In [1]:

```python
import pandas as pd
import numpy as np
import warnings
warnings.filterwarnings('ignore')
import matplotlib.pyplot as plt
import seaborn as sns
```

In [2]:

```python
Lead_data = pd.read_csv(r"C:\Users\Shirw\OneDrive\Desktop\Lead Scoring Assignment\Leads.csv")
Lead_data.head()
```

Out[2]:

	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	Asymm Profil
0	7927b2df-8bba-4d29-b9a2-b6e0beafe620	660737	API	Olark Chat	No	No	0	0.0	0	0.0	...	No	Select	Select	02.Medium	02.
1	2a272436-5132-4136-86fa-dcc88c88f482	660728	API	Organic Search	No	No	0	5.0	674	2.5	...	No	Select	Select	02.Medium	02.
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 [3]:

```python
Lead_data.describe()
```

Out[3]:

	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 [4]:

```
Lead_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 [5]:

```
Lead_data.shape
```

Out[5]:

```
(9240, 37)
```

In [6]:

```
# check for duplicate
Lead_data.duplicated(subset = ['Prospect ID'], keep = False).sum()
```

Out[6]:

```
0
```

No duplicate values in Prospect ID and Lead Number

Clearly Prospect ID & Lead Number are two variables that are just indicative of the ID number of the Contacted People & can be dropped.

EXPLORATORY DATA ANALYSIS
Data Cleaning & Treatment:

In [7]:

```
#dropping Lead Number and Prospect ID since they have all unique values

Lead_data.drop(['Prospect ID', 'Lead Number'], 1, inplace = True)
```

In [8]:

```
#Converting 'Select' values to NaN.

Lead_data = Lead_data.replace('Select', np.nan)
```

In [9]:

```
Lead_data.nunique()
```

Out[9]:

```
Lead Origin                                          5
Lead Source                                         21
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 [11]:

```
es About Our Courses','I agree to pay the amount through cheque','Get updates on DM Content','Update me on Supply Chain Content'],axis=1)
```

In [12]:

```
#checking null values in each rows

Lead_data.isnull().sum()
```

Out[12]:

```
Lead Origin                                         0
Lead Source                                        36
Do Not Email                                        0
Do Not Call                                         0
Converted                                           0
TotalVisits                                       137
Total Time Spent on Website                         0
Page Views Per Visit                              137
Last Activity                                     103
Country                                          2461
Specialization                                   3380
How did you hear about X Education               7250
What is your current occupation                  2690
What matters most to you in choosing a course    2709
Search                                              0
Newspaper Article                                   0
X Education Forums                                  0
Newspaper                                           0
Digital Advertisement                               0
Through Recommendations                             0
Tags                                             3353
Lead Quality                                     4767
Lead Profile                                     6855
City                                             3669
Asymmetrique Activity Index                      4218
Asymmetrique Profile Index                       4218
Asymmetrique Activity Score                      4218
Asymmetrique Profile Score                       4218
A free copy of Mastering The Interview              0
Last Notable Activity                               0
dtype: int64
```

In [13]:

```
# % of null value
round(100*(Lead_data.isnull().sum())/len(Lead_data.index),2)
```

Out[13]:

```
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 [14]:

```
#dropping cols with more than 45% missing values

Lead_data = Lead_data.drop(['Asymmetrique Profile Score','Asymmetrique Activity Score','Asymmetrique Profile Index','Asymmetrique Activity
```

In [15]:

```
Lead_data.shape
```

Out[15]:

```
(9240, 23)
```

In [16]:

```
#checking null values percentage

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

Out[16]:

```
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
Tags                                             36.29
City                                             39.71
A free copy of Mastering The Interview            0.00
Last Notable Activity                             0.00
dtype: float64
```

There is a huge value of null variables in some columns as seen above. But removing the rows with the null value will cost us a lot of data and they are important columns. So, instead we are going to replace the NaN values with 'not provided'. This way we have all the data and almost no null values. In case these come up in the model, it will be of no use and we can drop it off then.

In [17]:

```
Lead_data['Specialization'] = Lead_data['Specialization'].fillna('not provided')
Lead_data['City'] = Lead_data['City'].fillna('not provided')
Lead_data['Tags'] = Lead_data['Tags'].fillna('not provided')
Lead_data['What matters most to you in choosing a course'] = Lead_data['What matters most to you in choosing a course'].fillna('not provi
Lead_data['What is your current occupation'] = Lead_data['What is your current occupation'].fillna('not provided')
Lead_data['Country'] = Lead_data['Country'].fillna('not provided')
Lead_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9240 entries, 0 to 9239
Data columns (total 23 columns):
 #   Column                                         Non-Null Count  Dtype  
---  ------                                         --------------  -----  
 0   Lead Origin                                    9240 non-null   object 
 1   Lead Source                                    9204 non-null   object 
 2   Do Not Email                                   9240 non-null   object 
 3   Do Not Call                                    9240 non-null   object 
 4   Converted                                      9240 non-null   int64  
 5   TotalVisits                                    9103 non-null   float64
 6   Total Time Spent on Website                    9240 non-null   int64  
 7   Page Views Per Visit                           9103 non-null   float64
 8   Last Activity                                  9137 non-null   object 
 9   Country                                        9240 non-null   object 
 10  Specialization                                 9240 non-null   object 
 11  What is your current occupation                9240 non-null   object 
 12  What matters most to you in choosing a course  9240 non-null   object 
 13  Search                                         9240 non-null   object 
 14  Newspaper Article                              9240 non-null   object 
 15  X Education Forums                             9240 non-null   object 
 16  Newspaper                                      9240 non-null   object 
 17  Digital Advertisement                          9240 non-null   object 
 18  Through Recommendations                        9240 non-null   object 
 19  Tags                                           9240 non-null   object 
 20  City                                           9240 non-null   object 
 21  A free copy of Mastering The Interview         9240 non-null   object 
 22  Last Notable Activity                          9240 non-null   object 
dtypes: float64(2), int64(2), object(19)
memory usage: 1.6+ MB
```

In [18]:

```
#checking null values percentage

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

Out[18]:

```
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
Tags                                             0.00
City                                             0.00
A free copy of Mastering The Interview           0.00
Last Notable Activity                            0.00
dtype: float64
```

In [19]:

```
Lead_data.shape
```

Out[19]:

```
(9240, 23)
```

Categorical Attributes Analysis:

In [20]:

```
Lead_data['Country'].value_counts()
```

Out[20]:

```
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 [21]:

```
def slots(x):
    category = ""
    if x == "India":
        category = "India"
    elif x == "not provided":
        category = "not provided"
    else:
        category = "outside india"
    return category

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

Out[21]:

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

In [22]:

```
# Since India is the most common occurence among the non-missing values we can impute all not provided values with India

Lead_data['Country'] = Lead_data['Country'].replace('not provided','India')
Lead_data['Country'].value_counts()
```

Out[22]:

```
India            8953
outside india     287
Name: Country, dtype: int64
```

In [23]:

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

Out[23]:

```
1.48
```

In [24]:

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

In [25]:

```
# Rechecking the percentage of missing values
round(100*(Lead_data.isnull().sum()/len(Lead_data.index)), 2)
```

Out[25]:

```
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
Tags                                             0.0
City                                             0.0
A free copy of Mastering The Interview           0.0
Last Notable Activity                            0.0
dtype: float64
```

In [26]:

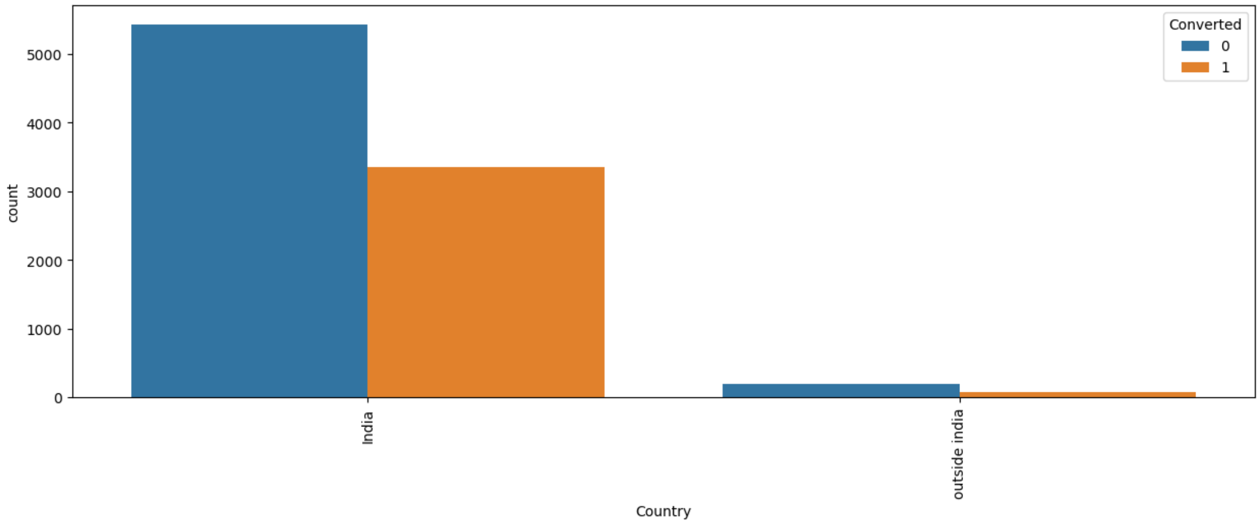
```
Lead_data.shape
```

Out[26]:

```
(9074, 23)
```

In [27]:

```
#plotting spread of Country columnn after replacing NaN values

plt.figure(figsize=(15,5))
s1=sns.countplot(Lead_data.Country, hue=Lead_data.Converted)
s1.set_xticklabels(s1.get_xticklabels(),rotation=90)
plt.show()
```



As we can see the Number of Values for India are quite high (nearly 97% of the Data), this column can be dropped

In [28]:

```
#creating a list of columns to be droppped

cols_to_drop=['Country']
```

In [29]:

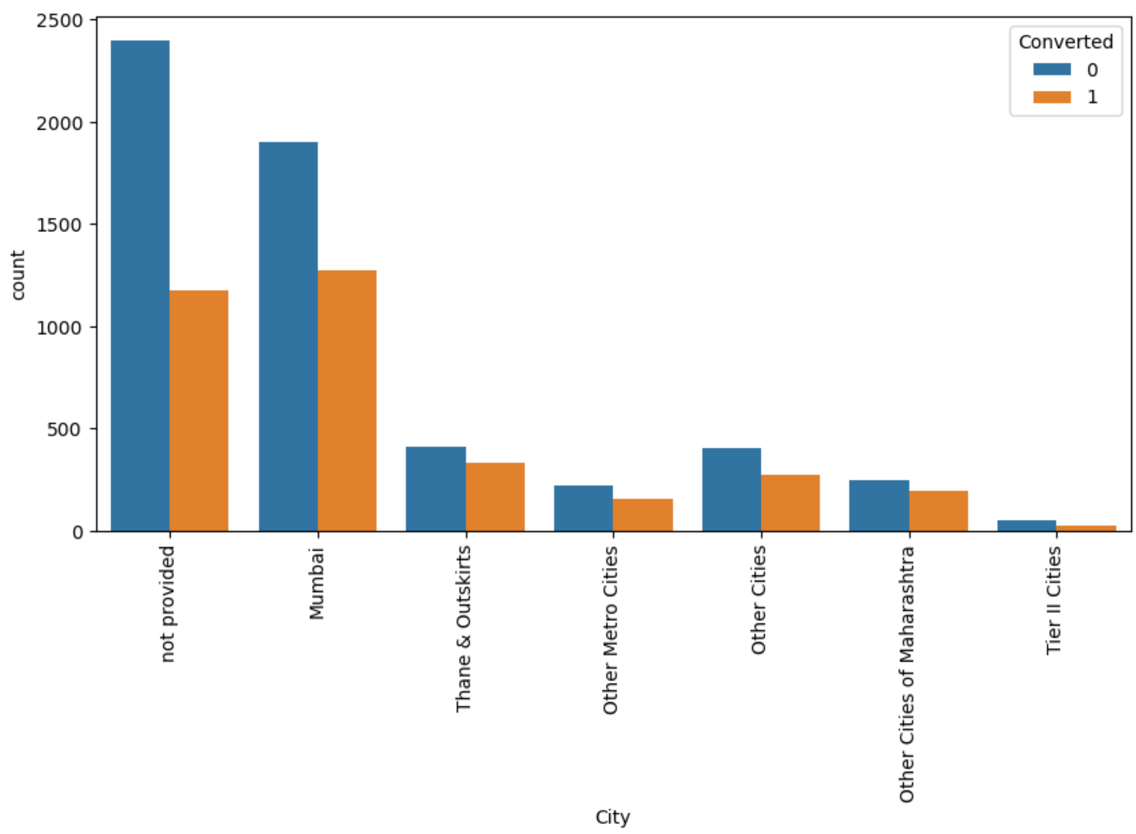
```
#checking value counts of "City" column

Lead_data['City'].value_counts(dropna=False)
```

Out[29]:

```
not provided                   3575
Mumbai                         3177
Thane & Outskirts               745
Other Cities                    680
Other Cities of Maharashtra     446
Other Metro Cities              377
Tier II Cities                   74
Name: City, dtype: int64
```

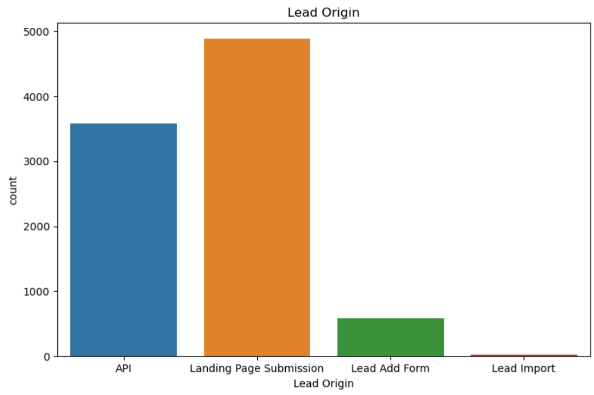
In [30]:

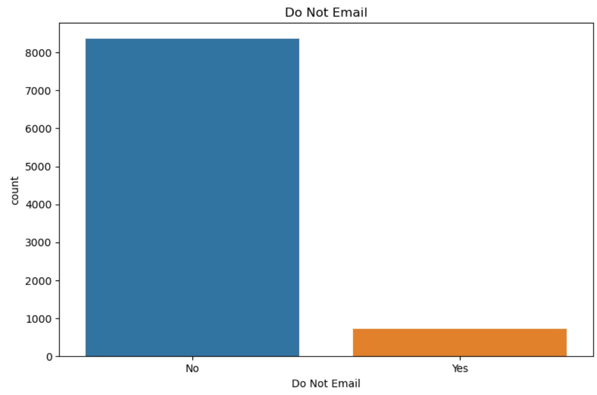
```
#plotting spread of City columnn

plt.figure(figsize=(10,5))
s1=sns.countplot(Lead_data.City, hue=Lead_data.Converted)
s1.set_xticklabels(s1.get_xticklabels(),rotation=90)
plt.show()
```



In [31]:

```
plt.figure(figsize = (20,40))

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

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

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

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

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

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

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

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

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

plt.subplot(6,2,11)
sns.countplot(Lead_data['A free copy of Mastering The Interview'])
plt.title('A free copy of Mastering The Interview')
plt.subplot(6,2,12)
sns.countplot(Lead_data['Last Notable Activity']).tick_params(axis='x', rotation = 90)
plt.title('Last Notable Activity')


plt.show()
```



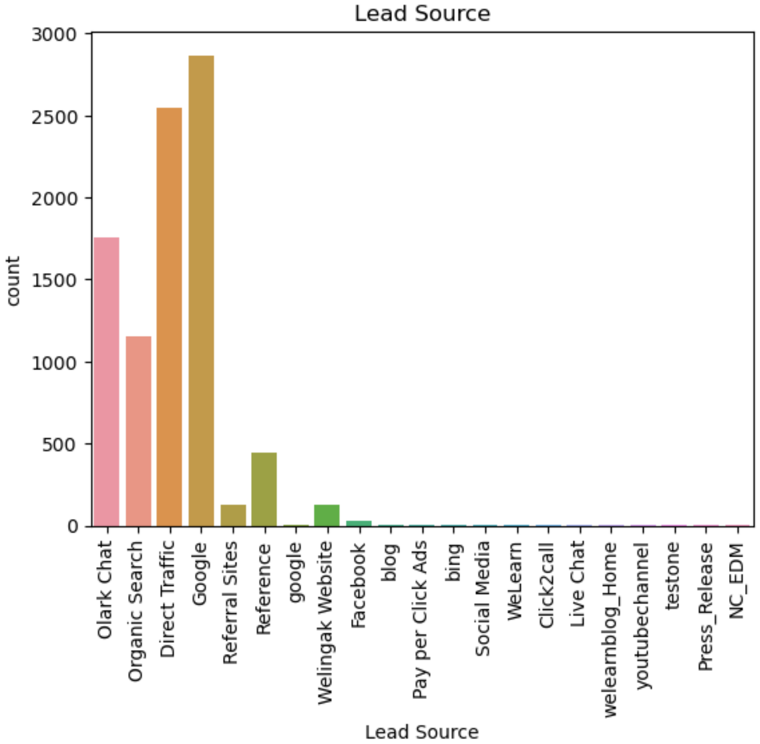


Do Not Call

Country

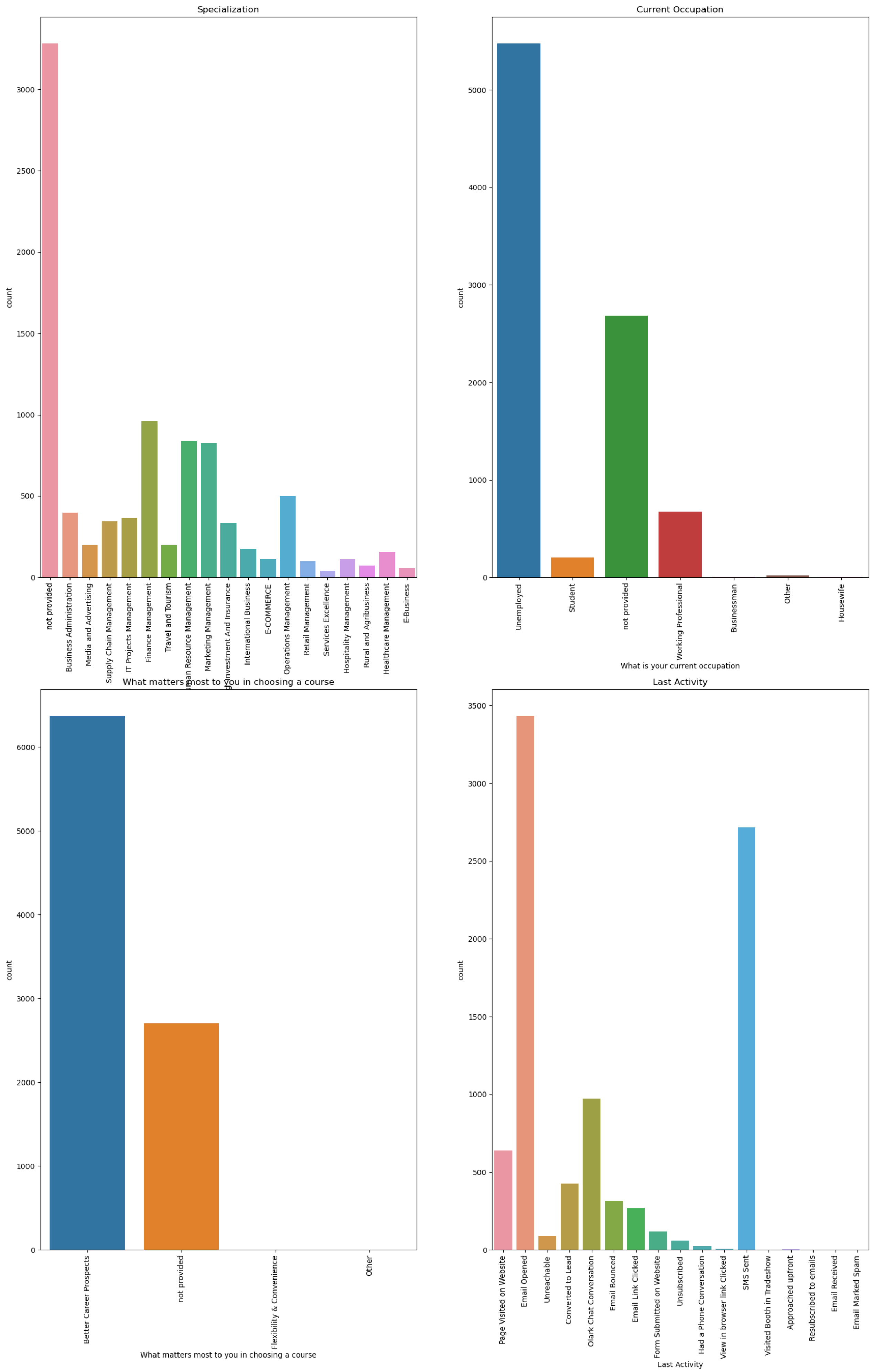
In [32]:

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



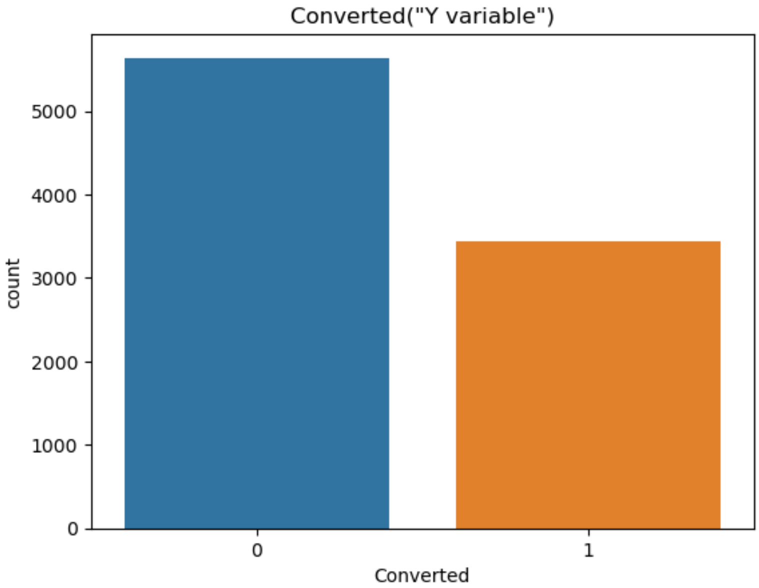
In [33]:

```
plt.figure(figsize = (20,30))
plt.subplot(2,2,1)
sns.countplot(Lead_data['Specialization']).tick_params(axis='x', rotation = 90)
plt.title('Specialization')
plt.subplot(2,2,2)
sns.countplot(Lead_data['What is your current occupation']).tick_params(axis='x', rotation = 90)
plt.title('Current Occupation')
plt.subplot(2,2,3)
sns.countplot(Lead_data['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(Lead_data['Last Activity']).tick_params(axis='x', rotation = 90)
plt.title('Last Activity')
plt.show()
```


Specialization
count
0
500
1000
1500
2000
2500
3000
not provided
Business Administration
Media and Advertising
Supply Chain Management
IT Projects Management
Finance Management
Travel and Tourism
Human Resource Management
Marketing Management
Banking, Investment And Insurance
International Business
E-COMMERCE
Operations Management
Retail Management
Services Excellence
Hospitality Management
Rural and Agribusiness
Healthcare Management
E-Business
Current Occupation
count
0
1000
2000
3000
4000
5000
Unemployed
Student
not provided
Working Professional
Businessman
Other
Housewife
What is your current occupation
What matters most to you in choosing a course
count
0
1000
2000
3000
4000
5000
6000
Better Career Prospects
not provided
Flexibility & Convenience
Other
What matters most to you in choosing a course
Last Activity
count
0
500
1000
1500
2000
2500
3000
3500
Page Visited on Website
Email Opened
Unreachable
Converted to Lead
Olark Chat Conversation
Email Bounced
Email Link Clicked
Form Submitted on Website
Unsubscribed
Had a Phone Conversation
View in browser link Clicked
SMS Sent
Visited Booth in Tradeshow
Approached upfront
Resubscribed to emails
Email Received
Email Marked Spam
Last Activity

In [34]:

```
sns.countplot(Lead_data['Converted'])
plt.title('Converted("Y variable")')
plt.show()
```

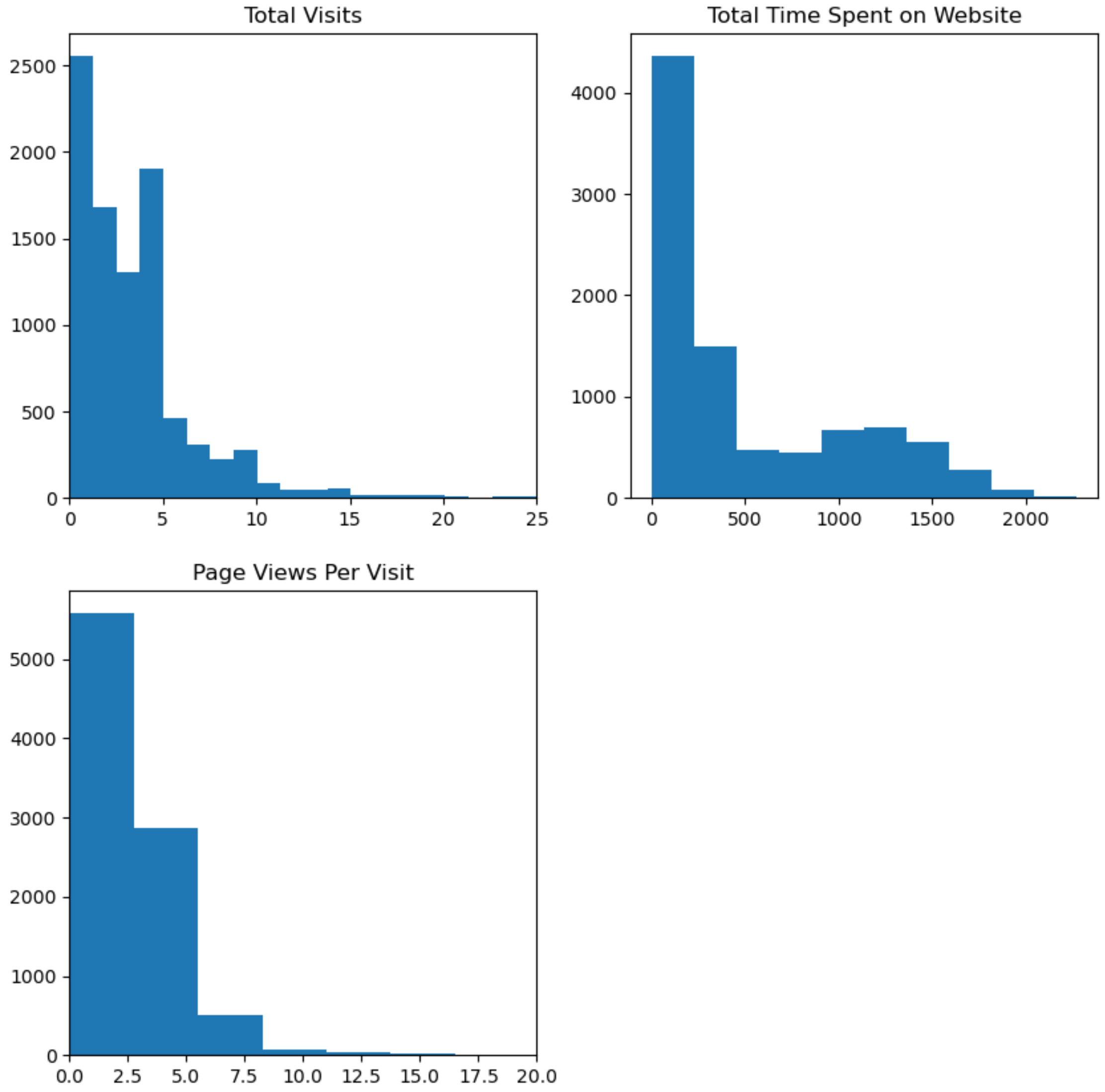


In [ ]:

```
##Numerical Variables
```

In [35]:

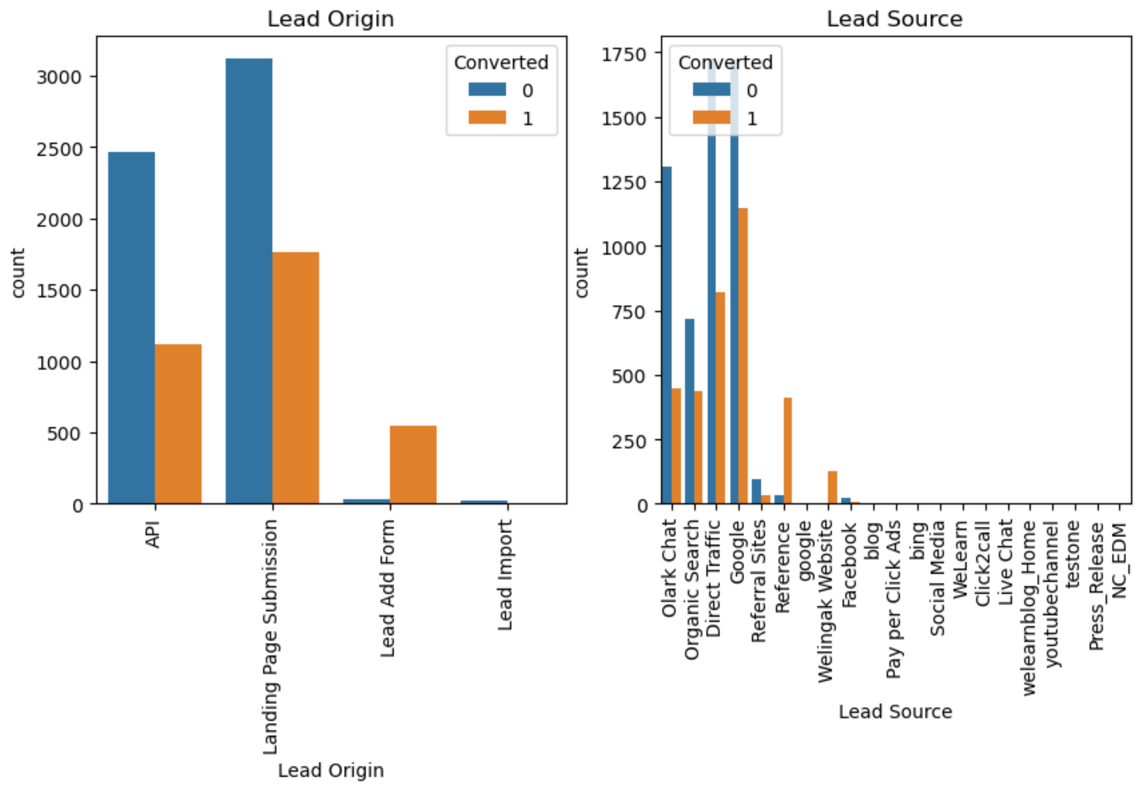
```
plt.figure(figsize = (10,10))
plt.subplot(221)
plt.hist(Lead_data['TotalVisits'], bins = 200)
plt.title('Total Visits')
plt.xlim(0,25)

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

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



Relating all the categorical variables to Converted

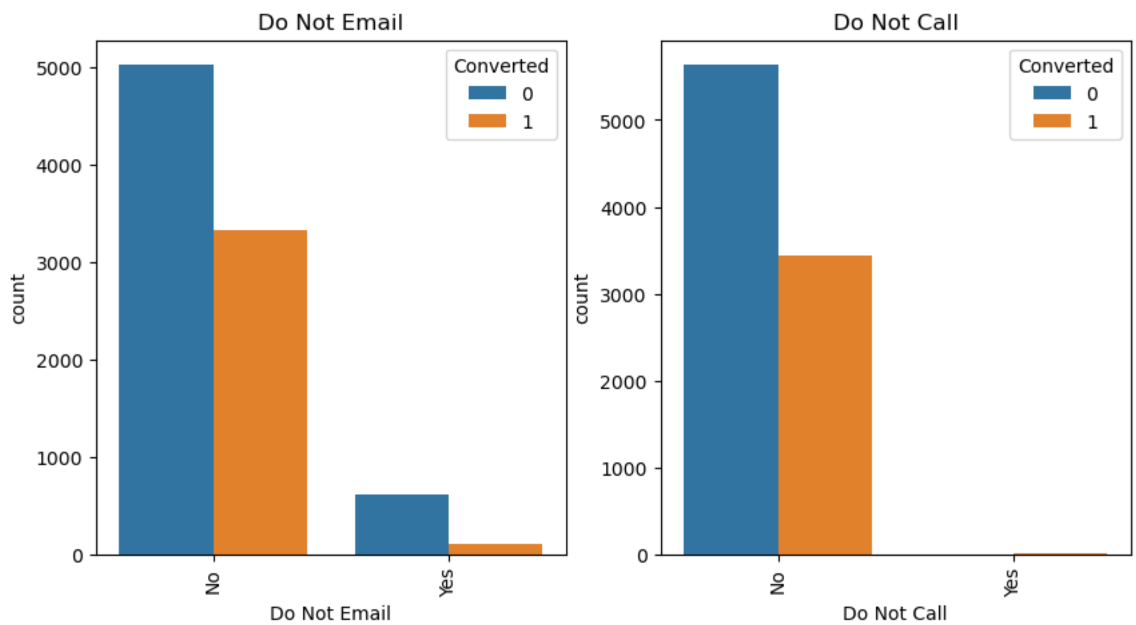
In [36]:

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

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

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



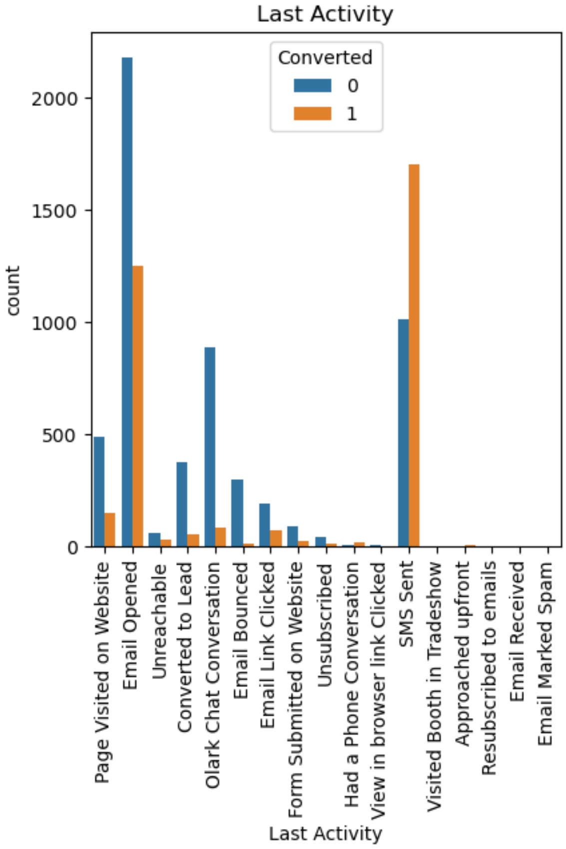
In [37]:

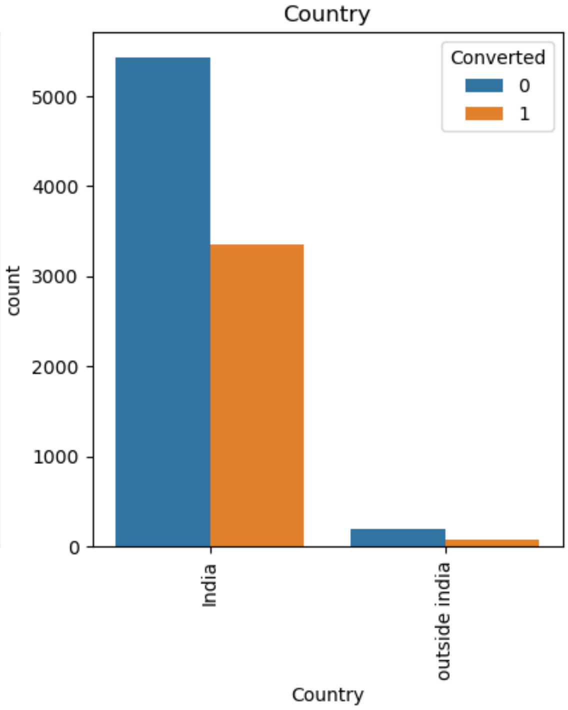
```
plt.figure(figsize=(10 ,5))
plt.subplot(1,2,1)
sns.countplot(x='Do Not Email', hue='Converted', data= Lead_data).tick_params(axis='x', rotation = 90)
plt.title('Do Not Email')

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



In [38]:

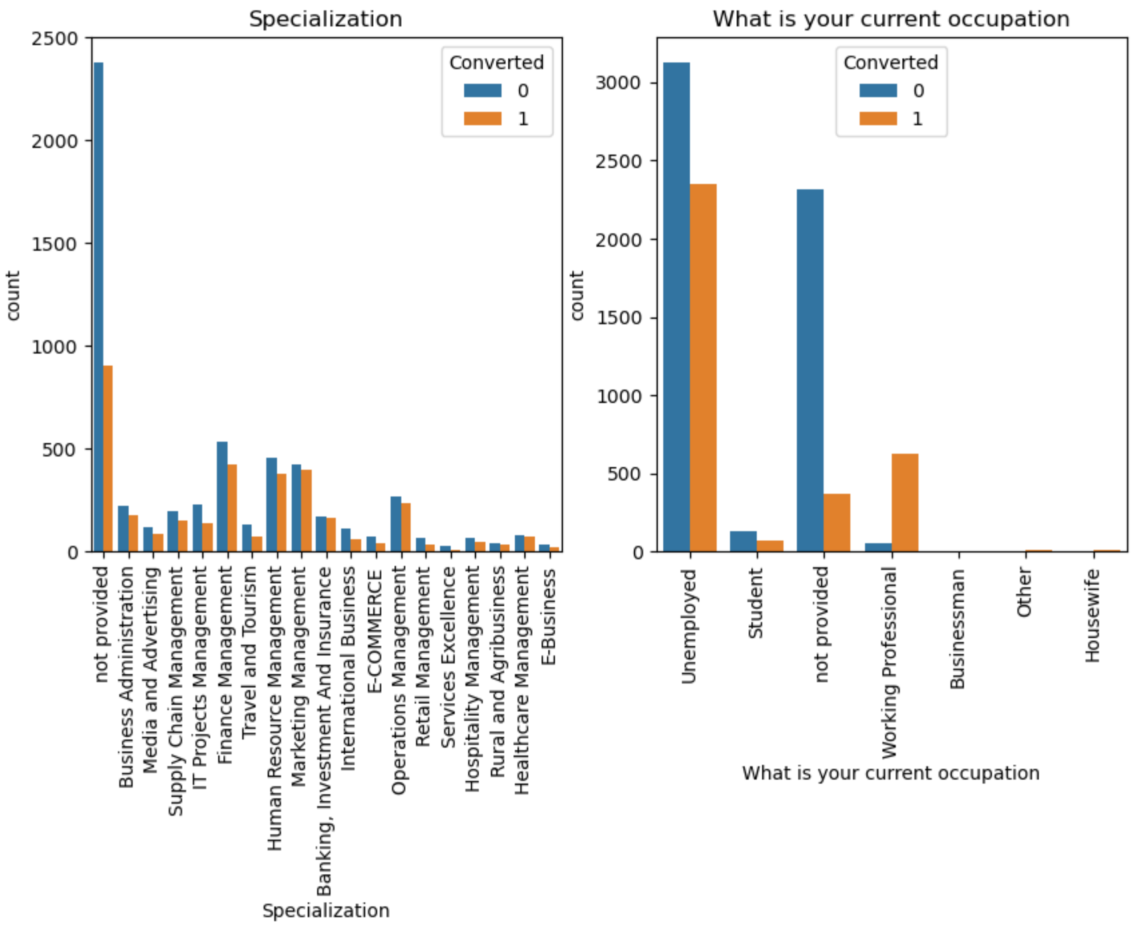
```
plt.figure(figsize = (10,5))

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

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





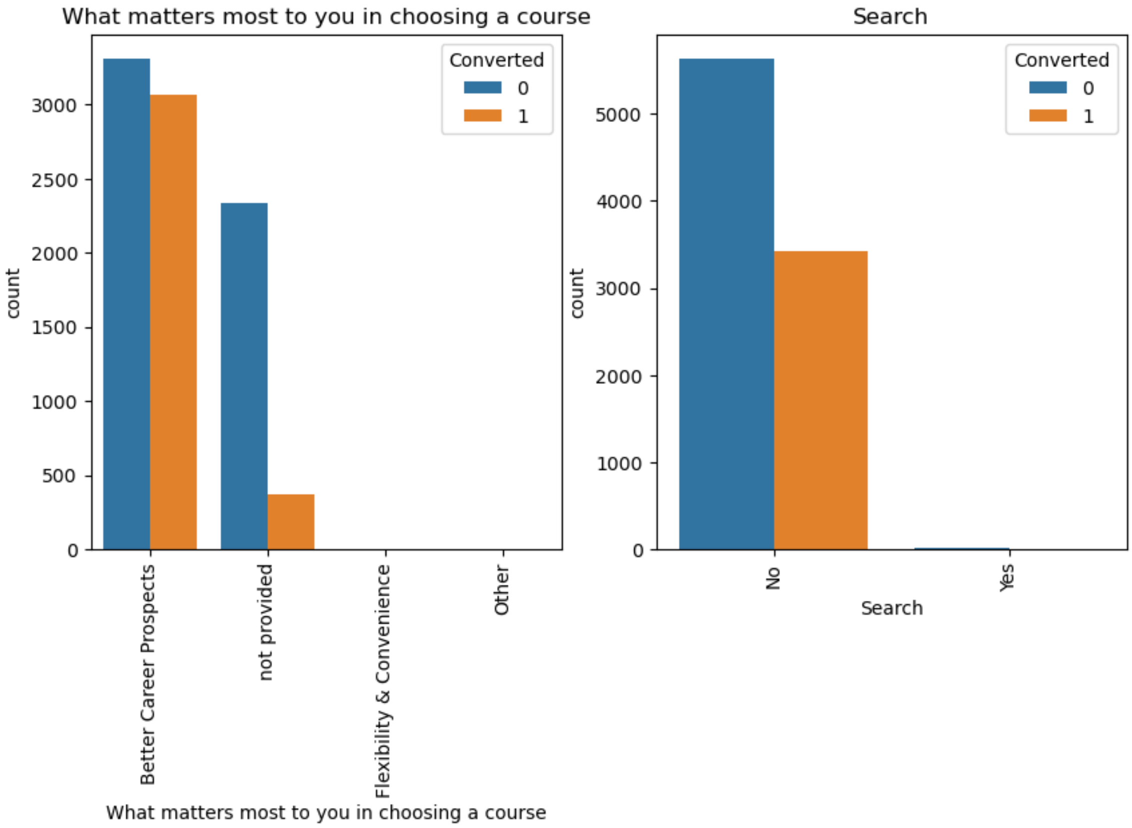
In [39]:

```
plt.figure(figsize = (10,5))

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



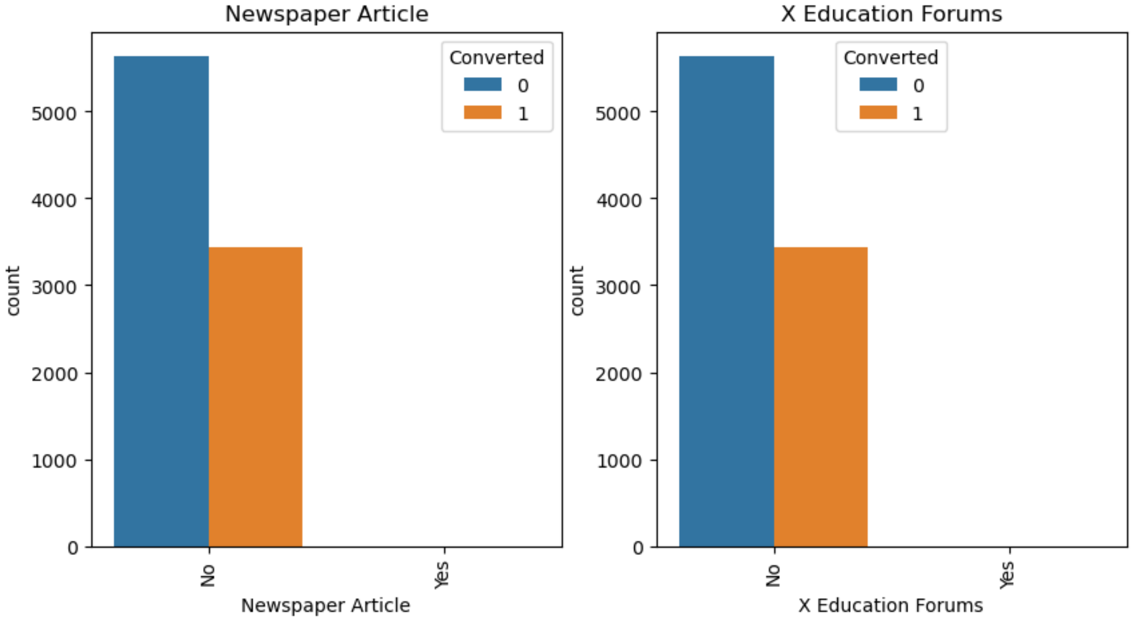
In [40]:

```
plt.figure(figsize = (10,5))

plt.subplot(1,2,1)
sns.countplot(x='What matters most to you in choosing a course', hue='Converted', data= Lead_data).tick_params(axis='x', rotation = 90)
plt.title('What matters most to you in choosing a course')

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



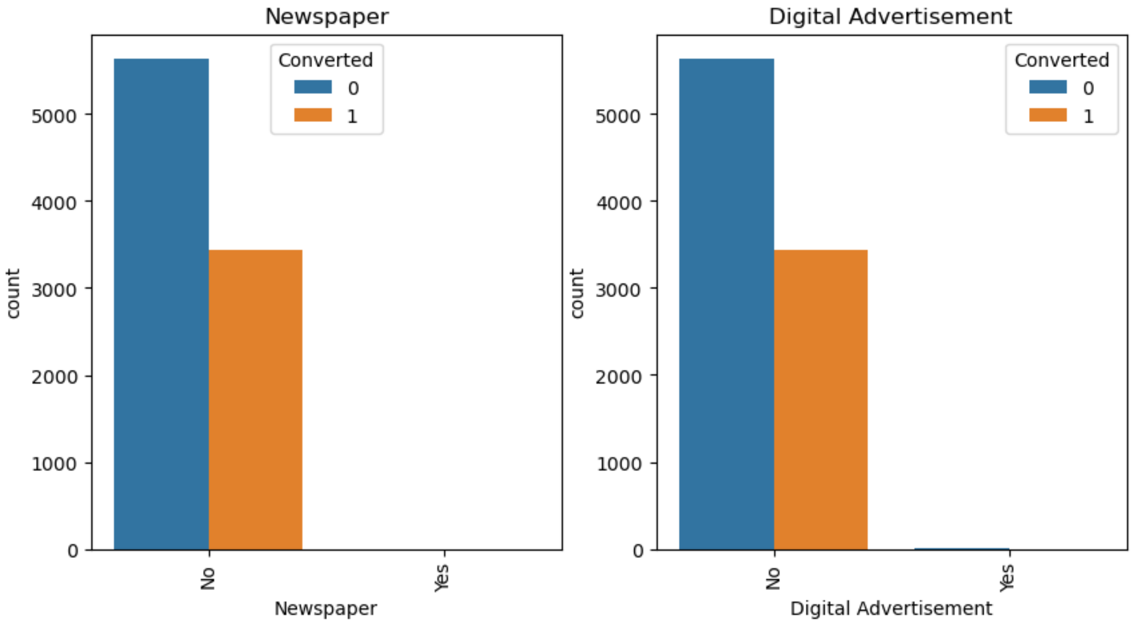
In [41]:

```
plt.figure(figsize = (10,5))

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

plt.subplot(1,2,2)
sns.countplot(x='X Education Forums', hue='Converted', data= Lead_data).tick_params(axis='x', rotation = 90)
plt.title('X Education Forums')
plt.show()
```



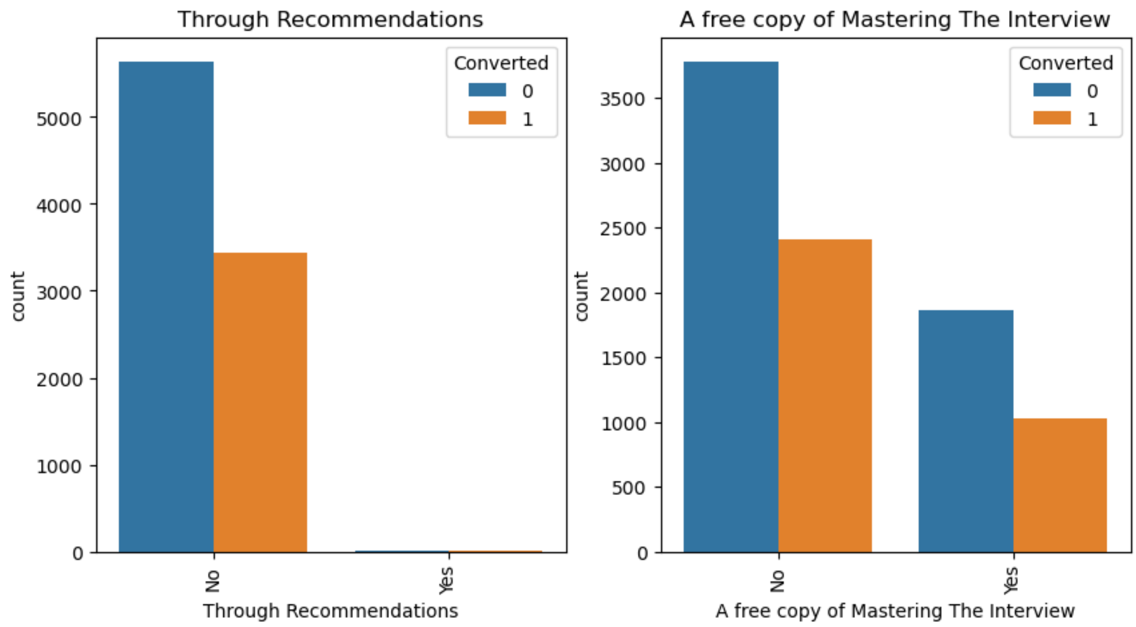
In [42]:

```
plt.figure(figsize = (10,5))

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

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



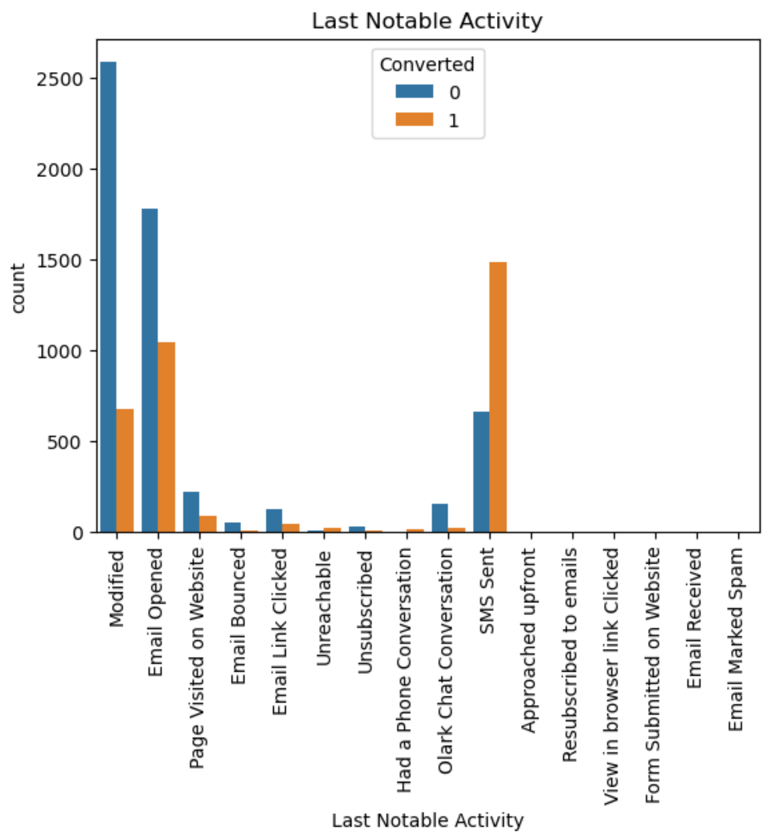
In [43]:

```
plt.figure(figsize = (10,5))

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



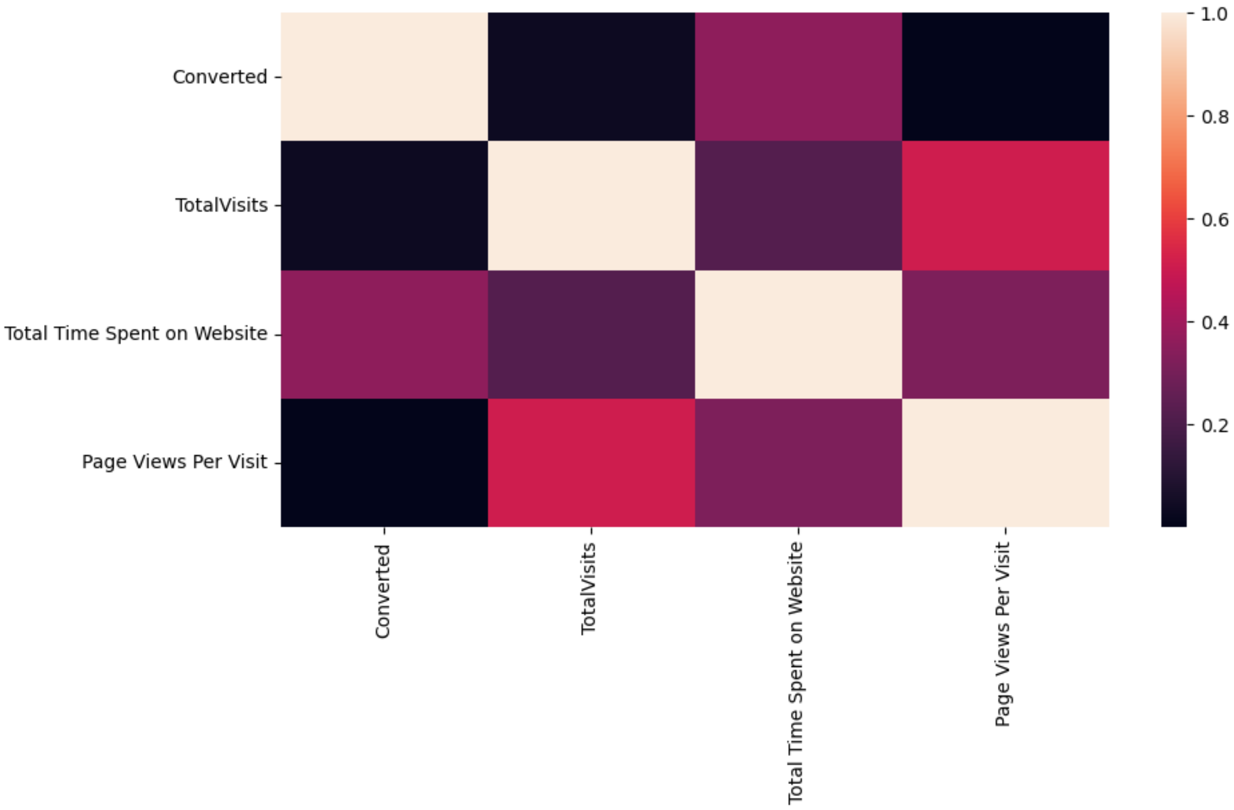
In [44]:

```
sns.countplot(x='Last Notable Activity', hue='Converted', data= Lead_data).tick_params(axis='x', rotation = 90)
plt.title('Last Notable Activity')
plt.show()
```



In [45]:

```
# To check the correlation among varibles
plt.figure(figsize=(10,5))
sns.heatmap(Lead_data.corr())
plt.show()
```



It is understandable from the above EDA that there are many elements that have very little data and so will be of less relevance to our analysis.

Outlier

In [46]:

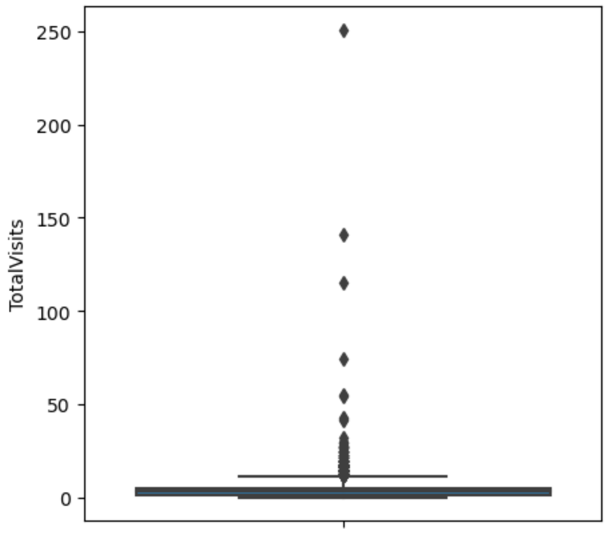
```
numeric = Lead_data[['TotalVisits','Total Time Spent on Website','Page Views Per Visit']]
numeric.describe(percentiles=[0.25,0.5,0.75,0.9,0.99])
```

Out[46]:

	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

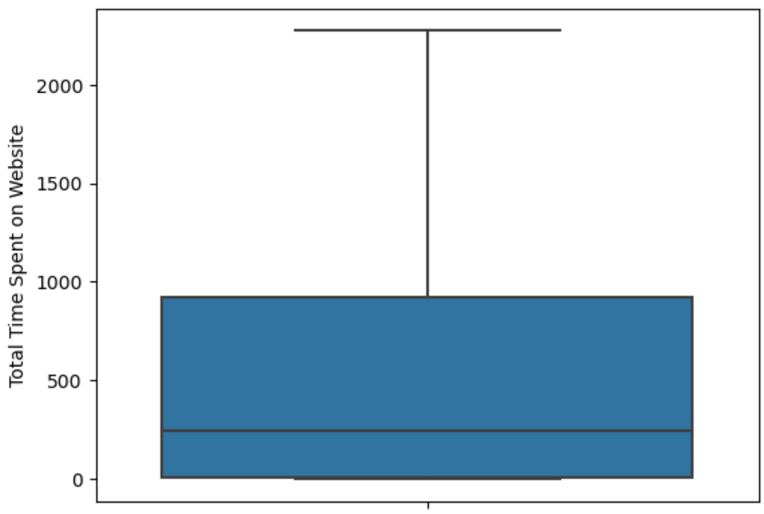
In [47]:

```
plt.figure(figsize = (5,5))
sns.boxplot(y=Lead_data['TotalVisits'])
plt.show()
```



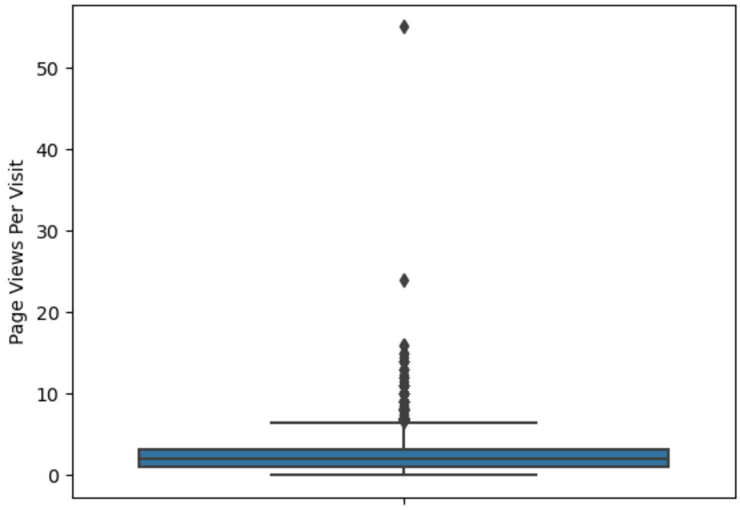
In [48]:

```
sns.boxplot(y=Lead_data['Total Time Spent on Website'])
plt.show()
```



In [49]:

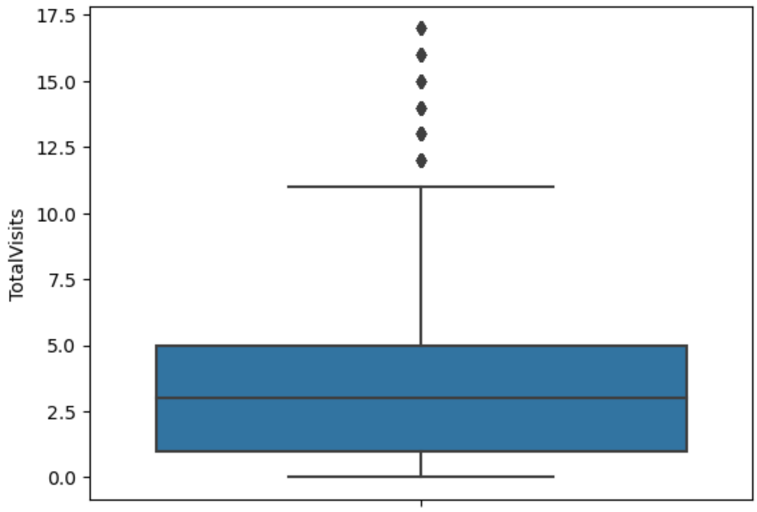
```
sns.boxplot(y=Lead_data['Page Views Per Visit'])
plt.show()
```



We can see presence of outliers in TotalVisits

In [50]:

```
#Outlier Treatment: Remove top & bottom 1% of the Column Outlier values

Q3 = Lead_data.TotalVisits.quantile(0.99)
Lead_data = Lead_data[(Lead_data.TotalVisits <= Q3)]
Q1 = Lead_data.TotalVisits.quantile(0.01)
Lead_data = Lead_data[(Lead_data.TotalVisits >= Q1)]
sns.boxplot(y=Lead_data['TotalVisits'])
plt.show()
```



Dummy Variables

In [51]:

```
#list of columns to be dropped
cols_to_drop=['Country','Tags']
```

We can drop "Tags" ,As tags variable is generated by the sales sales team after the disscussion with student otherwise it will increase the model accuracy .

In [52]:

```
#dropping columns
Lead_data = Lead_data.drop(cols_to_drop,1)
Lead_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 8991 entries, 0 to 9239
Data columns (total 21 columns):
 #   Column                                         Non-Null Count  Dtype  
---  ------                                         --------------  -----  
 0   Lead Origin                                    8991 non-null   object 
 1   Lead Source                                    8991 non-null   object 
 2   Do Not Email                                   8991 non-null   object 
 3   Do Not Call                                    8991 non-null   object 
 4   Converted                                      8991 non-null   int64  
 5   TotalVisits                                    8991 non-null   float64
 6   Total Time Spent on Website                    8991 non-null   int64  
 7   Page Views Per Visit                           8991 non-null   float64
 8   Last Activity                                  8991 non-null   object 
 9   Specialization                                 8991 non-null   object 
 10  What is your current occupation                8991 non-null   object 
 11  What matters most to you in choosing a course  8991 non-null   object 
 12  Search                                         8991 non-null   object 
 13  Newspaper Article                              8991 non-null   object 
 14  X Education Forums                             8991 non-null   object 
 15  Newspaper                                      8991 non-null   object 
 16  Digital Advertisement                          8991 non-null   object 
 17  Through Recommendations                        8991 non-null   object 
 18  City                                           8991 non-null   object 
 19  A free copy of Mastering The Interview         8991 non-null   object 
 20  Last Notable Activity                          8991 non-null   object 
dtypes: float64(2), int64(2), object(17)
memory usage: 1.5+ MB
```

In [53]:

```
#getting a list of categorical columns

cat_cols= Lead_data.select_dtypes(include=['object']).columns
cat_cols
```

Out[53]:

```
Index(['Lead Origin', 'Lead Source', 'Do Not Email', 'Do Not Call',
       'Last Activity', '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', 'City',
       'A free copy of Mastering The Interview', 'Last Notable Activity'],
      dtype='object')
```

In [54]:

```
# Create dummy variables using the 'get_dummies'
dummy = pd.get_dummies(Lead_data[['Lead Origin','Specialization' ,'Lead Source', 'Do Not Email', 'Last Activity', 'What is your current oc
# Add the results to the master dataframe
Lead_data_dum = pd.concat([Lead_data, dummy], axis=1)
Lead_data_dum
```

Out[54]:

	Lead Origin	Lead Source	Do Not Email	Do Not Call	Converted	TotalVisits	Total Time Spent on Website	Page Views Per Visit	Last Activity	Specialization	...	Last Notable Activity_Form Submitted on Website	Last Notable Activity_Had a Phone Conversation	Last Notal Activity_Modifi
0	API	Olark Chat	No	No	0	0.0	0	0.00	Page Visited on Website	not provided	...	0	0	
1	API	Organic Search	No	No	0	5.0	674	2.50	Email Opened	not provided	...	0	0	
2	Landing Page Submission	Direct Traffic	No	No	1	2.0	1532	2.00	Email Opened	Business Administration	...	0	0	
3	Landing Page Submission	Direct Traffic	No	No	0	1.0	305	1.00	Unreachable	Media and Advertising	...	0	0	
4	Landing Page Submission	Google	No	No	1	2.0	1428	1.00	Converted to Lead	not provided	...	0	0	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	
9235	Landing Page Submission	Direct Traffic	Yes	No	1	8.0	1845	2.67	Email Marked Spam	IT Projects Management	...	0	0	
9236	Landing Page Submission	Direct Traffic	No	No	0	2.0	238	2.00	SMS Sent	Media and Advertising	...	0	0	
9237	Landing Page Submission	Direct Traffic	Yes	No	0	2.0	199	2.00	SMS Sent	Business Administration	...	0	0	
9238	Landing Page Submission	Google	No	No	1	3.0	499	3.00	SMS Sent	Human Resource Management	...	0	0	
9239	Landing Page Submission	Direct Traffic	No	No	1	6.0	1279	3.00	SMS Sent	Supply Chain Management	...	0	0	

8991 rows × 101 columns

In [55]:

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

Out[55]:

	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	...
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	...

8991 rows × 82 columns

Test-Train Split

In [56]:

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

In [57]:

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

Out[57]:

	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_Finan Manageme
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	

5 rows × 81 columns

In [58]:

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

Out[58]:

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

In [59]:

```
# Split the dataset into 70% and 30% for train and test respectively
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 [60]:

```
# 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 Views
X_train.head()
```

Out[60]:

	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_F Mana
3523	0.117647	0.057218	0.0625	0	0	0	1	0	0	
3267	0.000000	0.000000	0.0000	0	1	0	0	0	0	
5653	0.117647	0.404049	0.1250	1	0	0	0	0	0	
5072	0.000000	0.000000	0.0000	0	0	0	0	0	0	
3704	0.235294	0.043134	0.2500	1	0	0	0	0	0	

5 rows × 81 columns

Model Building

In [61]:

```
# Import 'LogisticRegression'
from sklearn.linear_model import LogisticRegression
lr = LogisticRegression()
```

In [65]:

```
# Import RFE
from sklearn.feature_selection import RFE
```

In [68]:

```
# Running RFE with 15 variables as output
rfe = RFE(estimator=LogisticRegression(), n_features_to_select=20)
rfe = rfe.fit(X_train, y_train)
```

In [69]:

```
# Features that have been selected by RFE
list(zip(X_train.columns, rfe.support_, rfe.ranking_))
```

Out[69]:

```
[('TotalVisits', True, 1),
 ('Total Time Spent on Website', True, 1),
 ('Page Views Per Visit', True, 1),
 ('Lead Origin_Landing Page Submission', False, 22),
 ('Lead Origin_Lead Add Form', True, 1),
 ('Lead Origin_Lead Import', False, 34),
 ('Specialization_Business Administration', False, 24),
 ('Specialization_E-Business', False, 18),
 ('Specialization_E-COMMERCE', False, 29),
 ('Specialization_Finance Management', False, 21),
 ('Specialization_Healthcare Management', False, 20),
 ('Specialization_Hospitality Management', False, 51),
 ('Specialization_Human Resource Management', False, 23),
 ('Specialization_IT Projects Management', False, 27),
 ('Specialization_International Business', False, 26),
 ('Specialization_Marketing Management', False, 19),
 ('Specialization_Media and Advertising', False, 46),
 ('Specialization_Operations Management', False, 30),
 ('Specialization_Retail Management', False, 59),
 ('Specialization_Rural and Agribusiness', False, 25),
 ('Specialization_Services Excellence', False, 43),
 ('Specialization_Supply Chain Management', False, 28),
 ('Specialization_Travel and Tourism', False, 31),
 ('Lead Source_Direct Traffic', True, 1),
 ('Lead Source_Facebook', False, 45),
 ('Lead Source_Google', True, 1),
 ('Lead Source_Live Chat', False, 56),
 ('Lead Source_NC_EDM', False, 17),
 ('Lead Source_Olark Chat', False, 12),
 ('Lead Source_Organic Search', True, 1),
 ('Lead Source_Pay per Click Ads', False, 42),
 ('Lead Source_Press_Release', False, 55),
 ('Lead Source_Reference', False, 11),
 ('Lead Source_Referral Sites', True, 1),
 ('Lead Source_Social Media', False, 10),
 ('Lead Source_WeLearn', False, 41),
 ('Lead Source_Welingak Website', True, 1),
 ('Lead Source_bing', False, 48),
 ('Lead Source_blog', False, 36),
 ('Lead Source_google', False, 32),
 ('Lead Source_testone', False, 39),
 ('Lead Source_welearnblog_Home', False, 61),
 ('Lead Source_youtubechannel', False, 60),
 ('Do Not Email_Yes', True, 1),
 ('Last Activity_Converted to Lead', False, 6),
 ('Last Activity_Email Bounced', True, 1),
 ('Last Activity_Email Link Clicked', False, 53),
 ('Last Activity_Email Marked Spam', False, 33),
 ('Last Activity_Email Opened', False, 40),
 ('Last Activity_Email Received', False, 54),
 ('Last Activity_Form Submitted on Website', False, 7),
 ('Last Activity_Had a Phone Conversation', False, 14),
 ('Last Activity_Olark Chat Conversation', True, 1),
 ('Last Activity_Page Visited on Website', False, 8),
 ('Last Activity_Resubscribed to emails', False, 16),
 ('Last Activity_SMS Sent', False, 15),
 ('Last Activity_Unreachable', False, 5),
 ('Last Activity_Unsubscribed', False, 49),
 ('Last Activity_View in browser link Clicked', False, 47),
 ('Last Activity_Visited Booth in Tradeshow', False, 50),
 ('What is your current occupation_Housewife', True, 1),
 ('What is your current occupation_Other', False, 2),
 ('What is your current occupation_Student', False, 4),
 ('What is your current occupation_Unemployed', False, 3),
 ('What is your current occupation_Working Professional', True, 1),
 ('A free copy of Mastering The Interview_Yes', False, 52),
 ('Last Notable Activity_Email Bounced', False, 44),
 ('Last Notable Activity_Email Link Clicked', True, 1),
 ('Last Notable Activity_Email Marked Spam', False, 35),
 ('Last Notable Activity_Email Opened', True, 1),
 ('Last Notable Activity_Email Received', False, 57),
 ('Last Notable Activity_Form Submitted on Website', False, 58),
 ('Last Notable Activity_Had a Phone Conversation', True, 1),
 ('Last Notable Activity_Modified', True, 1),
 ('Last Notable Activity_Olark Chat Conversation', True, 1),
 ('Last Notable Activity_Page Visited on Website', True, 1),
 ('Last Notable Activity_Resubscribed to emails', False, 13),
 ('Last Notable Activity_SMS Sent', False, 37),
 ('Last Notable Activity_Unreachable', False, 9),
 ('Last Notable Activity_Unsubscribed', False, 38),
 ('Last Notable Activity_View in browser link Clicked', False, 62)]
```

In [70]:

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

```
All the variables selected by RFE, next statistics part (p-values and the VIFs)
```

In [71]:

```
# Selecting columns selected by RFE
X_train = X_train[col]
```

In [72]:

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

In [73]:

```
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[73]:

Generalized Linear Model Regression Results

Dep. Variable:	Converted	**No. Observations:**	6293
Model:	GLM	**Df Residuals:**	6272
Model Family:	Binomial	**Df Model:**	20
Link Function:	Logit	**Scale:**	1.0000
Method:	IRLS	**Log-Likelihood:**	-2573.2
Date:	Sun, 21 May 2023	**Deviance:**	5146.4
Time:	16:27:10	**Pearson chi2:**	6.52e+03
No. Iterations:	22	**Pseudo R-squ. (CS):**	0.3984
Covariance Type:	nonrobust		

	coef	std err	z	P>|z|	[0.025	0.975]
const	0.3920	0.101	3.876	0.000	0.194	0.590
TotalVisits	1.9299	0.301	6.405	0.000	1.339	2.520
Total Time Spent on Website	4.7035	0.170	27.617	0.000	4.370	5.037
Page Views Per Visit	-2.0243	0.444	-4.558	0.000	-2.895	-1.154
Lead Origin_Lead Add Form	3.0451	0.256	11.896	0.000	2.543	3.547
Lead Source_Direct Traffic	-1.5377	0.132	-11.617	0.000	-1.797	-1.278
Lead Source_Google	-1.1120	0.129	-8.598	0.000	-1.366	-0.859
Lead Source_Organic Search	-1.4285	0.165	-8.657	0.000	-1.752	-1.105
Lead Source_Referral Sites	-1.3511	0.334	-4.049	0.000	-2.005	-0.697
Lead Source_Welingak Website	2.4662	1.039	2.373	0.018	0.429	4.503
Do Not Email_Yes	-1.4273	0.206	-6.916	0.000	-1.832	-1.023
Last Activity_Email Bounced	-1.1159	0.396	-2.820	0.005	-1.891	-0.340
Last Activity_Olark Chat Conversation	-1.2987	0.193	-6.723	0.000	-1.677	-0.920
What is your current occupation_Housewife	23.3558	2.89e+04	0.001	0.999	-5.66e+04	5.66e+04
What is your current occupation_Working Professional	2.7793	0.191	14.523	0.000	2.404	3.154
Last Notable Activity_Email Link Clicked	-2.0672	0.266	-7.760	0.000	-2.589	-1.545
Last Notable Activity_Email Opened	-1.4274	0.090	-15.864	0.000	-1.604	-1.251
Last Notable Activity_Had a Phone Conversation	22.3270	2.19e+04	0.001	0.999	-4.28e+04	4.29e+04
Last Notable Activity_Modified	-1.8466	0.099	-18.632	0.000	-2.041	-1.652
Last Notable Activity_Olark Chat Conversation	-1.6187	0.372	-4.347	0.000	-2.348	-0.889
Last Notable Activity_Page Visited on Website	-2.1305	0.216	-9.849	0.000	-2.554	-1.707

In [74]:

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

In [75]:

```
# Make a VIF dataframe for all the variables present
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[75]:

	Features	VIF
2	Page Views Per Visit	6.32
0	TotalVisits	5.50
5	Lead Source_Google	3.56
4	Lead Source_Direct Traffic	3.15
6	Lead Source_Organic Search	2.43
1	Total Time Spent on Website	2.35
17	Last Notable Activity_Modified	2.34
9	Do Not Email_Yes	1.92
10	Last Activity_Email Bounced	1.86
11	Last Activity_Olark Chat Conversation	1.77
15	Last Notable Activity_Email Opened	1.70
3	Lead Origin_Lead Add Form	1.53
18	Last Notable Activity_Olark Chat Conversation	1.36
8	Lead Source_Welingak Website	1.34
19	Last Notable Activity_Page Visited on Website	1.18
13	What is your current occupation_Working Profes...	1.17
7	Lead Source_Referral Sites	1.15
14	Last Notable Activity_Email Link Clicked	1.03
12	What is your current occupation_Housewife	1.01
16	Last Notable Activity_Had a Phone Conversation	1.01

The VIF values seem fine but some p-values are 99 %. So removing ' What is your current occupation_Housewife','Last Notable Activity_Had a Phone Conversation'

In [76]:

```
X_train.drop(['What is your current occupation_Housewife','Last Notable Activity_Had a Phone Conversation'], axis = 1, inplace = True)
```

In [77]:

```
# Refit the model with the new set of features
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[77]:

Generalized Linear Model Regression Results

Dep. Variable:	Converted	**No. Observations:**	6293
Model:	GLM	**Df Residuals:**	6274
Model Family:	Binomial	**Df Model:**	18
Link Function:	Logit	**Scale:**	1.0000
Method:	IRLS	**Log-Likelihood:**	-2579.1
Date:	Sun, 21 May 2023	**Deviance:**	5158.3
Time:	16:28:08	**Pearson chi2:**	6.55e+03
No. Iterations:	7	**Pseudo R-squ. (CS):**	0.3973
Covariance Type:	nonrobust		

	coef	std err	z	P>|z|	[0.025	0.975]
const	0.3960	0.101	3.918	0.000	0.198	0.594
TotalVisits	1.9392	0.300	6.461	0.000	1.351	2.528
Total Time Spent on Website	4.7007	0.170	27.634	0.000	4.367	5.034
Page Views Per Visit	-2.0311	0.443	-4.582	0.000	-2.900	-1.162
Lead Origin_Lead Add Form	3.0728	0.256	12.020	0.000	2.572	3.574
Lead Source_Direct Traffic	-1.5326	0.132	-11.589	0.000	-1.792	-1.273
Lead Source_Google	-1.1040	0.129	-8.544	0.000	-1.357	-0.851
Lead Source_Organic Search	-1.4292	0.165	-8.664	0.000	-1.753	-1.106
Lead Source_Referral Sites	-1.3504	0.334	-4.048	0.000	-2.004	-0.697
Lead Source_Welingak Website	2.4399	1.039	2.348	0.019	0.403	4.477
Do Not Email_Yes	-1.4347	0.206	-6.949	0.000	-1.839	-1.030
Last Activity_Email Bounced	-1.1144	0.396	-2.816	0.005	-1.890	-0.339
Last Activity_Olark Chat Conversation	-1.2999	0.193	-6.730	0.000	-1.678	-0.921
What is your current occupation_Working Professional	2.7758	0.191	14.506	0.000	2.401	3.151
Last Notable Activity_Email Link Clicked	-2.0624	0.265	-7.796	0.000	-2.581	-1.544
Last Notable Activity_Email Opened	-1.4328	0.090	-15.945	0.000	-1.609	-1.257
Last Notable Activity_Modified	-1.8512	0.099	-18.701	0.000	-2.045	-1.657
Last Notable Activity_Olark Chat Conversation	-1.6244	0.372	-4.362	0.000	-2.354	-0.894
Last Notable Activity_Page Visited on Website	-2.1410	0.216	-9.903	0.000	-2.565	-1.717

In [78]:

```python
# Make a VIF dataframe for all the variables present
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[78]:

	Features	VIF
2	Page Views Per Visit	6.32
0	TotalVisits	5.50
5	Lead Source_Google	3.56
4	Lead Source_Direct Traffic	3.15
6	Lead Source_Organic Search	2.43
1	Total Time Spent on Website	2.34
15	Last Notable Activity_Modified	2.34
9	Do Not Email_Yes	1.92
10	Last Activity_Email Bounced	1.86
11	Last Activity_Olark Chat Conversation	1.77
14	Last Notable Activity_Email Opened	1.70
3	Lead Origin_Lead Add Form	1.53
16	Last Notable Activity_Olark Chat Conversation	1.36
8	Lead Source_Welingak Website	1.34
17	Last Notable Activity_Page Visited on Website	1.18
12	What is your current occupation_Working Profes...	1.17
7	Lead Source_Referral Sites	1.15
13	Last Notable Activity_Email Link Clicked	1.03

In [79]:

```python
X_train.drop('Page Views Per Visit', axis = 1, inplace = True)
```

In [80]:

```
# Refit the model with the new set of features
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[80]:

Generalized Linear Model Regression Results

Dep. Variable:	Converted	**No. Observations:**	6293
Model:	GLM	**Df Residuals:**	6275
Model Family:	Binomial	**Df Model:**	17
Link Function:	Logit	**Scale:**	1.0000
Method:	IRLS	**Log-Likelihood:**	-2589.8
Date:	Sun, 21 May 2023	**Deviance:**	5179.6
Time:	16:28:41	**Pearson chi2:**	6.56e+03
No. Iterations:	7	**Pseudo R-squ. (CS):**	0.3953
Covariance Type:	nonrobust		

	coef	std err	z	P>|z|	[0.025	0.975]
const	0.3575	0.100	3.558	0.000	0.161	0.554
TotalVisits	1.1670	0.249	4.688	0.000	0.679	1.655
Total Time Spent on Website	4.6833	0.170	27.604	0.000	4.351	5.016
Lead Origin_Lead Add Form	3.0865	0.256	12.074	0.000	2.585	3.588
Lead Source_Direct Traffic	-1.6995	0.128	-13.325	0.000	-1.949	-1.450
Lead Source_Google	-1.2811	0.124	-10.357	0.000	-1.523	-1.039
Lead Source_Organic Search	-1.6653	0.157	-10.607	0.000	-1.973	-1.358
Lead Source_Referral Sites	-1.5536	0.333	-4.672	0.000	-2.205	-0.902
Lead Source_Welingak Website	2.4405	1.039	2.348	0.019	0.403	4.478
Do Not Email_Yes	-1.4683	0.205	-7.156	0.000	-1.871	-1.066
Last Activity_Email Bounced	-1.0281	0.393	-2.613	0.009	-1.799	-0.257
Last Activity_Olark Chat Conversation	-1.2808	0.193	-6.632	0.000	-1.659	-0.902
What is your current occupation_Working Professional	2.7694	0.191	14.489	0.000	2.395	3.144
Last Notable Activity_Email Link Clicked	-2.0150	0.263	-7.668	0.000	-2.530	-1.500
Last Notable Activity_Email Opened	-1.4049	0.089	-15.727	0.000	-1.580	-1.230
Last Notable Activity_Modified	-1.8221	0.099	-18.493	0.000	-2.015	-1.629
Last Notable Activity_Olark Chat Conversation	-1.5389	0.368	-4.177	0.000	-2.261	-0.817
Last Notable Activity_Page Visited on Website	-1.9535	0.210	-9.324	0.000	-2.364	-1.543

In [81]:

```
# Make a VIF dataframe for all the variables present
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[81]:

	Features	VIF
0	TotalVisits	3.68
4	Lead Source_Google	3.09
3	Lead Source_Direct Traffic	2.77
1	Total Time Spent on Website	2.34
14	Last Notable Activity_Modified	2.34
5	Lead Source_Organic Search	2.10
8	Do Not Email_Yes	1.91
9	Last Activity_Email Bounced	1.85
10	Last Activity_Olark Chat Conversation	1.77
13	Last Notable Activity_Email Opened	1.70
2	Lead Origin_Lead Add Form	1.53
15	Last Notable Activity_Olark Chat Conversation	1.36
7	Lead Source_Welingak Website	1.34
11	What is your current occupation_Working Profes...	1.17
16	Last Notable Activity_Page Visited on Website	1.14
6	Lead Source_Referral Sites	1.12
12	Last Notable Activity_Email Link Clicked	1.03

```
All the VIF values are good and all the p-values are below 0.05. So we can fix model
```

Creating Prediction

In [82]:

```
# Predicting the probabilities on the train set
y_train_pred = res.predict(X_train_sm)
y_train_pred[:10]
```

Out[82]:

```
3523    0.257438
3267    0.997225
5653    0.327989
5072    0.259734
3704    0.135660
1790    0.116880
2482    0.180653
1694    0.187767
8768    0.119395
9225    0.004629
dtype: float64
```

In [83]:

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

Out[83]:

```
array([0.25743824, 0.99722548, 0.32798883, 0.2597343 , 0.1356595 ,
       0.11687958, 0.18065264, 0.1877671 , 0.11939502, 0.00462932])
```

In [84]:

```
# Data frame with given 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[84]:

	Converted	Conversion_Prob
0	0	0.257438
1	1	0.997225
2	1	0.327989
3	0	0.259734
4	0	0.135660

In [85]:

```
# Substituting 0 or 1 with the cut off as 0.5
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[85]:

	Converted	Conversion_Prob	Predicted
0	0	0.257438	0
1	1	0.997225	1
2	1	0.327989	0
3	0	0.259734	0
4	0	0.135660	0

Model Evaluation

In [86]:

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

In [87]:

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

Out[87]:

```
array([[3479,  436],
       [ 708, 1670]], dtype=int64)
```

In [ ]:

```
# Predicted         No          Yes
# Actual
# No               3498       417
# Yes              837       1541
```

In [88]:

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

Out[88]:

```
0.8182107103130463
```

That's around 82% accuracy with is a very good value

In [89]:

```
# 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 [90]:

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

Out[90]:

0.7022708158116064

In [91]:

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

Out[91]:

0.8886334610472542

With the current cut off as 0.5 we have around 82% accuracy, sensitivity of around 70% and specificity of around 88%.

Optimise Cut off (ROC Curve)

The previous cut off was randomely selected. Now to find the optimum one

In [92]:

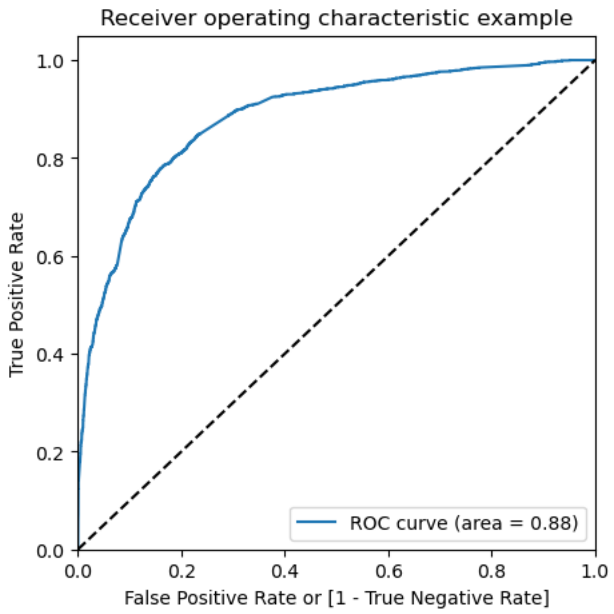
```
# ROC function
def draw_roc( actual, probs ):
    fpr, tpr, thresholds = metrics.roc_curve( actual, probs,
                                              drop_intermediate = False )
    auc_score = metrics.roc_auc_score( actual, probs )
    plt.figure(figsize=(5, 5))
    plt.plot( fpr, tpr, label='ROC 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 [93]:

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

In [94]:

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




```
The area under ROC curve is 0.88 which is a very good value
```

In [96]:

```
# 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[96]:

	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	0	0.257438	0	1	1	1	0	0	0	0	0	0	0
1	1	0.997225	1	1	1	1	1	1	1	1	1	1	1
2	1	0.327989	0	1	1	1	1	0	0	0	0	0	0
3	0	0.259734	0	1	1	1	0	0	0	0	0	0	0
4	0	0.135660	0	1	1	0	0	0	0	0	0	0	0

In [97]:

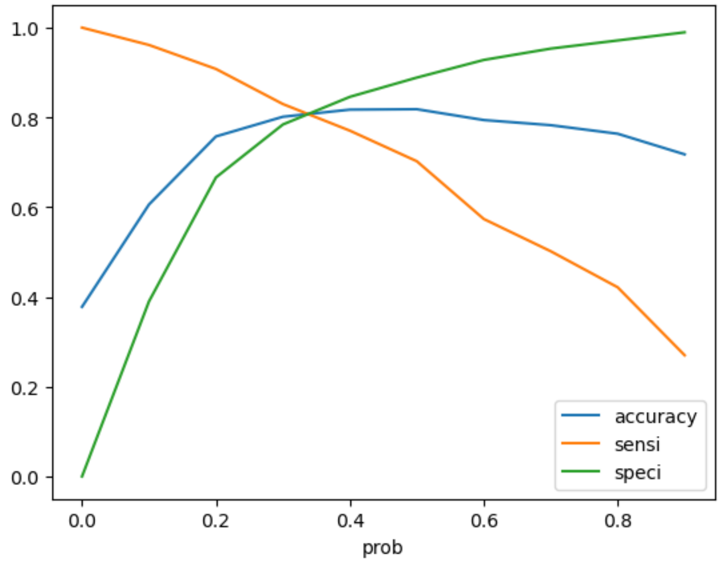
```
# Creating a dataframe to see the values of accuracy, sensitivity, and specificity at different values of probabiity cutoffs
cutoff_df = pd.DataFrame( columns = ['prob','accuracy','sensi','speci'])
# Making confusing matrix to find values of sensitivity, accurace and specificity for each level of 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[97]:

	prob	accuracy	sensi	speci
0.0	0.0	0.377880	1.000000	0.000000
0.1	0.1	0.605911	0.961312	0.390038
0.2	0.2	0.757508	0.907906	0.666156
0.3	0.3	0.801367	0.829689	0.784163
0.4	0.4	0.817257	0.770395	0.845722
0.5	0.5	0.818211	0.702271	0.888633
0.6	0.6	0.794057	0.573591	0.927969
0.7	0.7	0.782616	0.501682	0.953257
0.8	0.8	0.763547	0.421362	0.971392
0.9	0.9	0.717623	0.269975	0.989527

In [98]:

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



From the graph it is visible that the optimal cut off is at 0.35.

In [99]:

```
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[99]:

	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	0	0.257438	0	1	1	1	0	0	0	0	0	0	0	0
1	1	0.997225	1	1	1	1	1	1	1	1	1	1	1	1
2	1	0.327989	0	1	1	1	1	0	0	0	0	0	0	0
3	0	0.259734	0	1	1	1	0	0	0	0	0	0	0	0
4	0	0.135660	0	1	1	0	0	0	0	0	0	0	0	0

In [100]:

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

Out[100]:

```
0.8091530271730494
```

In [101]:

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

Out[101]:

```
array([[3191,  724],
       [ 477, 1901]], dtype=int64)
```

In [103]:

```
# 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 [104]:

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

Out[104]:

```
0.7994112699747687
```

In [105]:

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

Out[105]:

```
0.8150702426564496
```

With the current cut off as 0.35 we have accuracy, sensitivity and specificity of around 80%

Prediction on Test set

In [106]:

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

In [107]:

```
col = X_train.columns
```

In [108]:

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

Out[108]:

	const	TotalVisits	Total Time Spent on Website	Lead Origin_Lead Add Form	Lead Source_Direct Traffic	Lead Source_Google	Lead Source_Organic Search	Lead Source_Referral Sites	Lead Source_Welingak Website	Do Not Email_Yes	L Activity_En Bounc
3308	1.0	0.117647	0.050176	0	0	1	0	0	0	0	
4421	1.0	0.000000	0.000000	0	0	0	0	0	0	0	
8855	1.0	0.058824	0.547975	0	1	0	0	0	0	0	
5302	1.0	0.000000	0.000000	0	0	0	0	0	0	0	
2169	1.0	0.588235	0.390405	0	1	0	0	0	0	0	
...	...	...	...	...	...	...	...	...	...	...	
5655	1.0	0.058824	0.218310	0	1	0	0	0	0	0	
7836	1.0	0.588235	0.227113	0	0	1	0	0	0	0	
8378	1.0	0.588235	0.179577	0	0	0	1	0	0	1	
1263	1.0	0.117647	0.376320	0	1	0	0	0	0	0	
8633	1.0	0.058824	0.150088	0	1	0	0	0	0	1	

2698 rows × 18 columns

In [109]:

```
# 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[109]:

	Converted	Conversion_Prob
0	0	0.123887
1	1	0.588440
2	1	0.370721
3	0	0.060348
4	0	0.442248

In [110]:

```
# 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[110]:

	Converted	Conversion_Prob	final_predicted
0	0	0.123887	0
1	1	0.588440	1
2	1	0.370721	1
3	0	0.060348	0
4	0	0.442248	1
...	...	...	...
2693	1	0.111744	0
2694	1	0.829332	1
2695	0	0.039085	0
2696	1	0.965347	1
2697	0	0.007473	0

2698 rows × 3 columns

In [111]:

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

Out[111]:

```
0.8002223869532987
```

In [112]:

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

Out[112]:

```
array([[1350,  327],
       [ 212,  809]], dtype=int64)
```

In [113]:

```
# 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 [114]:

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

Out[114]:

```
0.7923604309500489
```

In [115]:

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

Out[115]:

```
0.8050089445438283
```

```
With the current cut off as 0.35 we have accuracy, sensitivity and specificity of around 80%
```

Precision-Recall

In [116]:

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

Out[116]:

```
array([[3479,  436],
       [ 708, 1670]], dtype=int64)
```

In [117]:

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

Out[117]:

```
0.7929724596391263
```

In [118]:

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

Out[118]:

```
0.7022708158116064
```

With the current cut off as 0.35 we have Precision around 79% and Recall around 70%

Precision and recall tradeoff

In [119]:

```
from sklearn.metrics import precision_recall_curve
```

In [120]:

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

Out[120]:

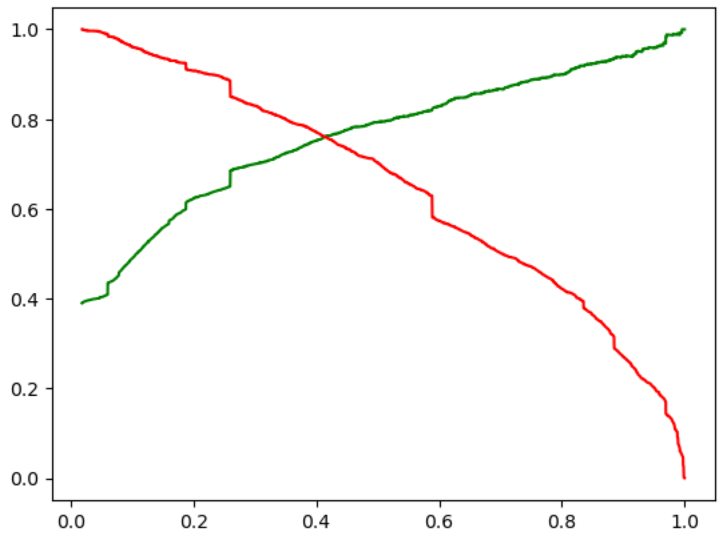
```
(0       0
 1       1
 2       1
 3       0
 4       0
        ..
 6288    1
 6289    1
 6290    1
 6291    0
 6292    1
 Name: Converted, Length: 6293, dtype: int64,
 0       0
 1       1
 2       0
 3       0
 4       0
        ..
 6288    1
 6289    0
 6290    1
 6291    0
 6292    1
 Name: Predicted, Length: 6293, dtype: int64)
```

In [121]:

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

In [122]:

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



In [123]:

```
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[123]:

	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	0	0.257438	0	1	1	1	0	0	0	0	0	0	0	0
1	1	0.997225	1	1	1	1	1	1	1	1	1	1	1	1
2	1	0.327989	0	1	1	1	1	0	0	0	0	0	0	0
3	0	0.259734	0	1	1	1	0	0	0	0	0	0	0	0
4	0	0.135660	0	1	1	0	0	0	0	0	0	0	0	0

In [124]:

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

Out[124]:

```
0.8180518035912919
```

In [125]:

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

Out[125]:

```
array([[3333,  582],
       [ 563, 1815]], dtype=int64)
```

In [126]:

```
# 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 [127]:

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

Out[127]:

0.7571964956195244

In [128]:

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

Out[128]:

0.763246425567704

With the current cut off as 0.44 we have Precision around 76% and Recall around 76.3% and accuracy 82 %.

Prediction on Test set

In [129]:

```
# 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[129]:

	Converted	Conversion_Prob
0	0	0.123887
1	1	0.588440
2	1	0.370721
3	0	0.060348
4	0	0.442248

In [130]:

```
# Making prediction using cut off 0.41
y_pred_final['final_predicted'] = y_pred_final.Conversion_Prob.map(lambda x: 1 if x > 0.44 else 0)
y_pred_final
```

Out[130]:

	Converted	Conversion_Prob	final_predicted
0	0	0.123887	0
1	1	0.588440	1
2	1	0.370721	0
3	0	0.060348	0
4	0	0.442248	1
...	...	...	...
2693	1	0.111744	0
2694	1	0.829332	1
2695	0	0.039085	0
2696	1	0.965347	1
2697	0	0.007473	0

2698 rows × 3 columns

Check the overall accuracy

In [131]:

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

Out[131]:

```
0.8057820607857672
```

In [132]:

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

Out[132]:

```
array([[1426,  251],
       [ 273,  748]], dtype=int64)
```

In [133]:

```
# 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 [134]:

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

Out[134]:

```
0.7487487487487487
```

In [135]:

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

Out[135]:

```
0.732615083251714
```

With the current cut off as 0.41 we have Precision around 75% , Recall around 73% and accuracy 80.5%.

The Model seems to predict the Conversion Rate very well and we should be able to give the CEO confidence in making good calls based on this model

Conclusion
It was found that the variables that mattered the most in the potential buyers are (In descending order) :

TotalVisits
The total time spend on the Website.
Lead Origin_Lead Add Form
Lead Source_Direct Traffic
Lead Source_Google
Lead Source_Welingak Website
Lead Source_Organic Search
Lead Source_Referral Sites
Lead Source_Welingak Website
Do Not Email_Yes
Last Activity_Email Bounced
Last Activity_Olark Chat Conversation
Keeping these in mind the X Education can flourish as they have a very high chance to get almost all the potential buyers to change their mind and buy their courses.