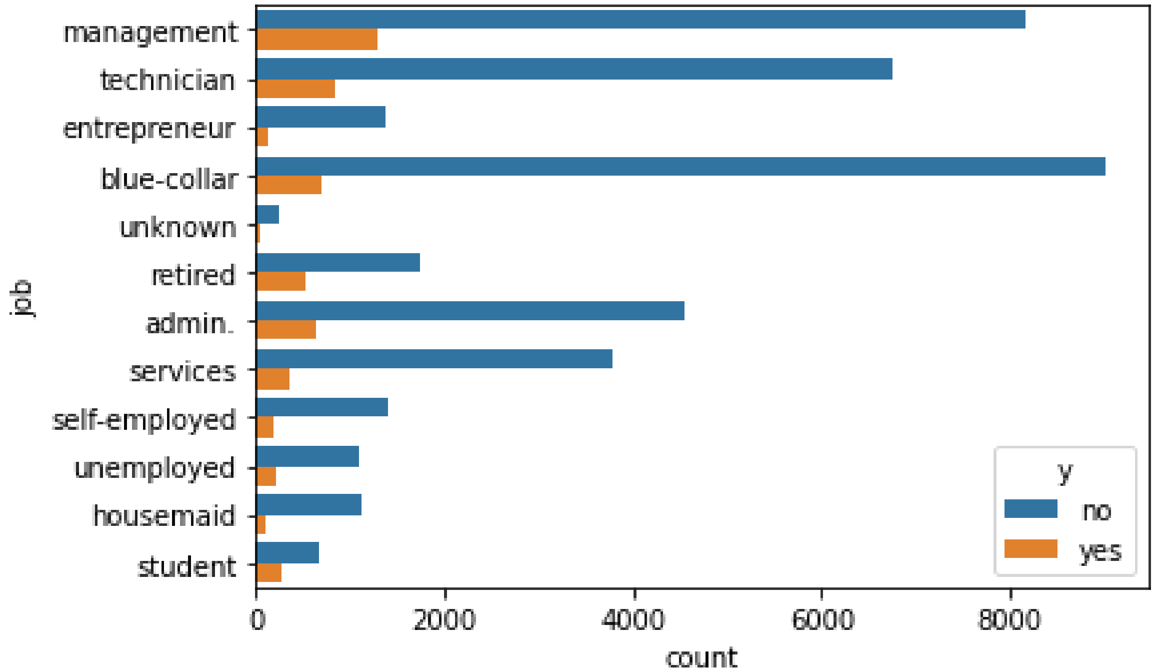
- **Categorical Variables & Target:**

Figure Age group Vs y (target variable)



Inferences: Customers in age group 3 are more likely to subscribe the Term deposit. Because they age range from 31-40, who can be professional with high paying jobs and this possibilities are high because of their experience.

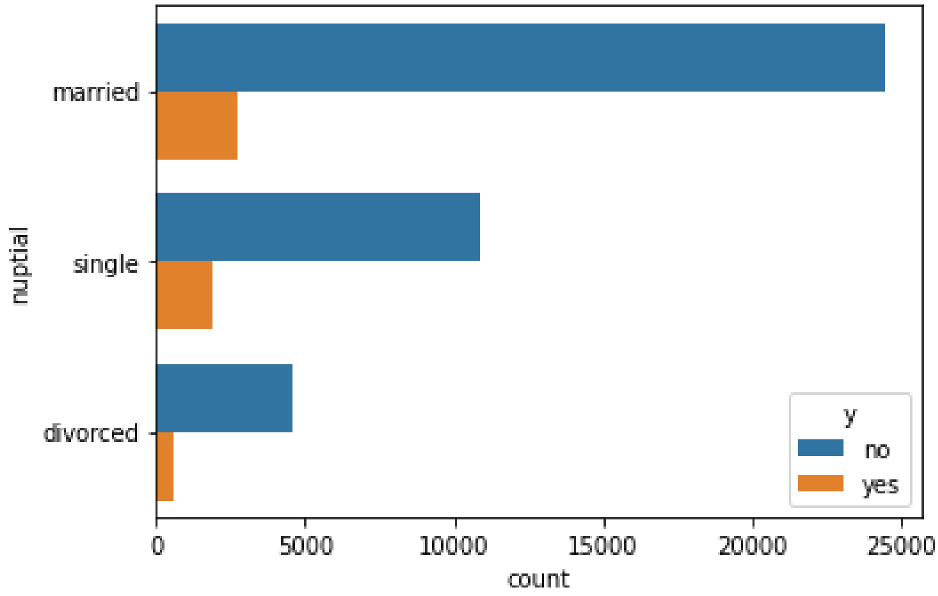
Figure Job Vs Target



Inferences: Customers with "Management", "Technician", and "Blue-collar" job categories are more likely to subscribe the Term deposit. It is because these professionals have regular income.

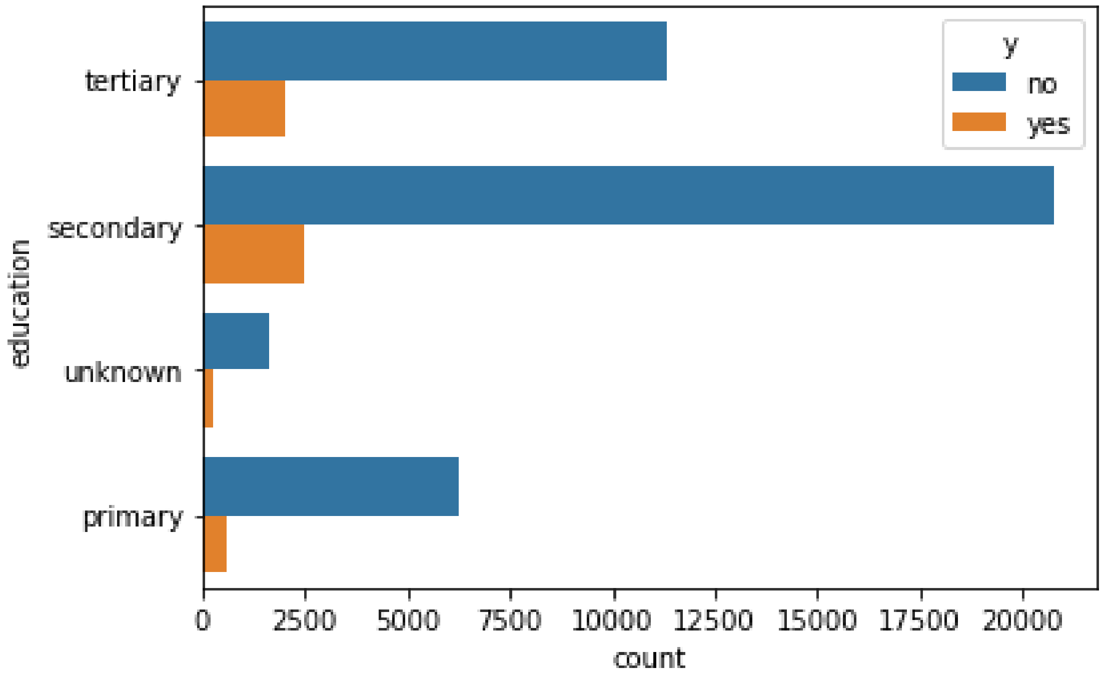
Other category workers may or may not have the regular income, which includes student (only few students work for part time jobs), housemaid (their earnings very less used to lead the normal life) etc..

Figure: nuptial Vs Target



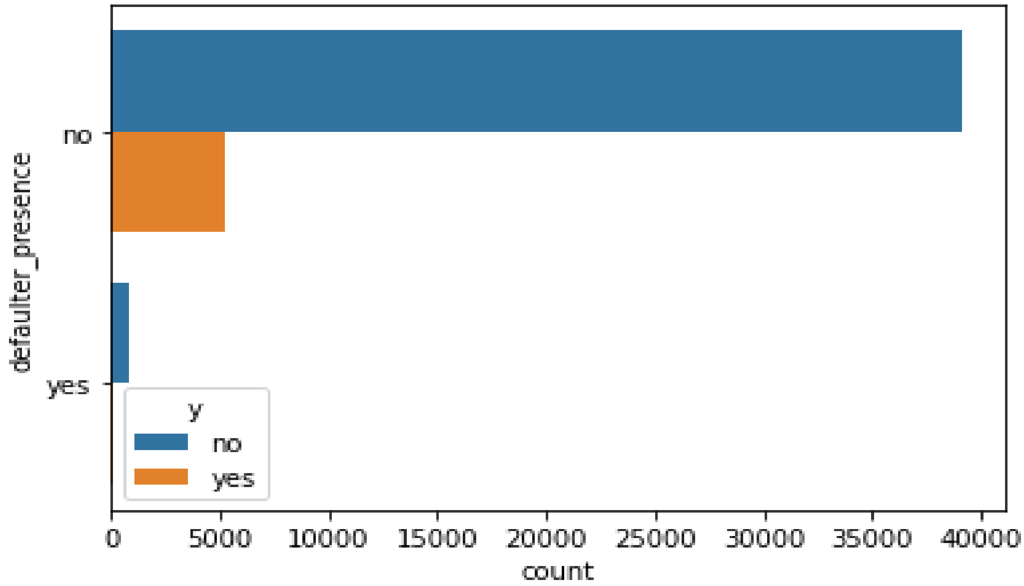
Inferences: Married Customers are more likely to subscribe the Term deposit, they saves money for their children education, marriage etc.. it is a long term investment.

Figure Education Vs Target



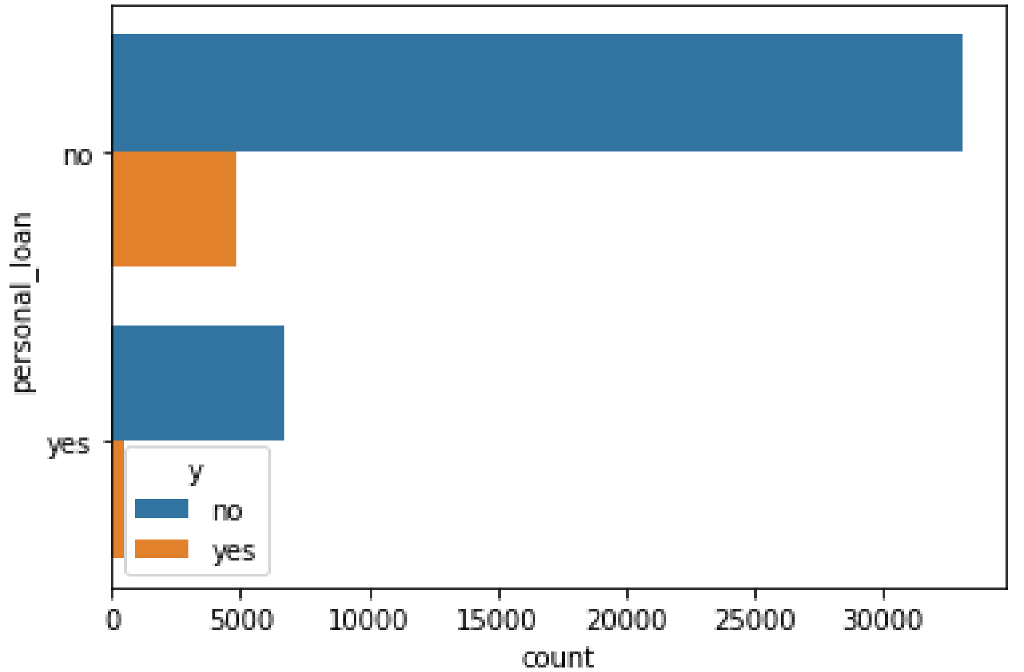
Inferences: Customers with "Secondary" & Tertiary" educations are more likely to subscribe the Term deposit, , it is mainly due to they have clarity about financial education and knows the value of long term investments.

Figure : Defaulter_Presence Vs Target Variable (y)



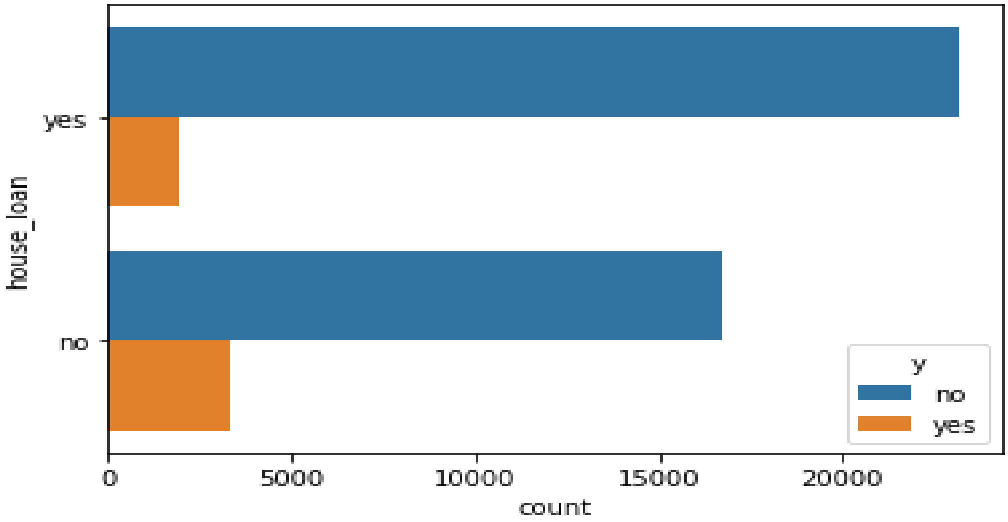
Inferences: those customers who are not Defaulter_presence are likely to get term deposit as we can see there are more than 5000 customer who are likely to get term deposit

Figure personal_loan Vs Target



- **Inference:** From the above observation we came to know that those who have not applied for personal_loan are likely to subscribe the term deposit plan as compared to those who have applied to personal_loan.

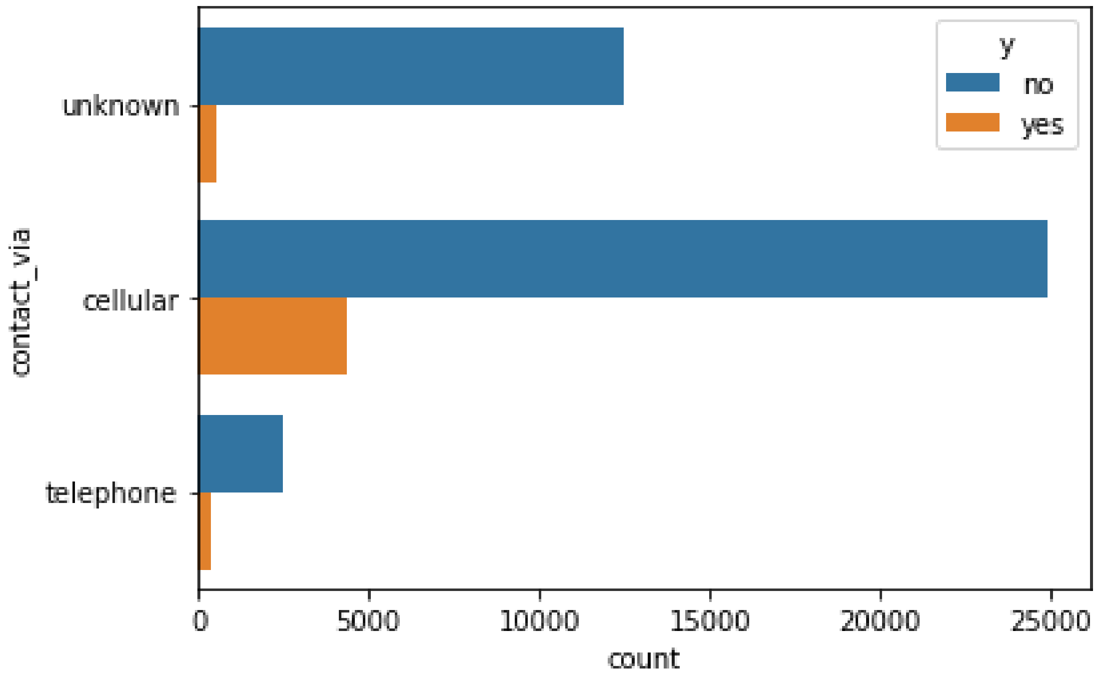
Figure: house_loan Vs Target



Inference: we came to know that those customers who have not applied for house loan are more likely to subscribe for term deposit as compared to those who have applied for house loan.

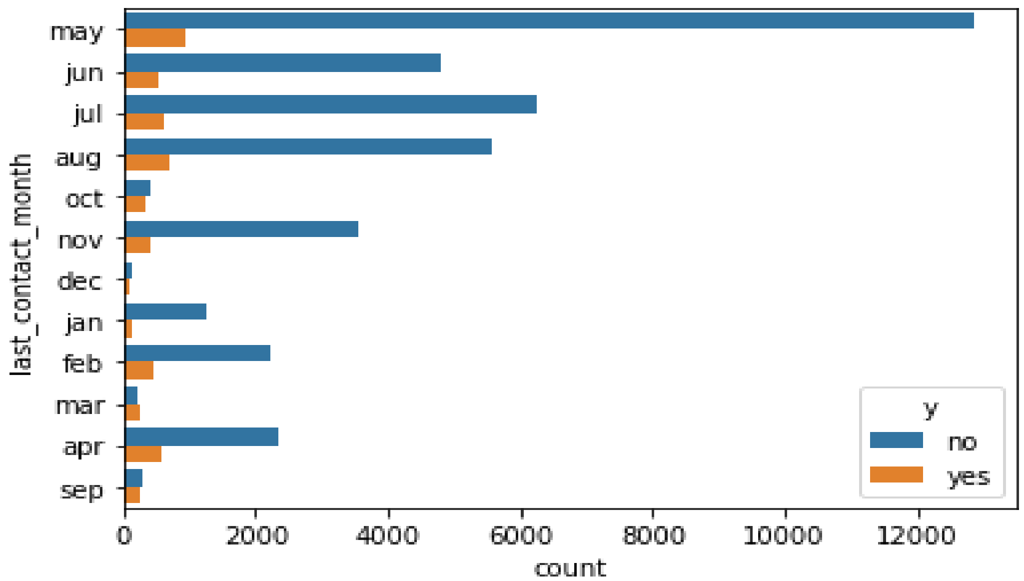
From the above data of personal loans, we can infer that those who have applied for any type of personal_loan may not subscribe for term deposit

Figure: contact_via vs Target Variable (y)



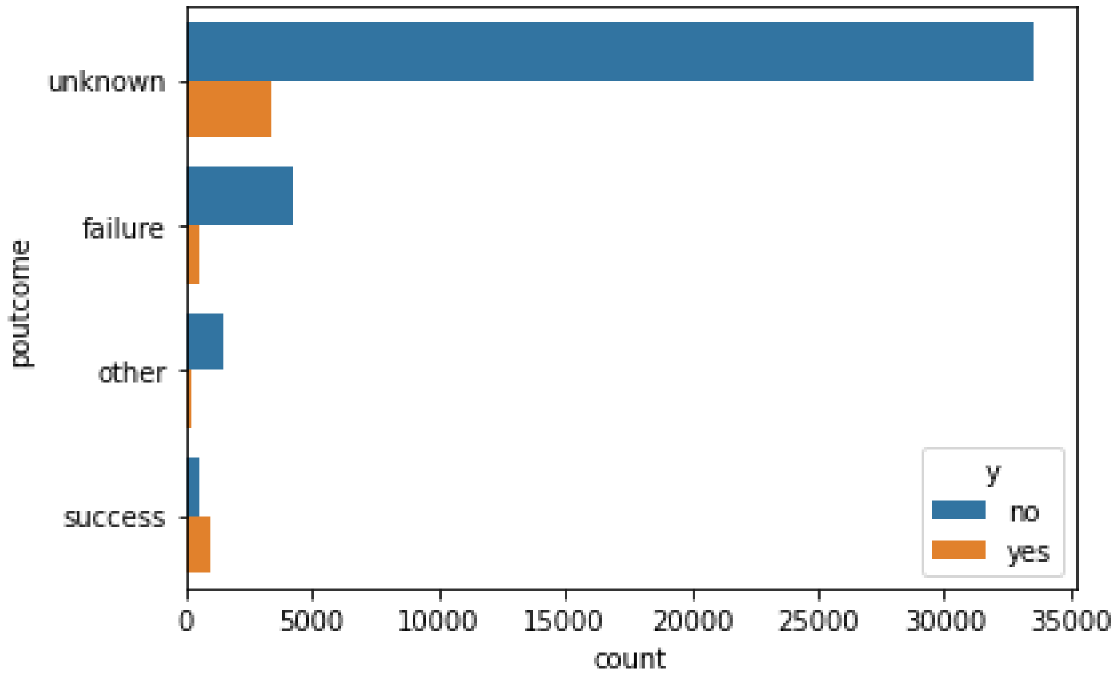
Inference: from the above observation we came to know that those customers who are approached by cellular device are likely to subscribe in term deposit.

Figure: Last._contact_month vs target variable (Y)



Inference: Those who get contacted during may last_contact_month are likely to subscribe in term deposit.

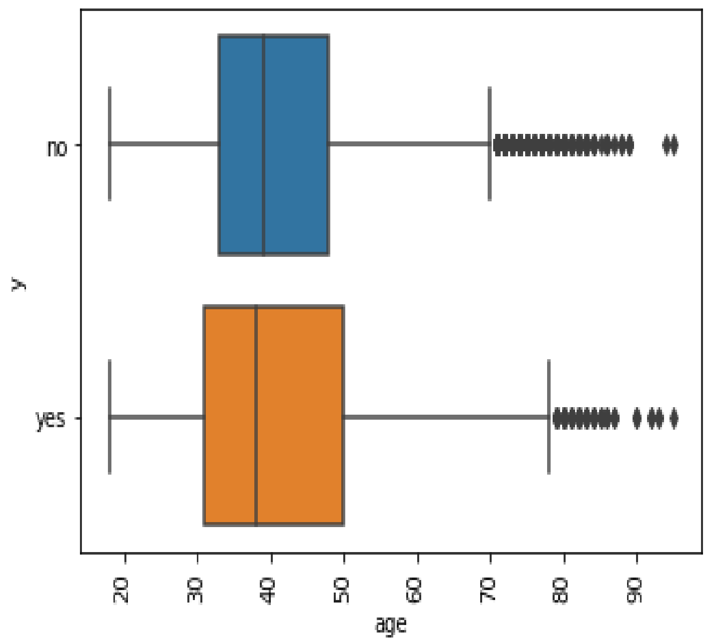
Figure: Poutcome vs target variable (Y)



Inference: from this campaign we came to know that unknown customers whose details are not given.They may be have chances to take term deposit in future

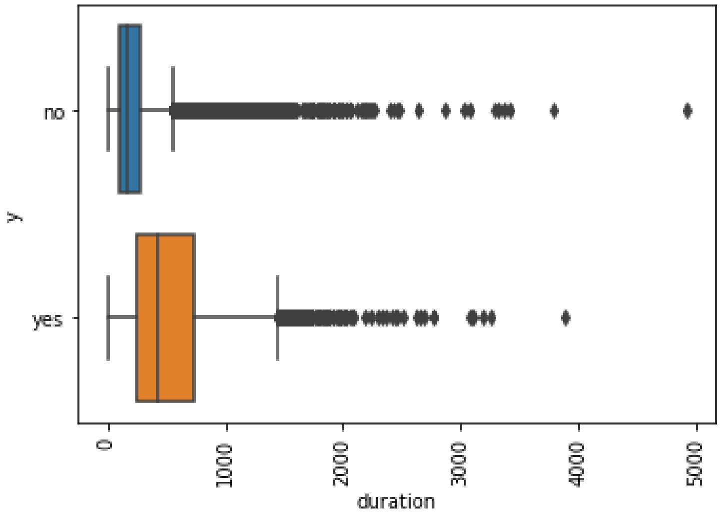
Numerical Variables & Target:

Figure



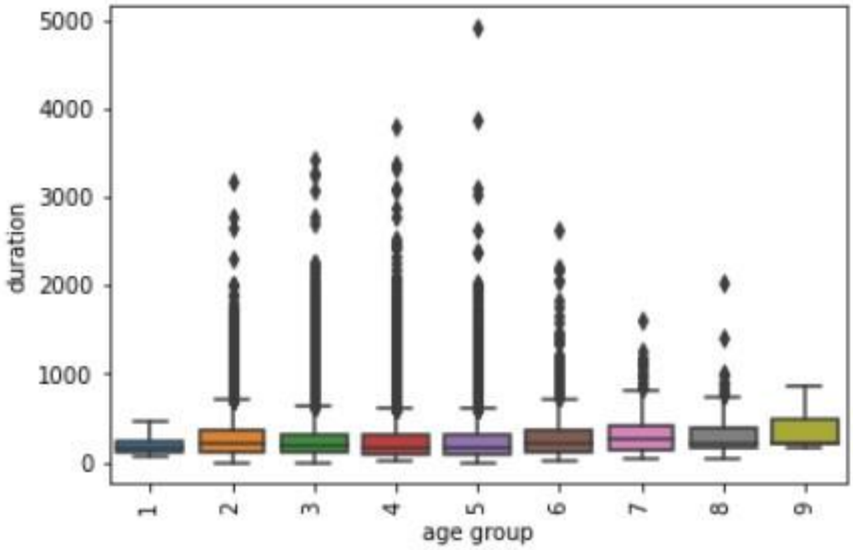
Inferences: There is maximum amount of outlier present in the age group of 65-90 who have not subscribed for term deposit

Figure : Duration Vs Target



Inferences: Customers with high Call durations are more likely to subscribe the Term deposit, from the call duration we can understand they are more likely to take the term deposit, they are interested to know about the plans.

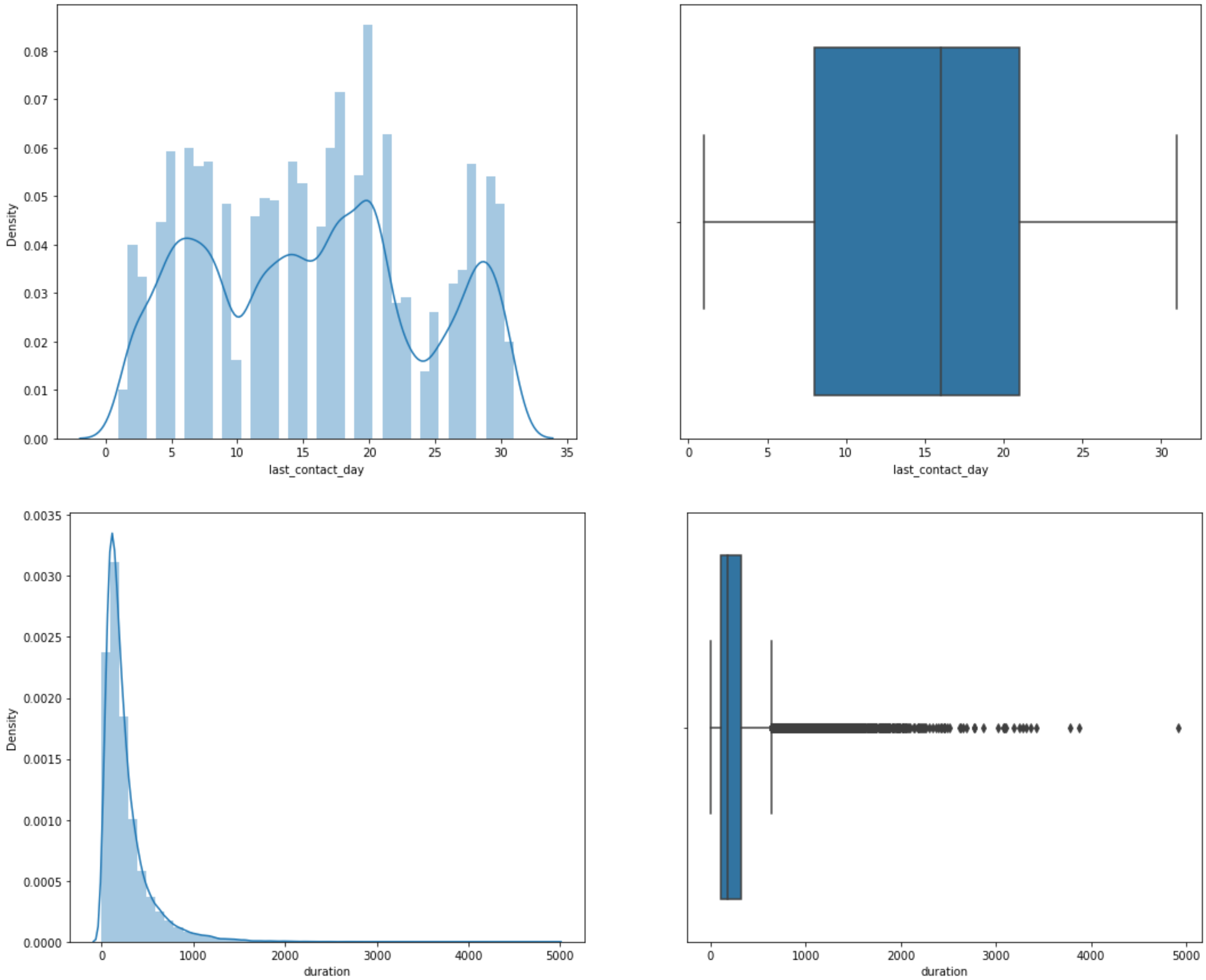
Figure : Duration Vs Age group



Inferences: Customers with Age group 5 (50~59 years) are contacted more and have higher call durations compared to other age groups.

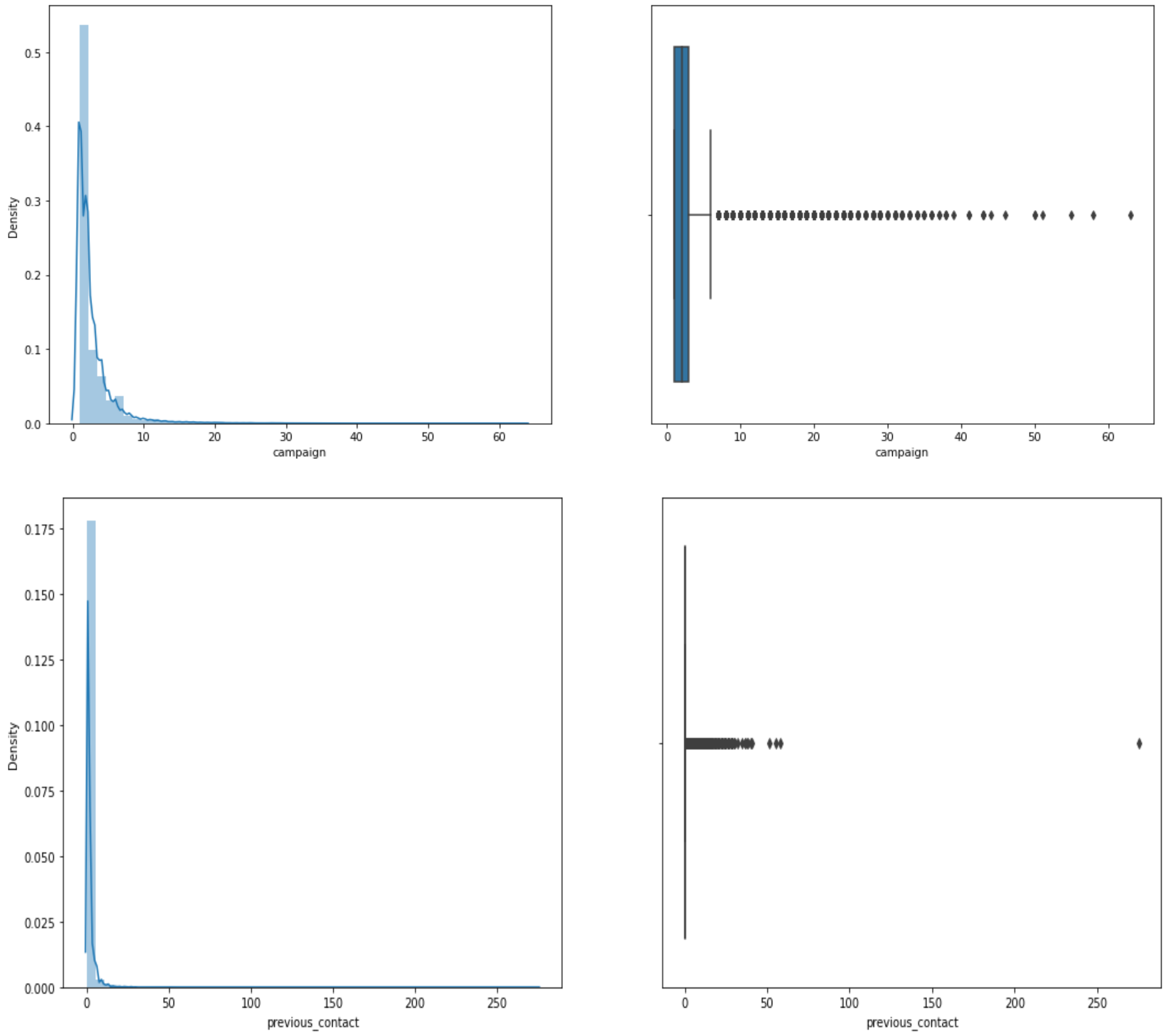
Distribution of Numerical variables:

Figure 16: Distribution of "Last_contact_day" & "Duration"



Inferences: Last_contact_day column is multi-modal with no outliers (skew=0.09) & Duration column is right skewed with outliers (skew=3.14)

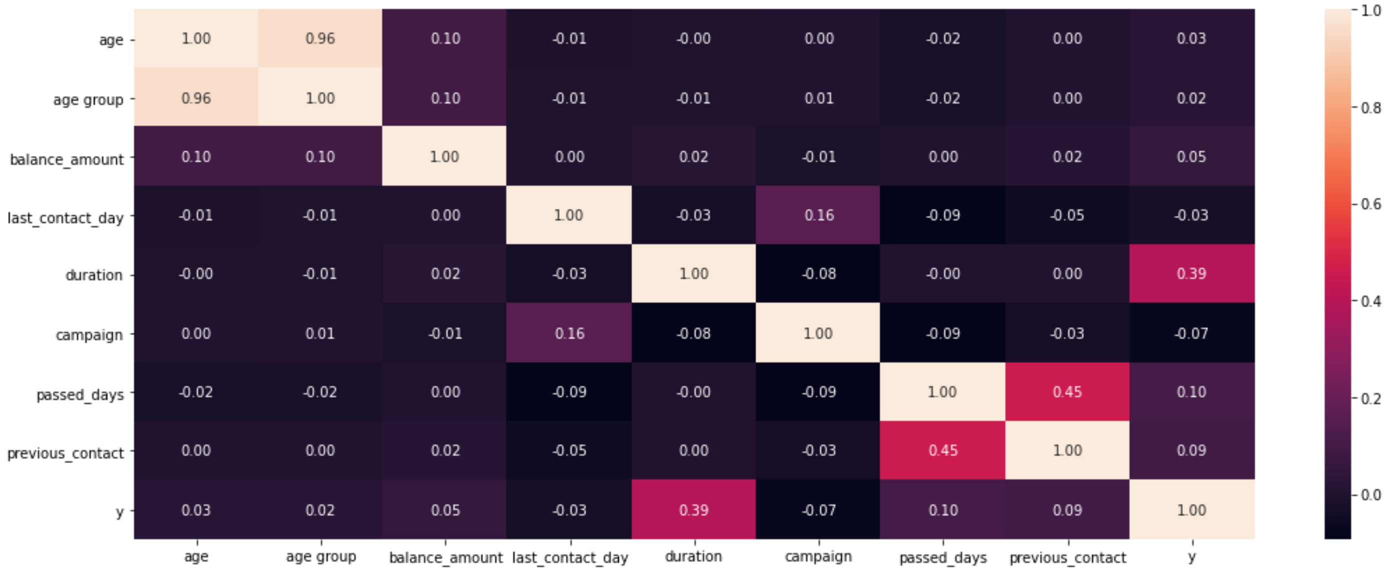
Figure : Distribution of “Campaign” & “Previous_contact”



Inferences: Campaign column is right skewed with outliers (skew=4.89) & Previous_contact column is right skewed with outliers (skew=41.84)

Relationship between variables & target:

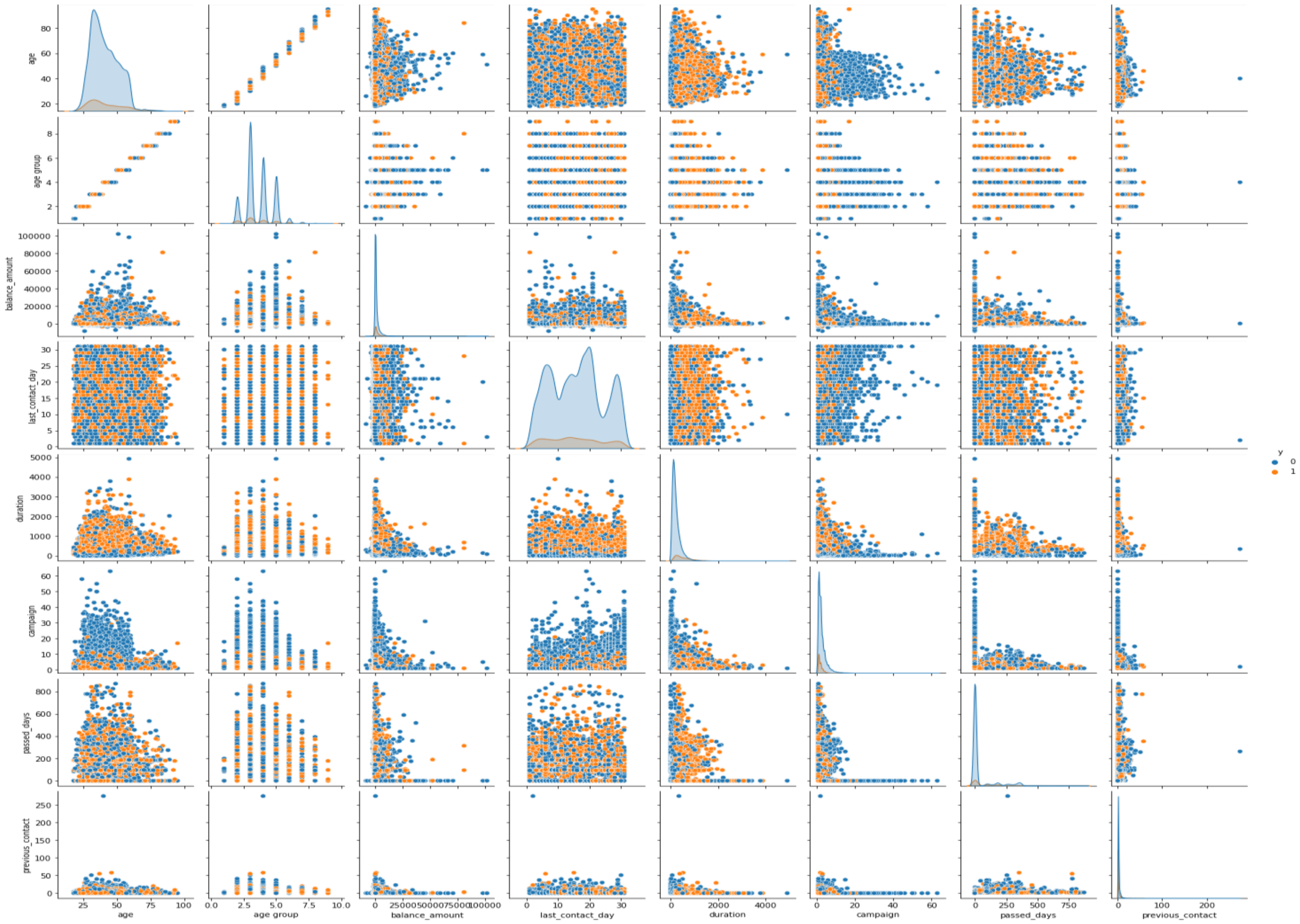
Figure : Correlation between variables



Inferences:

- Age & Age group are related with each other (corr = 0.96) – High Multicollinearity
- Passed_last_contact_days & Previous_contact are related with each other (corr = 0.45)
- Duration has positive relation with Response (corr=0.40)

Relationship between Variable:



Inferences: There is no pattern observed in the Scatterplot between variables and Target. No relation between variables and Target.

7. Methodology Followed:

- In this project, we use the CRISP-DM Framework. CRISP-DM has been the most favoured methodology in data mining domain. Therefore, we have chosen it as our reference model. By using CRISP-DM, we can find interesting patterns from within the data that we want to run. Where the data will be processed through the phases of the existing phase of the business understanding phase, understanding data, data preparation, modelling, evaluation and finally deployment. With this phase, it is expected that the results of this study will get the most appropriate modelling in the data mining process, so that the information generated is more efficient. The six phases of CRISP-DM and described briefly as follows:

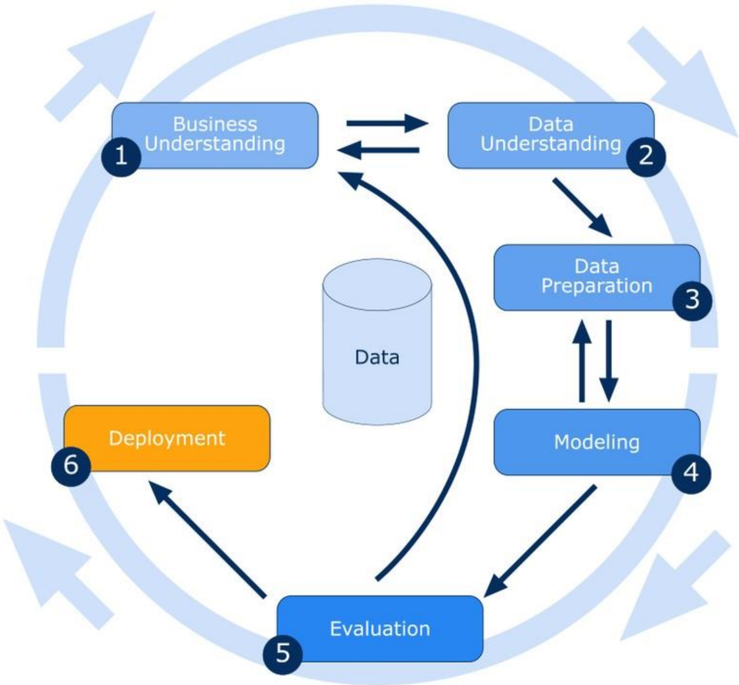


Figure: Methodology CRISPDM

8. Pre-Processing Steps:

- Data pre-processing is done to enhance the quality of data, to promote extraction of meaningful insights from the data. In simpler words, cleaning and organizing the raw data to make it suitable for building and training machine learning models. It is a data mining technique that transforms raw data into an understandable and readable format.
- Treated 'Unknown' Values in the education column in a logical perspective with job column taken as reference. For each category of the job column, based on the higher frequency in education categories,

the unknown values have been replaced. And other null values in the education column have been removed.

- Treated ‘Unknown’ Values in the contact column with mode replacement.
- Encoded the month column with the 1-12(month wise) order.

9. Algorithms Used:

- Here, our objective is to predict whether a customer is capable of making the purchase or not.
- So, the classification techniques we used are as follows :

  - **Logistic Regression**
  - **K-Nearest Neighbors Classifier**
  - **Random forest Classifier**
  - **Adaboost Classifier**
  - **Gradient Boost Classifier**
  - **Extreme Gradient Boost Classifier**

10. Assumptions for different models:

1.) Logistic Regression:

- The logistic regression assumptions are quite different from OLS regression in that:

1. There is no need for a linear relationship between the independent and dependent variables.
2. There is no need for residuals to be normal.
3. There is no need to meet the homoscedasticity assumption

- So what are the assumptions that need to be met for logistic regression?
- Here are the 5 key assumptions for logistic regression.

Assumption 1: Appropriate dependent variable structure

- This assumption simply states that a binary logistic regression requires your dependent variable to be dichotomous and an ordinal logistic regression requires it to be ordinal.
- In addition, the dependent variable should neither be an interval nor ratio scale.

Inference: Here, our Target is binary categorical type.

Assumption 2: No Multicollinearity

- Multicollinearity refers to the high correlation between your independent variables.
- Multicollinearity is a problem because it creates redundant information that will cause the results of your regression model to be unreliable.
- To circumvent this issue, we can deploy two techniques:
- Run a correlation analysis across all your independent variables.
- Remove independent variables with high variance inflation factor(VIF). As a general rule of thumb a vif>10 is a strong indication of multicollinearity
- VIF=1/(1-R^2)

Inference: Age and Age-group had multicollinearity, removed age feature to overcome the multicollinearity.

Assumption 3: No Influential Outliers

- Influential outliers are extreme data points that affect the quality of the logistic regression model.
- Not all outliers are influential.
- We need to check for which points are the influential ones before removing or transforming them for analysis.
- We have checked for it while checking for outliers in EDA section and found some of the independent variables has influential outliers so removed the m using IQR method.

Inference: Had outliers in 'balance','duration','campaign' variables. Treated them using IQR method.

Assumption 4: Observation Independence

- This assumption requires logistic regression observations to be independent of each other.
- That is, observations should not come from a repeated measure design.
- A repeated measure design refers to multiple measures of the same variable taken for the same person under different experimental conditions or across time. A good example of repeated measures is longitudinal studies — tracking progress of a subject over years.

Inference: There are no repeated measures in our feature variables

2.) <u>Tree Based models:</u>

- For tree-based models such as *Decision Trees*, *Random Forest* & *Gradient Boosting* there are no model assumptions to validate.
- Unlike OLS regression or logistic regression, tree-based models are robust to outliers and do not require the dependent variables to meet any normality assumptions.

3.) KNN:

- KNN is a non-parametric lazy learning algorithm**.** When you say a technique is nonparametric, it means that it does not make any assumptions on the underlying data distribution.

11. Step-by-Step Walkthrough of the solution:

Step 1-Import the Dataset into Jupiter Workbook:

- Before we import our sample dataset into the notebook, we will import the panda’s library. pandas is an open-source Python library that provides “high-performance, easy-to-use data structures and data analysis tools.”

```
df=pd.read_csv('bank-marketing.com')
df.head()
```

	age	age group	job	nuptial	education	defaulter_presence	balance_amount	house_loan	personal_loan	contact_via	last_contact_day	last_contact_mor
0	58	5	management	married	tertiary	no	2143.0	yes	no	unknown	5	m
1	44	4	technician	single	secondary	no	29.0	yes	no	unknown	5	m
2	33	3	entrepreneur	married	secondary	no	2.0	yes	yes	unknown	5	m
3	47	4	blue-collar	married	unknown	no	1506.0	yes	no	unknown	5	m
4	33	3	unknown	single	unknown	no	1.0	no	no	unknown	5	m

Step 2-Explore the data set:

Shape:

- shape is a tuple that gives you an indication of the number of dimensions in the array.
- We can get the shape(Total rows(45211) & Total columns(18)) of the data using .shape.

Describe:

	count	mean	std	min	25%	50%	75%	max
age	45211.0	40.936210	10.618762	18.0	33.0	39.0	48.0	95.0
age group	45211.0	3.645861	1.083271	1.0	3.0	3.0	4.0	9.0
balance_amount	45211.0	1362.272058	3044.765829	-8019.0	72.0	448.0	1428.0	102127.0
last_contact_day	45211.0	15.806419	8.322476	1.0	8.0	16.0	21.0	31.0
duration	45211.0	258.163080	257.527812	0.0	103.0	180.0	319.0	4918.0
campaign	45211.0	2.763841	3.098021	1.0	1.0	2.0	3.0	63.0
passed_days	45211.0	40.197828	100.128746	-1.0	-1.0	-1.0	-1.0	871.0
previous_contact	45211.0	0.580323	2.303441	0.0	0.0	0.0	0.0	275.0

- Age of the Customers start from 18 to 95.
- Minimum balance by a Customer is -8019 and max 102127.
- There are 1 to 31 days.
- Duration of calls lasts from 0 to 4918 seconds (82 Minutes or 1.3 Hrs).
- Campaign (No of Calls performed "during this campaign") ranges from 1 to 63 calls.
- Passed_days (No of days passed after the Customer was last contacted from a previous campaign) ranges from -1 to 871 days ---> -1 means the Customer is not contacted.
- Previous_contact (number of contacts performed "before this campaign") ranges from 0 to 275 calls.
- Response is the Target Variable (1-Subscribed a Term deposit, 0 - Not subscribed a Term deposit).

Null Values:

- isnull(). sum() returns the number of missing values in the data set. A simple way to deal with data containing missing values is to skip rows with missing values in the dataset.

```
age                    0
age group              0
job                    0
nuptial                0
education              0
defaulter_presence     0
balance_amount         0
house_loan             0
personal_loan          0
contact_via            0
last_contact_day       0
last_contact_month     0
duration               0
campaign               0
passed_days            0
previous_contact       0
poutcome               0
y                      0
dtype: int64
```

- We haven't found any null values in the given data, to know whether we have any unknown values present in the given data we can use .unique() .

Unique values:

- unique() Function to Get Unique Values from a Dataframe. The **.unique()** function returns the unique values present in a dataset. It basically uses a technique based on hash tables to return the non-redundant values from the set of values present in the data frame/series data structure.

```
for i in  df.select_dtypes(include=np.object).columns:
    print(i,'\n ',df[i].unique())
    print()
```

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

nuptial
  ['married' 'single' 'divorced']

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

defaulter_presence
  ['no' 'yes']

house_loan
  ['yes' 'no']

personal_loan
  ['no' 'yes']

contact_via
  ['unknown' 'cellular' 'telephone']

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

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

y
  ['no' 'yes']
```

- Using .unique() we came to know that we have some unknown values in the data, as we have unknown values we are changing the unknown values to NaN values.

Step 3-Treating 'Unknown' Values:

```
ind = df[(df['job']=='unknown') & (df['education']=='primary')]['job'].index
df.iloc[ind,3] = 'blue-collar'
```

```
ind = df[(df['job']=='unknown') & (df['education']=='secondary')]['job'].index
df.iloc[ind,3] = 'blue-collar'
```

```
ind = df[(df['job']=='unknown') & (df['education']=='tertiary')]['job'].index
df.iloc[ind,3] = 'management'
```

- As we have unknown values in the education column, based on each category in education column the unknown values have been imputed.
- Contact column has unknown values and those unknown values are imputed with mode.

Step 4- Dropping unwanted columns:

```
df.drop('age',1,inplace=True)
```

```
df.drop('poutcome',1,inplace=True)
```

- As we have age group column we can remove age column.
- Poutcome column has unknown values more than 80% we can remove that column.

Step 5-Feature engineering:

- It is the process of using domain knowledge of the data to create new features that make the machine learning model perform better.
- Feature engineering is the essential art in machine learning, which creates a massive difference between a good model and a bad model.

```
def pdays(x):
    if (x<=0):
        return 'Not.Previously.Contacted'
    elif (x>0 and x<=150):
        return '1-150 days'
    elif (x>150 and x<=300):
        return '151-300 days'
    else:
        return '>300 days'
```

```
df['passed_days'] = df['passed_days'].apply(pdays)
```

- Passed_last_contact_days columns contains values such as -1,0 or more than 500, to decrease complexity of the column, we are converting them to categorical column with four categories.
- Passed_last_contact_days <=0 Not previous_contactly connected.

Step 6-Outlier treatment:

- The difference between Q3 and Q1 is called the Inter-Quartile Range or IQR. Any data point less than the Lower Bound or more than the Upper Bound is considered as an outlier.

```
for i in num_out:
    q1 = df[i].quantile(0.25)
    q3 = df[i].quantile(0.75)
    iqr = q3 - q1
    ll = q1 - (3*iqr)
    ul = q3 + (3*iqr)
    df[i] = df[(df[i]>=ll) & (df[i]<=ul)][i]
```

Step 7- Statistical Test:

- The Chi-square test of independence is a statistical hypothesis test used to determine whether two categorical or nominal variables are likely to be related or not.

NOTE: Significance value of 0.05 is considered for Statistical testing

```python
# Chi-sqr Test of Independence
# Hypothesis Formation
# Ho : Variables are Independant (NO relation)
# Ha : Variables are Not independant (Relation)

def chi(obs):
    chi_stat,pval,df,exp_tab = stats.chi2_contingency(obs)
    return pval

not_sig_features = []
sig_features = []

for i in cat_col:
    obs = pd.crosstab(df[i],df['y'])
    pval = chi(obs)
    if (pval > 0.05):
        not_sig_features.append(i)   # Accept H0
    else:
        sig_features.append(i)       # Reject Ho
```

```python
sig_features
```

```
['job',
 'nuptial',
 'education',
 'defaulter_presence',
 'house_loan',
 'personal_loan',
 'contact_via',
 'last_contact_month',
 'passed_days']
```

- As we have target variable as categorical with 2 classes, so we can use chi-square test of independence, with that we have found the significant features.

Step 8-Multi Collinearity Check:

- Variance inflation factor (VIF) is a measure of the amount of multicollinearity in a set of multiple regression variables.

```
v = df[num_col]
vif = [VIF(v.values,i) for i in range(v.shape[1])]
vif_df = pd.DataFrame()
vif_df['numeric_features'] = v.columns
vif_df['VIF'] = vif
vif_df.sort_values('VIF',ascending=False)
```

	numeric_features	VIF
0	age	172.764911
1	age group	160.892434
3	last_contact_day	4.032780
4	duration	1.912673
5	campaign	1.827405
6	passed_days	1.456057
7	previous_contact	1.341648
2	balance_amount	1.212917

- To check whether multi-collinearity is present or not we have used VIF and we have found that only those variables are significant whose VIF factor is less than 10.

Step 9- Encoding:

- We use this categorical data encoding technique when the categorical feature is ordinal. In this case, retaining the order is important. Hence encoding should reflect the sequence.
- In Label encoding, each label is converted into an integer value. We will create a variable that contains the categories representation.

```
le = LabelEncoder()
```

```
for i in cat_le:
    df[i] = le.fit_transform(df[i])
```

- For all the categorical columns encoding has been done using Label Encoding method.

Step 10-Train Test Split:

- The train-test split is a technique for evaluating the performance of a machine learning algorithm.

- It can be used for classification or regression problems and can be used for any supervised learning algorithm.

- **Train Dataset**: Used to fit the machine learning model.
- **Test Dataset**: Used to evaluate the fit machine learning model.

```
x = df.drop('y',1)
y = df['y']

x_sm = df_sm.drop('y',1)
y_sm = df_sm['y']
```

```
x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.3,random_state=10)

x_sm_train,x_sm_test,y_sm_train,y_sm_test = train_test_split(x_sm,y_sm,test_size=0.3,random_state=10)
```

- Train test split has been done with the test size of 30%.

Step 11-Scalling:

- In many machine learning algorithms, to bring all features in the same standing, we need to do scaling so that one significant number doesn't impact the model just because of their large magnitude.

```
ss = StandardScaler()
```

```
sc = ['balance_amount','duration','campaign','previous_contact']
```

```
# Only used for Cross Validation Score

x_scaled = x.copy(deep=True)
x_sm_scaled = x_sm.copy(deep=True)

x_scaled[sc] = ss.fit_transform(x_scaled[sc])
x_sm_scaled[sc] = ss.fit_transform(x_sm_scaled[sc])
```

```
# For Model Building

x_train[sc] = ss.fit_transform(x_train[sc])
x_test[sc] = ss.fit_transform(x_test[sc])

x_sm_train[sc] = ss.fit_transform(x_sm_train[sc])
x_sm_test[sc] = ss.fit_transform(x_sm_test[sc])
```

- We have done standard scaler for the data.

Step 12-Class Imbalance:

- Imbalanced classification refers to a classification predictive modelling problem where the number of examples in the training dataset for each class label is not balanced.

```
x = df.drop('y',1)
y = df['y']
```

```
smote = SMOTE(sampling_strategy=0.5,random_state=10)
x_sm,y_sm = smote.fit_resample(x,y)
```

```
df_sm = pd.DataFrame(x_sm,columns=x.columns)
df_sm['y']=y_sm
df_sm.head()
```

- SMOTE is used to oversample the minority class to balance the class distribution.

Step 13-Model Building:

- A machine learning model is built by learning and generalizing from training data, then applying that acquired knowledge to new data it has never seen before to make predictions and fulfil its purpose.

- We have built following models,

1. **Base model - Logistic Regression(without smote):**

```
print(classification_report(y_test,y_test_pred))
```

	precision	recall	f1-score	support
0	0.91	0.98	0.94	10659
1	0.55	0.19	0.28	1273
accuracy			0.90	11932
macro avg	0.73	0.59	0.61	11932
weighted avg	0.87	0.90	0.87	11932

2. **Base model - Logistic Regression(with smote):**

```
print(classification_report(y_sm_test,y_sm_test_pred))
```

	precision	recall	f1-score	support
0	0.87	0.91	0.89	10672
1	0.80	0.73	0.76	5371
accuracy			0.85	16043
macro avg	0.83	0.82	0.83	16043
weighted avg	0.85	0.85	0.85	16043

3. **Knn Classifier:**

```
print(classification_report(y_sm_test,y_sm_test_pred))
```

	precision	recall	f1-score	support
0	0.89	0.92	0.91	10672
1	0.84	0.78	0.81	5371
accuracy			0.88	16043
macro avg	0.87	0.85	0.86	16043
weighted avg	0.88	0.88	0.88	16043

4. Random Forest Classifier:

```
print(classification_report(y_sm_test,y_sm_test_pred))
```

	precision	recall	f1-score	support
0	0.94	0.91	0.92	10672
1	0.83	0.88	0.86	5371
accuracy			0.90	16043
macro avg	0.89	0.89	0.89	16043
weighted avg	0.90	0.90	0.90	16043

5.Ada Boosting:

```
print(classification_report(y_sm_test,y_sm_test_pred))
```

```
              precision    recall  f1-score   support

           0       0.92      0.72      0.81     10672
           1       0.61      0.88      0.72      5371

    accuracy                           0.77     16043
   macro avg       0.76      0.80      0.76     16043
weighted avg       0.82      0.77      0.78     16043
```

6. Gradient Boosting:

```
print(classification_report(y_sm_test,y_sm_test_pred))
```

```
              precision    recall  f1-score   support

           0       0.93      0.79      0.85     10672
           1       0.68      0.88      0.77      5371

    accuracy                           0.82     16043
   macro avg       0.80      0.84      0.81     16043
weighted avg       0.85      0.82      0.82     16043
```

7. XGB (Hyperparameter tunning):

```
print(classification_report(y_sm_test,y_sm_test_pred))
```

	precision	recall	f1-score	support
0	0.96	0.86	0.90	10672
1	0.76	0.92	0.84	5371
accuracy			0.88	16043
macro avg	0.86	0.89	0.87	16043
weighted avg	0.89	0.88	0.88	16043

12. <u>Data Modelling and Evaluation:</u>

- The basic objective of the project is to analyse Tele-Marketing data collected from a Bank and predict whether the customer will subscribe the term deposit or not, also identify the driving factors behind this. This would inform the Bank's decisions on which Customers to target for their Marketing Campaign, which would ultimately increase their Product Sales. Understanding their customers is critical for effectively executing their Marketing campaign.

- As a part of data modelling, we are building a model which can whether the customer will subscribe the term deposit or not. As classification aims at categorizing the target data to which category it belongs, we have tried various base line models and ensemble models used for data modelling. We are interested in finding the customers who will subscribe the term deposit, so false negative is to be minimized and minority class in the dataset is to be balanced.

- Following Performance measures are to be used for our models:

- Recall [TP/(TP+FN)] – Sensitivity of the model
- F1 Weighted
- ROC_AUC

- Initially the dataset is split into predictors and dependent variable followed by dividing data into two groups, one for training the model and one for validating the model performance. The data is divided in different proportion of training and test data (that is, 70% data for training and 30% data for testing), In order to achieve higher accuracy, sampling techniques (over sampling - SMOTE) has been used (Since, our data set is imbalanced). The splitted training data is further fed into various classification algorithms. Using Cross-Validated score of recall, f1_weighted and roc_auc along with bias_error and variance_error of several models , we can choose the best model which gives higher Sensitivity (recall_score) with low bias and variance error.

The following table shows the cross-validated results of all the models, we have used.

	recall_score	f1_weighted	roc_auc (%)	bias_error %(roc_auc)	variance_error %(roc_auc)
Logistic Regression (without SMOTE)	0.22	0.88	87.28	12.72	0.83
Logistic Regression (with SMOTE)	0.76	0.86	92.80	7.20	0.33
KNN Classifier	0.81	0.88	93.58	6.42	0.23
Randm Forest	0.87	0.92	97.65	2.35	0.16
AdaBoost	0.81	0.90	95.89	4.11	0.27
Gradient Boost	0.82	0.91	96.72	3.28	0.17
XGB	0.88	0.93	98.07	1.93	0.11

- From the above Table, we can find that XGB model's performance is better when compared to other models.

For XGB model we attain the following metrics as:

- recall_score = 0.88
- f1_weighted = 0.93
- roc_auc (%) = 98.07 %
- bias_error (%) = 1.93 %
- variance_error (%) = 0.11%

- Our Final Model is XGB Classifier, which gives us a better results when compared to others. XGBoost uses decision trees as base learners, combining many weak learners to make a strong learner. As a result it is referred to as an ensemble learning method since it uses the output of many models in the final prediction. XGBoost or Extreme Gradient Boosting! It can be put into various use cases such as ranking, classification, regression and user-defined prediction problems. It can be referred to as an "ALL in One" algorithm. It is an ideal blend of software and hardware optimization techniques to yield prevalent outcomes by using fewer computing resources in shortest amount of time.
- There are many advantages of XGBoost, some of them are mentioned below:

- It is Highly Flexible
- It uses the power of parallel processing
- It is faster than Gradient Boosting
- It supports regularization
- It is designed to handle missing data with its in-build features.
- The user can run a cross-validation after each iteration.
- It Works well in small to medium dataset

13. Recommendations:

- Based on the key influencers, following recommendations are suggested to enhance our Business motive "**Identifying the right Customers** and **Increase the odds of subscribing our Term Deposit**".

1. Customers who do not prefer **"house loan & personal loans"** are more likely to subscribe our Term deposit. Thus, we can offer benefitting schemes coupled with house loan & personal loans to make them subscribe two of our products.
2. Customers with less **"passed days"** (Frequently contacted from the previous campaign) will definitely know about the latest Term deposit schemes as they are being contacted more frequently. Minimising the passed days is essential to make customers subscribe our Term deposit. This can be achieved by frequent reminders and calls to the customers to make them aware of our latest products.
3. Customers who do not prefer any **"personal loans"** are more likely to subscribe our Term deposit. We can use customer details, identify their background and then recommend them suitable personal loans coupled with Term deposit to make them subscribe two of our products.
4. **"Eligible"** Customers must be focussed and contacted more frequently. More Schemes and products can be developed targeting Eligible customers for maximum benefit.
5. Customers with more **"previous_contact"** (No of contacts performed) will definitely know about the latest Term deposit schemes as they are being contacted more frequently. Maximising the "previous_contact" is essential to make customers subscribe our Term deposit. This can be achieved by frequent reminders and calls to the customers to make them aware of our latest products.

14. Limitations:

- The Final Model was chosen to be Extreme Gradient Boosting Model by considering all the model performance metrics. Our final Model was tuned with only 'max-depth'.

- This final model was not a generalized model but this was business oriented model that is specifically designed to be used for targeting the customers who would subscribe to the term deposit. This Model will not be used for all the market analysis problems.

- We have used 'recall' as the important metric to determine the model performance which decreases the false negatives in the data specifically to correctly classify the customers who are possible to subscribe the product but were wrongly classified as not possible for subscription.

- This may not be applied in all the Marketing Analysis Problems. So, it only applies in targeting the customers.

15. Conclusion:

- The best model chosen from the previous section is XGBOOST. Hence it can be taken into production where a different set of test data will be used and the classification can be performed with the obtained confidence as told by the model's metrics.
- Hence we can expect our model to classify the customers who are about to take the term deposit with the confidence of 88%.
- From the analysis, it allows the bank to better anticipate and address the potential customers, while improving their strategic marketing campaigns.

16. References:

- Predicting the Success of Bank Telemarketing using various Classification Algorithms https://www.diva-portal.org/smash/get/diva2:1233529/FULLTEXT01.pdf
- Customer Profiling using classification approach for bank telemarketing http://www.joiv.org/index.php/joiv/article/view/68
- Analysis of bank customers and prediction of bank marketing strategies http://sersc.org/journals/index.php/IJAST/article/view/15497/7810