In [1]:

```
###Importing required packages
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [2]:

```
###Supress Warnings
import warnings
warnings.filterwarnings('ignore')
```

In [3]:

```
###importing the data
data= pd.read_csv('Leads.csv')
```

In [4]:

```
data.head()
```

Out[4]:

	Prospect ID	Lead Number	Lead Origin	Lead Source	Do Not Email	Do Not Call	Converted	TotalVisits	Total Time Spent on Website	Page Views Per Visit	...	Get updates on DM Content	Lead Profile	City	Asymmetrique Activity Index	As F
0	7927b2df-8bba-4d29-b9a2-b6e0beafe620	660737	API	Olark Chat	No	No	0	0.0	0	0.0	...	No	Select	Select	02.Medium	
1	2a272436-5132-4136-86fa-dcc88c88f482	660728	API	Organic Search	No	No	0	5.0	674	2.5	...	No	Select	Select	02.Medium	
2	8cc8c611-a219-4f35-ad23-fdfd2656bd8a	660727	Landing Page Submission	Direct Traffic	No	No	1	2.0	1532	2.0	...	No	Potential Lead	Mumbai	02.Medium	
3	0cc2df48-7cf4-4e39-9de9-19797f9b38cc	660719	Landing Page Submission	Direct Traffic	No	No	0	1.0	305	1.0	...	No	Select	Mumbai	02.Medium	
4	3256f628-e534-4826-9d63-4a8b88782852	660681	Landing Page Submission	Google	No	No	1	2.0	1428	1.0	...	No	Select	Mumbai	02.Medium	

5 rows × 37 columns

In [5]:

```
###Checking brief idea about the data
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9240 entries, 0 to 9239
Data columns (total 37 columns):
 #   Column                                         Non-Null Count  Dtype  
---  ------                                         --------------  -----  
 0   Prospect ID                                    9240 non-null   object 
 1   Lead Number                                    9240 non-null   int64  
 2   Lead Origin                                    9240 non-null   object 
 3   Lead Source                                    9204 non-null   object 
 4   Do Not Email                                   9240 non-null   object 
 5   Do Not Call                                    9240 non-null   object 
 6   Converted                                      9240 non-null   int64  
 7   TotalVisits                                    9103 non-null   float64
 8   Total Time Spent on Website                    9240 non-null   int64  
 9   Page Views Per Visit                           9103 non-null   float64
 10  Last Activity                                  9137 non-null   object 
 11  Country                                        6779 non-null   object 
 12  Specialization                                 7802 non-null   object 
 13  How did you hear about X Education             7033 non-null   object 
 14  What is your current occupation                6550 non-null   object 
 15  What matters most to you in choosing a course  6531 non-null   object 
 16  Search                                         9240 non-null   object 
 17  Magazine                                       9240 non-null   object 
 18  Newspaper Article                              9240 non-null   object 
 19  X Education Forums                             9240 non-null   object 
 20  Newspaper                                      9240 non-null   object 
 21  Digital Advertisement                          9240 non-null   object 
 22  Through Recommendations                        9240 non-null   object 
 23  Receive More Updates About Our Courses         9240 non-null   object 
 24  Tags                                           5887 non-null   object 
 25  Lead Quality                                   4473 non-null   object 
 26  Update me on Supply Chain Content              9240 non-null   object 
 27  Get updates on DM Content                      9240 non-null   object 
 28  Lead Profile                                   6531 non-null   object 
 29  City                                           7820 non-null   object 
 30  Asymmetrique Activity Index                    5022 non-null   object 
 31  Asymmetrique Profile Index                     5022 non-null   object 
 32  Asymmetrique Activity Score                    5022 non-null   float64
 33  Asymmetrique Profile Score                     5022 non-null   float64
 34  I agree to pay the amount through cheque       9240 non-null   object 
 35  A free copy of Mastering The Interview         9240 non-null   object 
 36  Last Notable Activity                          9240 non-null   object 
dtypes: float64(4), int64(3), object(30)
memory usage: 2.6+ MB
```

In [6]:

```
### Checking number of rows and columns
data.shape
```

Out[6]:

```
(9240, 37)
```

In [7]:

```
### there are 9240 rows and 37 columns present in the data
```

In [8]:

```
### To understand the numbers in data we use describe function
data.describe()
```

Out[8]:

	Lead Number	Converted	TotalVisits	Total Time Spent on Website	Page Views Per Visit	Asymmetrique Activity Score	Asymmetrique Profile Score
count	9240.000000	9240.000000	9103.000000	9240.000000	9103.000000	5022.000000	5022.000000
mean	617188.435606	0.385390	3.445238	487.698268	2.362820	14.306252	16.344883
std	23405.995698	0.486714	4.854853	548.021466	2.161418	1.386694	1.811395
min	579533.000000	0.000000	0.000000	0.000000	0.000000	7.000000	11.000000
25%	596484.500000	0.000000	1.000000	12.000000	1.000000	14.000000	15.000000
50%	615479.000000	0.000000	3.000000	248.000000	2.000000	14.000000	16.000000
75%	637387.250000	1.000000	5.000000	936.000000	3.000000	15.000000	18.000000
max	660737.000000	1.000000	251.000000	2272.000000	55.000000	18.000000	20.000000

In [9]:

```
from IPython.display import display
pd.options.display.max_columns = None
```

In [10]:

```
data.describe(include='all')
```

Out[10]:

	Prospect ID	Lead Number	Lead Origin	Lead Source	Do Not Email	Do Not Call	Converted	TotalVisits	Total Time Spent on Website	Page Views Per Visit	Last Activity	Country	Specia
count	9240	9240.000000	9240	9204	9240	9240	9240.000000	9103.000000	9240.000000	9103.000000	9137	6779	
unique	9240	NaN	5	21	2	2	NaN	NaN	NaN	NaN	17	38	
top	7927b2df-8bba-4d29-b9a2-b6e0beafe620	NaN	Landing Page Submission	Google	No	No	NaN	NaN	NaN	NaN	Email Opened	India	
freq	1	NaN	4886	2868	8506	9238	NaN	NaN	NaN	NaN	3437	6492	
mean	NaN	617188.435606	NaN	NaN	NaN	NaN	0.385390	3.445238	487.698268	2.362820	NaN	NaN	
std	NaN	23405.995698	NaN	NaN	NaN	NaN	0.486714	4.854853	548.021466	2.161418	NaN	NaN	
min	NaN	579533.000000	NaN	NaN	NaN	NaN	0.000000	0.000000	0.000000	0.000000	NaN	NaN	
25%	NaN	596484.500000	NaN	NaN	NaN	NaN	0.000000	1.000000	12.000000	1.000000	NaN	NaN	
50%	NaN	615479.000000	NaN	NaN	NaN	NaN	0.000000	3.000000	248.000000	2.000000	NaN	NaN	
75%	NaN	637387.250000	NaN	NaN	NaN	NaN	1.000000	5.000000	936.000000	3.000000	NaN	NaN	
max	NaN	660737.000000	NaN	NaN	NaN	NaN	1.000000	251.000000	2272.000000	55.000000	NaN	NaN	

In [11]:

```
### Cleaning the data before we start performing EDA
```

In [12]:

```
###Converting all the values to lower case
data = data.applymap(lambda s:s.lower() if type(s) == str else s)
```

In [13]:

```
### Replacing 'Select' with NaN (Since it means no option is selected)
data = data.replace('select',np.nan)
```

In [14]:

```
###Checking unique value in the data
data.nunique()
```

Out[14]:

```
Prospect ID                                      9240
Lead Number                                      9240
Lead Origin                                         5
Lead Source                                        20
Do Not Email                                        2
Do Not Call                                         2
Converted                                           2
TotalVisits                                        41
Total Time Spent on Website                      1731
Page Views Per Visit                              114
Last Activity                                      17
Country                                            38
Specialization                                     18
How did you hear about X Education                  9
What is your current occupation                     6
What matters most to you in choosing a course       3
Search                                              2
Magazine                                            1
Newspaper Article                                   2
X Education Forums                                  2
Newspaper                                           2
Digital Advertisement                               2
Through Recommendations                             2
Receive More Updates About Our Courses              1
Tags                                               26
Lead Quality                                        5
Update me on Supply Chain Content                   1
Get updates on DM Content                           1
Lead Profile                                        5
City                                                6
Asymmetrique Activity Index                         3
Asymmetrique Profile Index                          3
Asymmetrique Activity Score                        12
Asymmetrique Profile Score                         10
I agree to pay the amount through cheque            1
A free copy of Mastering The Interview              2
Last Notable Activity                              16
dtype: int64
```

In [15]:

```
###Dropping unique valued columns
data1 = data.drop(['Magazine', 'Receive More Updates About Our Courses', 'Update me on Supply Chain Content', 'I agree to pay the amou
```

In [16]:

```
###Checking if the coloumns are dropped
data1.nunique()
```

Out[16]:

```
Prospect ID                                      9240
Lead Number                                      9240
Lead Origin                                         5
Lead Source                                        20
Do Not Email                                        2
Do Not Call                                         2
Converted                                           2
TotalVisits                                        41
Total Time Spent on Website                      1731
Page Views Per Visit                              114
Last Activity                                      17
Country                                            38
Specialization                                     18
How did you hear about X Education                  9
What is your current occupation                     6
What matters most to you in choosing a course       3
Search                                              2
Newspaper Article                                   2
X Education Forums                                  2
Newspaper                                           2
Digital Advertisement                               2
Through Recommendations                             2
Tags                                               26
Lead Quality                                        5
Lead Profile                                        5
City                                                6
Asymmetrique Activity Index                         3
Asymmetrique Profile Index                          3
Asymmetrique Activity Score                        12
Asymmetrique Profile Score                         10
A free copy of Mastering The Interview              2
Last Notable Activity                              16
dtype: int64
```

In [17]:

```
####Let's now check the percentage of missing values in each column

round(100*(data1.isnull().sum()/len(data1.index)), 2)
```

Out[17]:

```
Prospect ID                                       0.00
Lead Number                                       0.00
Lead Origin                                       0.00
Lead Source                                       0.39
Do Not Email                                      0.00
Do Not Call                                       0.00
Converted                                         0.00
TotalVisits                                       1.48
Total Time Spent on Website                       0.00
Page Views Per Visit                              1.48
Last Activity                                     1.11
Country                                          26.63
Specialization                                   36.58
How did you hear about X Education               78.46
What is your current occupation                  29.11
What matters most to you in choosing a course    29.32
Search                                            0.00
Newspaper Article                                 0.00
X Education Forums                                0.00
Newspaper                                         0.00
Digital Advertisement                             0.00
Through Recommendations                           0.00
Tags                                             36.29
Lead Quality                                     51.59
Lead Profile                                     74.19
City                                             39.71
Asymmetrique Activity Index                      45.65
Asymmetrique Profile Index                       45.65
Asymmetrique Activity Score                      45.65
Asymmetrique Profile Score                       45.65
A free copy of Mastering The Interview            0.00
Last Notable Activity                             0.00
dtype: float64
```

In [18]:

```
###Removing all columns which have > 35% null value
data2= data1.drop(['Asymmetrique Profile Index','Asymmetrique Activity Index','Asymmetrique Activity Score','Asymmetrique Profile Scor
data2.head()
```

Out[18]:

	Prospect ID	Lead Origin	Lead Source	Do Not Email	Do Not Call	Converted	TotalVisits	Total Time Spent on Website	Page Views Per Visit	Last Activity	Country	Specialization	What is your current occupation	What matters most to you in choosing a course
0	7927b2df-8bba-4d29-b9a2-b6e0beafe620	api	olark chat	no	no	0	0.0	0	0.0	page visited on website	NaN	NaN	unemployed	better career prospects
1	2a272436-5132-4136-86fa-dcc88c88f482	api	organic search	no	no	0	5.0	674	2.5	email opened	india	NaN	unemployed	better career prospects
2	8cc8c611-a219-4f35-ad23-fdfd2656bd8a	landing page submission	direct traffic	no	no	1	2.0	1532	2.0	email opened	india	business administration	student	better career prospects
3	0cc2df48-7cf4-4e39-9de9-19797f9b38cc	landing page submission	direct traffic	no	no	0	1.0	305	1.0	unreachable	india	media and advertising	unemployed	better career prospects
4	3256f628-e534-4826-9d63-4a8b88782852	landing page submission	google	no	no	1	2.0	1428	1.0	converted to lead	india	NaN	unemployed	better career prospects

In [19]:

```
###Checking again if the columns has been dropped
round(100*(data2.isnull().sum()/len(data2.index)), 2)
```

Out[19]:

```
Prospect ID                                      0.00
Lead Origin                                      0.00
Lead Source                                      0.39
Do Not Email                                     0.00
Do Not Call                                      0.00
Converted                                        0.00
TotalVisits                                      1.48
Total Time Spent on Website                      0.00
Page Views Per Visit                             1.48
Last Activity                                    1.11
Country                                         26.63
Specialization                                  36.58
What is your current occupation                 29.11
What matters most to you in choosing a course   29.32
Search                                           0.00
Newspaper Article                                0.00
X Education Forums                               0.00
Newspaper                                        0.00
Digital Advertisement                            0.00
Through Recommendations                          0.00
A free copy of Mastering The Interview           0.00
Last Notable Activity                            0.00
dtype: float64
```

In [20]:

```
###Replacing the NaN values with 'not provided' in the four columns Country,Specialization,What is your current occupation,What matters
###Since dropping them will result in huge loss of data
data2['Specialization'] = data2['Specialization'].fillna('not provided')
data2['What matters most to you in choosing a course'] = data2['What matters most to you in choosing a course'].fillna('not provided')
data2['Country'] = data2['Country'].fillna('not provided')
data2['What is your current occupation'] = data2['What is your current occupation'].fillna('not provided')
data2.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9240 entries, 0 to 9239
Data columns (total 22 columns):
 #   Column                                         Non-Null Count  Dtype  
---  ------                                         --------------  -----  
 0   Prospect ID                                    9240 non-null   object 
 1   Lead Origin                                    9240 non-null   object 
 2   Lead Source                                    9204 non-null   object 
 3   Do Not Email                                   9240 non-null   object 
 4   Do Not Call                                    9240 non-null   object 
 5   Converted                                      9240 non-null   int64  
 6   TotalVisits                                    9103 non-null   float64
 7   Total Time Spent on Website                    9240 non-null   int64  
 8   Page Views Per Visit                           9103 non-null   float64
 9   Last Activity                                  9137 non-null   object 
 10  Country                                        9240 non-null   object 
 11  Specialization                                 9240 non-null   object 
 12  What is your current occupation                9240 non-null   object 
 13  What matters most to you in choosing a course  9240 non-null   object 
 14  Search                                         9240 non-null   object 
 15  Newspaper Article                              9240 non-null   object 
 16  X Education Forums                             9240 non-null   object 
 17  Newspaper                                      9240 non-null   object 
 18  Digital Advertisement                          9240 non-null   object 
 19  Through Recommendations                        9240 non-null   object 
 20  A free copy of Mastering The Interview         9240 non-null   object 
 21  Last Notable Activity                          9240 non-null   object 
dtypes: float64(2), int64(2), object(18)
memory usage: 1.6+ MB
```

In [21]:

```
###Checking the % of missing values after filling Nan
round(100*(data2.isnull().sum()/len(data2.index)), 2)
```

Out[21]:

```
Prospect ID                                      0.00
Lead Origin                                      0.00
Lead Source                                      0.39
Do Not Email                                     0.00
Do Not Call                                      0.00
Converted                                        0.00
TotalVisits                                      1.48
Total Time Spent on Website                      0.00
Page Views Per Visit                             1.48
Last Activity                                    1.11
Country                                          0.00
Specialization                                   0.00
What is your current occupation                  0.00
What matters most to you in choosing a course    0.00
Search                                           0.00
Newspaper Article                                0.00
X Education Forums                               0.00
Newspaper                                        0.00
Digital Advertisement                            0.00
Through Recommendations                          0.00
A free copy of Mastering The Interview           0.00
Last Notable Activity                            0.00
dtype: float64
```

In [22]:

```
###Checking the value counts for column country
data2["Country"].value_counts()
```

Out[22]:

```
india                   6492
not provided            2461
united states             69
united arab emirates      53
singapore                 24
saudi arabia              21
united kingdom            15
australia                 13
qatar                     10
bahrain                    7
hong kong                  7
oman                       6
france                     6
unknown                    5
kuwait                     4
south africa               4
canada                     4
nigeria                    4
germany                    4
sweden                     3
philippines                2
uganda                     2
italy                      2
bangladesh                 2
netherlands                2
asia/pacific region        2
china                      2
belgium                    2
ghana                      2
kenya                      1
sri lanka                  1
tanzania                   1
malaysia                   1
liberia                    1
switzerland                1
denmark                    1
russia                     1
vietnam                    1
indonesia                  1
Name: Country, dtype: int64
```

In [23]:

```
###Changing rest of the countries as outside india except India and not provided
def slots(x):
    category = ""
    if x == "india":
        category = "india"
    elif x == "not provided":
        category = "not provided"
    else:
        category = "outside india"
    return category

data2['Country'] = data2.apply(lambda x:slots(x['Country']), axis = 1)
data2['Country'].value_counts()
```

Out[23]:

```
india            6492
not provided     2461
outside india     287
Name: Country, dtype: int64
```

In [24]:

```
###Checking the percent of lose if the null values are removed
round(100*(sum(data2.isnull().sum(axis=1) > 1)/data2.shape[0]),2)
```

Out[24]:

```
1.48
```

In [25]:

```
data3 = data2[data2.isnull().sum(axis=1) <1]
```

In [26]:

```
###Checking % of missing values
round(100*(data3.isnull().sum()/len(data3.index)), 2)
```

Out[26]:

```
Prospect ID                                      0.0
Lead Origin                                      0.0
Lead Source                                      0.0
Do Not Email                                     0.0
Do Not Call                                      0.0
Converted                                        0.0
TotalVisits                                      0.0
Total Time Spent on Website                      0.0
Page Views Per Visit                             0.0
Last Activity                                    0.0
Country                                          0.0
Specialization                                   0.0
What is your current occupation                  0.0
What matters most to you in choosing a course    0.0
Search                                           0.0
Newspaper Article                                0.0
X Education Forums                               0.0
Newspaper                                        0.0
Digital Advertisement                            0.0
Through Recommendations                          0.0
A free copy of Mastering The Interview           0.0
Last Notable Activity                            0.0
dtype: float64
```

In [27]:

```
###To familiarize all the categorical values
for column in data3:
    print(data3[column].astype('category').value_counts())
    print('----------------------------------------------------------------------------------------')
```

```
1    3435
Name: Converted, dtype: int64
----------------------------------------------------------------------------------------
0.0      2161
2.0      1679
3.0      1306
4.0      1120
5.0       783
6.0       466
1.0       395
7.0       309
8.0       224
9.0       164
10.0      114
11.0       86
13.0       48
12.0       45
14.0       36
16.0       21
15.0       18
```

In [28]:

```
###Removing Prospect Id values since they are unique for everyone
data_final = data3.drop('Prospect ID',1)
data_final.shape
```

Out[28]:

```
(9074, 21)
```

In [29]:

```
data_final.info()
```

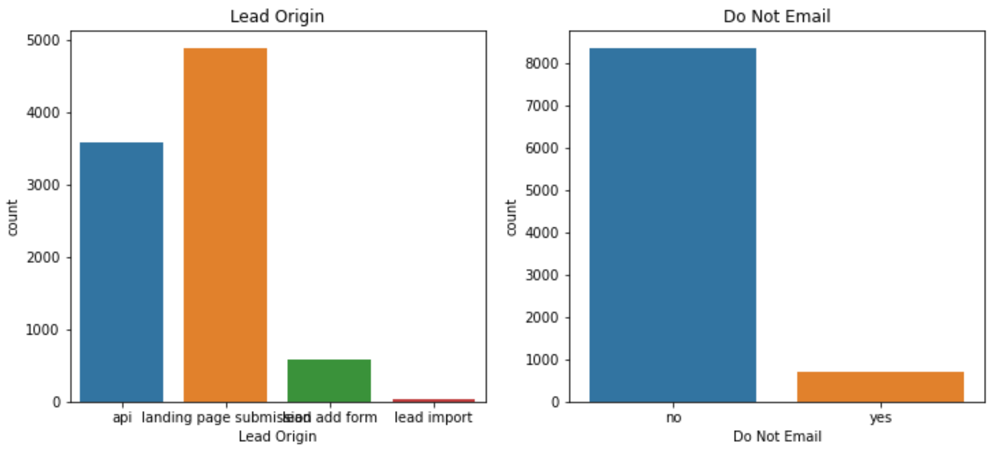
```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 9074 entries, 0 to 9239
Data columns (total 21 columns):
 #   Column                                         Non-Null Count  Dtype  
---  ------                                         --------------  -----  
 0   Lead Origin                                    9074 non-null   object 
 1   Lead Source                                    9074 non-null   object 
 2   Do Not Email                                   9074 non-null   object 
 3   Do Not Call                                    9074 non-null   object 
 4   Converted                                      9074 non-null   int64  
 5   TotalVisits                                    9074 non-null   float64
 6   Total Time Spent on Website                    9074 non-null   int64  
 7   Page Views Per Visit                           9074 non-null   float64
 8   Last Activity                                  9074 non-null   object 
 9   Country                                        9074 non-null   object 
 10  Specialization                                 9074 non-null   object 
 11  What is your current occupation                9074 non-null   object 
 12  What matters most to you in choosing a course  9074 non-null   object 
 13  Search                                         9074 non-null   object 
 14  Newspaper Article                              9074 non-null   object 
 15  X Education Forums                             9074 non-null   object 
 16  Newspaper                                      9074 non-null   object 
 17  Digital Advertisement                          9074 non-null   object 
 18  Through Recommendations                        9074 non-null   object 
 19  A free copy of Mastering The Interview         9074 non-null   object 
 20  Last Notable Activity                          9074 non-null   object 
dtypes: float64(2), int64(2), object(17)
memory usage: 1.5+ MB
```

In [30]:

```
### PERFORMING EDA
```

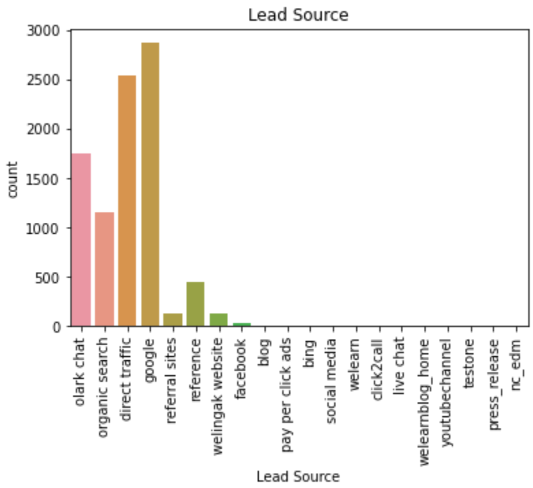
In [31]:

```
plt.figure(figsize = (12,35))

plt.subplot(6,2,1)
sns.countplot(data_final['Lead Origin'])
plt.title('Lead Origin')

plt.subplot(6,2,2)
sns.countplot(data_final['Do Not Email'])
plt.title('Do Not Email')

plt.subplot(6,2,3)
sns.countplot(data_final['Do Not Call'])
plt.title('Do Not Call')

plt.subplot(6,2,4)
sns.countplot(data_final['Country'])
plt.title('Country')

plt.subplot(6,2,5)
sns.countplot(data_final['Search'])
plt.title('Search')

plt.subplot(6,2,6)
sns.countplot(data_final['Newspaper Article'])
plt.title('Newspaper Article')

plt.subplot(6,2,7)
sns.countplot(data_final['X Education Forums'])
plt.title('X Education Forums')

plt.subplot(6,2,8)
sns.countplot(data_final['Newspaper'])
plt.title('Newspaper')

plt.subplot(6,2,9)
sns.countplot(data_final['Digital Advertisement'])
plt.title('Digital Advertisement')

plt.subplot(6,2,10)
sns.countplot(data_final['Through Recommendations'])
plt.title('Through Recommendations')

plt.subplot(6,2,11)
sns.countplot(data_final['A free copy of Mastering The Interview'])
plt.title('A free copy of Mastering The Interview')

plt.subplot(6,2,12)
sns.countplot(data_final['Last Notable Activity']).tick_params(axis='x', rotation = 90)
plt.title('Last Notable Activity')


plt.show()
```



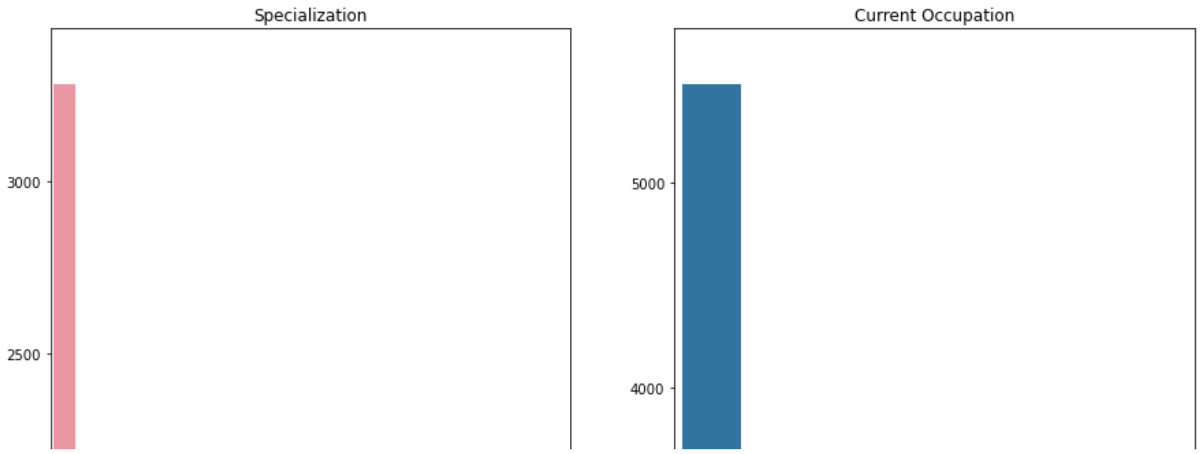
In [32]:

```
sns.countplot(data_final['Lead Source']).tick_params(axis='x', rotation = 90)
plt.title('Lead Source')
plt.show()
```



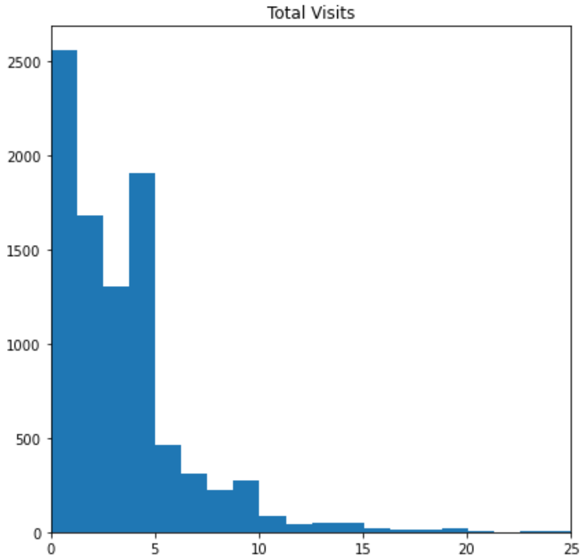
In [33]:

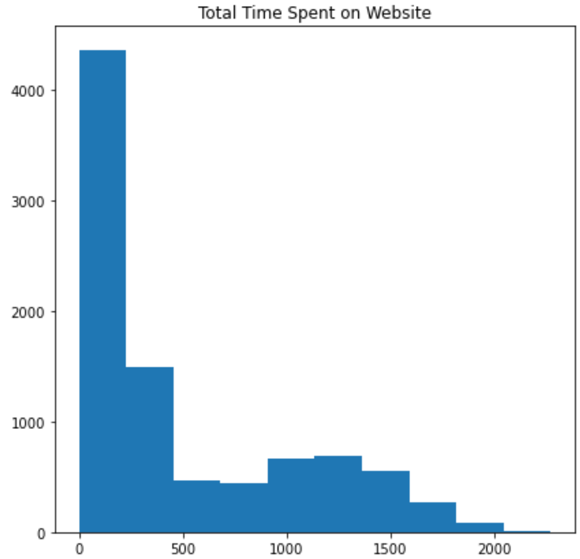
```
plt.figure(figsize = (15,35))
plt.subplot(2,2,1)
sns.countplot(data_final['Specialization']).tick_params(axis='x', rotation = 90)
plt.title('Specialization')
plt.subplot(2,2,2)
sns.countplot(data_final['What is your current occupation']).tick_params(axis='x', rotation = 90)
plt.title('Current Occupation')
plt.subplot(2,2,3)
sns.countplot(data_final['What matters most to you in choosing a course']).tick_params(axis='x', rotation = 90)
plt.title('What matters most to you in choosing a course')
plt.subplot(2,2,4)
sns.countplot(data_final['Last Activity']).tick_params(axis='x', rotation = 90)
plt.title('Last Activity')
plt.show()
```

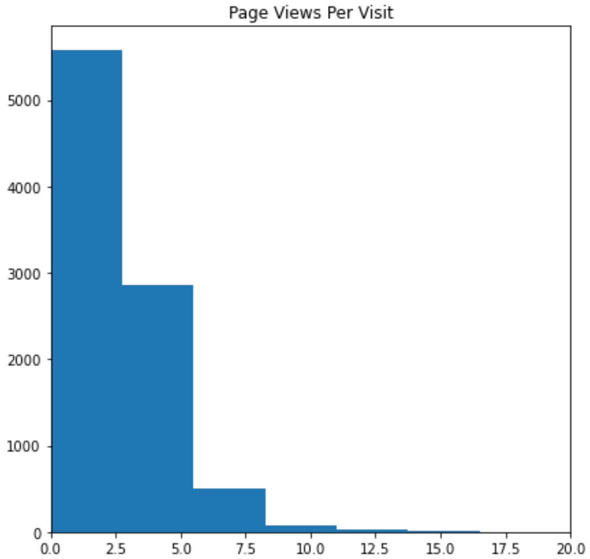


In [34]:

```
plt.figure(figsize = (15,15))
plt.subplot(221)
plt.hist(data_final['TotalVisits'], bins = 200)
plt.title('Total Visits')
plt.xlim(0,25)

plt.subplot(222)
plt.hist(data_final['Total Time Spent on Website'], bins = 10)
plt.title('Total Time Spent on Website')

plt.subplot(223)
plt.hist(data_final['Page Views Per Visit'], bins = 20)
plt.title('Page Views Per Visit')
plt.xlim(0,20)
plt.show()
```

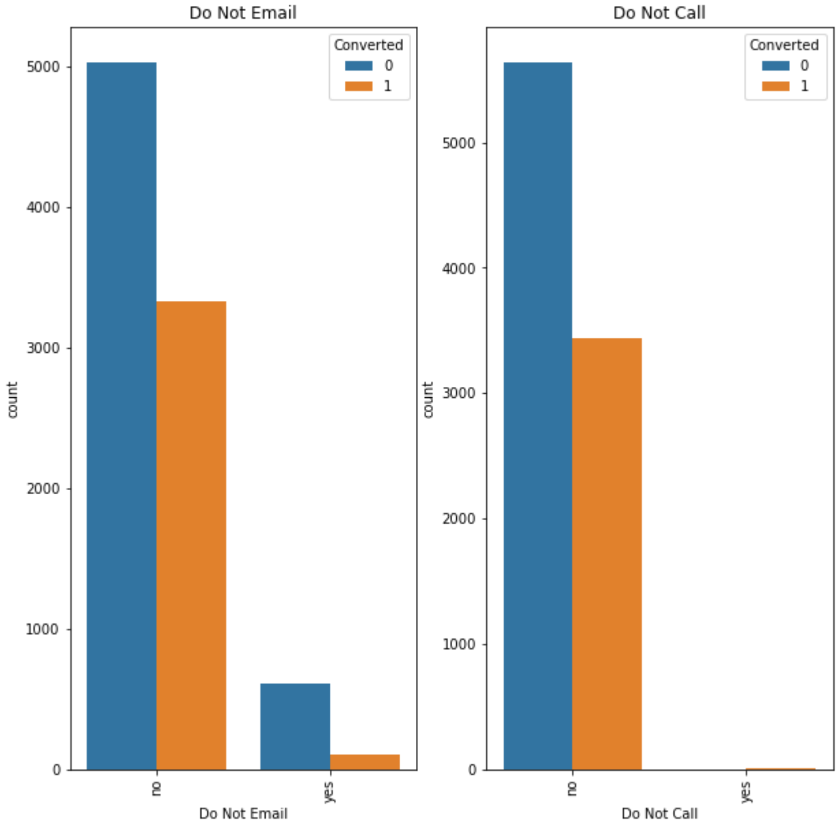






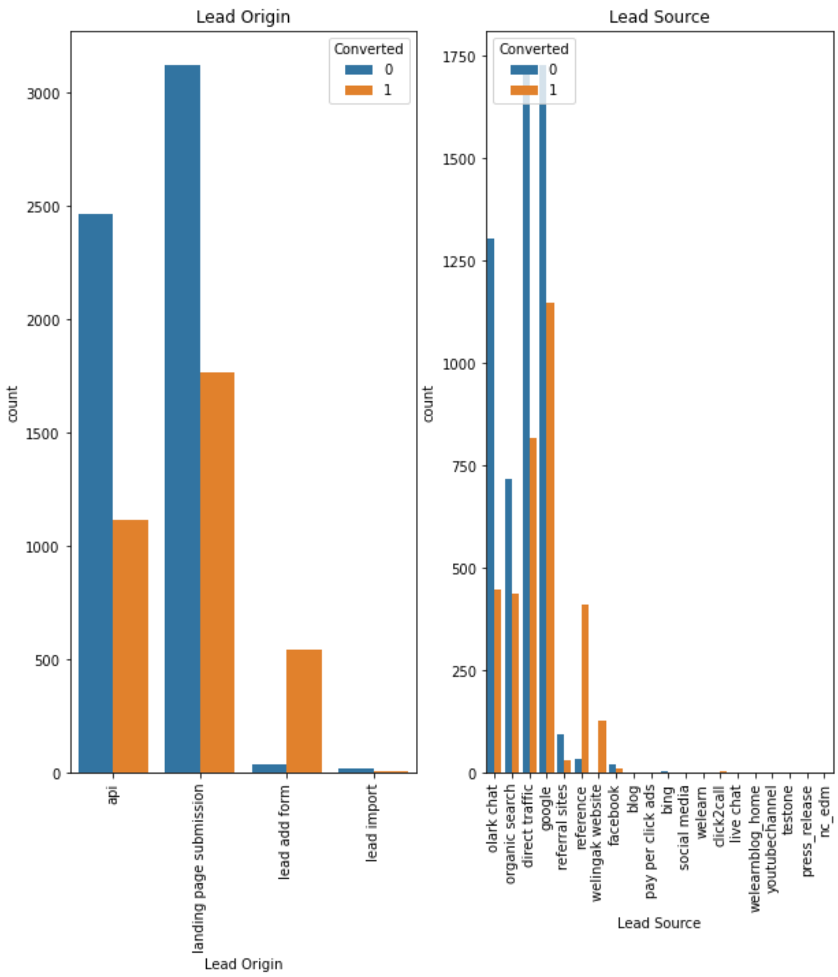
In [35]:

```
###Relating all Categorical variable to Converted
plt.figure(figsize = (10,10))

plt.subplot(1,2,1)
sns.countplot(x='Do Not Email', hue='Converted', data= data_final).tick_params(axis='x', rotation = 90)
plt.title('Do Not Email')

plt.subplot(1,2,2)
sns.countplot(x='Do Not Call', hue='Converted', data= data_final).tick_params(axis='x', rotation = 90)
plt.title('Do Not Call')
plt.show()
```



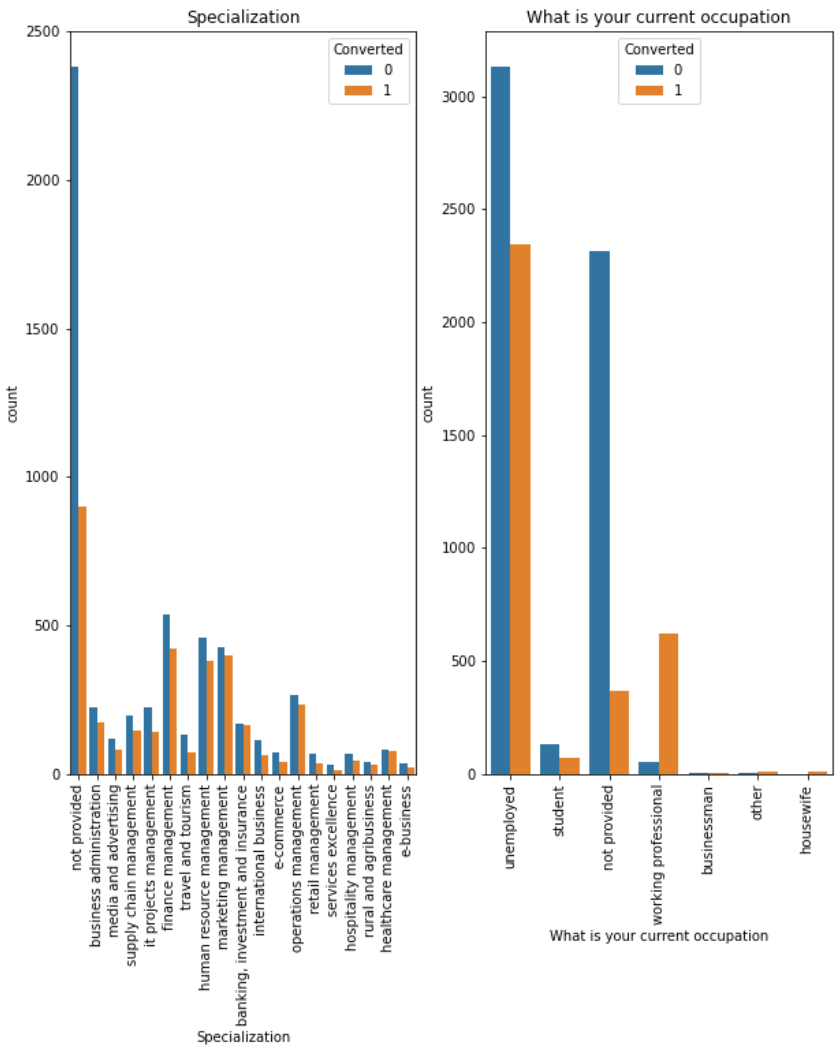
In [36]:

```
plt.figure(figsize = (10,10))

plt.subplot(1,2,1)
sns.countplot(x='Lead Origin', hue='Converted', data= data_final).tick_params(axis='x', rotation = 90)
plt.title('Lead Origin')

plt.subplot(1,2,2)
sns.countplot(x='Lead Source', hue='Converted', data= data_final).tick_params(axis='x', rotation = 90)
plt.title('Lead Source')
plt.show()
```



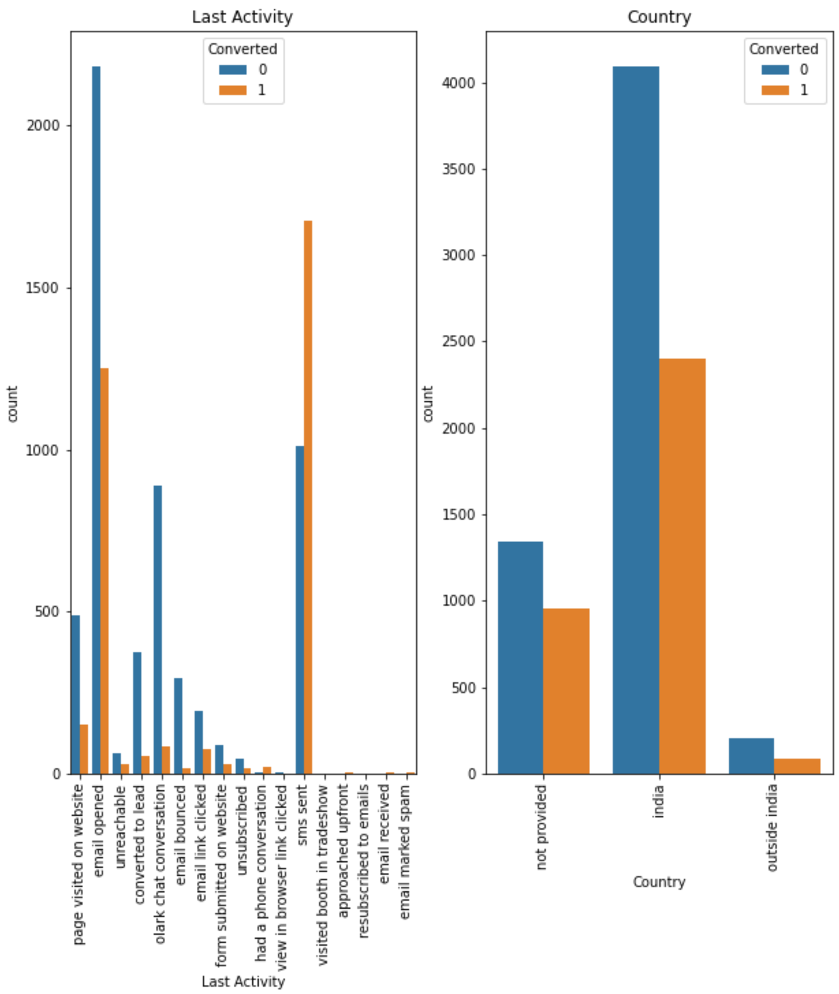
In [37]:

```
plt.figure(figsize = (10,10))

plt.subplot(1,2,1)
sns.countplot(x='Specialization', hue='Converted', data= data_final).tick_params(axis='x', rotation = 90)
plt.title('Specialization')

plt.subplot(1,2,2)
sns.countplot(x='What is your current occupation', hue='Converted', data= data_final).tick_params(axis='x', rotation = 90)
plt.title('What is your current occupation')
plt.show()
```



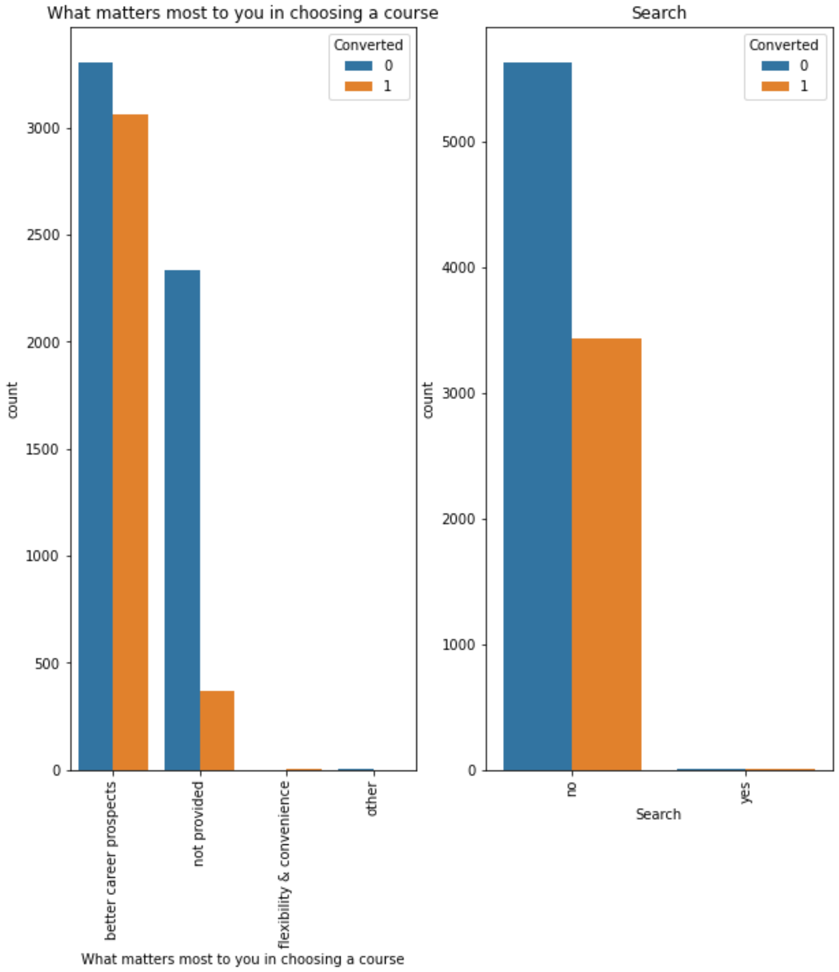
In [38]:

```
plt.figure(figsize = (10,10))

plt.subplot(1,2,1)
sns.countplot(x='Last Activity', hue='Converted', data= data_final).tick_params(axis='x', rotation = 90)
plt.title('Last Activity')

plt.subplot(1,2,2)
sns.countplot(x='Country', hue='Converted', data= data_final).tick_params(axis='x', rotation = 90)
plt.title('Country')
plt.show()
```



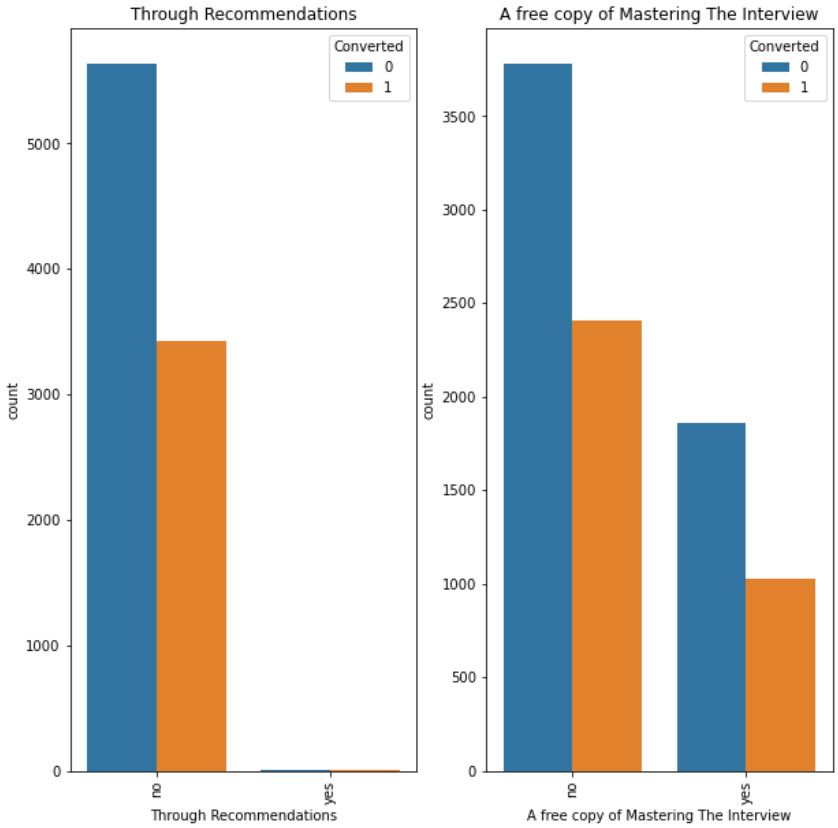
In [39]:

```
plt.figure(figsize = (10,10))

plt.subplot(1,2,1)
sns.countplot(x='What matters most to you in choosing a course', hue='Converted', data= data_final).tick_params(axis='x', rotation = 9(
plt.title('What matters most to you in choosing a course')

plt.subplot(1,2,2)
sns.countplot(x='Search', hue='Converted', data= data_final).tick_params(axis='x', rotation = 90)
plt.title('Search')
plt.show()
```



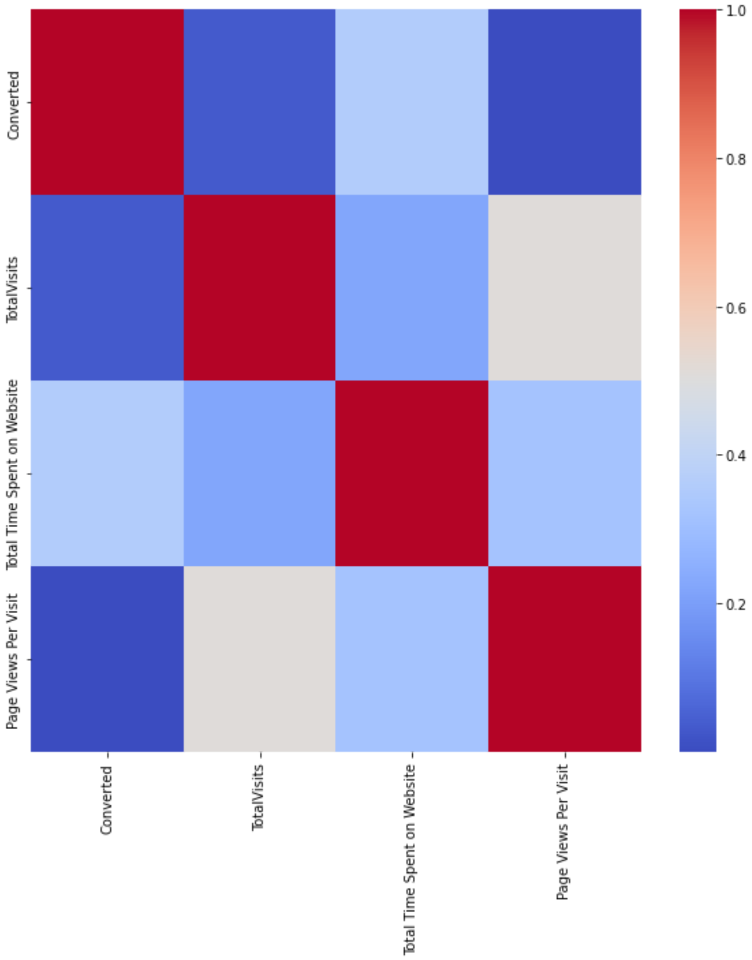
In [40]:

```
plt.figure(figsize = (10,10))

plt.subplot(1,2,1)
sns.countplot(x='Through Recommendations', hue='Converted', data= data_final).tick_params(axis='x', rotation = 90)
plt.title('Through Recommendations')

plt.subplot(1,2,2)
sns.countplot(x='A free copy of Mastering The Interview', hue='Converted', data= data_final).tick_params(axis='x', rotation = 90)
plt.title('A free copy of Mastering The Interview')
plt.show()
```



In [41]:

```
###Checking Correlations
plt.figure(figsize=(10,10))
sns.heatmap(data_final.corr(),cmap='coolwarm')
plt.show()
```



In [42]:

```
###Checking for outliers
numeric = data_final[['TotalVisits','Total Time Spent on Website','Page Views Per Visit']]
numeric.describe(percentiles=[0.25,0.5,0.75,0.9,0.99])
```

Out[42]:

	TotalVisits	Total Time Spent on Website	Page Views Per Visit
count	9074.000000	9074.000000	9074.000000
mean	3.456028	482.887481	2.370151
std	4.858802	545.256560	2.160871
min	0.000000	0.000000	0.000000
25%	1.000000	11.000000	1.000000
50%	3.000000	246.000000	2.000000
75%	5.000000	922.750000	3.200000
90%	7.000000	1373.000000	5.000000
99%	17.000000	1839.000000	9.000000
max	251.000000	2272.000000	55.000000

In [43]:

```
###Dummy Variables
```

In [44]:

```
data_final.loc[:, data_final.dtypes == 'object'].columns
```

Out[44]:

```
Index(['Lead Origin', 'Lead Source', 'Do Not Email', 'Do Not Call',
       'Last Activity', 'Country', 'Specialization',
       'What is your current occupation',
       'What matters most to you in choosing a course', 'Search',
       'Newspaper Article', 'X Education Forums', 'Newspaper',
       'Digital Advertisement', 'Through Recommendations',
       'A free copy of Mastering The Interview', 'Last Notable Activity'],
      dtype='object')
```

In [45]:

```
###Creating dummy variables using the 'get_dummies'
dummy = pd.get_dummies(data_final[['Lead Origin','Specialization' ,'Lead Source', 'Do Not Email', 'Last Activity', 'What is your curre
###Adding the results to the master dataframe
data_final_dum = pd.concat([data_final, dummy], axis=1)
data_final_dum
```

Out[45]:

	Lead Origin	Lead Source	Do Not Email	Do Not Call	Converted	TotalVisits	Total Time Spent on Website	Page Views Per Visit	Last Activity	Country	Specialization	What is your current occupation	What matters most to you in choosing a course	Search	Ne
0	api	olark chat	no	no	0	0.0	0	0.00	page visited on website	not provided	not provided	unemployed	better career prospects	no	
1	api	organic search	no	no	0	5.0	674	2.50	email opened	india	not provided	unemployed	better career prospects	no	
2	landing page submission	direct traffic	no	no	1	2.0	1532	2.00	email opened	india	business administration	student	better career prospects	no	
3	landing page submission	direct traffic	no	no	0	1.0	305	1.00	unreachable	india	media and advertising	unemployed	better career prospects	no	
4	landing page submission	google	no	no	1	2.0	1428	1.00	converted to lead	india	not provided	unemployed	better career prospects	no	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
9235	landing page submission	direct traffic	yes	no	1	8.0	1845	2.67	email marked spam	outside india	it projects management	unemployed	better career prospects	no	
9236	landing page submission	direct traffic	no	no	0	2.0	238	2.00	sms sent	india	media and advertising	unemployed	better career prospects	no	
9237	landing page submission	direct traffic	yes	no	0	2.0	199	2.00	sms sent	india	business administration	unemployed	better career prospects	no	
9238	landing page submission	google	no	no	1	3.0	499	3.00	sms sent	india	human resource management	not provided	not provided	no	
9239	landing page submission	direct traffic	no	no	1	6.0	1279	3.00	sms sent	outside india	supply chain management	unemployed	better career prospects	no	

9074 rows × 100 columns

In [46]:

```
data_final_dum = data_final_dum.drop(['What is your current occupation_not provided','Lead Origin', 'Lead Source', 'Do Not Email', 'Do
data_final_dum
```

Out[46]:

	Converted	TotalVisits	Total Time Spent on Website	Page Views Per Visit	Lead Origin_landing page submission	Lead Origin_lead add form	Lead Origin_lead import	Specialization_business administration	Specialization_e-business	Specialization_e-commerce	Spe
0	0	0.0	0	0.00	0	0	0	0	0	0	
1	0	5.0	674	2.50	0	0	0	0	0	0	
2	1	2.0	1532	2.00	1	0	0	1	0	0	
3	0	1.0	305	1.00	1	0	0	0	0	0	
4	1	2.0	1428	1.00	1	0	0	0	0	0	
...	...	...	...	...	...	...	...	...	...	...	
9235	1	8.0	1845	2.67	1	0	0	0	0	0	
9236	0	2.0	238	2.00	1	0	0	0	0	0	
9237	0	2.0	199	2.00	1	0	0	1	0	0	
9238	1	3.0	499	3.00	1	0	0	0	0	0	
9239	1	6.0	1279	3.00	1	0	0	0	0	0	

9074 rows × 81 columns

In [47]:

```
###TRAIN-TEST SPLIT
```

In [48]:

```
###Importing the required library
from sklearn.model_selection import train_test_split
```

In [49]:

```
X = data_final_dum.drop(['Converted'], 1)
X.head()
```

Out[49]:

	TotalVisits	Total Time Spent on Website	Page Views Per Visit	Lead Origin_landing page submission	Lead Origin_lead add form	Lead Origin_lead import	Specialization_business administration	Specialization_e-business	Specialization_e-commerce	Specialization_finai managem
0	0.0	0	0.0	0	0	0	0	0	0	
1	5.0	674	2.5	0	0	0	0	0	0	
2	2.0	1532	2.0	1	0	0	1	0	0	
3	1.0	305	1.0	1	0	0	0	0	0	
4	2.0	1428	1.0	1	0	0	0	0	0	

In [50]:

```
###Putting the target variable in y
y = data_final_dum['Converted']
y.head()
```

Out[50]:

```
0    0
1    0
2    1
3    0
4    1
Name: Converted, dtype: int64
```

In [51]:

```
###Spliting the dataset into 70% and 30% ratio
X_train, X_test, y_train, y_test = train_test_split(X, y, train_size=0.7, test_size=0.3, random_state=10)
```

In [52]:

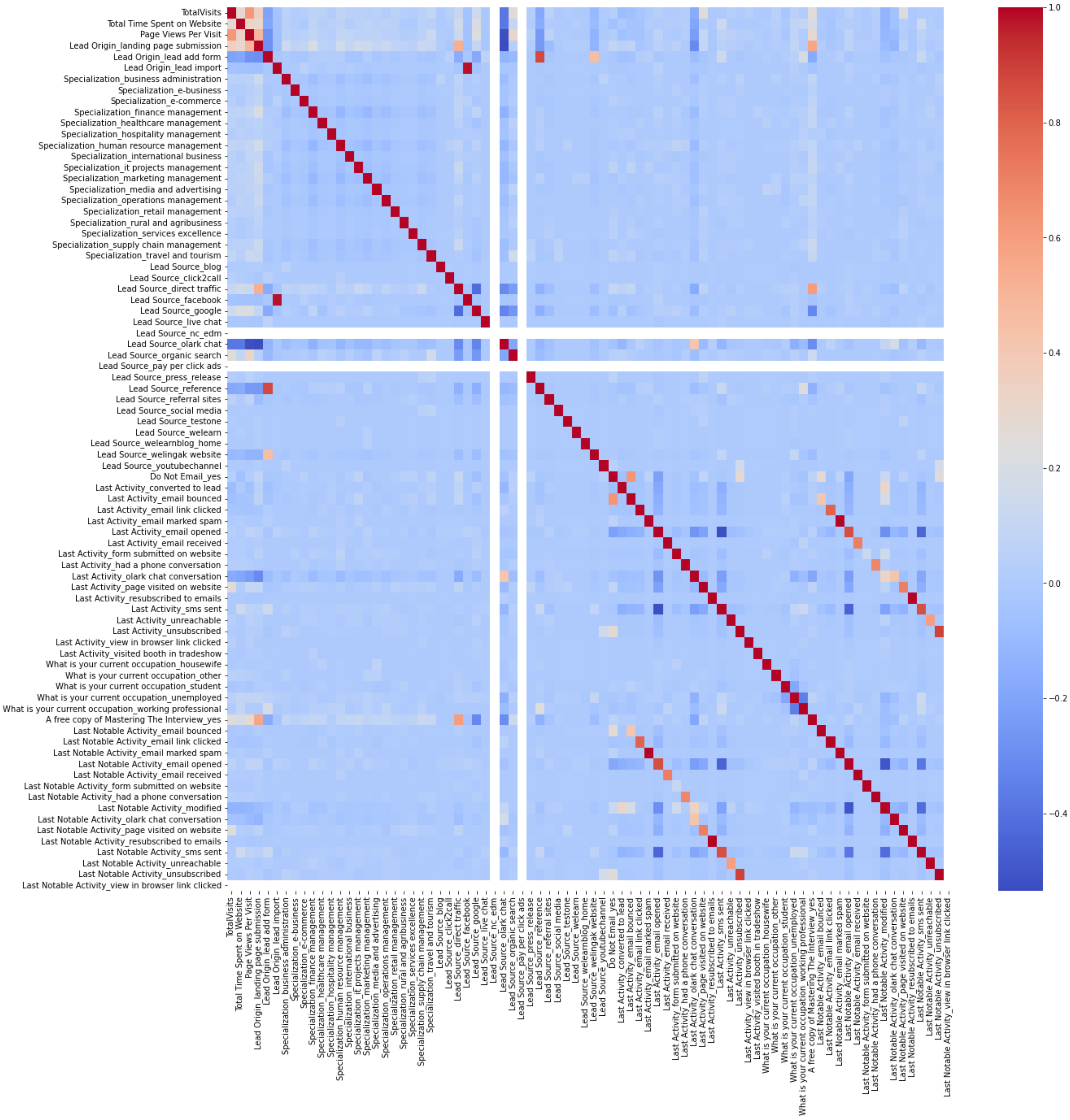
```
###Import MinMax scaler
from sklearn.preprocessing import MinMaxScaler
###Scale the three numeric features
scaler = MinMaxScaler()
X_train[['TotalVisits', 'Page Views Per Visit', 'Total Time Spent on Website']] = scaler.fit_transform(X_train[['TotalVisits', 'Page V
X_train.head()
```

Out[52]:

	TotalVisits	Total Time Spent on Website	Page Views Per Visit	Lead Origin_landing page submission	Lead Origin_lead add form	Lead Origin_lead import	Specialization_business administration	Specialization_e-business	Specialization_e-commerce	Specializatic ma
1289	0.014184	0.612676	0.083333	1	0	0	0	0	0	
3604	0.000000	0.000000	0.000000	0	0	0	0	0	0	
5584	0.042553	0.751761	0.250000	1	0	0	0	0	0	
7679	0.000000	0.000000	0.000000	0	0	0	0	0	0	
7563	0.014184	0.787852	0.083333	1	0	0	0	0	0	

In [53]:

```
###Checking the correlation among varibles
plt.figure(figsize=(20,20))
sns.heatmap(X_train.corr(),cmap='coolwarm')
plt.show()
```



In [54]:

```
###Will use RFE to understand better
### Model Building
```

In [55]:

```
###Import 'LogisticRegression'
from sklearn.linear_model import LogisticRegression
logreg = LogisticRegression()
###Import RFE
from sklearn.feature_selection import RFE
```

In [56]:

```
###Running RFE with 15 variables as output
rfe = RFE(logreg,n_features_to_select=15)
rfe = rfe.fit(X_train, y_train)
```

In [57]:

```python
###Feature that has been selected by RFE
list(zip(X_train.columns, rfe.support_, rfe.ranking_))
```

Out[57]:

```
[('TotalVisits', True, 1),
 ('Total Time Spent on Website', True, 1),
 ('Page Views Per Visit', False, 4),
 ('Lead Origin_landing page submission', False, 26),
 ('Lead Origin_lead add form', True, 1),
 ('Lead Origin_lead import', False, 40),
 ('Specialization_business administration', False, 33),
 ('Specialization_e-business', False, 32),
 ('Specialization_e-commerce', False, 23),
 ('Specialization_finance management', False, 30),
 ('Specialization_healthcare management', False, 25),
 ('Specialization_hospitality management', False, 45),
 ('Specialization_human resource management', False, 31),
 ('Specialization_international business', False, 37),
 ('Specialization_it projects management', False, 28),
 ('Specialization_marketing management', False, 22),
 ('Specialization_media and advertising', False, 42),
 ('Specialization_operations management', False, 27),
 ('Specialization_retail management', False, 63),
 ('Specialization_rural and agribusiness', False, 24),
 ('Specialization_services excellence', False, 21),
 ('Specialization_supply chain management', False, 29),
 ('Specialization_travel and tourism', False, 36),
 ('Lead Source_blog', False, 43),
 ('Lead Source_click2call', False, 62),
 ('Lead Source_direct traffic', False, 14),
 ('Lead Source_facebook', False, 41),
 ('Lead Source_google', False, 16),
 ('Lead Source_live chat', False, 49),
 ('Lead Source_nc_edm', False, 64),
 ('Lead Source_olark chat', True, 1),
 ('Lead Source_organic search', False, 15),
 ('Lead Source_pay per click ads', False, 65),
 ('Lead Source_press_release', False, 52),
 ('Lead Source_reference', False, 3),
 ('Lead Source_referral sites', False, 17),
 ('Lead Source_social media', False, 20),
 ('Lead Source_testone', False, 44),
 ('Lead Source_welearn', False, 46),
 ('Lead Source_welearnblog_home', False, 47),
 ('Lead Source_welingak website', True, 1),
 ('Lead Source_youtubechannel', False, 50),
 ('Do Not Email_yes', True, 1),
 ('Last Activity_converted to lead', False, 11),
 ('Last Activity_email bounced', False, 8),
 ('Last Activity_email link clicked', False, 56),
 ('Last Activity_email marked spam', False, 34),
 ('Last Activity_email opened', False, 61),
 ('Last Activity_email received', False, 55),
 ('Last Activity_form submitted on website', False, 39),
 ('Last Activity_had a phone conversation', False, 2),
 ('Last Activity_olark chat conversation', True, 1),
 ('Last Activity_page visited on website', False, 18),
 ('Last Activity_resubscribed to emails', False, 12),
 ('Last Activity_sms sent', True, 1),
 ('Last Activity_unreachable', False, 19),
 ('Last Activity_unsubscribed', False, 57),
 ('Last Activity_view in browser link clicked', False, 53),
 ('Last Activity_visited booth in tradeshow', False, 54),
 ('What is your current occupation_housewife', True, 1),
 ('What is your current occupation_other', True, 1),
 ('What is your current occupation_student', True, 1),
 ('What is your current occupation_unemployed', True, 1),
 ('What is your current occupation_working professional', True, 1),
 ('A free copy of Mastering The Interview_yes', False, 59),
 ('Last Notable Activity_email bounced', False, 48),
 ('Last Notable Activity_email link clicked', False, 7),
 ('Last Notable Activity_email marked spam', False, 38),
 ('Last Notable Activity_email opened', False, 10),
 ('Last Notable Activity_email received', False, 60),
 ('Last Notable Activity_form submitted on website', False, 58),
 ('Last Notable Activity_had a phone conversation', True, 1),
 ('Last Notable Activity_modified', False, 5),
 ('Last Notable Activity_olark chat conversation', False, 6),
 ('Last Notable Activity_page visited on website', False, 9),
 ('Last Notable Activity_resubscribed to emails', False, 13),
 ('Last Notable Activity_sms sent', False, 51),
 ('Last Notable Activity_unreachable', True, 1),
 ('Last Notable Activity_unsubscribed', False, 35),
 ('Last Notable Activity_view in browser link clicked', False, 66)]
```

In [58]:

```
###PutTING all the columns selected by RFE in the variable 'col'
col = X_train.columns[rfe.support_]
```

In [59]:

```
X_train = X_train[col]
```

In [60]:

```
###Importing statsmodels
import statsmodels.api as sm
```

In [61]:

```
X_train_sm = sm.add_constant(X_train)
logm1 = sm.GLM(y_train, X_train_sm, family = sm.families.Binomial())
res = logm1.fit()
res.summary()
```

Out[61]:

Generalized Linear Model Regression Results

Dep. Variable:	Converted	**No. Observations:**	6351
Model:	GLM	**Df Residuals:**	6335
Model Family:	Binomial	**Df Model:**	15
Link Function:	Logit	**Scale:**	1.0000
Method:	IRLS	**Log-Likelihood:**	-2635.0
Date:	Sun, 16 Apr 2023	**Deviance:**	5270.1
Time:	11:29:52	**Pearson chi2:**	6.48e+03
No. Iterations:	22	**Pseudo R-squ. (CS):**	0.3963
Covariance Type:	nonrobust		

	coef	std err	z	P>|z|	[0.025	0.975]
const	-3.4876	0.114	-30.661	0.000	-3.711	-3.265
TotalVisits	5.4367	1.437	3.782	0.000	2.619	8.254
Total Time Spent on Website	4.6247	0.167	27.689	0.000	4.297	4.952
Lead Origin_lead add form	3.7433	0.225	16.616	0.000	3.302	4.185
Lead Source_olark chat	1.5954	0.112	14.288	0.000	1.377	1.814
Lead Source_welingak website	2.5982	1.033	2.515	0.012	0.574	4.623
Do Not Email_yes	-1.4275	0.170	-8.376	0.000	-1.762	-1.093
Last Activity_olark chat conversation	-1.3875	0.168	-8.281	0.000	-1.716	-1.059
Last Activity_sms sent	1.2834	0.074	17.331	0.000	1.138	1.428
What is your current occupation_housewife	25.4080	3.09e+04	0.001	0.999	-6.05e+04	6.06e+04
What is your current occupation_other	2.1868	0.755	2.895	0.004	0.706	3.667
What is your current occupation_student	1.2705	0.227	5.604	0.000	0.826	1.715
What is your current occupation_unemployed	1.1800	0.086	13.680	0.000	1.011	1.349
What is your current occupation_working professional	3.7057	0.205	18.098	0.000	3.304	4.107
Last Notable Activity_had a phone conversation	24.0110	2.17e+04	0.001	0.999	-4.25e+04	4.26e+04
Last Notable Activity_unreachable	1.8344	0.601	3.051	0.002	0.656	3.013

In [62]:

```
###Importing 'variance_inflation_factor'
from statsmodels.stats.outliers_influence import variance_inflation_factor
```

In [63]:

```
###Calculating the VIF
vif = pd.DataFrame()
vif['Features'] = X_train.columns
vif['VIF'] = [variance_inflation_factor(X_train.values, i) for i in range(X_train.shape[1])]
vif['VIF'] = round(vif['VIF'], 2)
vif = vif.sort_values(by = "VIF", ascending = False)
vif
```

Out[63]:

	Features	VIF
11	What is your current occupation_unemployed	2.30
1	Total Time Spent on Website	2.07
0	TotalVisits	1.85
2	Lead Origin_lead add form	1.59
7	Last Activity_sms sent	1.54
3	Lead Source_olark chat	1.51
6	Last Activity_olark chat conversation	1.37
12	What is your current occupation_working profes...	1.32
4	Lead Source_welingak website	1.31
5	Do Not Email_yes	1.06
10	What is your current occupation_student	1.05
9	What is your current occupation_other	1.01
14	Last Notable Activity_unreachable	1.01
8	What is your current occupation_housewife	1.00
13	Last Notable Activity_had a phone conversation	1.00

In [64]:

```
###VIF seems to be fine but pvalues are not ok
###Dropping last notable column
X_train.drop('Last Notable Activity_had a phone conversation', axis = 1, inplace = True)
```

In [65]:

```
###Running the model again after dropping the column
X_train_sm = sm.add_constant(X_train)
logm2 = sm.GLM(y_train, X_train_sm, family = sm.families.Binomial())
res = logm2.fit()
res.summary()
```

Out[65]:

Generalized Linear Model Regression Results

Dep. Variable:	Converted	**No. Observations:**	6351
Model:	GLM	**Df Residuals:**	6336
Model Family:	Binomial	**Df Model:**	14
Link Function:	Logit	**Scale:**	1.0000
Method:	IRLS	**Log-Likelihood:**	-2642.8
Date:	Sun, 16 Apr 2023	**Deviance:**	5285.6
Time:	11:29:52	**Pearson chi2:**	6.48e+03
No. Iterations:	20	**Pseudo R-squ. (CS):**	0.3948
Covariance Type:	nonrobust		

	coef	std err	z	P>|z|	[0.025	0.975]
const	-3.4831	0.114	-30.629	0.000	-3.706	-3.260
TotalVisits	5.6046	1.450	3.866	0.000	2.763	8.446
Total Time Spent on Website	4.6104	0.167	27.675	0.000	4.284	4.937
Lead Origin_lead add form	3.7375	0.225	16.591	0.000	3.296	4.179
Lead Source_olark chat	1.5910	0.112	14.249	0.000	1.372	1.810
Lead Source_welingak website	2.5984	1.033	2.516	0.012	0.574	4.623
Do Not Email_yes	-1.4324	0.170	-8.409	0.000	-1.766	-1.099
Last Activity_olark chat conversation	-1.3919	0.168	-8.310	0.000	-1.720	-1.064
Last Activity_sms sent	1.2754	0.074	17.245	0.000	1.130	1.420
What is your current occupation_housewife	23.4021	1.14e+04	0.002	0.998	-2.23e+04	2.23e+04
What is your current occupation_other	2.1799	0.755	2.887	0.004	0.700	3.660
What is your current occupation_student	1.2690	0.227	5.600	0.000	0.825	1.713
What is your current occupation_unemployed	1.1852	0.086	13.753	0.000	1.016	1.354
What is your current occupation_working professional	3.7035	0.205	18.099	0.000	3.302	4.105
Last Notable Activity_unreachable	1.8251	0.601	3.036	0.002	0.647	3.003

In [66]:

```
###Calculating the VIF
vif = pd.DataFrame()
vif['Features'] = X_train.columns
vif['VIF'] = [variance_inflation_factor(X_train.values, i) for i in range(X_train.shape[1])]
vif['VIF'] = round(vif['VIF'], 2)
vif = vif.sort_values(by = "VIF", ascending = False)
vif
```

Out[66]:

	Features	VIF
11	What is your current occupation_unemployed	2.30
1	Total Time Spent on Website	2.06
0	TotalVisits	1.85
2	Lead Origin_lead add form	1.59
7	Last Activity_sms sent	1.54
3	Lead Source_olark chat	1.51
6	Last Activity_olark chat conversation	1.37
12	What is your current occupation_working profes...	1.32
4	Lead Source_welingak website	1.31
5	Do Not Email_yes	1.06
10	What is your current occupation_student	1.05
9	What is your current occupation_other	1.01
13	Last Notable Activity_unreachable	1.01
8	What is your current occupation_housewife	1.00

In [67]:

```
###VIF seems fine but Pvalue of What is your current occupation_housewife is not ok
### So dropping the column
X_train.drop('What is your current occupation_housewife', axis = 1, inplace = True)
```

In [68]:

```
###Running the model again after dropping the column
X_train_sm = sm.add_constant(X_train)
logm3 = sm.GLM(y_train, X_train_sm, family = sm.families.Binomial())
res = logm3.fit()
res.summary()
```

Out[68]:

Generalized Linear Model Regression Results

Dep. Variable:	Converted	**No. Observations:**	6351
Model:	GLM	**Df Residuals:**	6337
Model Family:	Binomial	**Df Model:**	13
Link Function:	Logit	**Scale:**	1.0000
Method:	IRLS	**Log-Likelihood:**	-2651.3
Date:	Sun, 16 Apr 2023	**Deviance:**	5302.6
Time:	11:29:52	**Pearson chi2:**	6.50e+03
No. Iterations:	7	**Pseudo R-squ. (CS):**	0.3932
Covariance Type:	nonrobust		

	coef	std err	z	P>|z|	[0.025	0.975]
const	-3.4533	0.113	-30.579	0.000	-3.675	-3.232
TotalVisits	5.5427	1.444	3.838	0.000	2.712	8.373
Total Time Spent on Website	4.6048	0.166	27.690	0.000	4.279	4.931
Lead Origin_lead add form	3.7501	0.225	16.651	0.000	3.309	4.192
Lead Source_olark chat	1.5802	0.111	14.187	0.000	1.362	1.798
Lead Source_welingak website	2.5821	1.033	2.500	0.012	0.558	4.607
Do Not Email_yes	-1.4360	0.170	-8.437	0.000	-1.770	-1.102
Last Activity_olark chat conversation	-1.3974	0.167	-8.348	0.000	-1.725	-1.069
Last Activity_sms sent	1.2672	0.074	17.164	0.000	1.123	1.412
What is your current occupation_other	2.1567	0.755	2.857	0.004	0.677	3.636
What is your current occupation_student	1.2456	0.226	5.502	0.000	0.802	1.689
What is your current occupation_unemployed	1.1632	0.086	13.582	0.000	0.995	1.331
What is your current occupation_working professional	3.6797	0.204	18.008	0.000	3.279	4.080
Last Notable Activity_unreachable	1.8153	0.601	3.022	0.003	0.638	2.993

In [69]:

```
###Calculating the VIF
vif = pd.DataFrame()
vif['Features'] = X_train.columns
vif['VIF'] = [variance_inflation_factor(X_train.values, i) for i in range(X_train.shape[1])]
vif['VIF'] = round(vif['VIF'], 2)
vif = vif.sort_values(by = "VIF", ascending = False)
vif
```

Out[69]:

	Features	VIF
10	What is your current occupation_unemployed	2.30
1	Total Time Spent on Website	2.06
0	TotalVisits	1.85
2	Lead Origin_lead add form	1.58
7	Last Activity_sms sent	1.53
3	Lead Source_olark chat	1.51
6	Last Activity_olark chat conversation	1.37
11	What is your current occupation_working profes...	1.32
4	Lead Source_welingak website	1.31
5	Do Not Email_yes	1.06
9	What is your current occupation_student	1.05
8	What is your current occupation_other	1.01
12	Last Notable Activity_unreachable	1.01

In [70]:

```
###VIF seems fine but Pvalue of What is your current occupation_other is not ok
### So dropping the column
X_train.drop('What is your current occupation_other', axis = 1, inplace = True)
```

In [71]:

```
###Running the model again after dropping the column
X_train_sm = sm.add_constant(X_train)
logm4 = sm.GLM(y_train, X_train_sm, family = sm.families.Binomial())
res = logm4.fit()
res.summary()
```

Out[71]:

Generalized Linear Model Regression Results

Dep. Variable:	Converted	**No. Observations:**	6351
Model:	GLM	**Df Residuals:**	6338
Model Family:	Binomial	**Df Model:**	12
Link Function:	Logit	**Scale:**	1.0000
Method:	IRLS	**Log-Likelihood:**	-2655.8
Date:	Sun, 16 Apr 2023	**Deviance:**	5311.7
Time:	11:29:53	**Pearson chi2:**	6.51e+03
No. Iterations:	7	**Pseudo R-squ. (CS):**	0.3923
Covariance Type:	nonrobust		

	coef	std err	z	P>|z|	[0.025	0.975]
const	-3.4345	0.113	-30.511	0.000	-3.655	-3.214
TotalVisits	5.7276	1.459	3.926	0.000	2.868	8.587
Total Time Spent on Website	4.6142	0.166	27.753	0.000	4.288	4.940
Lead Origin_lead add form	3.7570	0.225	16.676	0.000	3.315	4.199
Lead Source_olark chat	1.5780	0.111	14.159	0.000	1.360	1.796
Lead Source_welingak website	2.5828	1.033	2.501	0.012	0.558	4.607
Do Not Email_yes	-1.4412	0.170	-8.470	0.000	-1.775	-1.108
Last Activity_olark chat conversation	-1.3929	0.167	-8.330	0.000	-1.721	-1.065
Last Activity_sms sent	1.2616	0.074	17.108	0.000	1.117	1.406
What is your current occupation_student	1.2218	0.226	5.401	0.000	0.778	1.665
What is your current occupation_unemployed	1.1394	0.085	13.408	0.000	0.973	1.306
What is your current occupation_working professional	3.6555	0.204	17.914	0.000	3.256	4.055
Last Notable Activity_unreachable	1.8066	0.601	3.008	0.003	0.629	2.984

In [72]:

```
###Calculating the VIF
vif = pd.DataFrame()
vif['Features'] = X_train.columns
vif['VIF'] = [variance_inflation_factor(X_train.values, i) for i in range(X_train.shape[1])]
vif['VIF'] = round(vif['VIF'], 2)
vif = vif.sort_values(by = "VIF", ascending = False)
vif
```

Out[72]:

	Features	VIF
9	What is your current occupation_unemployed	2.29
1	Total Time Spent on Website	2.06
0	TotalVisits	1.84
2	Lead Origin_lead add form	1.58
7	Last Activity_sms sent	1.53
3	Lead Source_olark chat	1.51
6	Last Activity_olark chat conversation	1.37
10	What is your current occupation_working profes...	1.32
4	Lead Source_welingak website	1.31
5	Do Not Email_yes	1.06
8	What is your current occupation_student	1.05
11	Last Notable Activity_unreachable	1.01

In [73]:

```
###VIF & Pvalues seems to be ok
###Creating Predictions
y_train_pred = res.predict(X_train_sm)
y_train_pred[:10]
```

Out[73]:

```
1289    0.648651
3604    0.135107
5584    0.238085
7679    0.135107
7563    0.495064
7978    0.778219
7780    0.169048
7863    0.982785
838     0.772810
708     0.149226
dtype: float64
```

In [74]:

```
###Reshaping to array
y_train_pred = y_train_pred.values.reshape(-1)
y_train_pred[:10]
```

Out[74]:

```
array([0.64865119, 0.135107  , 0.23808524, 0.135107  , 0.49506379,
       0.77821892, 0.16904797, 0.98278528, 0.77281013, 0.14922632])
```

In [75]:

```
###Convertion rate and probablity of predicted ones
y_train_pred_final = pd.DataFrame({'Converted':y_train.values, 'Conversion_Prob':y_train_pred})
y_train_pred_final.head()
```

Out[75]:

	Converted	Conversion_Prob
0	1	0.648651
1	0	0.135107
2	0	0.238085
3	0	0.135107
4	0	0.495064

In [76]:

```
###Cut off as 0.5 & Substituting 0 or 1
y_train_pred_final['Predicted'] = y_train_pred_final.Conversion_Prob.map(lambda x: 1 if x > 0.5 else 0)
y_train_pred_final.head()
```

Out[76]:

	Converted	Conversion_Prob	Predicted
0	1	0.648651	1
1	0	0.135107	0
2	0	0.238085	0
3	0	0.135107	0
4	0	0.495064	0

In [ ]:

```
###Model Evaluation
```

In [77]:

```
###Importing metrics from sklearn for evaluation
from sklearn import metrics
```

In [78]:

```
###Creating confusion matrix
confusion = metrics.confusion_matrix(y_train_pred_final.Converted, y_train_pred_final.Predicted )
confusion
```

Out[78]:

```
array([[3438,  457],
       [ 748, 1708]], dtype=int64)
```

In [ ]:

```
###not churn=3438
###Churn=1708
###Churn but Predicted as non churn=729
###Not Churn but predicted as Churn=457
```

In [80]:

```
###Check the overall accuracy
metrics.accuracy_score(y_train_pred_final.Converted, y_train_pred_final.Predicted)
```

Out[80]:

```
0.810266099826799
```

In [ ]:

```
###81% Acuuracy
```

In [81]:

```
###Substituting the value of true positive
TP = confusion[1,1]
###Substituting the value of true negatives
TN = confusion[0,0]
###Substituting the value of false positives
FP = confusion[0,1]
###Substituting the value of false negatives
FN = confusion[1,0]
```

In [85]:

```
###Calculating sensitivity
TP/(TP+FN)
```

Out[85]:

```
0.6954397394136808
```

In [84]:

```
###Calculating specificity
TN/(TN+FP)
```

Out[84]:

```
0.8826700898587934
```

In [ ]:

```
###Sensitivity=70% &Specificty=88% CutOff=0.5
```

In [ ]:

```
###Optimizing ROC curve
```

In [86]:

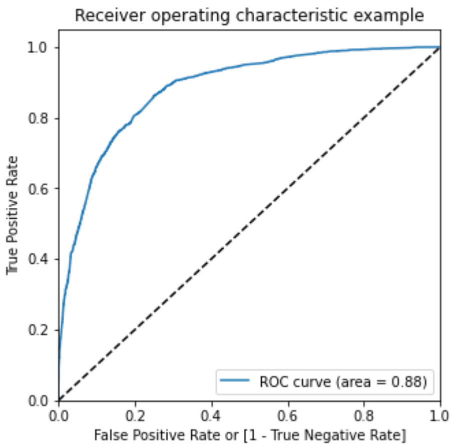
```
###ROC function
def draw_roc( actual, probs ):
    fpr, tpr, thresholds = metrics.roc_curve( actual, probs,
                                              drop_intermediate = False )
    auc_score = metrics.roc_auc_score( actual, probs )
    plt.figure(figsize=(5, 5))
    plt.plot( fpr, tpr, label='ROC curve (area = %0.2f)' % auc_score )
    plt.plot([0, 1], [0, 1], 'k--')
    plt.xlim([0.0, 1.0])
    plt.ylim([0.0, 1.05])
    plt.xlabel('False Positive Rate or [1 - True Negative Rate]')
    plt.ylabel('True Positive Rate')
    plt.title('Receiver operating characteristic example')
    plt.legend(loc="lower right")
    plt.show()

    return None
```

In [87]:

```
pr, tpr, thresholds = metrics.roc_curve( y_train_pred_final.Converted, y_train_pred_final.Conversion_Prob, drop_intermediate = False )
```

In [88]:

```
###Calling the ROC function
draw_roc(y_train_pred_final.Converted, y_train_pred_final.Conversion_Prob)
```



In [ ]:

```
###ROC CURVE IS 0.87
```

In [89]:

```
###Creating columns with different probability cutoffs
numbers = [float(x)/10 for x in range(10)]
for i in numbers:
    y_train_pred_final[i]= y_train_pred_final.Conversion_Prob.map(lambda x: 1 if x > i else 0)
y_train_pred_final.head()
```

Out[89]:

	Converted	Conversion_Prob	Predicted	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
0	1	0.648651	1	1	1	1	1	1	1	1	0	0	0
1	0	0.135107	0	1	1	0	0	0	0	0	0	0	0
2	0	0.238085	0	1	1	1	0	0	0	0	0	0	0
3	0	0.135107	0	1	1	0	0	0	0	0	0	0	0
4	0	0.495064	0	1	1	1	1	1	0	0	0	0	0

In [90]:

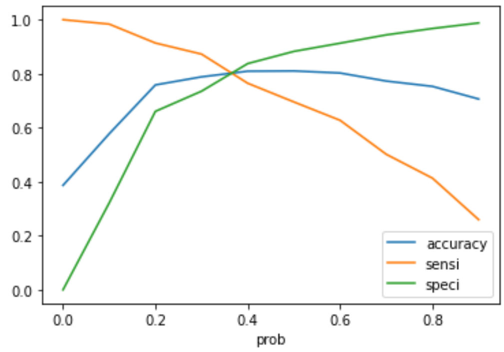
```
###Creating a dataframe to see the values of accuracy, sensitivity, and specificity at different values of probabiity cutoffs
cutoff_df = pd.DataFrame( columns = ['prob','accuracy','sensi','speci'])
###Making confusing matrix to find values of sensitivity, accurace and specificity for each level of probablity
from sklearn.metrics import confusion_matrix
num = [0.0,0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9]
for i in num:
    cm1 = metrics.confusion_matrix(y_train_pred_final.Converted, y_train_pred_final[i] )
    total1=sum(sum(cm1))
    accuracy = (cm1[0,0]+cm1[1,1])/total1

    speci = cm1[0,0]/(cm1[0,0]+cm1[0,1])
    sensi = cm1[1,1]/(cm1[1,0]+cm1[1,1])
    cutoff_df.loc[i] =[ i ,accuracy,sensi,speci]
cutoff_df
```

Out[90]:

	prob	accuracy	sensi	speci
0.0	0.0	0.386711	1.000000	0.000000
0.1	0.1	0.577547	0.983713	0.321438
0.2	0.2	0.758463	0.913681	0.660591
0.3	0.3	0.788380	0.872557	0.735302
0.4	0.4	0.809321	0.764658	0.837484
0.5	0.5	0.810266	0.695440	0.882670
0.6	0.6	0.802551	0.627443	0.912965
0.7	0.7	0.772792	0.501629	0.943774
0.8	0.8	0.753110	0.413274	0.967394
0.9	0.9	0.706345	0.259772	0.987933

In [91]:

```
###Plotting it
cutoff_df.plot.line(x='prob', y=['accuracy','sensi','speci'])
plt.show()
```



In [ ]:

```
### OPTIMAL CUT IS 0.35
```

In [92]:

```
y_train_pred_final['final_predicted'] = y_train_pred_final.Conversion_Prob.map( lambda x: 1 if x > 0.35 else 0)
y_train_pred_final.head()
```

Out[92]:

	Converted	Conversion_Prob	Predicted	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	final_predicted
0	1	0.648651	1	1	1	1	1	1	1	1	0	0	0	1
1	0	0.135107	0	1	1	0	0	0	0	0	0	0	0	0
2	0	0.238085	0	1	1	1	0	0	0	0	0	0	0	0
3	0	0.135107	0	1	1	0	0	0	0	0	0	0	0	0
4	0	0.495064	0	1	1	1	1	1	0	0	0	0	0	1

In [93]:

```
###Checking for overall accuracy
metrics.accuracy_score(y_train_pred_final.Converted, y_train_pred_final.final_predicted)
```

Out[93]:

```
0.8031806014800819
```

In [94]:

```
###Creating confusion matrix
confusion2 = metrics.confusion_matrix(y_train_pred_final.Converted, y_train_pred_final.final_predicted )
confusion2
```

Out[94]:

```
array([[3126,  769],
       [ 481, 1975]], dtype=int64)
```

In [95]:

```
###Substituting the value of true positive
TP = confusion2[1,1]
###Substituting the value of true negatives
TN = confusion2[0,0]
###Substituting the value of false positives
FP = confusion2[0,1]
###Substituting the value of false negatives
FN = confusion2[1,0]
```

In [96]:

```
###Calculating sensitivity
TP/(TP+FN)
```

Out[96]:

```
0.8041530944625407
```

In [97]:

```
###Calculating specificity
TN/(TN+FP)
```

Out[97]:

```
0.8025673940949936
```

In [ ]:

```
###Sensitivity=80% &Specificty=80% Cut0ff=0.35
```

In [ ]:

```
###PREDICTION ON TRAIN SET
```

In [98]:

```
###Scaling numeric values
X_test[['TotalVisits', 'Page Views Per Visit', 'Total Time Spent on Website']] = scaler.transform(X_test[['TotalVisits', 'Page Views Pe
```

In [99]:

```
###Substituting all the columns in the final train model
col = X_train.columns
```

In [100]:

```
###Select the columns in X_train for X_test as well
X_test = X_test[col]
###Adding a constant to X_test
X_test_sm = sm.add_constant(X_test[col])
X_test_sm
X_test_sm
```

Out[100]:

	const	TotalVisits	Total Time Spent on Website	Lead Origin_lead add form	Lead Source_olark chat	Lead Source_welingak website	Do Not Email_yes	Last Activity_olark chat conversation	Last Activity_sms sent	What is your current occupation_student	What i occupatio
8308	1.0	0.035461	0.416813	0	0	0	0	0	0	0	
7212	1.0	0.028369	0.001320	0	0	0	0	0	1	0	
2085	1.0	0.000000	0.000000	1	0	1	0	0	0	0	
4048	1.0	0.028369	0.617077	0	0	0	0	0	1	0	
4790	1.0	0.028369	0.005282	0	0	0	0	0	0	0	
...	...	...	...	...	...	...	...	...	...	...	
3261	1.0	0.000000	0.000000	0	1	0	0	1	0	0	
8179	1.0	0.170213	0.148768	0	0	0	0	0	1	0	
6236	1.0	0.000000	0.000000	0	1	0	0	0	0	0	
5240	1.0	0.078014	0.458627	0	0	0	0	0	1	0	
7243	1.0	0.035461	0.499560	0	0	0	0	0	0	0	

2723 rows × 13 columns

In [101]:

```
###Storing prediction of test set in the variable 'y_test_pred'
y_test_pred = res.predict(X_test_sm)
###Coverting it to df
y_pred_df = pd.DataFrame(y_test_pred)
###Converting y_test to dataframe
y_test_df = pd.DataFrame(y_test)
###Remove index for both dataframes to append them side by side
y_pred_df.reset_index(drop=True, inplace=True)
y_test_df.reset_index(drop=True, inplace=True)
###Append y_test_df and y_pred_df
y_pred_final = pd.concat([y_test_df, y_pred_df],axis=1)
###Renaming column
y_pred_final= y_pred_final.rename(columns = {0 : 'Conversion_Prob'})
y_pred_final.head()
```

Out[101]:

	Converted	Conversion_Prob
0	0	0.457908
1	1	0.839048
2	1	0.982785
3	1	0.878283
4	0	0.108296

In [102]:

```
###Making prediction using cut off 0.35
y_pred_final['final_predicted'] = y_pred_final.Conversion_Prob.map(lambda x: 1 if x > 0.35 else 0)
y_pred_final
```

Out[102]:

	Converted	Conversion_Prob	final_predicted
0	0	0.457908	1
1	1	0.839048	1
2	1	0.982785	1
3	1	0.878283	1
4	0	0.108296	0
...	...	...	...
2718	1	0.108126	0
2719	0	0.374824	1
2720	0	0.135107	0
2721	1	0.821933	1
2722	1	0.553060	1

2723 rows × 3 columns

In [103]:

```
###Check the overall accuracy
metrics.accuracy_score(y_pred_final['Converted'], y_pred_final.final_predicted)
```

Out[103]:

```
0.8094013955196474
```

In [104]:

```
###Creating confusion matrix
confusion2 = metrics.confusion_matrix(y_pred_final['Converted'], y_pred_final.final_predicted )
confusion2
```

Out[104]:

```
array([[1406,  338],
       [ 181,  798]], dtype=int64)
```

In [105]:

```
###Substituting the value of true positive
TP = confusion2[1,1]
###Substituting the value of true negatives
TN = confusion2[0,0]
###Substituting the value of false positives
FP = confusion2[0,1]
###Substituting the value of false negatives
FN = confusion2[1,0]
```

In [106]:

```
###Calculating sensitivity
TP/(TP+FN)
```

Out[106]:

```
0.81511746680286
```

In [107]:

```
###Calculating specificity
TN/(TN+FP)
```

Out[107]:

```
0.8061926605504587
```

In [ ]:

```
###Sensitivity=80% &Specificty=80% CutOff=0.35
```

In [ ]:

```
### PRECISION RECALL
```

In [108]:

```
confusion = metrics.confusion_matrix(y_train_pred_final.Converted, y_train_pred_final.Predicted )
confusion
```

Out[108]:

```
array([[3438,  457],
       [ 748, 1708]], dtype=int64)
```

In [109]:

```
###Precision = TP / TP + FP
confusion[1,1]/(confusion[0,1]+confusion[1,1])
```

Out[109]:

```
0.7889145496535797
```

In [110]:

```
###Recall = TP / TP + FN
confusion[1,1]/(confusion[1,0]+confusion[1,1])
```

Out[110]:

```
0.6954397394136808
```

In [ ]:

```
###Precision at 79% & Recall at 70% with cutoff 0.35
```

In [ ]:

```
### PRECISION & RECALL TRADE OFF
```

In [111]:

```
###Importing libraries
from sklearn.metrics import precision_recall_curve
```

In [112]:

```
y_train_pred_final.Converted, y_train_pred_final.Predicted
```

Out[112]:

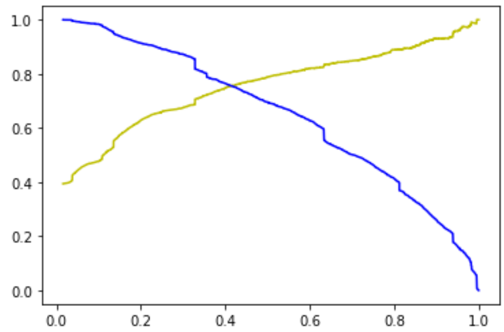
```
(0       1
 1       0
 2       0
 3       0
 4       0
        ..
 6346    0
 6347    0
 6348    0
 6349    0
 6350    1
 Name: Converted, Length: 6351, dtype: int64,
 0       1
 1       0
 2       0
 3       0
 4       0
        ..
 6346    0
 6347    0
 6348    0
 6349    0
 6350    0
 Name: Predicted, Length: 6351, dtype: int64)
```

In [113]:

```
p, r, thresholds = precision_recall_curve(y_train_pred_final.Converted, y_train_pred_final.Conversion_Prob)
```

In [114]:

```
plt.plot(thresholds, p[:-1], "y-")
plt.plot(thresholds, r[:-1], "b-")
plt.show()
```



In [115]:

```
###Making cut off 0.41
y_train_pred_final['final_predicted'] = y_train_pred_final.Conversion_Prob.map(lambda x: 1 if x > 0.41 else 0)
y_train_pred_final.head()
```

Out[115]:

	Converted	Conversion_Prob	Predicted	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	final_predicted
0	1	0.648651	1	1	1	1	1	1	1	1	0	0	0	1
1	0	0.135107	0	1	1	0	0	0	0	0	0	0	0	0
2	0	0.238085	0	1	1	1	0	0	0	0	0	0	0	0
3	0	0.135107	0	1	1	0	0	0	0	0	0	0	0	0
4	0	0.495064	0	1	1	1	1	1	0	0	0	0	0	1

In [116]:

```
###Accuracy
metrics.accuracy_score(y_train_pred_final.Converted, y_train_pred_final.final_predicted)
```

Out[116]:

```
0.8112108329396945
```

In [117]:

```
###Creating confusion matrix again
confusion2 = metrics.confusion_matrix(y_train_pred_final.Converted, y_train_pred_final.final_predicted )
confusion2
```

Out[117]:

```
array([[3289,  606],
       [ 593, 1863]], dtype=int64)
```

In [118]:

```
###Substituting the value of true positive
TP = confusion2[1,1]
###Substituting the value of true negatives
TN = confusion2[0,0]
###Substituting the value of false positives
FP = confusion2[0,1]
###Substituting the value of false negatives
FN = confusion2[1,0]
```

In [119]:

```
###Precision = TP / TP + FP
TP / (TP + FP)
```

Out[119]:

```
0.7545565006075334
```

In [120]:

```
###Recall = TP / TP + FN
TP / (TP + FN)
```

Out[120]:

0.7585504885993485

In [ ]:

```
###Precision at 76% & Recall at 76% with cutoff 0.41
```

In [121]:

```
###PREDICTION ON TEST SET
###Storing prediction of test set in the variable 'y_test_pred'
y_test_pred = res.predict(X_test_sm)
###Coverting it to df
y_pred_df = pd.DataFrame(y_test_pred)
###Converting y_test to dataframe
y_test_df = pd.DataFrame(y_test)
###Remove index for both dataframes to append them side by side
y_pred_df.reset_index(drop=True, inplace=True)
y_test_df.reset_index(drop=True, inplace=True)
###Append y_test_df and y_pred_df
y_pred_final = pd.concat([y_test_df, y_pred_df],axis=1)
###Renaming column
y_pred_final= y_pred_final.rename(columns = {0 : 'Conversion_Prob'})
y_pred_final.head()
```

Out[121]:

	Converted	Conversion_Prob
0	0	0.457908
1	1	0.839048
2	1	0.982785
3	1	0.878283
4	0	0.108296

In [122]:

```
###Making prediction using cut off 0.41
y_pred_final['final_predicted'] = y_pred_final.Conversion_Prob.map(lambda x: 1 if x > 0.41 else 0)
y_pred_final
```

Out[122]:

	Converted	Conversion_Prob	final_predicted
0	0	0.457908	1
1	1	0.839048	1
2	1	0.982785	1
3	1	0.878283	1
4	0	0.108296	0
...	...	...	...
2718	1	0.108126	0
2719	0	0.374824	0
2720	0	0.135107	0
2721	1	0.821933	1
2722	1	0.553060	1

2723 rows × 3 columns

In [123]:

```
###Checking the overall accuracy
metrics.accuracy_score(y_pred_final['Converted'], y_pred_final.final_predicted)
```

Out[123]:

0.8149100257069408

In [124]:

```
###Creating confusion matrix
confusion2 = metrics.confusion_matrix(y_pred_final['Converted'], y_pred_final.final_predicted )
confusion2
```

Out[124]:

```
array([[1472,  272],
       [ 232,  747]], dtype=int64)
```

In [125]:

```
###Substituting the value of true positive
TP = confusion2[1,1]
###Substituting the value of true negatives
TN = confusion2[0,0]
###Substituting the value of false positives
FP = confusion2[0,1]
###Substituting the value of false negatives
FN = confusion2[1,0]
```

In [126]:

```
###Precision = TP / TP + FP
TP / (TP + FP)
```

Out[126]:

```
0.7330716388616291
```

In [127]:

```
###Recall = TP / TP + FN
TP / (TP + FN)
```

Out[127]:

```
0.763023493360572
```

In [ ]:

```
###Precision at 73% & Recall at 76% with cutoff 0.41
```