In [1]:
```
## Importing libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
from sklearn.preprocessing import StandardScaler
```

In [2]:
```
##Importing dataset to CSV
Leads_df = pd.read_csv(r"C:\Users\Keshav\OneDrive\Desktop\Leads.csv")
Leads_df
```

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	...	upda on Cont
0	7927b2df-8bba-4d29-b9a2-b6e0beafe620	660737	API	Olark Chat	No	No	0	0.0	0	0.00	...	
1	2a272436-5132-4136-86fa-dcc88c88f482	660728	API	Organic Search	No	No	0	5.0	674	2.50	...	
2	8cc8c611-a219-4f35-ad23-fdfd2656bd8a	660727	Landing Page Submission	Direct Traffic	No	No	1	2.0	1532	2.00	...	
3	0cc2df48-7cf4-4e39-9de9-19797f9b38cc	660719	Landing Page Submission	Direct Traffic	No	No	0	1.0	305	1.00	...	
4	3256f628-e534-4826-9d63-4a8b88782852	660681	Landing Page Submission	Google	No	No	1	2.0	1428	1.00	...	
...	...	...	...	...	...	...	...	...	...	...	...	
9235	19d6451e-fcd6-407c-b83b-48e1af805ea9	579564	Landing Page Submission	Direct Traffic	Yes	No	1	8.0	1845	2.67	...	
9236	82a7005b-7196-4d56-95ce-a79f937a158d	579546	Landing Page Submission	Direct Traffic	No	No	0	2.0	238	2.00	...	
9237	aac550fe-a586-452d-8d3c-f1b62c94e02c	579545	Landing Page Submission	Direct Traffic	Yes	No	0	2.0	199	2.00	...	
9238	5330a7d1-2f2b-4df4-85d6-64ca2f6b95b9	579538	Landing Page Submission	Google	No	No	1	3.0	499	3.00	...	
9239	571b5c8e-a5b2-4d57-8574-f2ffb06fdeff	579533	Landing Page Submission	Direct Traffic	No	No	1	6.0	1279	3.00	...	

9240 rows × 37 columns

In [3]:
```
#checking total rows and cols in dataset
Leads_df.shape
```

Out[3]:
```
(9240, 37)
```

In [4]:
```
#basic data check
Leads_df.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]:
```
Leads_df.describe()
```

Out[5]:

	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 [6]:
```
#dropping Lead Number and Prospect ID since they have all unique values
Leads_df.drop(['Prospect ID', 'Lead Number'], 1, inplace = True)
```

In [7]:
```
Leads_df
```

Out[7]:

	Lead Origin	Lead Source	Do Not Email	Do Not Call	Converted	TotalVisits	Total Time Spent on Website	Page Views Per Visit	Last Activity	Country	...	upd on Cor
0	API	Olark Chat	No	No	0	0.0	0	0.00	Page Visited on Website	NaN	...	
1	API	Organic Search	No	No	0	5.0	674	2.50	Email Opened	India	...	
2	Landing Page Submission	Direct Traffic	No	No	1	2.0	1532	2.00	Email Opened	India	...	
3	Landing Page Submission	Direct Traffic	No	No	0	1.0	305	1.00	Unreachable	India	...	
4	Landing Page Submission	Google	No	No	1	2.0	1428	1.00	Converted to Lead	India	...	
...	...	...	...	...	...	...	...	...	...	...	...	
9235	Landing Page Submission	Direct Traffic	Yes	No	1	8.0	1845	2.67	Email Marked Spam	Saudi Arabia	...	
9236	Landing Page Submission	Direct Traffic	No	No	0	2.0	238	2.00	SMS Sent	India	...	
9237	Landing Page Submission	Direct Traffic	Yes	No	0	2.0	199	2.00	SMS Sent	India	...	
9238	Landing Page Submission	Google	No	No	1	3.0	499	3.00	SMS Sent	India	...	
9239	Landing Page Submission	Direct Traffic	No	No	1	6.0	1279	3.00	SMS Sent	Bangladesh	...	

9240 rows × 35 columns

In [8]:
```
#Converting 'Select' values to NaN.
Leads_df = Leads_df .replace('Select', np.nan)
Leads_df
```

Out[8]:

	Lead Origin	Lead Source	Do Not Email	Do Not Call	Converted	TotalVisits	Total Time Spent on Website	Page Views Per Visit	Last Activity	Country	...	upc or Cor
0	API	Olark Chat	No	No	0	0.0	0	0.00	Page Visited on Website	NaN	...	
1	API	Organic Search	No	No	0	5.0	674	2.50	Email Opened	India	...	
2	Landing Page Submission	Direct Traffic	No	No	1	2.0	1532	2.00	Email Opened	India	...	
3	Landing Page Submission	Direct Traffic	No	No	0	1.0	305	1.00	Unreachable	India	...	
4	Landing Page Submission	Google	No	No	1	2.0	1428	1.00	Converted to Lead	India	...	
...	...	...	...	...	...	...	...	...	...	...	...	
9235	Landing Page Submission	Direct Traffic	Yes	No	1	8.0	1845	2.67	Email Marked Spam	Saudi Arabia	...	
9236	Landing Page Submission	Direct Traffic	No	No	0	2.0	238	2.00	SMS Sent	India	...	
9237	Landing Page Submission	Direct Traffic	Yes	No	0	2.0	199	2.00	SMS Sent	India	...	
9238	Landing Page Submission	Google	No	No	1	3.0	499	3.00	SMS Sent	India	...	
9239	Landing Page Submission	Direct Traffic	No	No	1	6.0	1279	3.00	SMS Sent	Bangladesh	...	

9240 rows × 35 columns

In [9]:
```
#checking null values in each rows
Leads_df.isnull().sum()
```

Out[9]:

```
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
Magazine                                          0
Newspaper Article                                 0
X Education Forums                                0
Newspaper                                         0
Digital Advertisement                             0
Through Recommendations                           0
Receive More Updates About Our Courses            0
Tags                                           3353
Lead Quality                                   4767
Update me on Supply Chain Content                 0
Get updates on DM Content                         0
Lead Profile                                   6855
City                                           3669
Asymmetrique Activity Index                    4218
Asymmetrique Profile Index                     4218
Asymmetrique Activity Score                    4218
Asymmetrique Profile Score                     4218
I agree to pay the amount through cheque          0
A free copy of Mastering The Interview            0
Last Notable Activity                             0
dtype: int64
```

In [10]:

```
#checking percentage of null values in each column
round(100*(Leads_df.isnull().sum()/len(Leads_df.index)), 2)
```

Out[10]:
```
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
Magazine                                       0.00
Newspaper Article                              0.00
X Education Forums                             0.00
Newspaper                                      0.00
Digital Advertisement                          0.00
Through Recommendations                        0.00
Receive More Updates About Our Courses         0.00
Tags                                          36.29
Lead Quality                                  51.59
Update me on Supply Chain Content              0.00
Get updates on DM Content                      0.00
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
I agree to pay the amount through cheque       0.00
A free copy of Mastering The Interview         0.00
Last Notable Activity                          0.00
dtype: float64
```

In [11]:
```
#dropping cols with more than 45% missing values

cols=Leads_df.columns

for i in cols:
    if((100*(Leads_df[i].isnull().sum()/len(Leads_df.index))) >= 45):
        Leads_df.drop(i, 1, inplace = True)
```

In [12]:
```
#checking null values percentage
round(100*(Leads_df.isnull().sum()/len(Leads_df.index)), 2)
```

Out[12]:
```
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
Magazine                                         0.00
Newspaper Article                                0.00
X Education Forums                               0.00
Newspaper                                        0.00
Digital Advertisement                            0.00
Through Recommendations                          0.00
Receive More Updates About Our Courses           0.00
Tags                                            36.29
Update me on Supply Chain Content                0.00
Get updates on DM Content                        0.00
City                                            39.71
I agree to pay the amount through cheque         0.00
A free copy of Mastering The Interview           0.00
Last Notable Activity                            0.00
dtype: float64
```

In [13]:
```
#checking value counts of Country column
Leads_df['Country'].value_counts(dropna=False)
```

Out[13]:
```
India                   6492
NaN                     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 [14]:
```
# Since India is the most common occurence among the non-missing values we can impute al
Leads_df['Country'] = Leads_df['Country'].replace(np.nan,'India')
```

In [15]:
```
###Number of values for India are quite high, so this column can be dropped
cols_to_drop=['Country']
```

In [16]:
```
#checking value counts of "City" column
Leads_df['City'].value_counts(dropna=False)
```

Out[16]:
```
NaN                         3669
Mumbai                      3222
Thane & Outskirts            752
Other Cities                 686
Other Cities of Maharashtra  457
Other Metro Cities           380
Tier II Cities                74
Name: City, dtype: int64
```

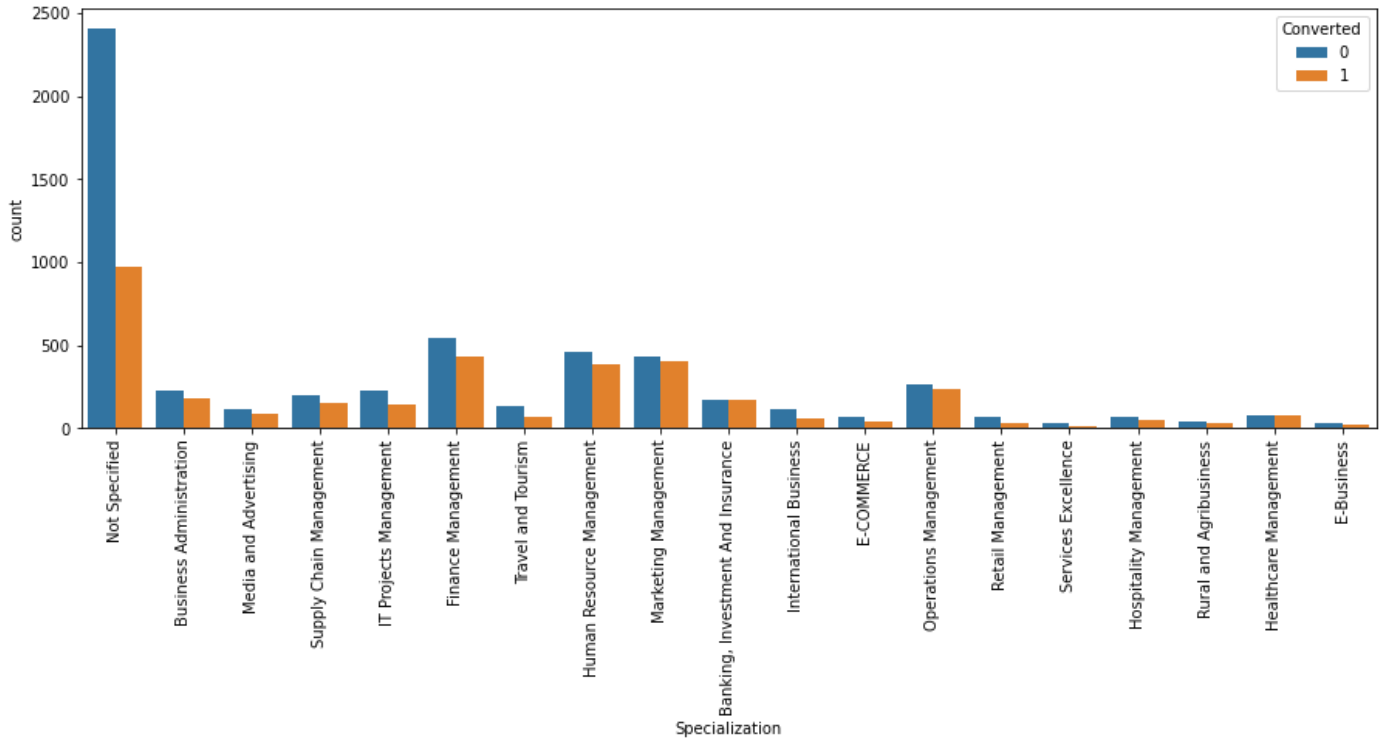
In [17]:
```
Leads_df['City'] = Leads_df['City'].replace(np.nan,'Mumbai')
```

In [18]:
```
#checking value counts of Specialization column
Leads_df['Specialization'].value_counts(dropna=False)
```

Out[18]:
```
NaN                                  3380
Finance Management                    976
Human Resource Management             848
Marketing Management                  838
Operations Management                 503
Business Administration               403
IT Projects Management                366
Supply Chain Management               349
Banking, Investment And Insurance     338
Travel and Tourism                    203
Media and Advertising                 203
International Business                178
Healthcare Management                 159
Hospitality Management                114
E-COMMERCE                            112
Retail Management                     100
Rural and Agribusiness                 73
E-Business                             57
Services Excellence                    40
Name: Specialization, dtype: int64
```

In [19]:
```
##we will replace NaN values here with 'Not Specified'
Leads_df['Specialization'] = Leads_df['Specialization'].replace(np.nan, 'Not Specified')
```

In [20]:
```
plt.figure(figsize=(15,5))
s1=sns.countplot(Leads_df.Specialization, hue=Leads_df.Converted)
s1.set_xticklabels(s1.get_xticklabels(),rotation=90)
plt.show()
```



In [21]:
```
Leads_df['What is your current occupation'].value_counts(dropna=False)
```

Out[21]:
```
Unemployed              5600
NaN                     2690
Working Professional     706
Student                  210
Other                     16
Housewife                 10
Businessman                8
Name: What is your current occupation, dtype: int64
```

In [22]:
```
#imputing Nan values with mode "Unemployed"
Leads_df['What is your current occupation'] = Leads_df['What is your current occupation'
```

In [23]:
```
Leads_df['What is your current occupation'].value_counts(dropna=False)
```

Out[23]:
```
Unemployed              8290
Working Professional     706
Student                  210
Other                     16
Housewife                 10
Businessman                8
Name: What is your current occupation, dtype: int64
```

In [24]:
```
Leads_df['What matters most to you in choosing a course'].value_counts(dropna=False)
```

Out[24]:
```
Better Career Prospects      6528
NaN                          2709
Flexibility & Convenience       2
Other                           1
Name: What matters most to you in choosing a course, dtype: int64
```

In [25]:
```
#replacing Nan values with Mode "Better Career Prospects"
Leads_df['What matters most to you in choosing a course'] = Leads_df['What matters most
```

In [26]:
```
Leads_df['What matters most to you in choosing a course'].value_counts(dropna=False)
```

Out[26]:
```
Better Career Prospects      9237
Flexibility & Convenience       2
Other                           1
Name: What matters most to you in choosing a course, dtype: int64
```

In [27]:
```
##we Append to the cols_to_drop List
cols_to_drop.append('What matters most to you in choosing a course')
cols_to_drop
```

Out[27]:
```
['Country', 'What matters most to you in choosing a course']
```

In [28]:
```
Leads_df['Tags'].value_counts(dropna=False)
```

Out[28]:
```
NaN                                                 3353
Will revert after reading the email                 2072
Ringing                                             1203
Interested in other courses                          513
Already a student                                    465
Closed by Horizzon                                   358
switched off                                         240
Busy                                                 186
Lost to EINS                                         175
Not doing further education                          145
Interested  in full time MBA                         117
Graduation in progress                               111
invalid number                                        83
Diploma holder (Not Eligible)                         63
wrong number given                                    47
opp hangup                                            33
number not provided                                   27
in touch with EINS                                    12
Lost to Others                                         7
Still Thinking                                         6
Want to take admission but has financial problems      6
In confusion whether part time or DLP                  5
Interested in Next batch                               5
Lateral student                                        3
Shall take in the next coming month                    2
University not recognized                              2
Recognition issue (DEC approval)                       1
Name: Tags, dtype: int64
```

In [29]:
```
#replacing Nan values with "Not Specified"
Leads_df['Tags'] = Leads_df['Tags'].replace(np.nan,'Not Specified')
```

In [30]:
```
#replacing tags with low frequency with "Other Tags"
Leads_df['Tags'] = Leads_df['Tags'].replace(['In confusion whether part time or DLP', 'i
                                     'Approached upfront','Graduation in progress','numb
                                    'Lost to Others','Shall take in the next coming mont
                                    'Recognition issue (DEC approval)','Want to take adm
                                    'University not recognized'], 'Other_Tags')

Leads_df['Tags'] = Leads_df['Tags'].replace(['switched off',
                                      'Already a student',
                                       'Not doing further education',
                                       'invalid number',
                                       'wrong number given',
                                       'Interested  in full time MBA'] , 'Other_Tags')
```

In [31]:
```
round(100*(Leads_df.isnull().sum()/len(Leads_df.index)), 2)
```

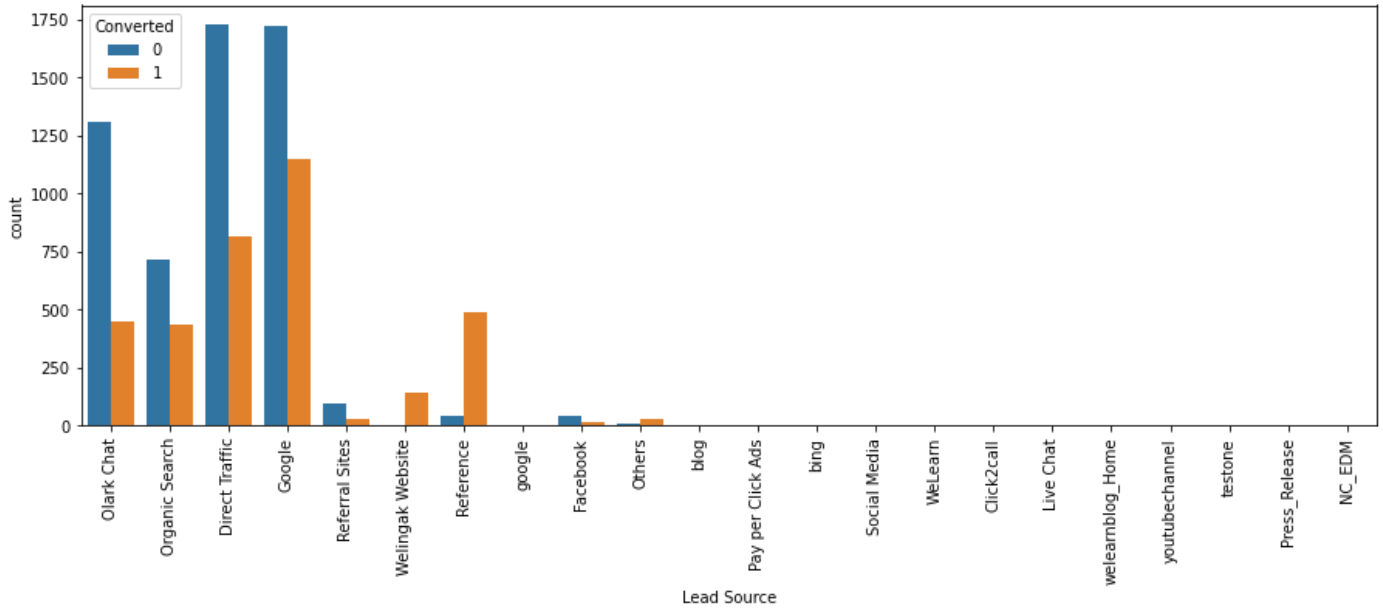
Out[31]:
```
Lead Origin                                      0.00
Lead Source                                      0.39
Do Not Email                                     0.00
Do Not Call                                      0.00
Converted                                        0.00
TotalVisits                                      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
Magazine                                         0.00
Newspaper Article                                0.00
X Education Forums                               0.00
Newspaper                                        0.00
Digital Advertisement                            0.00
Through Recommendations                          0.00
Receive More Updates About Our Courses           0.00
Tags                                             0.00
Update me on Supply Chain Content                0.00
Get updates on DM Content                        0.00
City                                             0.00
I agree to pay the amount through cheque         0.00
A free copy of Mastering The Interview           0.00
Last Notable Activity                            0.00
dtype: float64
```

In [32]:
```
Leads_df['Lead Source'].value_counts(dropna=False)
```

Out[32]:
```
Google               2868
Direct Traffic       2543
Olark Chat           1755
Organic Search       1154
Reference             534
Welingak Website      142
Referral Sites        125
Facebook               55
NaN                    36
bing                    6
google                  5
Click2call              4
Press_Release           2
Social Media            2
Live Chat               2
youtubechannel          1
testone                 1
Pay per Click Ads       1
welearnblog_Home        1
WeLearn                 1
blog                    1
NC_EDM                  1
Name: Lead Source, dtype: int64
```

In [33]:
```
Leads_df['Lead Source'] = Leads_df['Lead Source'].replace(np.nan,'Others')
```

In [34]:
```
plt.figure(figsize=(15,5))
s1=sns.countplot(Leads_df['Lead Source'], hue=Leads_df.Converted)
s1.set_xticklabels(s1.get_xticklabels(),rotation=90)
plt.show()
```



In [35]:
```
Leads_df['Last Activity'].value_counts(dropna=False)
```

Out[35]:
```
Email Opened                    3437
SMS Sent                        2745
Olark Chat Conversation          973
Page Visited on Website          640
Converted to Lead                428
Email Bounced                    326
Email Link Clicked               267
Form Submitted on Website        116
NaN                              103
Unreachable                       93
Unsubscribed                      61
Had a Phone Conversation          30
Approached upfront                 9
View in browser link Clicked       6
Email Received                     2
Email Marked Spam                  2
Visited Booth in Tradeshow         1
Resubscribed to emails             1
Name: Last Activity, dtype: int64
```

In [36]:
```
Leads_df['Last Activity'] = Leads_df['Last Activity'].replace(np.nan,'Others')
Leads_df['Last Activity'] = Leads_df['Last Activity'].replace(['Unreachable','Unsubscrib
```

In [37]:
```
Leads_df['Last Activity'].value_counts(dropna=False)
```

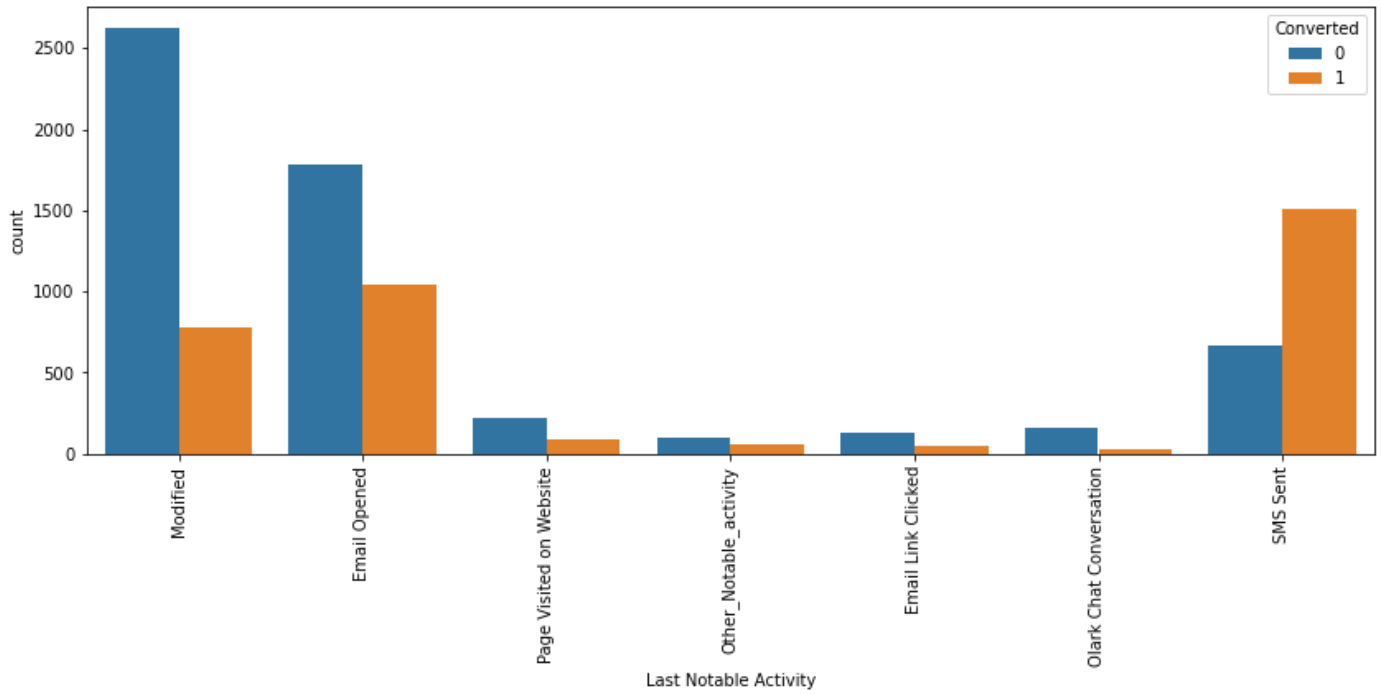
Out[37]:
```
Email Opened                 3437
SMS Sent                     2745
Olark Chat Conversation       973
Page Visited on Website       640
Converted to Lead             428
Email Bounced                 326
Others                        308
Email Link Clicked            267
Form Submitted on Website     116
Name: Last Activity, dtype: int64
```

In [38]:
```
Leads_df['Last Notable Activity'].value_counts()
```

Out[38]:
```
Modified                        3407
Email Opened                    2827
SMS Sent                        2172
Page Visited on Website          318
Olark Chat Conversation          183
Email Link Clicked               173
Email Bounced                     60
Unsubscribed                      47
Unreachable                       32
Had a Phone Conversation          14
Email Marked Spam                  2
Approached upfront                 1
Resubscribed to emails             1
View in browser link Clicked       1
Form Submitted on Website          1
Email Received                     1
Name: Last Notable Activity, dtype: int64
```

In [39]:
```
Leads_df['Last Notable Activity'] = Leads_df['Last Notable Activity'].replace(['Had a Ph
```

In [40]:
```
plt.figure(figsize = (14,5))
ax1=sns.countplot(x = "Last Notable Activity", hue = "Converted", data = Leads_df)
ax1.set_xticklabels(ax1.get_xticklabels(),rotation=90)
plt.show()
```



In [41]:
```
Leads_df['Last Notable Activity'].value_counts(dropna=False)
```

Out[41]:
```
Modified                 3407
Email Opened             2827
SMS Sent                 2172
Page Visited on Website   318
Olark Chat Conversation   183
Email Link Clicked        173
Other_Notable_activity    160
Name: Last Notable Activity, dtype: int64
```

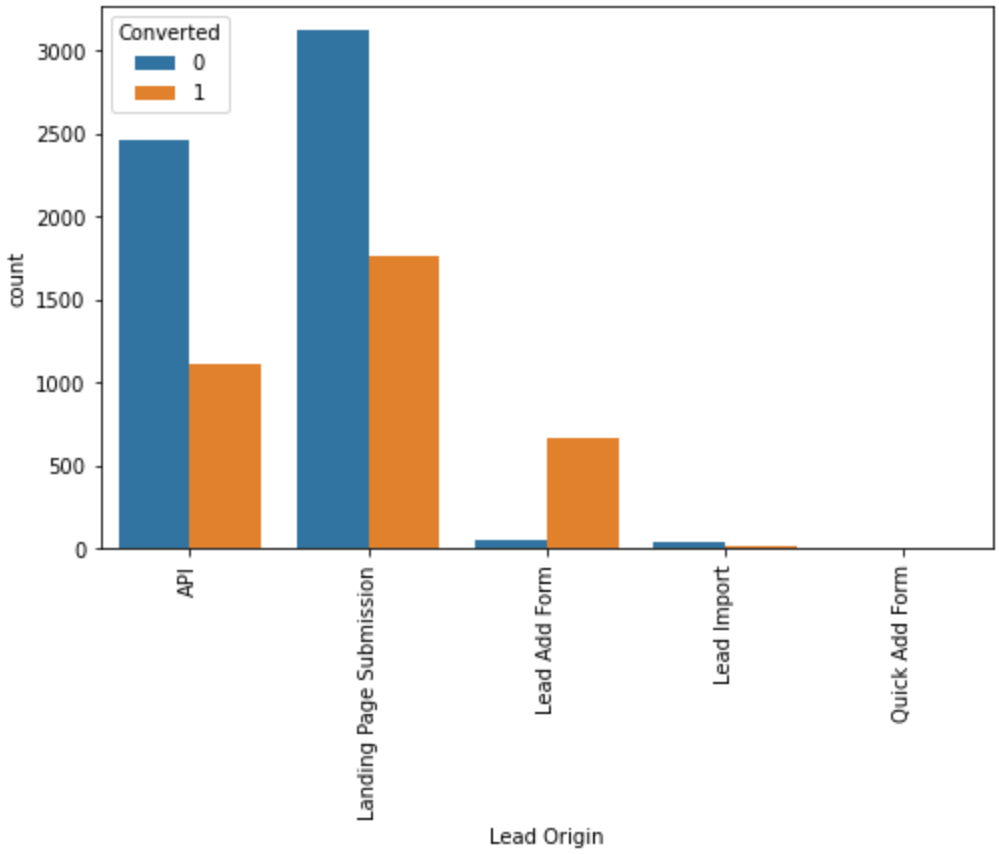
In [42]:
```
round(100*(Leads_df.isnull().sum()/len(Leads_df.index)), 2)
```

Out[42]:
```
Lead Origin                                      0.00
Lead Source                                      0.00
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                                    0.00
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
Magazine                                         0.00
Newspaper Article                                0.00
X Education Forums                               0.00
Newspaper                                        0.00
Digital Advertisement                            0.00
Through Recommendations                          0.00
Receive More Updates About Our Courses           0.00
Tags                                             0.00
Update me on Supply Chain Content                0.00
Get updates on DM Content                        0.00
City                                             0.00
I agree to pay the amount through cheque         0.00
A free copy of Mastering The Interview           0.00
Last Notable Activity                            0.00
dtype: float64
```

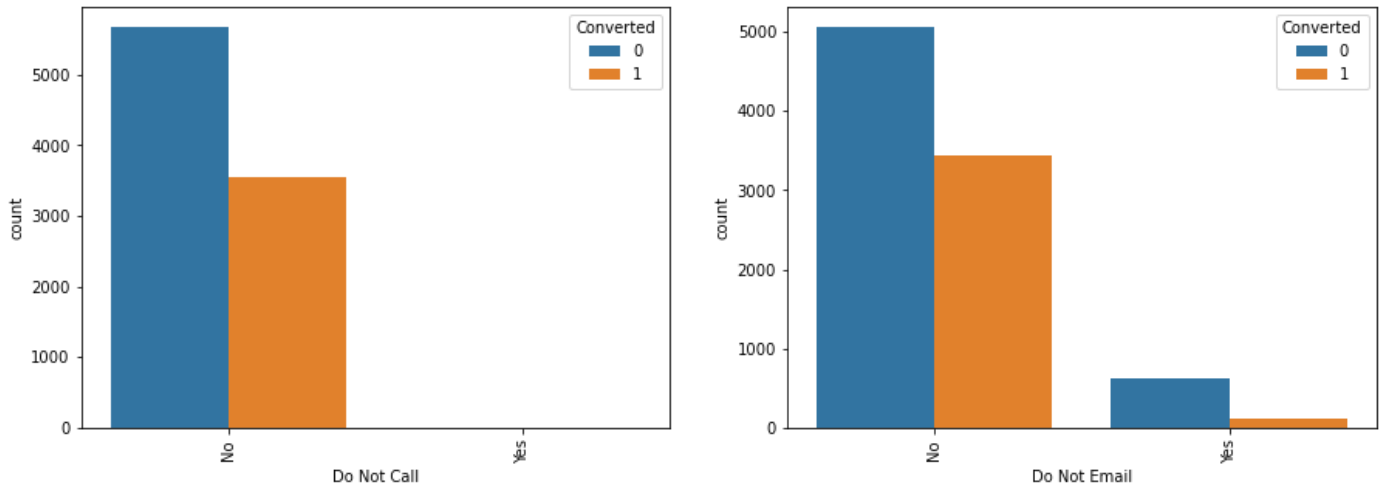
In [43]:
```
Leads_df['Lead Origin'].value_counts(dropna=False)
```

Out[43]:
```
Landing Page Submission    4886
API                        3580
Lead Add Form               718
Lead Import                  55
Quick Add Form                1
Name: Lead Origin, dtype: int64
```

In [44]:
```
plt.figure(figsize=(8,5))
s1=sns.countplot(Leads_df['Lead Origin'], hue=Leads_df.Converted)
s1.set_xticklabels(s1.get_xticklabels(),rotation=90)
plt.show()
```



In [45]:
```
plt.figure(figsize=(15,5))

ax1=plt.subplot(1, 2, 1)
ax1=sns.countplot(Leads_df['Do Not Call'], hue=Leads_df.Converted)
ax1.set_xticklabels(ax1.get_xticklabels(),rotation=90)

ax2=plt.subplot(1, 2, 2)
ax2=sns.countplot(Leads_df['Do Not Email'], hue=Leads_df.Converted)
ax2.set_xticklabels(ax2.get_xticklabels(),rotation=90)
plt.show()
```



In [46]:
```
Leads_df['Do Not Call'].value_counts(dropna=False)
```

Out[46]:
```
No     9238
Yes       2
Name: Do Not Call, dtype: int64
```

In [47]:
```
cols_to_drop.append('Do Not Call')
cols_to_drop
```

Out[47]:
```
['Country', 'What matters most to you in choosing a course', 'Do Not Call']
```

In [48]:
```
Leads_df.Search.value_counts(dropna=False)
```

Out[48]:
```
No     9226
Yes      14
Name: Search, dtype: int64
```

In [49]:
```
Leads_df.Magazine.value_counts(dropna=False)
```

Out[49]:
```
No    9240
Name: Magazine, dtype: int64
```

In [50]:
```
Leads_df['Newspaper Article'].value_counts(dropna=False)
```

Out[50]:
```
No     9238
Yes       2
Name: Newspaper Article, dtype: int64
```

In [51]:
```
Leads_df['X Education Forums'].value_counts(dropna=False)
```

Out[51]:
```
No     9239
Yes       1
Name: X Education Forums, dtype: int64
```

In [52]:
```
Leads_df['Newspaper'].value_counts(dropna=False)
```

Out[52]:
```
No     9239
Yes       1
Name: Newspaper, dtype: int64
```

In [53]:
```
Leads_df['Digital Advertisement'].value_counts(dropna=False)
```

Out[53]:
```
No     9236
Yes       4
Name: Digital Advertisement, dtype: int64
```

In [54]:
```
Leads_df['Through Recommendations'].value_counts(dropna=False)
```

Out[54]:
```
No     9233
Yes       7
Name: Through Recommendations, dtype: int64
```

In [55]:
```
Leads_df['Receive More Updates About Our Courses'].value_counts(dropna=False)
```

Out[55]:
```
No    9240
Name: Receive More Updates About Our Courses, dtype: int64
```

In [56]:
```
Leads_df['Update me on Supply Chain Content'].value_counts(dropna=False)
```

Out[56]:
```
No    9240
Name: Update me on Supply Chain Content, dtype: int64
```

In [57]:
```
Leads_df['Get updates on DM Content'].value_counts(dropna=False)
```

Out[57]:
```
No    9240
Name: Get updates on DM Content, dtype: int64
```

In [58]:
```
Leads_df['I agree to pay the amount through cheque'].value_counts(dropna=False)
```

Out[58]:
```
No    9240
Name: I agree to pay the amount through cheque, dtype: int64
```

In [59]:
```
Leads_df['A free copy of Mastering The Interview'].value_counts(dropna=False)
```

Out[59]:
```
No     6352
Yes    2888
Name: A free copy of Mastering The Interview, dtype: int64
```

In [60]:
```
cols_to_drop.extend(['Search','Magazine','Newspaper Article','X Education Forums','Newsp
```

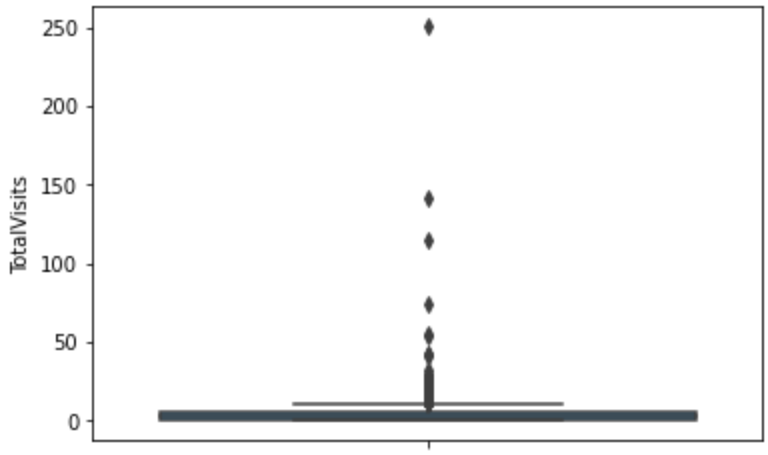
In [61]:
```
#list of columns to be dropped
cols_to_drop
```

Out[61]:
```
['Country',
 'What matters most to you in choosing a course',
 'Do Not Call',
 'Search',
 'Magazine',
 'Newspaper Article',
 'X Education Forums',
 'Newspaper',
 'Digital Advertisement',
 'Through Recommendations',
 'Receive More Updates About Our Courses',
 'Update me on Supply Chain Content',
 'Get updates on DM Content',
 'I agree to pay the amount through cheque']
```

In [62]:
```
Leads_df = Leads_df.drop(cols_to_drop,1)
Leads_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9240 entries, 0 to 9239
Data columns (total 14 columns):
 #   Column                                  Non-Null Count  Dtype
---  ------                                  --------------  -----
 0   Lead Origin                             9240 non-null   object
 1   Lead Source                             9240 non-null   object
 2   Do Not Email                            9240 non-null   object
 3   Converted                               9240 non-null   int64
 4   TotalVisits                             9103 non-null   float64
 5   Total Time Spent on Website             9240 non-null   int64
 6   Page Views Per Visit                    9103 non-null   float64
 7   Last Activity                           9240 non-null   object
 8   Specialization                          9240 non-null   object
 9   What is your current occupation         9240 non-null   object
 10  Tags                                    9240 non-null   object
 11  City                                    9240 non-null   object
 12  A free copy of Mastering The Interview  9240 non-null   object
 13  Last Notable Activity                   9240 non-null   object
dtypes: float64(2), int64(2), object(10)
memory usage: 1010.8+ KB
```

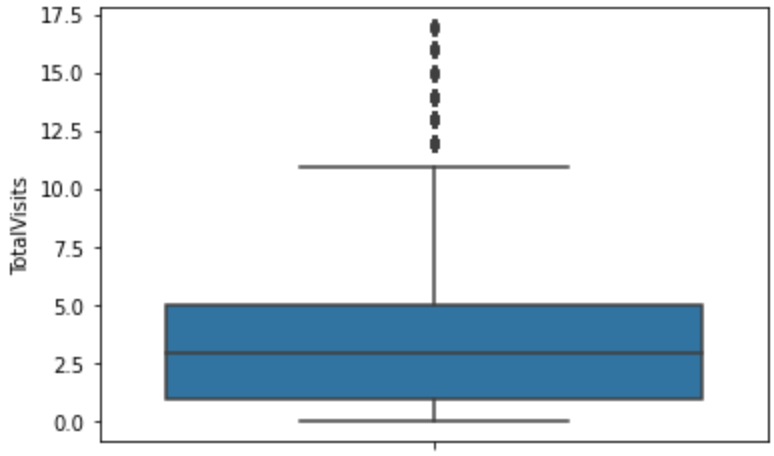
In [63]:
```
#Total Visits
#visualizing spread of variable

plt.figure(figsize=(6,4))
sns.boxplot(y=Leads_df ['TotalVisits'])
plt.show()
```



In [64]:
```
#checking percentile values for "Total Visits"
Leads_df['TotalVisits'].describe(percentiles=[0.05,.25, .5, .75, .90, .95, .99])
```

Out[64]:
```
count    9103.000000
mean        3.445238
std         4.854853
min         0.000000
5%          0.000000
25%         1.000000
50%         3.000000
75%         5.000000
90%         7.000000
95%        10.000000
99%        17.000000
max       251.000000
Name: TotalVisits, dtype: float64
```

In [65]:
```
#Outlier Treatment: Remove top & bottom 1% of the Column Outlier values

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



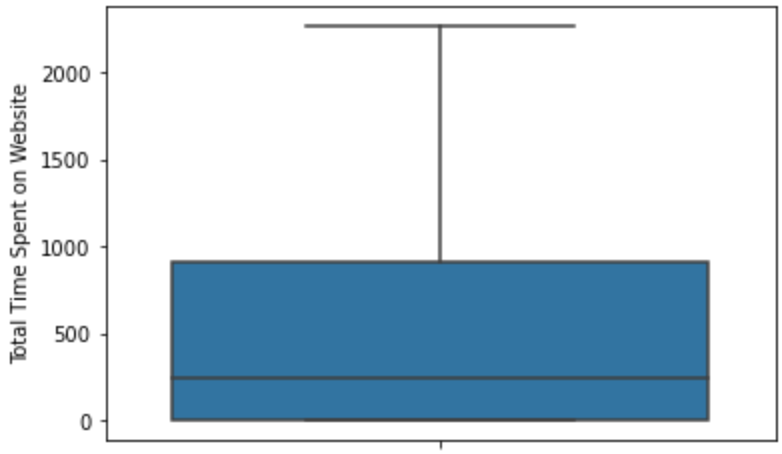
In [66]:
```
Leads_df.shape
```

Out[66]:
```
(9020, 14)
```

In [67]:
```
#checking percentiles for "Total Time Spent on Website"

Leads_df['Total Time Spent on Website'].describe(percentiles=[0.05,.25, .5, .75, .90, .9
```

Out[67]:
```
count    9020.000000
mean      479.759534
std       544.688157
min         0.000000
5%          0.000000
25%         7.000000
50%       243.000000
75%       915.250000
90%      1371.000000
95%      1554.050000
99%      1836.620000
max      2272.000000
Name: Total Time Spent on Website, dtype: float64
```

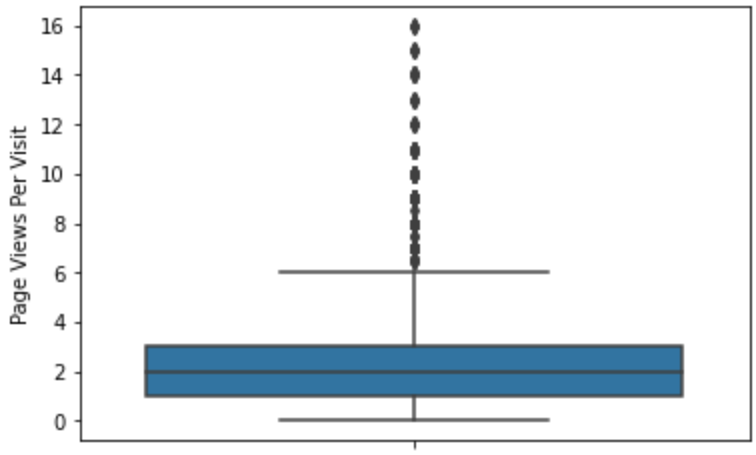
In [68]:
```
#visualizing spread of numeric variable

plt.figure(figsize=(6,4))
sns.boxplot(y=Leads_df['Total Time Spent on Website'])
plt.show()
```



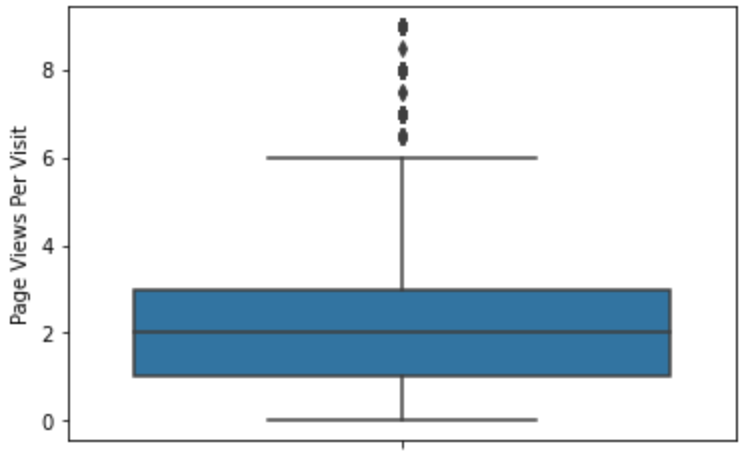
In [69]:
```
Leads_df['Page Views Per Visit'].describe()
```

Out[69]:
```
count    9020.000000
mean        2.337271
std         2.062363
min         0.000000
25%         1.000000
50%         2.000000
75%         3.000000
max        16.000000
Name: Page Views Per Visit, dtype: float64
```

In [70]:
```
#visualizing spread of numeric variable

plt.figure(figsize=(6,4))
sns.boxplot(y=Leads_df['Page Views Per Visit'])
plt.show()
```



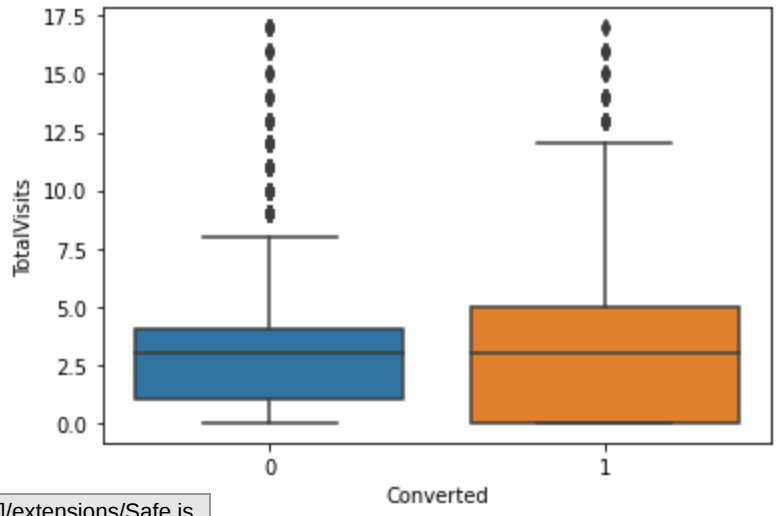
In [71]:
```
#Outlier Treatment: Remove top & bottom 1%

Q3 = Leads_df['Page Views Per Visit'].quantile(0.99)
Leads_df = Leads_df[Leads_df['Page Views Per Visit'] <= Q3]
Q1 = Leads_df['Page Views Per Visit'].quantile(0.01)
Leads_df = Leads_df[Leads_df['Page Views Per Visit'] >= Q1]
sns.boxplot(y=Leads_df['Page Views Per Visit'])
plt.show()
```



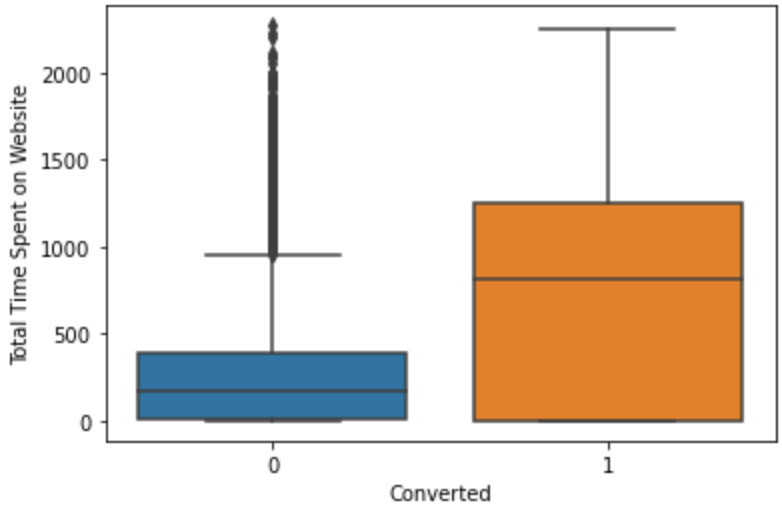
In [72]:
```
Leads_df.shape
```

Out[72]:
```
(8953, 14)
```

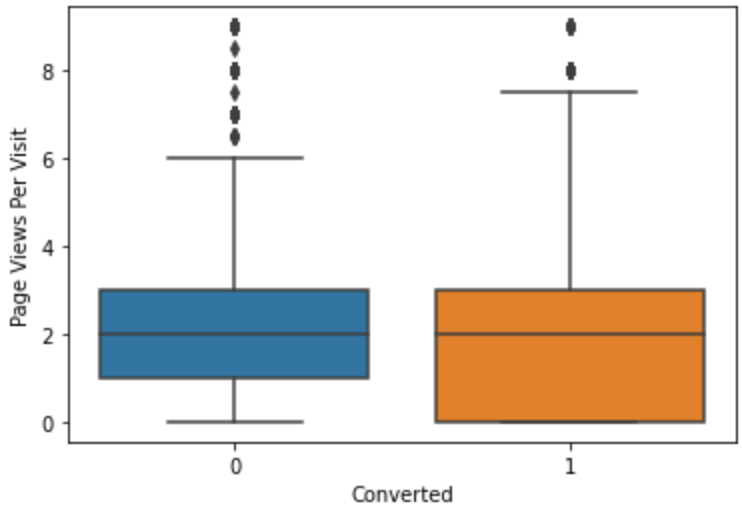
In [73]:
```
#checking Spread of "Total Visits" vs Converted variable
sns.boxplot(y = 'TotalVisits', x = 'Converted', data = Leads_df)
plt.show()
```



In [74]:
```
#checking Spread of "Total Time Spent on Website" vs Converted variable
sns.boxplot(x=Leads_df.Converted, y=Leads_df['Total Time Spent on Website'])
plt.show()
```



In [75]:
```
#checking Spread of "Page Views Per Visit" vs Converted variable
sns.boxplot(x=Leads_df.Converted,y=Leads_df['Page Views Per Visit'])
plt.show()
```



In [76]:
```
round(100*(Leads_df.isnull().sum()/len(Leads_df.index)),2)
```

Out[76]:
```
Lead Origin                              0.0
Lead Source                              0.0
Do Not Email                             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
Specialization                           0.0
What is your current occupation          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 [77]:
```
#getting a list of categorical columns
cat_cols= Leads_df.select_dtypes(include=['object']).columns
cat_cols
```

Out[77]:
```
Index(['Lead Origin', 'Lead Source', 'Do Not Email', 'Last Activity',
       'Specialization', 'What is your current occupation', 'Tags', 'City',
       'A free copy of Mastering The Interview', 'Last Notable Activity'],
      dtype='object')
```

In [78]:
```
# List of variables to map
varlist =  ['A free copy of Mastering The Interview','Do Not Email']
def binary_map(x):
    return x.map({'Yes': 1, "No": 0})

Leads_df[varlist] = Leads_df[varlist].apply(binary_map)
```

In [79]:
```
#getting dummies and dropping the first column and adding the results to the master data
dummy = pd.get_dummies(Leads_df[['Lead Origin','What is your current occupation',
                                 'City']], drop_first=True)

Leads_df = pd.concat([Leads_df,dummy],1)
```

In [80]:
```
dummy = pd.get_dummies(Leads_df['Specialization'], prefix  = 'Specialization')
dummy = dummy.drop(['Specialization_Not Specified'], 1)
Leads_df = pd.concat([Leads_df, dummy], axis = 1)
```

In [81]:
```
dummy = pd.get_dummies(Leads_df['Lead Source'], prefix  = 'Lead Source')
dummy = dummy.drop(['Lead Source_Others'], 1)
Leads_df = pd.concat([Leads_df, dummy], axis = 1)
```

In [82]:
```
dummy = pd.get_dummies(Leads_df['Last Activity'], prefix  = 'Last Activity')
dummy = dummy.drop(['Last Activity_Others'], 1)
Leads_df = pd.concat([Leads_df, dummy], axis = 1)
```

In [83]:
```
dummy = pd.get_dummies(Leads_df['Last Notable Activity'], prefix  = 'Last Notable Activi
dummy = dummy.drop(['Last Notable Activity_Other_Notable_activity'], 1)
Leads_df = pd.concat([Leads_df, dummy], axis = 1)
```

In [84]:
```
dummy = pd.get_dummies(Leads_df['Tags'], prefix  = 'Tags')
dummy = dummy.drop(['Tags_Not Specified'], 1)
Leads_df = pd.concat([Leads_df, dummy], axis = 1)
```

In [85]:
```
#dropping the original columns after dummy variable creation
Leads_df.drop(cat_cols,1,inplace = True)
```

In [86]:
```
Leads_df.head()
```

Out[86]:

	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	What is your current occupation_Housewife	oc
0	0	0.0	0	0.0	0	0	0	0	
1	0	5.0	674	2.5	0	0	0	0	
2	1	2.0	1532	2.0	1	0	0	0	
3	0	1.0	305	1.0	1	0	0	0	
4	1	2.0	1428	1.0	1	0	0	0	

5 rows × 77 columns

In [87]:
```
from sklearn.model_selection import train_test_split
```

```
# Putting response variable to y
y = Leads_df['Converted']

y.head()

X=Leads_df.drop('Converted', axis=1)
```

In [88]:
```
# Splitting the data into train and test
X_train, X_test, y_train, y_test = train_test_split(X, y, train_size=0.7, test_size=0.3,
```

In [89]:
```
X_train.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 6267 entries, 9196 to 5825
Data columns (total 76 columns):
 #   Column                                            Non-Null Count  Dtype  
---  ------                                            --------------  -----  
 0   TotalVisits                                       6267 non-null   float64
 1   Total Time Spent on Website                       6267 non-null   int64  
 2   Page Views Per Visit                              6267 non-null   float64
 3   Lead Origin_Landing Page Submission               6267 non-null   uint8  
 4   Lead Origin_Lead Add Form                         6267 non-null   uint8  
 5   Lead Origin_Lead Import                           6267 non-null   uint8  
 6   What is your current occupation_Housewife         6267 non-null   uint8  
 7   What is your current occupation_Other             6267 non-null   uint8  
 8   What is your current occupation_Student           6267 non-null   uint8  
 9   What is your current occupation_Unemployed        6267 non-null   uint8  
 10  What is your current occupation_Working Professional  6267 non-null   uint8  
 11  City_Other Cities                                 6267 non-null   uint8  
 12  City_Other Cities of Maharashtra                  6267 non-null   uint8  
 13  City_Other Metro Cities                           6267 non-null   uint8  
 14  City_Thane & Outskirts                            6267 non-null   uint8  
 15  City_Tier II Cities                               6267 non-null   uint8  
 16  Specialization_Banking, Investment And Insurance  6267 non-null   uint8  
 17  Specialization_Business Administration            6267 non-null   uint8  
 18  Specialization_E-Business                         6267 non-null   uint8  
 19  Specialization_E-COMMERCE                         6267 non-null   uint8  
 20  Specialization_Finance Management                 6267 non-null   uint8  
 21  Specialization_Healthcare Management              6267 non-null   uint8  
 22  Specialization_Hospitality Management             6267 non-null   uint8  
 23  Specialization_Human Resource Management          6267 non-null   uint8  
 24  Specialization_IT Projects Management             6267 non-null   uint8  
 25  Specialization_International Business             6267 non-null   uint8  
 26  Specialization_Marketing Management               6267 non-null   uint8  
 27  Specialization_Media and Advertising              6267 non-null   uint8  
 28  Specialization_Operations Management              6267 non-null   uint8  
 29  Specialization_Retail Management                  6267 non-null   uint8  
 30  Specialization_Rural and Agribusiness             6267 non-null   uint8  
 31  Specialization_Services Excellence                6267 non-null   uint8  
 32  Specialization_Supply Chain Management            6267 non-null   uint8  
 33  Specialization_Travel and Tourism                 6267 non-null   uint8  
 34  Lead Source_Click2call                            6267 non-null   uint8  
 35  Lead Source_Direct Traffic                        6267 non-null   uint8  
 36  Lead Source_Facebook                              6267 non-null   uint8  
 37  Lead Source_Google                                6267 non-null   uint8  
 38  Lead Source_Live Chat                             6267 non-null   uint8  
 39  Lead Source_NC_EDM                                6267 non-null   uint8  
 40  Lead Source_Olark Chat                            6267 non-null   uint8  
 41  Lead Source_Organic Search                        6267 non-null   uint8  
 42  Lead Source_Pay per Click Ads                     6267 non-null   uint8  
 43  Lead Source_Press_Release                         6267 non-null   uint8  
 44  Lead Source_Reference                             6267 non-null   uint8  
 45  Lead Source_Referral Sites                        6267 non-null   uint8  
 46  Lead Source_Social Media                          6267 non-null   uint8  
 47  Lead Source_WeLearn                               6267 non-null   uint8  
 48  Lead Source_Welingak Website                      6267 non-null   uint8  
 49  Lead Source_bing                                  6267 non-null   uint8  
 50  Lead Source_blog                                  6267 non-null   uint8  
 51  Lead Source_google                                6267 non-null   uint8  
 52  Lead Source_testone                               6267 non-null   uint8  
 53  Lead Source_welearnblog_Home                      6267 non-null   uint8  
 54  Lead Source_youtubechannel                        6267 non-null   uint8  
 55  Last Activity_Converted to Lead                   6267 non-null   uint8  
 56  Last Activity_Email Bounced                       6267 non-null   uint8  
 57  Last Activity_Email Link Clicked                  6267 non-null   uint8  
 58  Last Activity_Email Opened                        6267 non-null   uint8  
```

```
 59  Last Activity_Form Submitted on Website                6267 non-null   uint8
 60  Last Activity_Olark Chat Conversation                  6267 non-null   uint8
 61  Last Activity_Page Visited on Website                  6267 non-null   uint8
 62  Last Activity_SMS Sent                                 6267 non-null   uint8
 63  Last Notable Activity_Email Link Clicked               6267 non-null   uint8
 64  Last Notable Activity_Email Opened                     6267 non-null   uint8
 65  Last Notable Activity_Modified                         6267 non-null   uint8
 66  Last Notable Activity_Olark Chat Conversation          6267 non-null   uint8
 67  Last Notable Activity_Page Visited on Website          6267 non-null   uint8
 68  Last Notable Activity_SMS Sent                         6267 non-null   uint8
 69  Tags_Busy                                              6267 non-null   uint8
 70  Tags_Closed by Horizzon                                6267 non-null   uint8
 71  Tags_Interested in other courses                       6267 non-null   uint8
 72  Tags_Lost to EINS                                      6267 non-null   uint8
 73  Tags_Other_Tags                                        6267 non-null   uint8
 74  Tags_Ringing                                           6267 non-null   uint8
 75  Tags_Will revert after reading the email               6267 non-null   uint8
dtypes: float64(2), int64(1), uint8(73)
memory usage: 642.6 KB
```

In [90]:
```
#scaling numeric columns
from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()

num_cols=X_train.select_dtypes(include=['float64', 'int64']).columns

X_train[num_cols] = scaler.fit_transform(X_train[num_cols])

X_train.head()
```

Out[90]:

	TotalVisits	Total Time Spent on Website	Page Views Per Visit	Lead Origin_Landing Page Submission	Lead Origin_Lead Add Form	Lead Origin_Lead Import	What is your current occupation_Housewife	W occupa
9196	0.668862	1.848117	1.455819	1	0	0	0	
4696	-0.030697	-0.037832	0.399961	1	0	0	0	
3274	0.319082	-0.642138	-0.127967	1	0	0	0	
2164	-0.380477	-0.154676	-0.127967	0	0	0	0	
1667	0.319082	1.258415	-0.481679	0	0	0	0	

5 rows × 76 columns

In [91]:
```
import statsmodels.api as sm
```

In [92]:
```
from sklearn.linear_model import LogisticRegression
logreg = LogisticRegression()

from sklearn.feature_selection import RFE
rfe = RFE(estimator=LogisticRegression(), n_features_to_select=15)
rfe = rfe.fit(X_train, y_train)
```

In [93]:
```
rfe.support_
```

Out[93]:
```
array([False,  True, False, False,  True, False, False, False, False,
       False, False, False, False, False, False, False, False, False,
       False, False, False, False, False, False, False, False, False,
       False, False, False, False, False, False, False, False,  True,
       False, False, False, False, False, False, False, False, False,
        True, False, False,  True, False, False, False, False, False,
       False, False, False, False, False, False, False, False,  True,
       False, False,  True,  True, False,  True, False,  True,  True,
        True,  True,  True,  True])
```

In [94]:
```
list(zip(X_train.columns, rfe.support_, rfe.ranking_))
```

Out[94]:
```
[('TotalVisits', False, 34),
 ('Total Time Spent on Website', True, 1),
 ('Page Views Per Visit', False, 33),
 ('Lead Origin_Landing Page Submission', False, 10),
 ('Lead Origin_Lead Add Form', True, 1),
 ('Lead Origin_Lead Import', False, 21),
 ('What is your current occupation_Housewife', False, 36),
 ('What is your current occupation_Other', False, 46),
 ('What is your current occupation_Student', False, 45),
 ('What is your current occupation_Unemployed', False, 27),
 ('What is your current occupation_Working Professional', False, 8),
 ('City_Other Cities', False, 31),
 ('City_Other Cities of Maharashtra', False, 48),
 ('City_Other Metro Cities', False, 57),
 ('City_Thane & Outskirts', False, 51),
 ('City_Tier II Cities', False, 38),
 ('Specialization_Banking, Investment And Insurance', False, 19),
 ('Specialization_Business Administration', False, 52),
 ('Specialization_E-Business', False, 43),
 ('Specialization_E-COMMERCE', False, 29),
 ('Specialization_Finance Management', False, 50),
 ('Specialization_Healthcare Management', False, 14),
 ('Specialization_Hospitality Management', False, 15),
 ('Specialization_Human Resource Management', False, 44),
 ('Specialization_IT Projects Management', False, 28),
 ('Specialization_International Business', False, 53),
 ('Specialization_Marketing Management', False, 22),
 ('Specialization_Media and Advertising', False, 41),
 ('Specialization_Operations Management', False, 35),
 ('Specialization_Retail Management', False, 49),
 ('Specialization_Rural and Agribusiness', False, 47),
 ('Specialization_Services Excellence', False, 39),
 ('Specialization_Supply Chain Management', False, 16),
 ('Specialization_Travel and Tourism', False, 5),
 ('Lead Source_Click2call', False, 54),
 ('Lead Source_Direct Traffic', True, 1),
 ('Lead Source_Facebook', False, 20),
 ('Lead Source_Google', False, 3),
 ('Lead Source_Live Chat', False, 59),
 ('Lead Source_NC_EDM', False, 17),
 ('Lead Source_Olark Chat', False, 23),
 ('Lead Source_Organic Search', False, 2),
 ('Lead Source_Pay per Click Ads', False, 58),
 ('Lead Source_Press_Release', False, 60),
 ('Lead Source_Reference', False, 13),
 ('Lead Source_Referral Sites', True, 1),
 ('Lead Source_Social Media', False, 61),
 ('Lead Source_WeLearn', False, 55),
 ('Lead Source_Welingak Website', True, 1),
 ('Lead Source_bing', False, 30),
 ('Lead Source_blog', False, 37),
 ('Lead Source_google', False, 18),
 ('Lead Source_testone', False, 56),
 ('Lead Source_welearnblog_Home', False, 40),
 ('Lead Source_youtubechannel', False, 62),
 ('Last Activity_Converted to Lead', False, 11),
 ('Last Activity_Email Bounced', False, 6),
 ('Last Activity_Email Link Clicked', False, 42),
 ('Last Activity_Email Opened', False, 25),
 ('Last Activity_Form Submitted on Website', False, 24),
 ('Last Activity_Olark Chat Conversation', False, 7),
 ('Last Activity_Page Visited on Website', False, 12),
 ('Last Activity_SMS Sent', True, 1),
 ('Last Notable Activity_Email Link Clicked', False, 4),
```

```
('Last Notable Activity_Email Opened', False, 26),
('Last Notable Activity_Modified', True, 1),
('Last Notable Activity_Olark Chat Conversation', True, 1),
('Last Notable Activity_Page Visited on Website', False, 32),
('Last Notable Activity_SMS Sent', True, 1),
('Tags_Busy', False, 9),
('Tags_Closed by Horizzon', True, 1),
('Tags_Interested in other courses', True, 1),
('Tags_Lost to EINS', True, 1),
('Tags_Other_Tags', True, 1),
('Tags_Ringing', True, 1),
('Tags_Will revert after reading the email', True, 1)]
```

In [95]:
```
#list of RFE supported columns
col = X_train.columns[rfe.support_]
col
```

Out[95]:
```
Index(['Total Time Spent on Website', 'Lead Origin_Lead Add Form',
       'Lead Source_Direct Traffic', 'Lead Source_Referral Sites',
       'Lead Source_Welingak Website', 'Last Activity_SMS Sent',
       'Last Notable Activity_Modified',
       'Last Notable Activity_Olark Chat Conversation',
       'Last Notable Activity_SMS Sent', 'Tags_Closed by Horizzon',
       'Tags_Interested in other courses', 'Tags_Lost to EINS',
       'Tags_Other_Tags', 'Tags_Ringing',
       'Tags_Will revert after reading the email'],
      dtype='object')
```

In [96]:
```
X_train.columns[~rfe.support_]
```

Out[96]:
```
Index(['TotalVisits', 'Page Views Per Visit',
       'Lead Origin_Landing Page Submission', 'Lead Origin_Lead Import',
       'What is your current occupation_Housewife',
       'What is your current occupation_Other',
       'What is your current occupation_Student',
       'What is your current occupation_Unemployed',
       'What is your current occupation_Working Professional',
       'City_Other Cities', 'City_Other Cities of Maharashtra',
       'City_Other Metro Cities', 'City_Thane & Outskirts',
       'City_Tier II Cities',
       'Specialization_Banking, Investment And Insurance',
       'Specialization_Business Administration', 'Specialization_E-Business',
       'Specialization_E-COMMERCE', 'Specialization_Finance Management',
       'Specialization_Healthcare Management',
       'Specialization_Hospitality Management',
       'Specialization_Human Resource Management',
       'Specialization_IT Projects Management',
       'Specialization_International Business',
       'Specialization_Marketing Management',
       'Specialization_Media and Advertising',
       'Specialization_Operations Management',
       'Specialization_Retail Management',
       'Specialization_Rural and Agribusiness',
       'Specialization_Services Excellence',
       'Specialization_Supply Chain Management',
       'Specialization_Travel and Tourism', 'Lead Source_Click2call',
       'Lead Source_Facebook', 'Lead Source_Google', 'Lead Source_Live Chat',
       'Lead Source_NC_EDM', 'Lead Source_Olark Chat',
       'Lead Source_Organic Search', 'Lead Source_Pay per Click Ads',
       'Lead Source_Press_Release', 'Lead Source_Reference',
       'Lead Source_Social Media', 'Lead Source_WeLearn', 'Lead Source_bing',
       'Lead Source_blog', 'Lead Source_google', 'Lead Source_testone',
       'Lead Source_welearnblog_Home', 'Lead Source_youtubechannel',
       'Last Activity_Converted to Lead', 'Last Activity_Email Bounced',
       'Last Activity_Email Link Clicked', 'Last Activity_Email Opened',
       'Last Activity_Form Submitted on Website',
       'Last Activity_Olark Chat Conversation',
       'Last Activity_Page Visited on Website',
       'Last Notable Activity_Email Link Clicked',
       'Last Notable Activity_Email Opened',
       'Last Notable Activity_Page Visited on Website', 'Tags_Busy'],
      dtype='object')
```

In [97]:
```
#BUILDING MODEL #1
X_train_sm = sm.add_constant(X_train[col])
logm1 = sm.GLM(y_train,X_train_sm, family = sm.families.Binomial())
res = logm1.fit()
res.summary()
```

Out[97]:

Generalized Linear Model Regression Results

Dep. Variable:	Converted	**No. Observations:**	6267
Model:	GLM	**Df Residuals:**	6251
Model Family:	Binomial	**Df Model:**	15
Link Function:	Logit	**Scale:**	1.0000
Method:	IRLS	**Log-Likelihood:**	-1254.7
Date:	Tue, 25 Apr 2023	**Deviance:**	2509.3
Time:	11:53:11	**Pearson chi2:**	8.34e+03
No. Iterations:	8	**Pseudo R-squ. (CS):**	0.6048
Covariance Type:	nonrobust		

	coef	std err	z	P>|z|	[0.025	0.975]
const	-1.1899	0.088	-13.480	0.000	-1.363	-1.017
Total Time Spent on Website	0.8970	0.053	16.999	0.000	0.794	1.000
Lead Origin_Lead Add Form	1.6712	0.450	3.714	0.000	0.789	2.553
Lead Source_Direct Traffic	-0.8320	0.129	-6.471	0.000	-1.084	-0.580
Lead Source_Referral Sites	-0.5284	0.465	-1.138	0.255	-1.439	0.382
Lead Source_Welingak Website	3.9043	1.110	3.518	0.000	1.729	6.079
Last Activity_SMS Sent	1.2373	0.223	5.555	0.000	0.801	1.674
Last Notable Activity_Modified	-1.2839	0.150	-8.532	0.000	-1.579	-0.989
Last Notable Activity_Olark Chat Conversation	-1.7123	0.490	-3.496	0.000	-2.672	-0.752
Last Notable Activity_SMS Sent	1.0151	0.257	3.943	0.000	0.511	1.520
Tags_Closed by Horizzon	6.9834	1.019	6.853	0.000	4.986	8.981
Tags_Interested in other courses	-2.1641	0.407	-5.321	0.000	-2.961	-1.367
Tags_Lost to EINS	5.7302	0.608	9.419	0.000	4.538	6.923
Tags_Other_Tags	-2.4417	0.210	-11.633	0.000	-2.853	-2.030
Tags_Ringing	-3.5858	0.243	-14.752	0.000	-4.062	-3.109
Tags_Will revert after reading the email	4.4263	0.185	23.989	0.000	4.065	4.788

In [98]:
```
#dropping column with high p-value
col = col.drop('Lead Source_Referral Sites',1)
```

In [99]:
```
#BUILDING MODEL #2
X_train_sm = sm.add_constant(X_train[col])
logm2 = sm.GLM(y_train,X_train_sm, family = sm.families.Binomial())
res = logm2.fit()
res.summary()
```

Out[99]:

Generalized Linear Model Regression Results

Dep. Variable:	Converted	**No. Observations:**	6267
Model:	GLM	**Df Residuals:**	6252
Model Family:	Binomial	**Df Model:**	14
Link Function:	Logit	**Scale:**	1.0000
Method:	IRLS	**Log-Likelihood:**	-1255.3
Date:	Tue, 25 Apr 2023	**Deviance:**	2510.7
Time:	11:53:11	**Pearson chi2:**	8.34e+03
No. Iterations:	8	**Pseudo R-squ. (CS):**	0.6047
Covariance Type:	nonrobust		

	coef	std err	z	P>|z|	[0.025	0.975]
const	-1.2029	0.088	-13.729	0.000	-1.375	-1.031
Total Time Spent on Website	0.8963	0.053	16.979	0.000	0.793	1.000
Lead Origin_Lead Add Form	1.6795	0.450	3.735	0.000	0.798	2.561
Lead Source_Direct Traffic	-0.8224	0.128	-6.409	0.000	-1.074	-0.571
Lead Source_Welingak Website	3.9060	1.110	3.520	0.000	1.731	6.081
Last Activity_SMS Sent	1.2437	0.223	5.584	0.000	0.807	1.680
Last Notable Activity_Modified	-1.2791	0.150	-8.501	0.000	-1.574	-0.984
Last Notable Activity_Olark Chat Conversation	-1.7079	0.489	-3.491	0.000	-2.667	-0.749
Last Notable Activity_SMS Sent	1.0150	0.257	3.943	0.000	0.510	1.520
Tags_Closed by Horizzon	6.9868	1.019	6.857	0.000	4.990	8.984
Tags_Interested in other courses	-2.2028	0.409	-5.391	0.000	-3.004	-1.402
Tags_Lost to EINS	5.7337	0.608	9.426	0.000	4.541	6.926
Tags_Other_Tags	-2.4401	0.210	-11.625	0.000	-2.852	-2.029
Tags_Ringing	-3.5818	0.243	-14.740	0.000	-4.058	-3.106
Tags_Will revert after reading the email	4.4234	0.184	23.993	0.000	4.062	4.785

In [100…

```
# Check for the VIF values of the feature variables.
from statsmodels.stats.outliers_influence import variance_inflation_factor
```

In [101…

```
# Create a dataframe that will contain the names of all the feature variables and their
vif = pd.DataFrame()
vif['Features'] = X_train[col].columns
vif['VIF'] = [variance_inflation_factor(X_train[col].values, i) for i in range(X_train[c
vif['VIF'] = round(vif['VIF'], 2)
vif = vif.sort_values(by = "VIF", ascending = False)
vif
```

Out[101]:

	Features	VIF
7	Last Notable Activity_SMS Sent	6.22
4	Last Activity_SMS Sent	6.12
1	Lead Origin_Lead Add Form	1.82
5	Last Notable Activity_Modified	1.69
13	Tags_Will revert after reading the email	1.61
2	Lead Source_Direct Traffic	1.38
3	Lead Source_Welingak Website	1.34
11	Tags_Other_Tags	1.26
0	Total Time Spent on Website	1.22
8	Tags_Closed by Horizzon	1.21
12	Tags_Ringing	1.18
9	Tags_Interested in other courses	1.13
10	Tags_Lost to EINS	1.06
6	Last Notable Activity_Olark Chat Conversation	1.01

In [102…

```
#dropping variable with high VIF
col = col.drop('Last Notable Activity_SMS Sent',1)
```

In [103…

```
#BUILDING MODEL #3
X_train_sm = sm.add_constant(X_train[col])
logm3 = sm.GLM(y_train,X_train_sm, family = sm.families.Binomial())
res = logm3.fit()
res.summary()
```

Out[103]:

Generalized Linear Model Regression Results

Dep. Variable:	Converted	**No. Observations:**	6267
Model:	GLM	**Df Residuals:**	6253
Model Family:	Binomial	**Df Model:**	13
Link Function:	Logit	**Scale:**	1.0000
Method:	IRLS	**Log-Likelihood:**	-1263.3
Date:	Tue, 25 Apr 2023	**Deviance:**	2526.6
Time:	11:53:12	**Pearson chi2:**	8.51e+03
No. Iterations:	8	**Pseudo R-squ. (CS):**	0.6037
Covariance Type:	nonrobust		

	coef	std err	z	P>|z|	[0.025	0.975]
const	-1.1179	0.084	-13.382	0.000	-1.282	-0.954
Total Time Spent on Website	0.8896	0.053	16.907	0.000	0.786	0.993
Lead Origin_Lead Add Form	1.6630	0.455	3.657	0.000	0.772	2.554
Lead Source_Direct Traffic	-0.8212	0.127	-6.471	0.000	-1.070	-0.572
Lead Source_Welingak Website	3.8845	1.114	3.488	0.000	1.701	6.068
Last Activity_SMS Sent	1.9981	0.113	17.718	0.000	1.777	2.219
Last Notable Activity_Modified	-1.6525	0.124	-13.279	0.000	-1.896	-1.409
Last Notable Activity_Olark Chat Conversation	-1.8023	0.491	-3.669	0.000	-2.765	-0.839
Tags_Closed by Horizzon	7.1955	1.020	7.053	0.000	5.196	9.195
Tags_Interested in other courses	-2.1318	0.406	-5.253	0.000	-2.927	-1.336
Tags_Lost to EINS	5.9177	0.611	9.689	0.000	4.721	7.115
Tags_Other_Tags	-2.3737	0.206	-11.507	0.000	-2.778	-1.969
Tags_Ringing	-3.4531	0.238	-14.532	0.000	-3.919	-2.987
Tags_Will revert after reading the email	4.5070	0.188	24.002	0.000	4.139	4.875

In [104…

```
# Create a dataframe that will contain the names of all the feature variables and their
vif = pd.DataFrame()
vif['Features'] = X_train[col].columns
vif['VIF'] = [variance_inflation_factor(X_train[col].values, i) for i in range(X_train[c
vif['VIF'] = round(vif['VIF'], 2)
vif = vif.sort_values(by = "VIF", ascending = False)
vif
```

Out[104]:

	Features	VIF
1	Lead Origin_Lead Add Form	1.82
12	Tags_Will revert after reading the email	1.56
4	Last Activity_SMS Sent	1.46
5	Last Notable Activity_Modified	1.40
2	Lead Source_Direct Traffic	1.38
3	Lead Source_Welingak Website	1.34
10	Tags_Other_Tags	1.25
0	Total Time Spent on Website	1.22
7	Tags_Closed by Horizzon	1.21
11	Tags_Ringing	1.16
8	Tags_Interested in other courses	1.12
9	Tags_Lost to EINS	1.06
6	Last Notable Activity_Olark Chat Conversation	1.01

In [105…

```
# Getting the Predicted values on the train set
y_train_pred = res.predict(X_train_sm)
y_train_pred[:10]
```

Out[105]:

```
9196    0.283149
4696    0.031440
3274    0.576636
2164    0.006433
1667    0.989105
7024    0.130813
8018    0.024219
778     0.205594
6942    0.002678
4440    0.096716
dtype: float64
```

In [106…

```
y_train_pred = y_train_pred.values.reshape(-1)
y_train_pred[:10]
```

Out[106]:

```
array([0.28314859, 0.0314396 , 0.57663553, 0.00643284, 0.98910464,
       0.13081306, 0.02421913, 0.20559401, 0.00267787, 0.09671623])
```

In [107…

```
y_train_pred_final = pd.DataFrame({'Converted':y_train.values, 'Converted_prob':y_train_
y_train_pred_final['Prospect ID'] = y_train.index
y_train_pred_final.head()
```

Out[107]:

	Converted	Converted_prob	Prospect ID
0	1	0.283149	9196
1	0	0.031440	4696
2	0	0.576636	3274
3	0	0.006433	2164
4	1	0.989105	1667

In [108…

```
y_train_pred_final['Predicted'] = y_train_pred_final.Converted_prob.map(lambda x: 1 if x
```

```
# Let's see the head
y_train_pred_final.head()
```

Out[108]:

	Converted	Converted_prob	Prospect ID	Predicted
0	1	0.283149	9196	0
1	0	0.031440	4696	0
2	0	0.576636	3274	1
3	0	0.006433	2164	0
4	1	0.989105	1667	1

In [109…
```
from sklearn import metrics

# Confusion matrix
confusion = metrics.confusion_matrix(y_train_pred_final.Converted, y_train_pred_final.Pr
print(confusion)
```

```
[[3693  189]
 [ 281 2104]]
```

In [110…
```
##overall accuracy
print(metrics.accuracy_score(y_train_pred_final.Converted, y_train_pred_final.Predicted)
```

```
0.9250039891495133
```

In [111…
```
TP = confusion[1,1] # true positive
TN = confusion[0,0] # true negatives
FP = confusion[0,1] # false positives
FN = confusion[1,0] # false negatives
```

In [112…
```
##the sensitivity of our logistic regression model
TP / float(TP+FN)
```

Out[112]:
```
0.8821802935010482
```

In [113…
```
#calculate specificity
TN / float(TN+FP)
```

Out[113]:
```
0.9513137557959814
```

In [114…
```
# Calculate False Positive Rate
print(FP/ float(TN+FP))
```

```
0.04868624420401855
```

In [115…
```
# positive predictive value
print (TP / float(TP+FP))
```

```
0.9175752289576974
```

In [116…
```
# Negative predictive value
print (TN / float(TN+ FN))
```

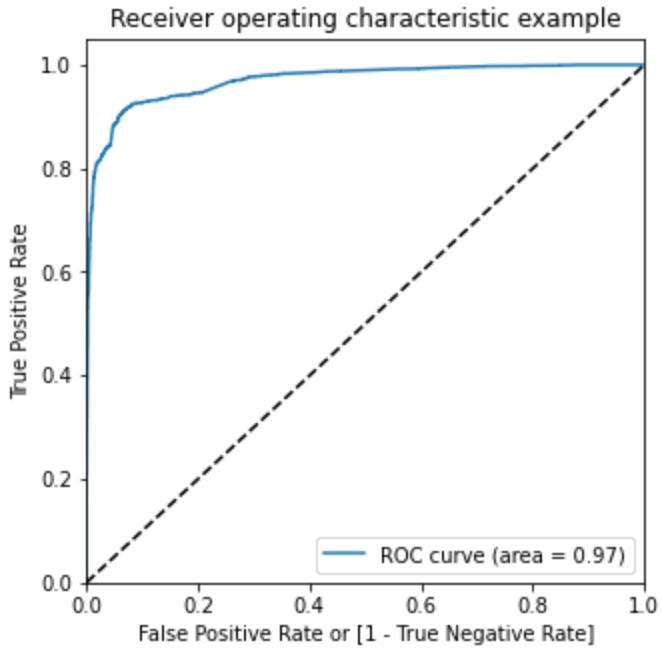
```
0.9292903875188727
```

In [117…
```
## Plotting ROC curve
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))
    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 [118…
```
fpr, tpr, thresholds = metrics.roc_curve( y_train_pred_final.Converted, y_train_pred_fin
```

In [119…
```
draw_roc(y_train_pred_final.Converted, y_train_pred_final.Converted_prob)
```



In [120…
```
# create 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.Converted_prob.map(lambda x: 1 if x > i el
y_train_pred_final.head()
```

Out[120]:

	Converted	Converted_prob	Prospect ID	Predicted	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
0	1	0.283149	9196	0	1	1	1	0	0	0	0	0	0	0
1	0	0.031440	4696	0	1	0	0	0	0	0	0	0	0	0
2	0	0.576636	3274	1	1	1	1	1	1	1	0	0	0	0
3	0	0.006433	2164	0	1	0	0	0	0	0	0	0	0	0
4	1	0.989105	1667	1	1	1	1	1	1	1	1	1	1	1

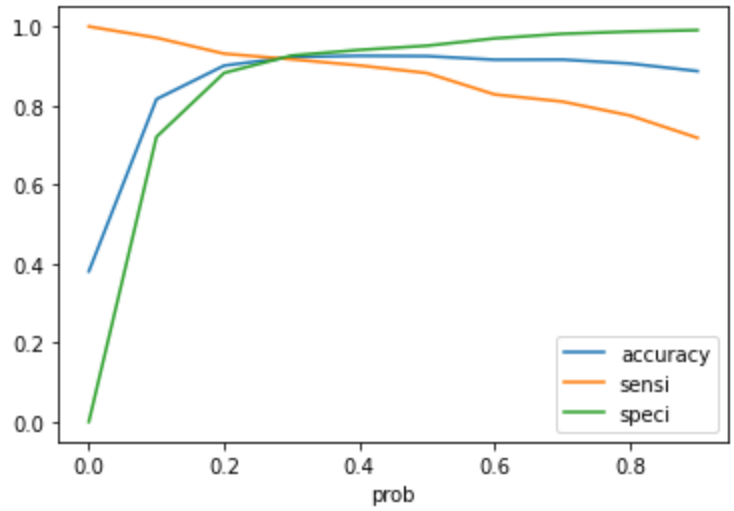
In [121…
```
# calculate accuracy sensitivity and specificity for various probability cutoffs.
cutoff_df = pd.DataFrame( columns = ['prob','accuracy','sensi','speci'])
from sklearn.metrics import confusion_matrix

# TP = confusion[1,1] # true positive
# TN = confusion[0,0] # true negatives
# FP = confusion[0,1] # false positives
# FN = confusion[1,0] # false negatives

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]
print(cutoff_df)
```

```
     prob  accuracy     sensi     speci
0.0   0.0  0.380565  1.000000  0.000000
0.1   0.1  0.816180  0.971488  0.720762
0.2   0.2  0.901069  0.931237  0.882535
0.3   0.3  0.922930  0.916981  0.926584
0.4   0.4  0.925802  0.901468  0.940752
0.5   0.5  0.925004  0.882180  0.951314
0.6   0.6  0.915909  0.828092  0.969861
0.7   0.7  0.916228  0.810063  0.981453
0.8   0.8  0.906335  0.774843  0.987120
0.9   0.9  0.887027  0.718239  0.990726
```

In [122…

```
#  plot accuracy sensitivity and specificity for various probabilities.
cutoff_df.plot.line(x='prob', y=['accuracy','sensi','speci'])
plt.show()
```



In [123…

```
y_train_pred_final['final_Predicted'] = y_train_pred_final.Converted_prob.map( lambda x:

y_train_pred_final.head()
```

Out[123]:

	Converted	Converted_prob	Prospect ID	Predicted	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	final_Predict
0	1	0.283149	9196	0	1	1	1	0	0	0	0	0	0	0	
1	0	0.031440	4696	0	1	0	0	0	0	0	0	0	0	0	
2	0	0.576636	3274	1	1	1	1	1	1	1	0	0	0	0	
3	0	0.006433	2164	0	1	0	0	0	0	0	0	0	0	0	
4	1	0.989105	1667	1	1	1	1	1	1	1	1	1	1	1	

In [124…

```
y_train_pred_final['Lead_Score'] = y_train_pred_final.Converted_prob.map( lambda x: roun

y_train_pred_final[['Converted','Converted_prob','Prospect ID','final_Predicted','Lead_S
```

Out[124]:

	Converted	Converted_prob	Prospect ID	final_Predicted	Lead_Score
0	1	0.283149	9196	0	28
1	0	0.031440	4696	0	3
2	0	0.576636	3274	1	58
3	0	0.006433	2164	0	1
4	1	0.989105	1667	1	99

In [125…
```
#  check the overall accuracy.
metrics.accuracy_score(y_train_pred_final.Converted, y_train_pred_final.final_Predicted)
```

Out[125]:
```
0.922929631402585
```

In [126…
```
confusion2 = metrics.confusion_matrix(y_train_pred_final.Converted, y_train_pred_final.f
confusion2
```

Out[126]:
```
array([[3597,  285],
       [ 198, 2187]], dtype=int64)
```

In [127…
```
TP = confusion2[1,1] # true positive
TN = confusion2[0,0] # true negatives
FP = confusion2[0,1] # false positives
FN = confusion2[1,0] # false negatives
```

In [128…
```
#   the sensitivity of our logistic regression model
TP / float(TP+FN)
```

Out[128]:
```
0.9169811320754717
```

In [129…
```
#  calculate specificity
TN / float(TN+FP)
```

Out[129]:
```
0.9265842349304482
```

In [130…
```
#  False Positive Rate
print(FP/ float(TN+FP))
```

```
0.07341576506955177
```

In [131…
```
print (TP / float(TP+FP))
```

```
0.8847087378640777
```

In [132…
```
# Negative predictive value
print (TN / float(TN+ FN))
```

```
0.9478260869565217
```

In [133…
```
confusion = metrics.confusion_matrix(y_train_pred_final.Converted, y_train_pred_final.fi
confusion
```

Out[133]:
```
array([[3597,  285],
       [ 198, 2187]], dtype=int64)
```

In [134…
```
##### Precision
TP / TP + FP

confusion[1,1]/(confusion[0,1]+confusion[1,1])
```

Out[134]:
```
0.8847087378640777
```

In [135…
```
##### Recall
TP / TP + FN

confusion[1,1]/(confusion[1,0]+confusion[1,1])
```

Out[135]:
```
0.9169811320754717
```

In [136…
```
from sklearn.metrics import precision_score, recall_score
```

In [137…
```
precision_score(y_train_pred_final.Converted , y_train_pred_final.final_Predicted)
```

Out[137]:
```
0.8847087378640777
```

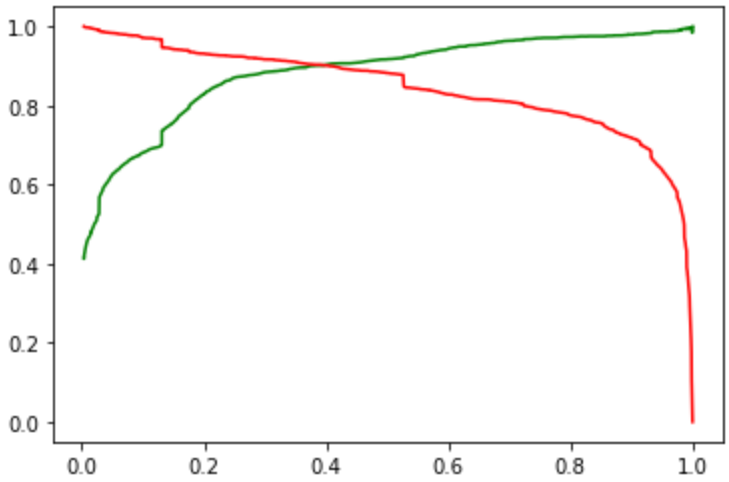
In [138…
```
recall_score(y_train_pred_final.Converted, y_train_pred_final.final_Predicted)
```

Out[138]:
```
0.9169811320754717
```

In [139…
```
from sklearn.metrics import precision_recall_curve
```

In [140…
```
y_train_pred_final.Converted, y_train_pred_final.final_Predicted
p, r, thresholds = precision_recall_curve(y_train_pred_final.Converted, y_train_pred_fin
```

In [141…
```
plt.plot(thresholds, p[:-1], "g-")
plt.plot(thresholds, r[:-1], "r-")
plt.show()
```



In [142…
```
#scaling test set
num_cols=X_test.select_dtypes(include=['float64', 'int64']).columns

X_test[num_cols] = scaler.fit_transform(X_test[num_cols])

X_test.head()
```

Out[142]:

	TotalVisits	Total Time Spent on Website	Page Views Per Visit	Lead Origin_Landing Page Submission	Lead Origin_Lead Add Form	Lead Origin_Lead Import	What is your current occupation_Housewife	occup
7681	0.575687	-0.311318	0.092860	1	0	0	0	
984	-0.090676	-0.550262	0.356568	1	0	0	0	
8135	-0.423857	0.812462	-0.170849	1	0	0	0	
6915	0.242505	-0.628665	-0.170849	1	0	0	0	
2712	-0.090676	-0.421456	0.356568	0	0	0	0	

5 rows × 76 columns

In [143…
```
X_test = X_test[col]
X_test.head()
```

Out[143]:

	Total Time Spent on Website	Lead Origin_Lead Add Form	Lead Source_Direct Traffic	Lead Source_Welingak Website	Last Activity_SMS Sent	Last Notable Activity_Modified	Last Notable Activity_Olark Chat Conversation
7681	-0.311318	0	1	0	1	0	0
984	-0.550262	0	0	0	1	1	0
8135	0.812462	0	1	0	1	0	0
6915	-0.628665	0	0	0	0	0	0
2712	-0.421456	0	0	0	0	0	0

In [144…
```
X_test_sm = sm.add_constant(X_test)
```

In [145…
```
###prediction on test set
y_test_pred = res.predict(X_test_sm)
```

In [146…
```
y_test_pred[:10]
```

Out[146]:
```
7681    0.024819
984     0.025692
8135    0.686054
6915    0.005880
2712    0.953208
244     0.002398
4698    0.014697
8287    0.027549
6791    0.981608
8970    0.005703
dtype: float64
```

In [147…
```
# Converting y_pred to a dataframe which is an array
y_pred_1 = pd.DataFrame(y_test_pred)
```

In [148…
```
y_pred_1.head()
```

Out[148]:

	0
7681	0.024819
984	0.025692
8135	0.686054
6915	0.005880
2712	0.953208

In [149…
```
# Converting y_test to dataframe
y_test_df = pd.DataFrame(y_test)
```

In [150…
```
# Putting CustID to index
y_test_df['Prospect ID'] = y_test_df.index
```

In [151…
```
# Removing index for both dataframes to append them side by side
y_pred_1.reset_index(drop=True, inplace=True)
y_test_df.reset_index(drop=True, inplace=True)
```

In [152…
```
# Appending y_test_df and y_pred_1
y_pred_final = pd.concat([y_test_df, y_pred_1],axis=1)
```

In [153…
```
y_pred_final.head()
```

Out[153]:

	Converted	Prospect ID	0
0	0	7681	0.024819
1	0	984	0.025692
2	0	8135	0.686054
3	0	6915	0.005880
4	1	2712	0.953208

In [154…
```
# Renaming the column
y_pred_final= y_pred_final.rename(columns={ 0 : 'Converted_prob'})
```

In [155…
```
y_pred_final.head()
```

Out[155]:

	Converted	Prospect ID	Converted_prob
0	0	7681	0.024819
1	0	984	0.025692
2	0	8135	0.686054
3	0	6915	0.005880
4	1	2712	0.953208

In [156…
```
# Rearranging the columns
y_pred_final = y_pred_final[['Prospect ID','Converted','Converted_prob']]
y_pred_final['Lead_Score'] = y_pred_final.Converted_prob.map( lambda x: round(x*100))
```

In [157…
```
y_pred_final.head()
```

Out[157]:

	Prospect ID	Converted	Converted_prob	Lead_Score
0	7681	0	0.024819	2
1	984	0	0.025692	3
2	8135	0	0.686054	69
3	6915	0	0.005880	1
4	2712	1	0.953208	95

In [158…
```
y_pred_final['final_Predicted'] = y_pred_final.Converted_prob.map(lambda x: 1 if x > 0.3
```

In [159…
```
y_pred_final.head()
```

Out[159]:

	Prospect ID	Converted	Converted_prob	Lead_Score	final_Predicted
0	7681	0	0.024819	2	0
1	984	0	0.025692	3	0
2	8135	0	0.686054	69	1
3	6915	0	0.005880	1	0
4	2712	1	0.953208	95	1

In [160…
```
# check the overall accuracy.
metrics.accuracy_score(y_pred_final.Converted, y_pred_final.final_Predicted)
```

Out[160]:
```
0.9277736411020104
```

In [161…
```
confusion2 = metrics.confusion_matrix(y_pred_final.Converted, y_pred_final.final_Predict
confusion2
```

Out[161]:
```
array([[1563,  113],
       [  81,  929]], dtype=int64)
```

In [162…
```
TP = confusion2[1,1] # true positive
TN = confusion2[0,0] # true negatives
FP = confusion2[0,1] # false positives
FN = confusion2[1,0] # false negatives
```

In [163…
```
#  see the sensitivity of our logistic regression model
TP / float(TP+FN)
```

Out[163]:
```
0.9198019801980198
```

In [164…
```
#calculate specificity
TN / float(TN+FP)
```

Out[164]:
```
0.9325775656324582
```

In [165…
```
precision_score(y_pred_final.Converted , y_pred_final.final_Predicted)
```

Out[165]:
```
0.8915547024952015
```

In [166…
```
recall_score(y_pred_final.Converted, y_pred_final.final_Predicted)
```

Out[166]:
```
0.9198019801980198
```

In [ ]:

In [ ]:

In [ ]: