Logistic Regresssion Assignment

Problem statement:

• An education company named X Education sells online courses to industry professionals. On any given day, many professionals who are interested in the courses land on their website and browse for courses.

• The company markets its courses on several websites and search engines like Google. Once these people land on the website, they might browse the courses or fill up a form for the course or watch some videos. When these people fill up a form providing their email address or phone number, they are classified to be a lead. Moreover, the company also gets leads through past referrals. Once these leads are acquired, employees from the sales team start making calls, writing emails, etc. Through this process, some of the leads get converted while most do not. The typical lead conversion rate at X education is around 30%. •

• Now, although X Education gets a lot of leads, its lead conversion rate is very poor. For example, if, say, they acquire 100 leads in a day, only about 30 of them are converted. To make this process more efficient, the company wishes to identify the most potential leads, also known as 'Hot Leads'. If they successfully identify this set of leads, the lead conversion rate should go up as the sales team will now be focusing more on communicating with the potential leads rather than making calls to everyone. A typical lead conversion process can be represented using the following funnel: •

• As you can see, there are a lot of leads generated in the initial stage (top) but only a few of them come out as paying customers from the bottom. In the middle stage, you need to nurture the potential leads well (i.e. educating the leads about the product, constantly communicating etc.) in order to get a higher lead conversion.

• X Education has appointed you to help them select the most promising leads, i.e. the leads that are most likely to convert into paying customers. The company requires you to build a model wherein you need to assign a lead score to each of the leads such that the customers with higher lead score have a higher conversion chance and the customers with lower lead score have a lower conversion chance. The CEO, in particular, has given a ballpark of the target lead conversion rate to be around 80%. Data

• You have been provided with a leads dataset from the past with around 9000 data points. This dataset consists of various attributes such as Lead Source, Total Time Spent on Website, Total Visits, Last Activity, etc. which may or may not be useful in ultimately deciding whether a lead will be converted or not. The target variable, in this case, is the column 'Converted' which tells whether a past lead was converted or not wherein 1 means it was converted and 0 means it wasn't converted. You can learn more about the dataset from the data dictionary provided in the zip folder at the end of the page. Another thing that you also need to check out for are the levels present in the categorical variables. Many of the categorical variables have a level called 'Select' which needs to be handled because it is as good as a null value (think why?). Goals of the Case Study

• There are quite a few goals for this case study.

Build a logistic regression model to assign a lead score between 0 and 100 to each of the leads which can be used by the company to target potential leads. A higher score would mean that the lead is hot, i.e. is most likely to convert whereas a lower score would mean that the lead is cold and will mostly not get converted.

There are some more problems presented by the company which your model should be able to adjust to if the company's requirement changes in the future so you will need to handle these as well. These problems are provided in a separate doc file. Please fill it based on the logistic regression model you got in the first step. Also, make sure you include this in your final PPT where you'll make recommendations.

Results Expected A well-commented Jupyter note with at least the logistic regression model

1.Importing Data

In [1]:

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

In [2]:

```
# Importing Pandas and NumPy
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

```python
from matplotlib.pyplot import xticks
%matplotlib inline
```

In [3]:

```python
lead_data=pd.read_csv("E:\dsw\Leads.csv")
lead_data.head()
```

Out[3]:

	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	As A
0	7927b2df-8bba-4d29-b9a2-b6e0beafe620	660737	API	Olark Chat	No	No	0	0.0	0	0.0	...	No	Select	Select	
1	2a272436-5132-4136-86fa-dcc88c88f482	660728	API	Organic Search	No	No	0	5.0	674	2.5	...	No	Select	Select	
2	8cc8c611-a219-4f35-ad23-fdfd2656bd8a	660727	Landing Page Submission	Direct Traffic	No	No	1	2.0	1532	2.0	...	No	Potential Lead	Mumbai	
3	0cc2df48-7cf4-4e39-9de9-19797f9b38cc	660719	Landing Page Submission	Direct Traffic	No	No	0	1.0	305	1.0	...	No	Select	Mumbai	
4	3256f628-e534-4826-9d63-4a8b88782852	660681	Landing Page Submission	Google	No	No	1	2.0	1428	1.0	...	No	Select	Mumbai	

5 rows × 37 columns

In [4]:

```python
sum(lead_data.duplicated(subset = 'Prospect ID')) == 0
```

Out[4]:

```
True
```

In [5]:

```python
lead_data.shape
```

Out[5]:

```
(9240, 37)
```

In [6]:

```python
lead_data.describe()
```

Out[6]:

	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

	Lead Number	Converted	TotalVisits	Total Time Spent on Website	Page Views Per Visit	Asymmetrique Activity Score	Asymmetrique Profile Score
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 [7]:

```
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 [8]:

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

In [9]:

```
lead_data.isnull().sum()
```

Out[9]:

```
Prospect ID                                         0
Lead Number                                         0
Lead Origin                                         0
Lead Source                                        36
Do Not Email                                        0
Do Not Call                                         0
Converted                                           0
TotalVisits                                       137
Total Time Spent on Website                         0
```

```
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]:

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

Out[10]:

```
Prospect ID                                       0.00
Lead Number                                       0.00
Lead Origin                                       0.00
Lead Source                                       0.39
Do Not Email                                      0.00
Do Not Call                                       0.00
Converted                                         0.00
TotalVisits                                       1.48
Total Time Spent on Website                       0.00
Page Views Per Visit                              1.48
Last Activity                                     1.11
Country                                          26.63
Specialization                                   36.58
How did you hear about X Education               78.46
What is your current occupation                  29.11
What matters most to you in choosing a course    29.32
Search                                            0.00
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]:

```
# we will drop the columns having more than 70% NA values.
lead_data =lead_data.drop(lead_data.loc[:,list(round(100*(lead_data.isnull().sum()/len(lead_data.i
ndex)), 2)>70)].columns, 1)
```

Now we will take care of null values in each column one by one.

In [12]:

```
lead_data['Lead Quality'].describe()
```

Out[12]:

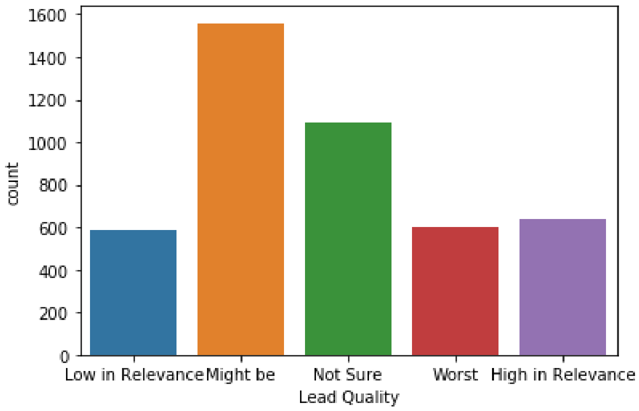
```
count          4473
unique            5
top        Might be
freq           1560
Name: Lead Quality, dtype: object
```

In [13]:

```
sns.countplot(lead_data['Lead Quality'])
```

Out[13]:

<matplotlib.axes._subplots.AxesSubplot at 0x2241f212220>



In [14]:

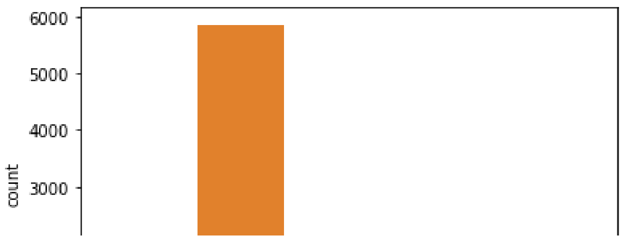
```
# As Lead quality is based on the intution of employee, so if left blank we can impute 'Not Sure'
in NaN safely.
lead_data['Lead Quality'] = lead_data['Lead Quality'].replace(np.nan, 'Not Sure')
```

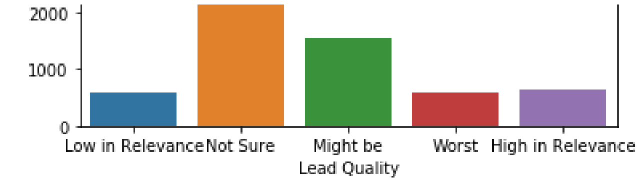
In [15]:

```
sns.countplot(lead_data['Lead Quality'])
```

Out[15]:

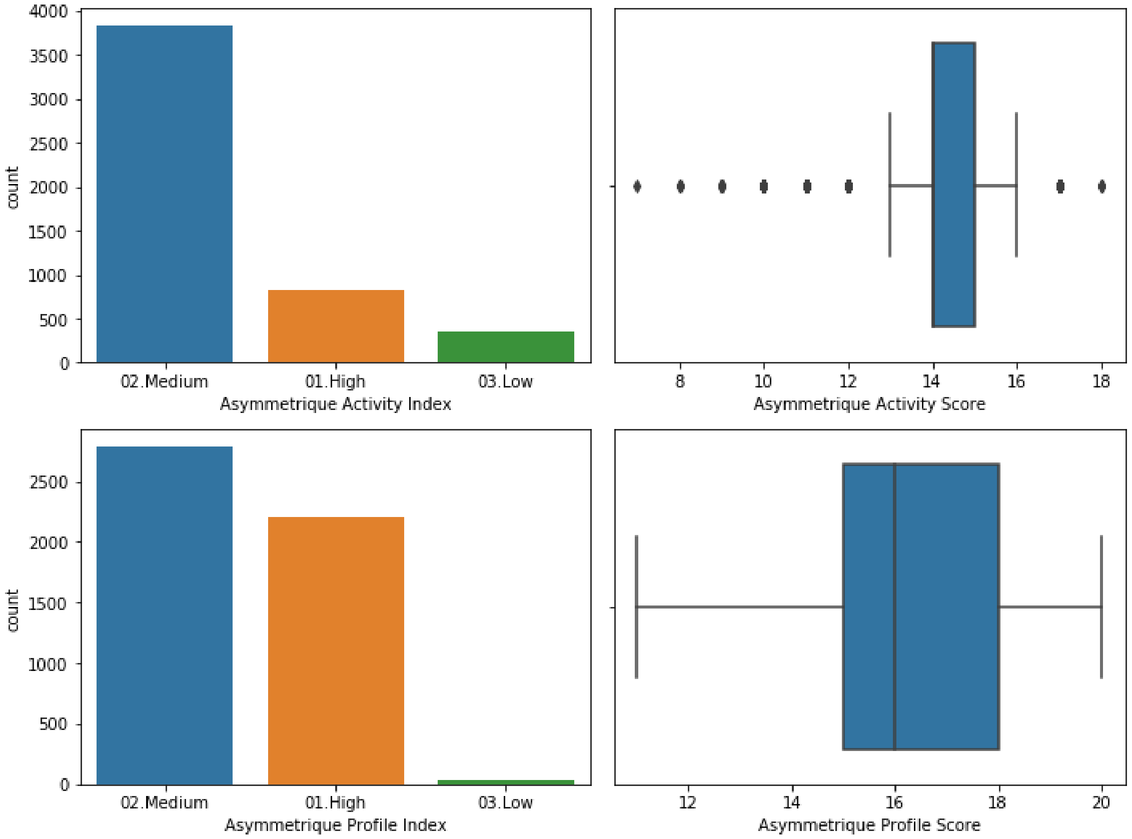
<matplotlib.axes._subplots.AxesSubplot at 0x2241f2b7820>





In [16]:

```
fig, axs = plt.subplots(2,2, figsize = (10,7.5))
plt1 = sns.countplot(lead_data['Asymmetrique Activity Index'], ax = axs[0,0])
plt2 = sns.boxplot(lead_data['Asymmetrique Activity Score'], ax = axs[0,1])
plt3 = sns.countplot(lead_data['Asymmetrique Profile Index'], ax = axs[1,0])
plt4 = sns.boxplot(lead_data['Asymmetrique Profile Score'], ax = axs[1,1])
plt.tight_layout()
```



In [17]:

```
lead_data = lead_data.drop(['Asymmetrique Activity Index','Asymmetrique Activity Score','Asymmetri
que Profile Index','Asymmetrique Profile Score'],1)
```

In [18]:

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

Out[18]:

Prospect ID	0.00
Lead Number	0.00
Lead Origin	0.00
Lead Source	0.39
Do Not Email	0.00
Do Not Call	0.00
Converted	0.00
TotalVisits	1.48
Total Time Spent on Website	0.00
Page Views Per Visit	1.48
Last Activity	1.11
Country	26.63
Specialization	36.58

```
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                                      0.00
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 [19]:

```
lead_data.City.describe()
```

Out[19]:

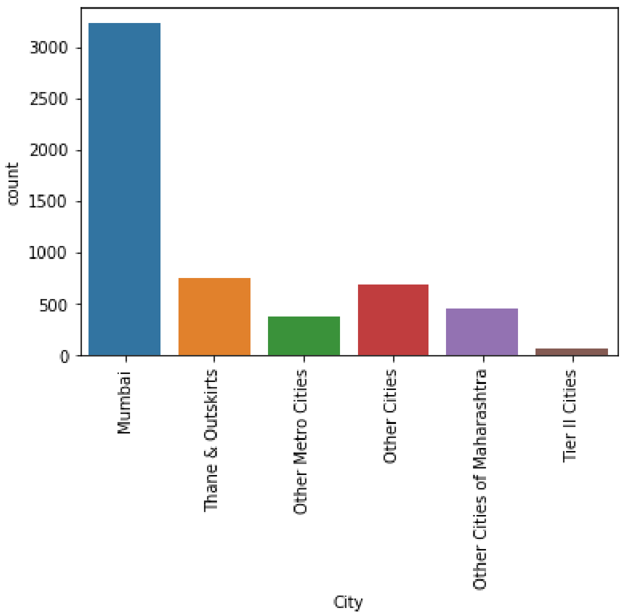
```
count       5571
unique         6
top       Mumbai
freq        3222
Name: City, dtype: object
```

In [20]:

```
sns.countplot(lead_data.City)
xticks(rotation = 90)
```

Out[20]:

```
(array([0, 1, 2, 3, 4, 5]), <a list of 6 Text xticklabel objects>)
```



In [21]:

```
lead_data['City'] = lead_data['City'].replace(np.nan, 'Mumbai')
```

In [22]:

```
lead_data.Specialization.describe()
```

Out[22]:

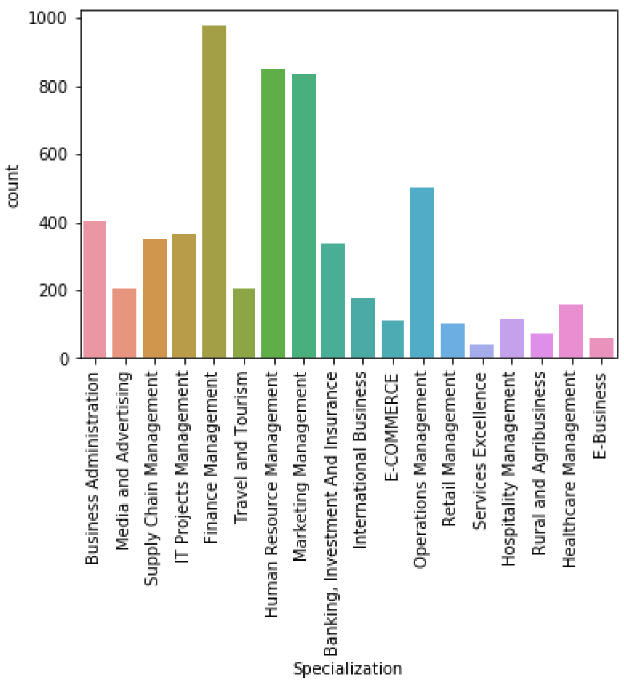
```
count                      5860
unique                       18
top          Finance Management
freq                        976
Name: Specialization, dtype: object
```

In [23]:

```
sns.countplot(lead_data.Specialization)
xticks(rotation = 90)
```

Out[23]:

```
(array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16,
        17]),
 <a list of 18 Text xticklabel objects>)
```



In [24]:

```
lead_data['Specialization'] = lead_data['Specialization'].replace(np.nan, 'Others')
```

In [25]:

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

Out[25]:

```
Prospect ID                                      0.00
Lead Number                                      0.00
Lead Origin                                      0.00
Lead Source                                      0.39
Do Not Email                                     0.00
Do Not Call                                      0.00
Converted                                        0.00
TotalVisits                                      1.48
Total Time Spent on Website                      0.00
Page Views Per Visit                             1.48
Last Activity                                    1.11
Country                                         26.63
Specialization                                   0.00
What is your current occupation                 29.11
What matters most to you in choosing a course   29.32
Search                                           0.00
Magazine                                         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                                      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 [26]:

```
lead_data.Tags.describe()
```

Out[26]:

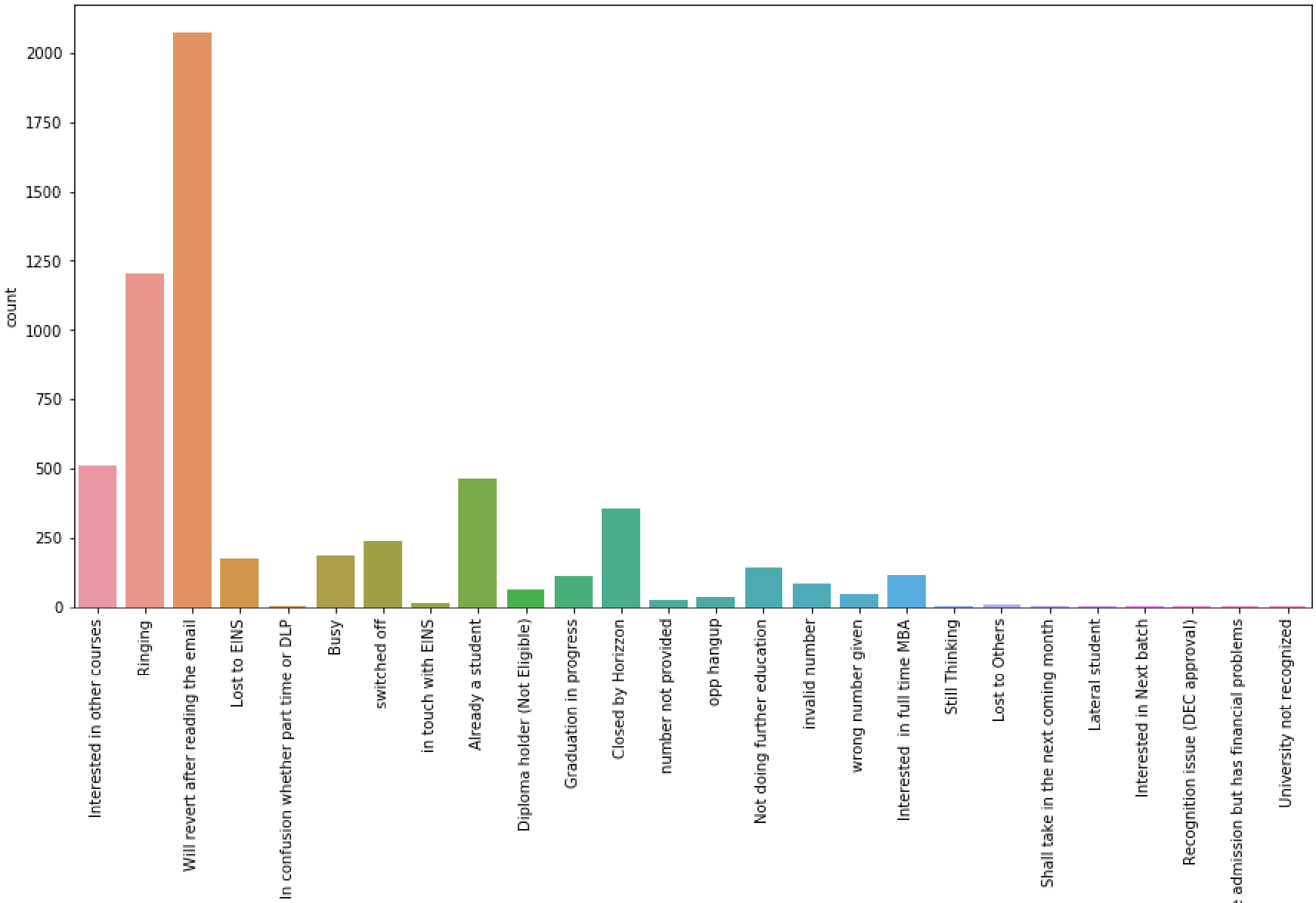
```
count                                5887
unique                                 26
top       Will revert after reading the email
freq                                 2072
Name: Tags, dtype: object
```

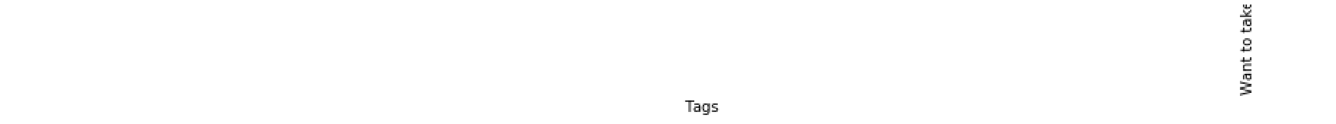
In [27]:

```
fig, axs = plt.subplots(figsize = (15,7.5))
sns.countplot(lead_data.Tags)
xticks(rotation = 90)
```

Out[27]:

```
(array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16,
        17, 18, 19, 20, 21, 22, 23, 24, 25]),
 <a list of 26 Text xticklabel objects>)
```





In [28]:

```
lead_data['Tags'] = lead_data['Tags'].replace(np.nan, 'Will revert after reading the email')
```

In [29]:

```
lead_data['What matters most to you in choosing a course'].describe()
```

Out[29]:

```
count                         6531
unique                           3
top        Better Career Prospects
freq                          6528
Name: What matters most to you in choosing a course, dtype: object
```

In [30]:

```
lead_data['What matters most to you in choosing a course'] = lead_data['What matters most to you i
n choosing a course'].replace(np.nan, 'Better Career Prospects')
```

In [31]:

```
lead_data['What is your current occupation'].describe()
```

Out[31]:

```
count           6550
unique             6
top       Unemployed
freq            5600
Name: What is your current occupation, dtype: object
```

In [32]:

```
lead_data['What is your current occupation'] = lead_data['What is your current
occupation'].replace(np.nan, 'Unemployed')
```

In [33]:

```
# Country is India for most values so let's impute the same in missing values.
lead_data['Country'] = lead_data['Country'].replace(np.nan, 'India')
```

In [34]:

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

Out[34]:

```
Prospect ID                                      0.00
Lead Number                                      0.00
Lead Origin                                      0.00
Lead Source                                      0.39
Do Not Email                                     0.00
Do Not Call                                      0.00
Converted                                        0.00
TotalVisits                                      1.48
Total Time Spent on Website                      0.00
Page Views Per Visit                             1.48
Last Activity                                    1.11
Country                                          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
Lead Quality                                     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 [35]:

```
# Rest missing values are under 2% so we can drop these rows.
lead_data.dropna(inplace = True)
```

In [36]:

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

Out[36]:

```
Prospect ID                                      0.0
Lead Number                                      0.0
Lead Origin                                      0.0
Lead Source                                      0.0
Do Not Email                                     0.0
Do Not Call                                      0.0
Converted                                        0.0
TotalVisits                                      0.0
Total Time Spent on Website                      0.0
Page Views Per Visit                             0.0
Last Activity                                    0.0
Country                                          0.0
Specialization                                   0.0
What is your current occupation                  0.0
What matters most to you in choosing a course    0.0
Search                                           0.0
Magazine                                         0.0
Newspaper Article                                0.0
X Education Forums                               0.0
Newspaper                                        0.0
Digital Advertisement                            0.0
Through Recommendations                          0.0
Receive More Updates About Our Courses           0.0
Tags                                             0.0
Lead Quality                                     0.0
Update me on Supply Chain Content                0.0
Get updates on DM Content                        0.0
City                                             0.0
I agree to pay the amount through cheque         0.0
A free copy of Mastering The Interview           0.0
Last Notable Activity                            0.0
dtype: float64
```

In [37]:

```
lead_data.to_csv('Leads_cleaned')
```

The Data is cleaned now and now we will see the analysis part

Exploratory Data Analysis

In [38]:

```
Converted = (sum(lead_data['Converted'])/len(lead_data['Converted'].index))*100
Converted
```

Out[38]:

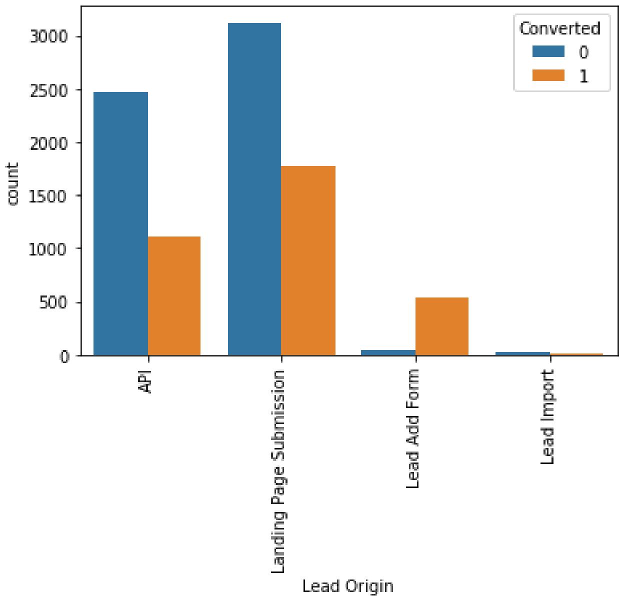
```
37.85541106458012
```

In [39]:

```
#lead origin
sns.countplot(x = "Lead Origin", hue = "Converted", data = lead_data)
xticks(rotation = 90)
```

Out[39]:

```
(array([0, 1, 2, 3]), <a list of 4 Text xticklabel objects>)
```



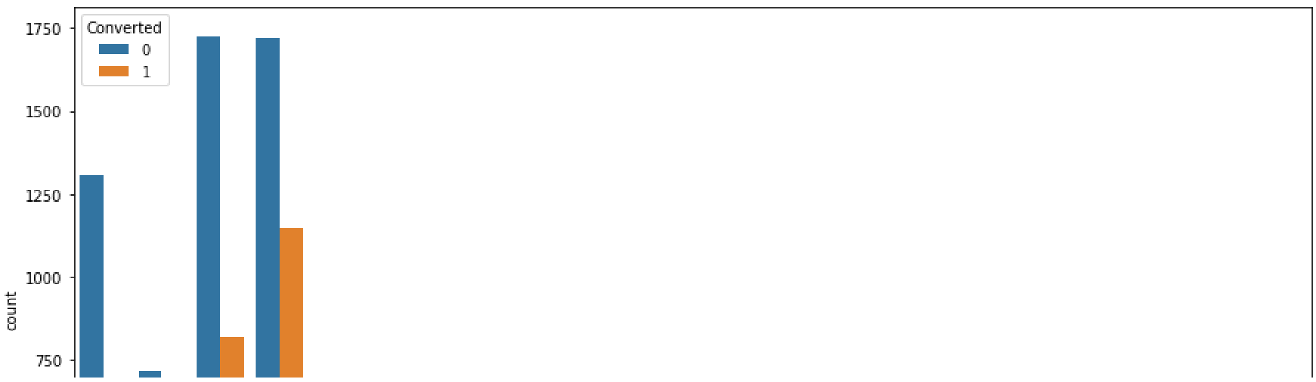
To improve overall lead conversion rate, we need to focus more on improving lead converion of API and Landing Page Submission origin and generate more leads from Lead Add Form.

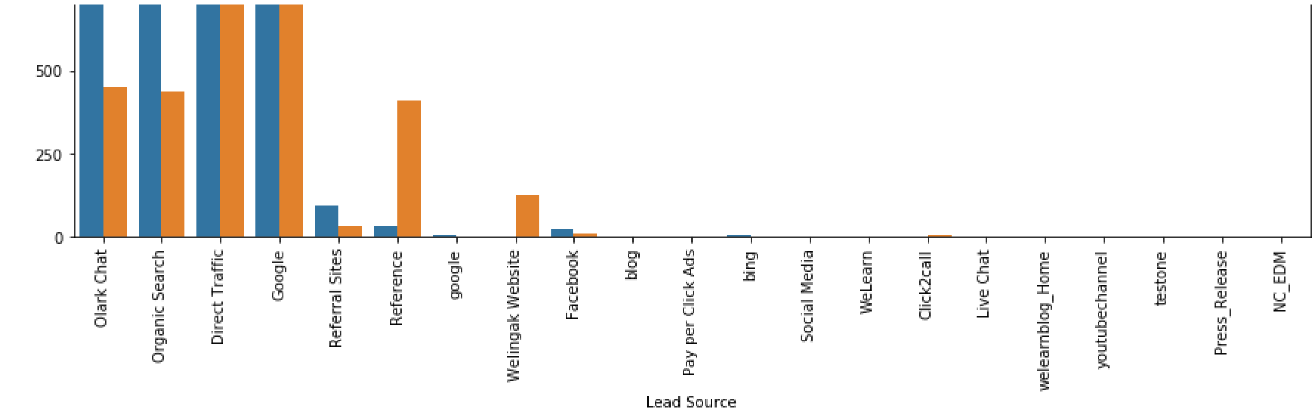
In [40]:

```
#lead source
fig, axs = plt.subplots(figsize = (15,7.5))
sns.countplot(x = "Lead Source", hue = "Converted", data = lead_data)
xticks(rotation = 90)
```

Out[40]:

```
(array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16,
        17, 18, 19, 20]),
 <a list of 21 Text xticklabel objects>)
```





In [41]:

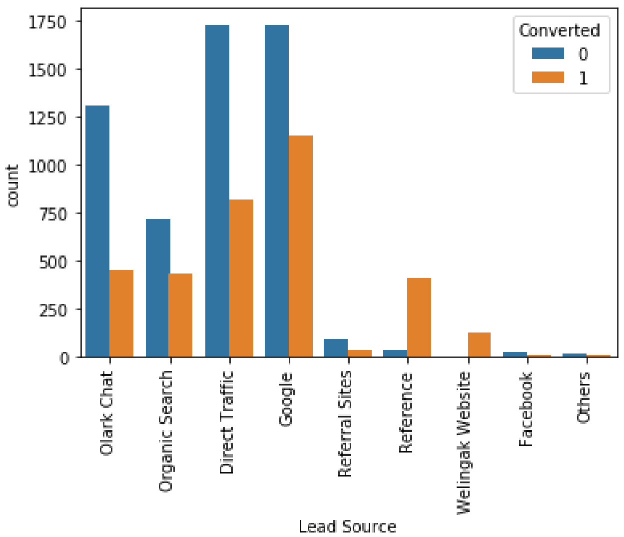
```
lead_data['Lead Source'] = lead_data['Lead Source'].replace(['google'], 'Google')
lead_data['Lead Source'] = lead_data['Lead Source'].replace(['Click2call', 'Live Chat', 'NC_EDM', '
Pay per Click Ads', 'Press_Release',
  'Social Media', 'WeLearn', 'bing', 'blog', 'testone', 'welearnblog_Home', 'youtubechannel'], 'Oth
ers')
```

In [42]:

```
sns.countplot(x = "Lead Source", hue = "Converted", data = lead_data)
xticks(rotation = 90)
```

Out[42]:

```
(array([0, 1, 2, 3, 4, 5, 6, 7, 8]), <a list of 9 Text xticklabel objects>)
```

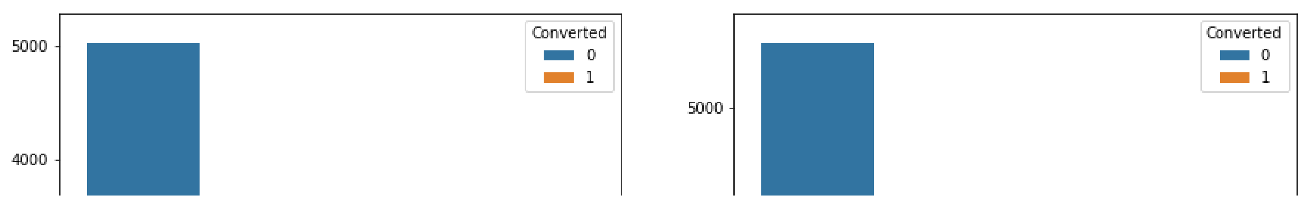


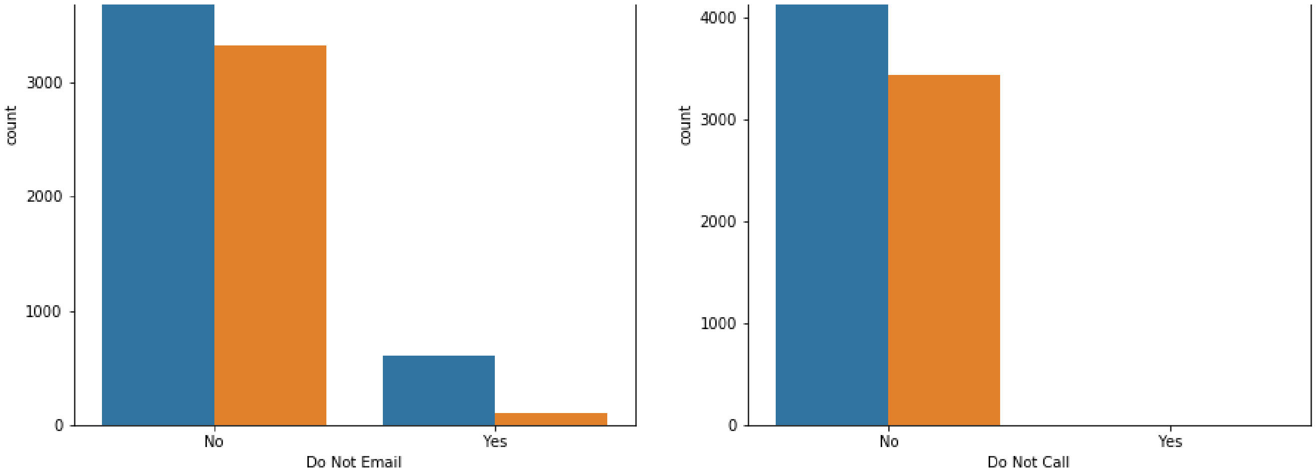
In [43]:

```
#Do Not Email & Do nOt Call
fig, axs = plt.subplots(1,2,figsize = (15,7.5))
sns.countplot(x = "Do Not Email", hue = "Converted", data = lead_data, ax = axs[0])
sns.countplot(x = "Do Not Call", hue = "Converted", data = lead_data, ax = axs[1])
```

Out[43]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x22421b48e20>
```





In [44]:

```
#Total visits
lead_data['TotalVisits'].describe(percentiles=[0.05,.25, .5, .75, .90, .95, .99])
```

Out[44]:

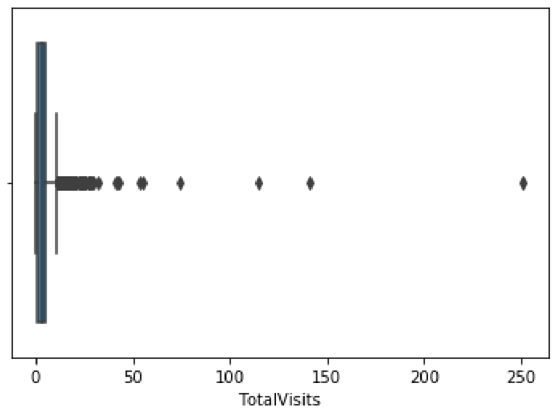
```
count    9074.000000
mean        3.456028
std         4.858802
min         0.000000
5%          0.000000
25%         1.000000
50%         3.000000
75%         5.000000
90%         7.000000
95%        10.000000
99%        17.000000
max       251.000000
Name: TotalVisits, dtype: float64
```

In [45]:

```
sns.boxplot(lead_data['TotalVisits'])
```

Out[45]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x22421bac2b0>
```



In [46]:

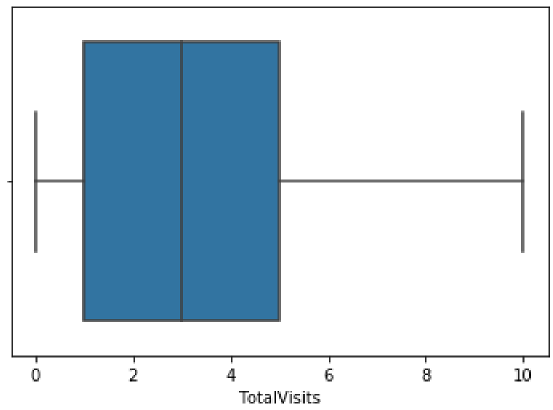
```
percentiles = lead_data['TotalVisits'].quantile([0.05,0.95]).values
lead_data['TotalVisits'][lead_data['TotalVisits'] <= percentiles[0]] = percentiles[0]
lead_data['TotalVisits'][lead_data['TotalVisits'] >= percentiles[1]] = percentiles[1]
```

In [47]:

```python
sns.boxplot(lead_data['TotalVisits'])
```

Out[47]:

<matplotlib.axes._subplots.AxesSubplot at 0x22421c16910>

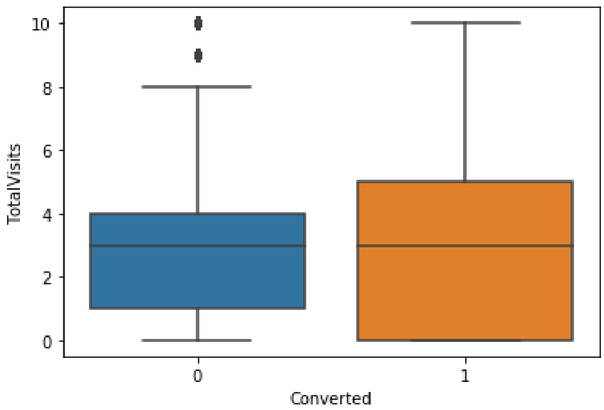


In [48]:

```python
sns.boxplot(y = 'TotalVisits', x = 'Converted', data = lead_data)
```

Out[48]:

<matplotlib.axes._subplots.AxesSubplot at 0x22421c73400>



In [49]:

```python
lead_data['Total Time Spent on Website'].describe()
```

Out[49]:

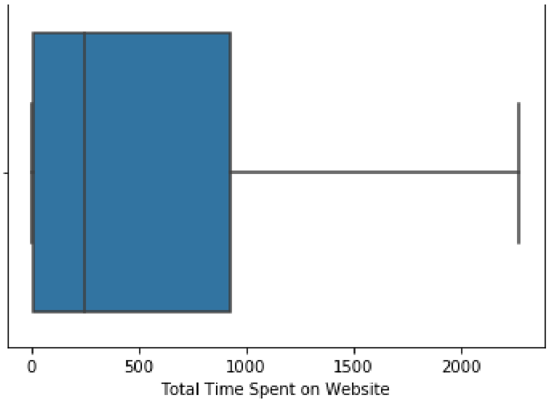
```
count    9074.000000
mean      482.887481
std       545.256560
min         0.000000
25%        11.000000
50%       246.000000
75%       922.750000
max      2272.000000
Name: Total Time Spent on Website, dtype: float64
```

In [50]:

```python
sns.boxplot(lead_data['Total Time Spent on Website'])
```

Out[50]:

<matplotlib.axes._subplots.AxesSubplot at 0x22421c77790>

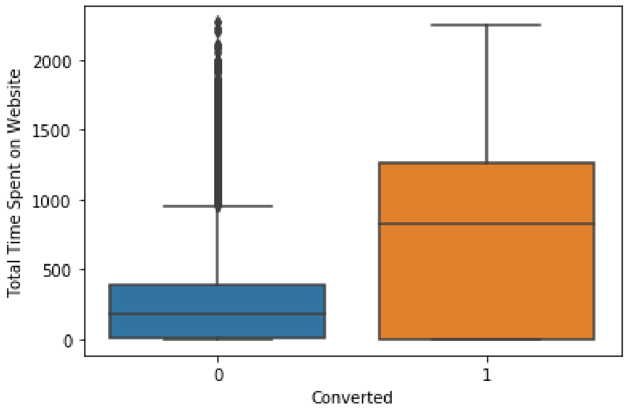


In [51]:

```
sns.boxplot(y = 'Total Time Spent on Website', x = 'Converted', data = lead_data)
```

Out[51]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x22421f202e0>
```



Website should be made more engaging to make leads spend more time.

In [52]:

```
#pages views per visit
lead_data['Page Views Per Visit'].describe()
```

Out[52]:

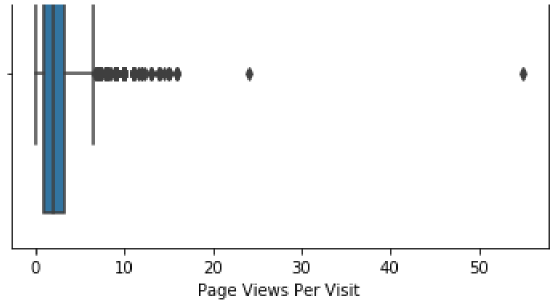
```
count    9074.000000
mean        2.370151
std         2.160871
min         0.000000
25%         1.000000
50%         2.000000
75%         3.200000
max        55.000000
Name: Page Views Per Visit, dtype: float64
```

In [53]:

```
sns.boxplot(lead_data['Page Views Per Visit'])
```

Out[53]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x22421f16a60>
```



In [54]:

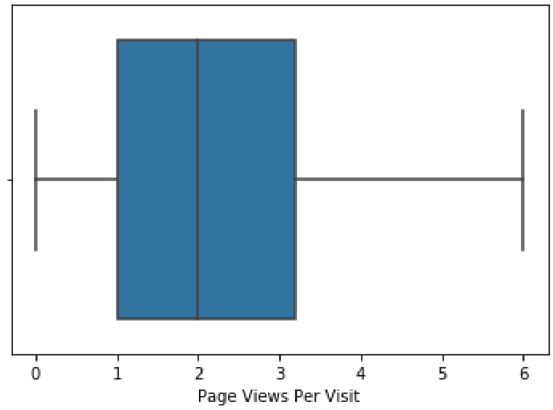
```
# As we can see there are a number of outliers in the data.
# We will cap the outliers to 95% value for analysis.
percentiles = lead_data['Page Views Per Visit'].quantile([0.05,0.95]).values
lead_data['Page Views Per Visit'][lead_data['Page Views Per Visit'] <= percentiles[0]] =
percentiles[0]
lead_data['Page Views Per Visit'][lead_data['Page Views Per Visit'] >= percentiles[1]] =
percentiles[1]
```

In [55]:

```
sns.boxplot(lead_data['Page Views Per Visit'])
```

Out[55]:

<matplotlib.axes._subplots.AxesSubplot at 0x22422258f10>

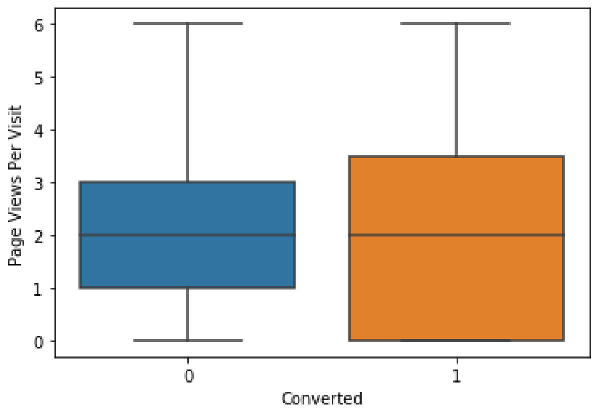


In [56]:

```
sns.boxplot(y = 'Page Views Per Visit', x = 'Converted', data = lead_data)
```

Out[56]:

<matplotlib.axes._subplots.AxesSubplot at 0x224222ac160>



In [57]:

```
#last activity
lead_data['Last Activity'].describe()
```

Out[57]:

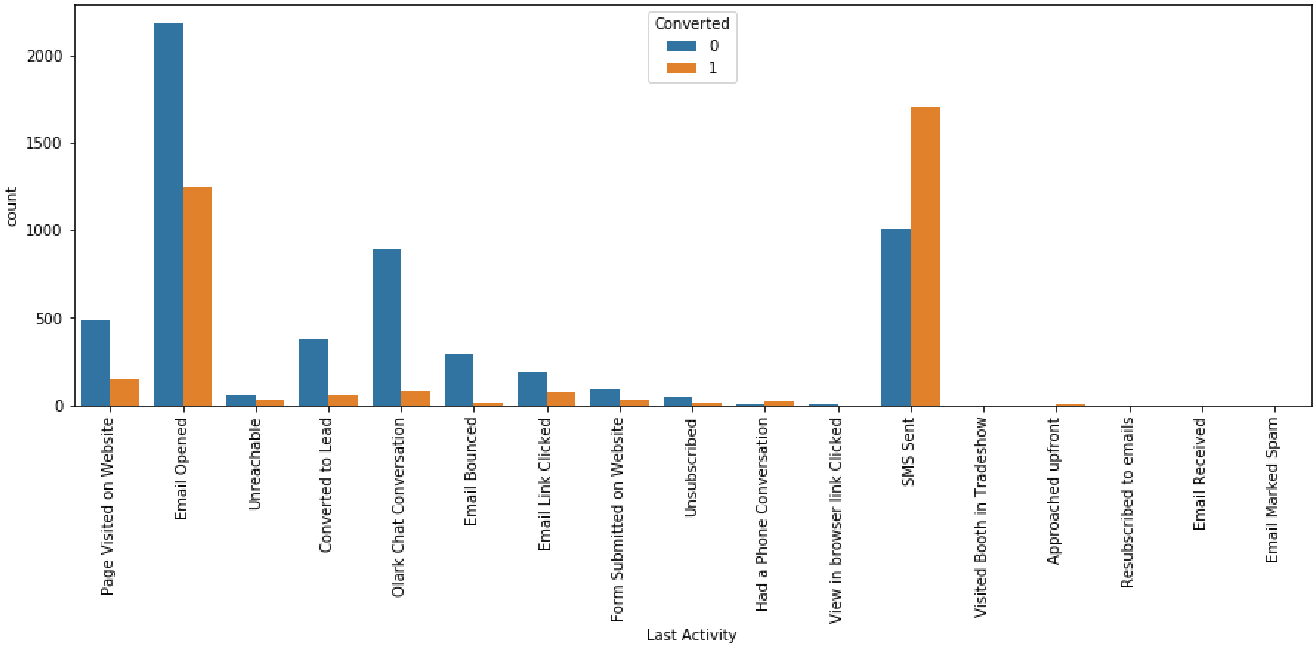
```
count             9074
unique              17
top       Email Opened
freq              3432
Name: Last Activity, dtype: object
```

In [58]:

```
fig, axs = plt.subplots(figsize = (15,5))
sns.countplot(x = "Last Activity", hue = "Converted", data = lead_data)
xticks(rotation = 90)
```

Out[58]:

```
(array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16]),
 <a list of 17 Text xticklabel objects>)
```



In [59]:

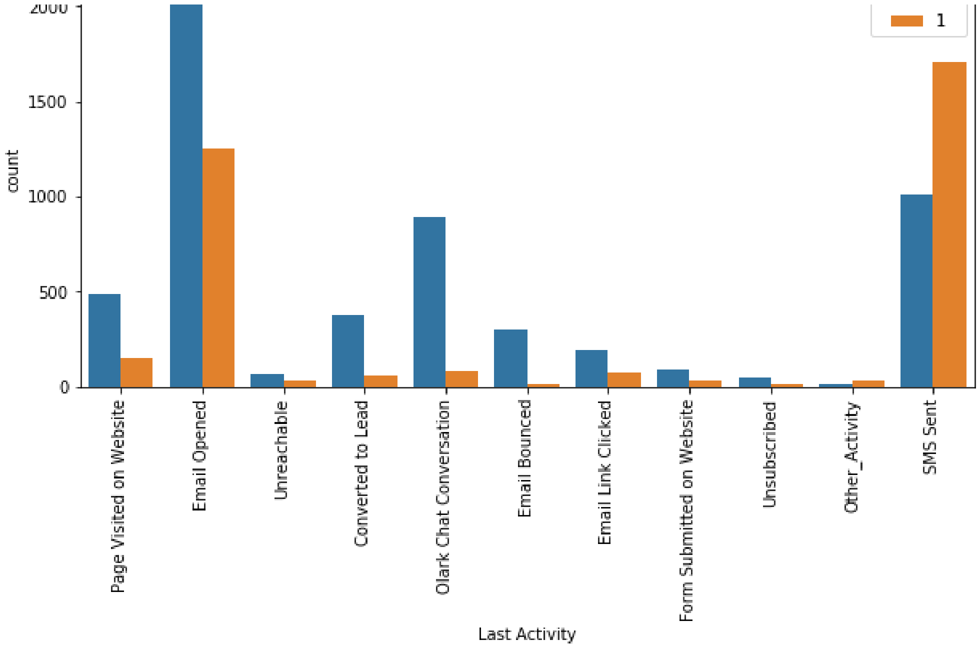
```
# Let's keep considerable last activities as such and club all others to "Other_Activity"
lead_data['Last Activity'] = lead_data['Last Activity'].replace(['Had a Phone Conversation', 'View in browser link Clicked',
                                                       'Visited Booth in Tradeshow', 'Approached upfront',
                                                       'Resubscribed to emails','Email Received', 'Email Marked Spam'], 'Other_Activity')
```

In [60]:

```
fig, axs = plt.subplots(figsize = (10,5))
sns.countplot(x = "Last Activity", hue = "Converted", data = lead_data)
xticks(rotation = 90)
```

Out[60]:

```
(array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10]),
 <a list of 11 Text xticklabel objects>)
```



In [61]:

```
#country
lead_data.Country.describe()
```

Out[61]:

```
count       9074
unique        38
top        India
freq        8787
Name: Country, dtype: object
```

In [62]:

```
lead_data.Specialization.describe()
```

Out[62]:

```
count        9074
unique         19
top        Others
freq         3282
Name: Specialization, dtype: object
```

In [63]:

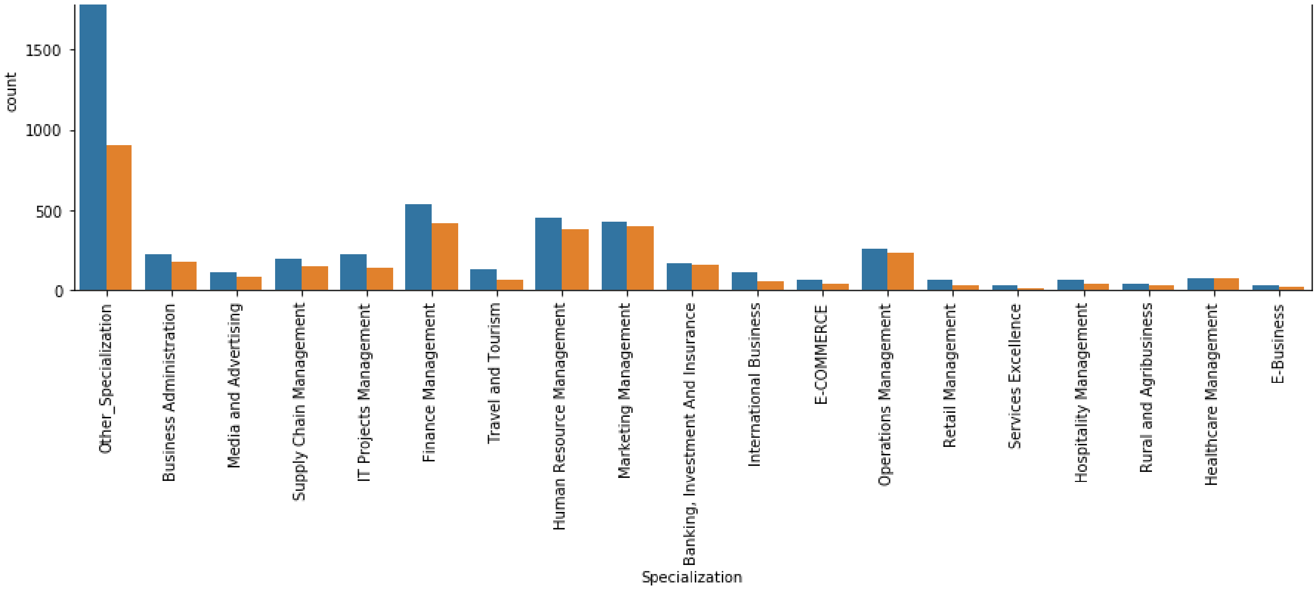
```
lead_data['Specialization'] = lead_data['Specialization'].replace(['Others'],
'Other_Specialization')
```

In [64]:

```
fig, axs = plt.subplots(figsize = (15,5))
sns.countplot(x = "Specialization", hue = "Converted", data = lead_data)
xticks(rotation = 90)
```

Out[64]:

```
(array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16,
        17, 18]),
 <a list of 19 Text xticklabel objects>)
```



In [65]:

```
#Occupation
lead_data['What is your current occupation'].describe()
```

Out[65]:

```
count            9074
unique              6
top        Unemployed
freq             8159
Name: What is your current occupation, dtype: object
```

In [66]:

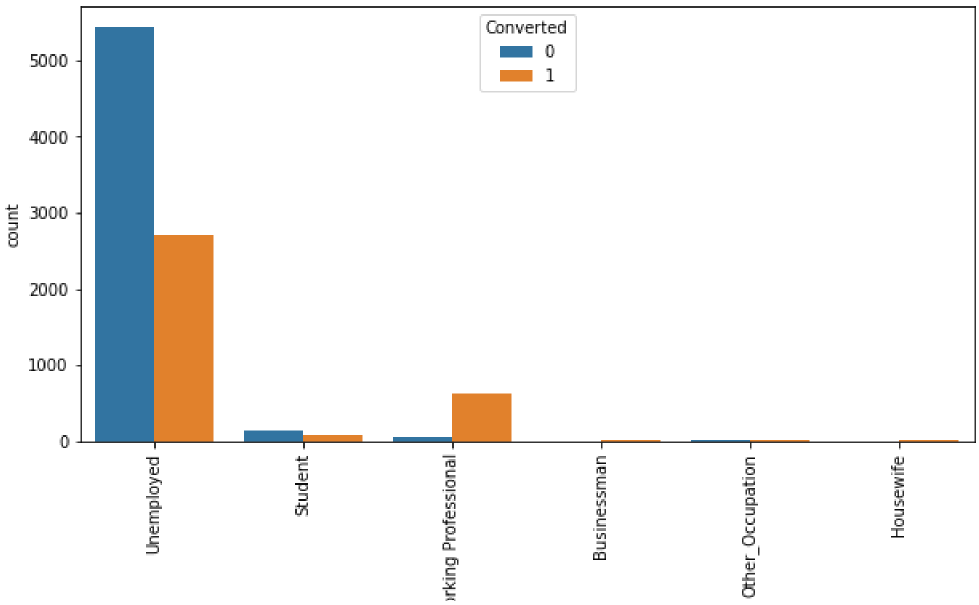
```
lead_data['What is your current occupation'] = lead_data['What is your current
occupation'].replace(['Other'], 'Other_Occupation')
```

In [67]:

```
fig, axs = plt.subplots(figsize = (10,5))
sns.countplot(x = "What is your current occupation", hue = "Converted", data = lead_data)
xticks(rotation = 90)
```

Out[67]:

```
(array([0, 1, 2, 3, 4, 5]), <a list of 6 Text xticklabel objects>)
```



In [68]:

```
#What matters most to you in choosing a course
lead_data['What matters most to you in choosing a course'].describe()
```

Out[68]:

```
count                          9074
unique                            3
top        Better Career Prospects
freq                           9072
Name: What matters most to you in choosing a course, dtype: object
```

In [69]:

```
#search
lead_data.Search.describe()
```

Out[69]:

```
count     9074
unique       2
top         No
freq      9060
Name: Search, dtype: object
```

In [70]:

```
#Magazine
lead_data.Magazine.describe()
```

Out[70]:

```
count     9074
unique       1
top         No
freq      9074
Name: Magazine, dtype: object
```

In [71]:

```
#Newspaper Article
lead_data['Newspaper Article'].describe()
```

Out[71]:

```
count     9074
unique       2
top         No
freq      9072
Name: Newspaper Article, dtype: object
```

In [72]:

```
#X Education Forms
lead_data['X Education Forums'].describe()
```

Out[72]:

```
count     9074
unique       2
top         No
freq      9073
Name: X Education Forums, dtype: object
```

In [73]:

```
#Newspaper
lead_data['Newspaper'].describe()
```

Out[73]:

```
count     9074
unique       2
top         No
freq      9073
Name: Newspaper, dtype: object
```

In [74]:

```
#Digital Advertisement
lead_data['Digital Advertisement'].describe()
```

Out[74]:

```
count     9074
unique       2
top         No
freq      9070
Name: Digital Advertisement, dtype: object
```

In [75]:

```
#Through recommendations
lead_data['Through Recommendations'].describe()
```

Out[75]:

```
count     9074
unique       2
top         No
freq      9067
Name: Through Recommendations, dtype: object
```

In [76]:

```
#Recieve More Updates About Our Courses
lead_data['Receive More Updates About Our Courses'].describe()
```

Out[76]:

```
count     9074
unique       1
top         No
freq      9074
Name: Receive More Updates About Our Courses, dtype: object
```

In [77]:

```
#tags
lead_data.Tags.describe()
```

Out[77]:

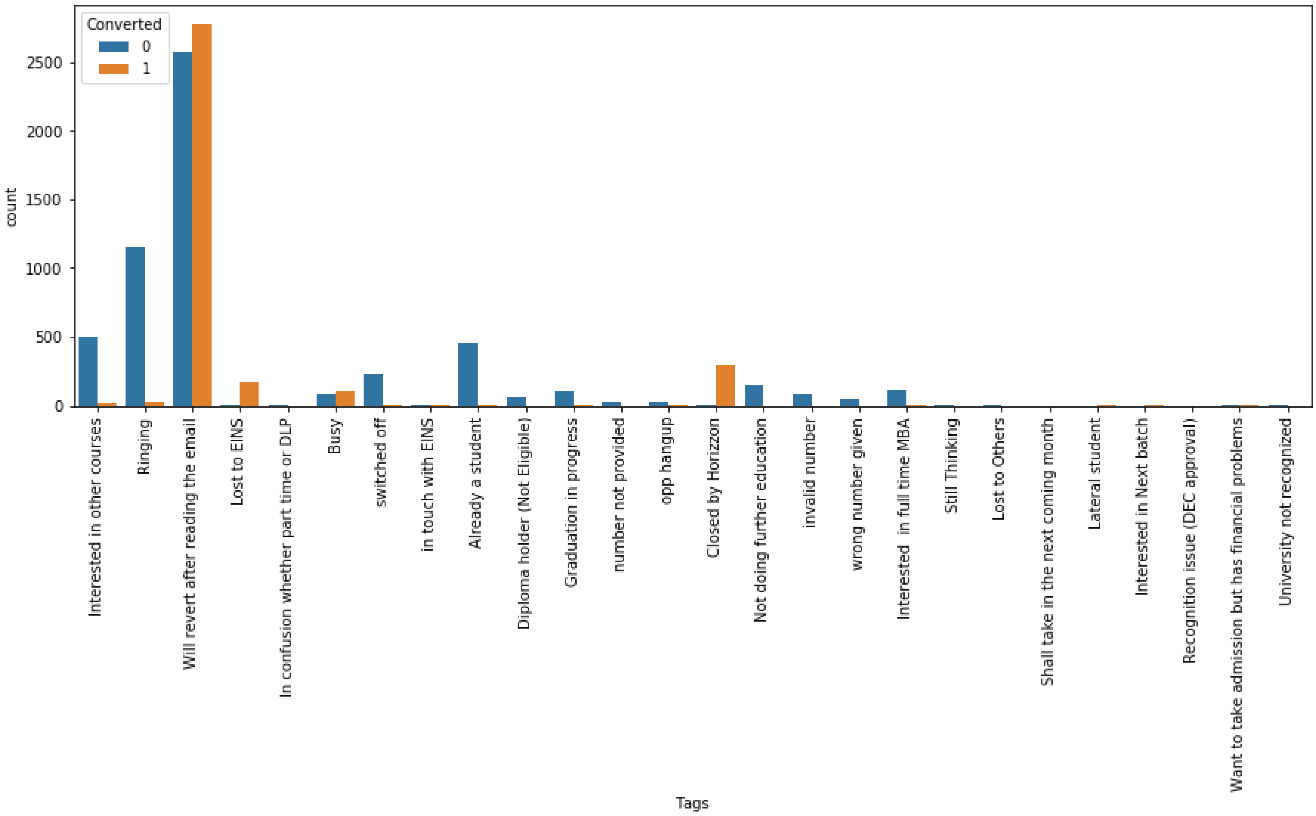
```
count                                9074
unique                                 26
top       Will revert after reading the email
freq                                 5343
Name: Tags, dtype: object
```

In [78]:

```
fig, axs = plt.subplots(figsize = (15,5))
sns.countplot(x = "Tags", hue = "Converted", data = lead_data)
xticks(rotation = 90)
```

Out[78]:

```
(array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16,
        17, 18, 19, 20, 21, 22, 23, 24, 25]),
 <a list of 26 Text xticklabel objects>)
```



In [79]:

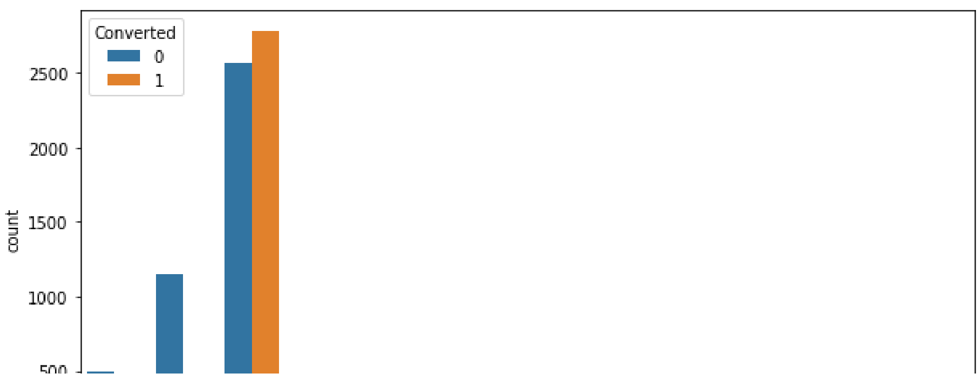
```
# Let's keep considerable last activities as such and club all others to "Other_Activity"
lead_data['Tags'] = lead_data['Tags'].replace(['In confusion whether part time or DLP', 'in touch
with EINS','Diploma holder (Not Eligible)',
                                                 'Approached upfront','Graduation in progress','number not prov
ded', 'opp hangup','Still Thinking',
                                                'Lost to Others','Shall take in the next coming month','Lateral
student','Interested in Next batch',
                                                'Recognition issue (DEC approval)','Want to take admission but
as financial problems',
                                                'University not recognized'], 'Other_Tags')
```

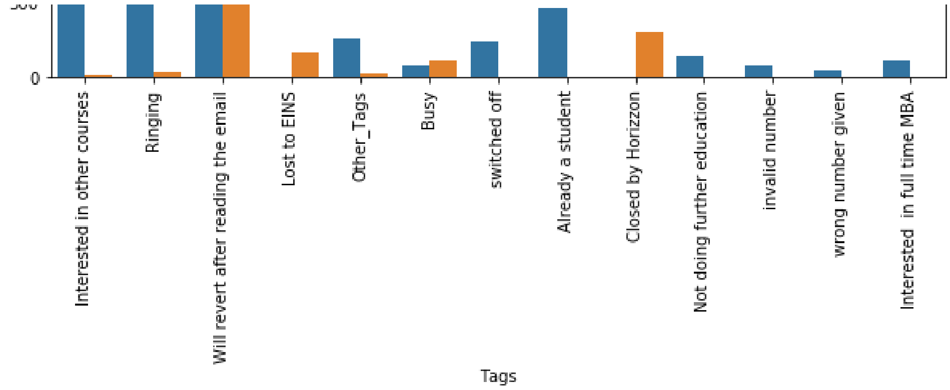
In [80]:

```
fig, axs = plt.subplots(figsize = (10,5))
sns.countplot(x = "Tags", hue = "Converted", data = lead_data)
xticks(rotation = 90)
```

Out[80]:

```
(array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12]),
 <a list of 13 Text xticklabel objects>)
```





In [81]:

```
#Lead Quality
lead_data['Lead Quality'].describe()
```

Out[81]:

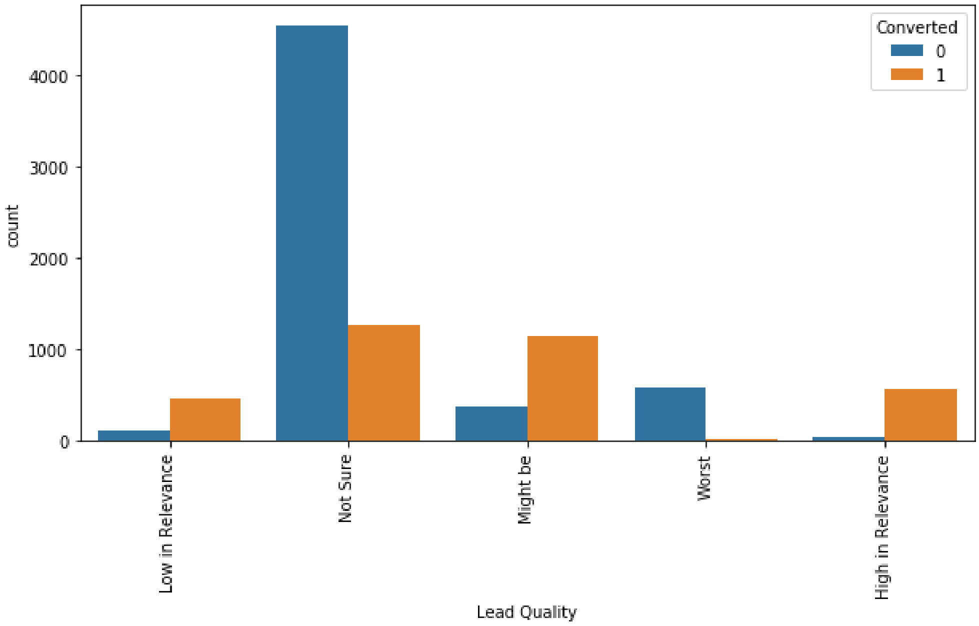
```
count          9074
unique            5
top        Not Sure
freq           5806
Name: Lead Quality, dtype: object
```

In [82]:

```
fig, axs = plt.subplots(figsize = (10,5))
sns.countplot(x = "Lead Quality", hue = "Converted", data = lead_data)
xticks(rotation = 90)
```

Out[82]:

```
(array([0, 1, 2, 3, 4]), <a list of 5 Text xticklabel objects>)
```



In [83]:

```
#Update me on Supply Chain Content
lead_data['Update me on Supply Chain Content'].describe()
```

Out[83]:

```
count      9074
unique        1
top          No
```

```
freq      9074
Name: Update me on Supply Chain Content, dtype: object
```

Most entries are 'No'. No Inference can be drawn with this parameter.

In [84]:

```
#Get updates on DM Content
lead_data['Get updates on DM Content'].describe()
```

Out[84]:

```
count     9074
unique       1
top         No
freq      9074
Name: Get updates on DM Content, dtype: object
```

Most entries are 'No'. No Inference can be drawn with this parameter.

In [85]:

```
#I agree to pay the amount through cheque
lead_data['I agree to pay the amount through cheque'].describe()
```

Out[85]:

```
count     9074
unique       1
top         No
freq      9074
Name: I agree to pay the amount through cheque, dtype: object
```

Most entries are 'No'. No Inference can be drawn with this parameter.

In [86]:

```
#A free copy of Mastering The Interview
lead_data['A free copy of Mastering The Interview'].describe()
```

Out[86]:

```
count     9074
unique       2
top         No
freq      6186
Name: A free copy of Mastering The Interview, dtype: object
```

Most entries are 'No'. No Inference can be drawn with this parameter.

In [87]:

```
lead_data.City.describe()
```

Out[87]:

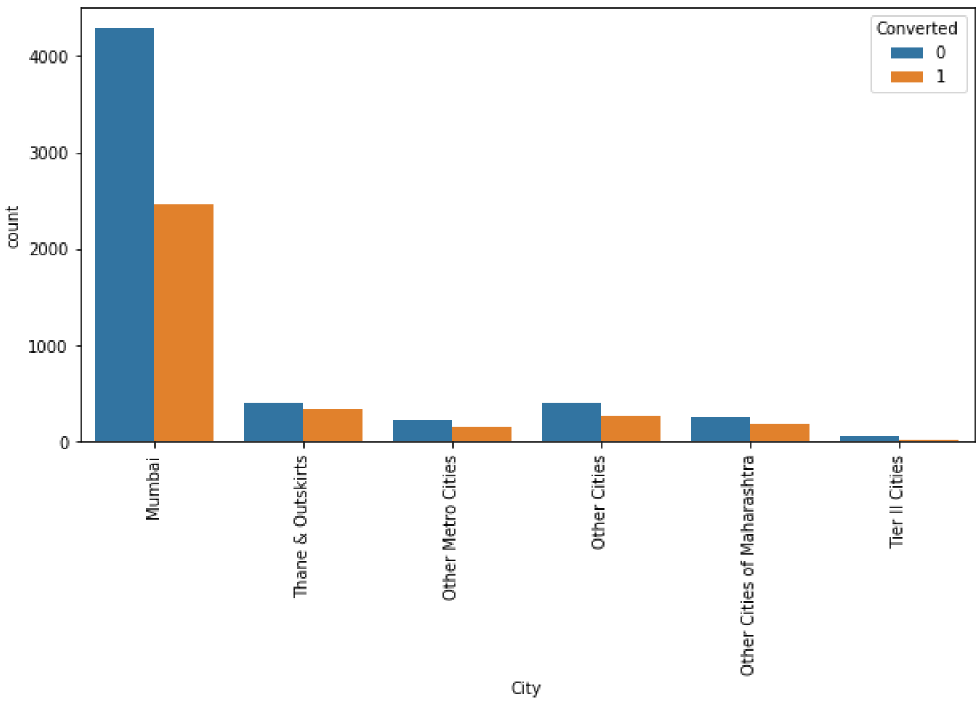
```
count       9074
unique         6
top       Mumbai
freq        6752
Name: City, dtype: object
```

In [88]:

```
fig, axs = plt.subplots(figsize = (10,5))
sns.countplot(x = "City", hue = "Converted", data = lead_data)
xticks(rotation = 90)
```

Out[88]:

```
(array([0, 1, 2, 3, 4, 5]), <a list of 6 Text xticklabel objects>)
```



In [89]:

```
#Last Notable Activity
lead_data['Last Notable Activity'].describe()
```

Out[89]:

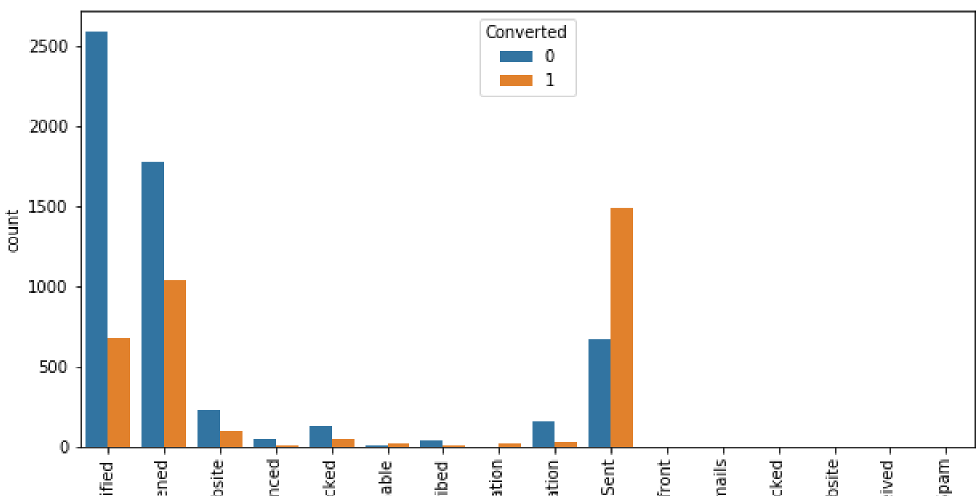
```
count          9074
unique           16
top        Modified
freq           3267
Name: Last Notable Activity, dtype: object
```

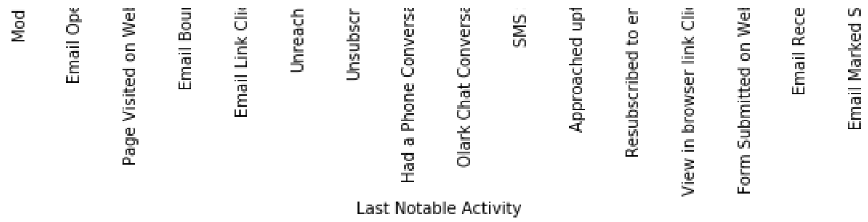
In [90]:

```
fig, axs = plt.subplots(figsize = (10,5))
sns.countplot(x = "Last Notable Activity", hue = "Converted", data = lead_data)
xticks(rotation = 90)
```

Out[90]:

```
(array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15]),
 <a list of 16 Text xticklabel objects>)
```





Results

Based on the univariate analysis we have seen that many columns are not adding any information to the model, heance we can drop them for frther analysis

In [91]:

```
lead_data = lead_data.drop(['Lead Number','What matters most to you in choosing a course','Search'
,'Magazine','Newspaper Article','X Education Forums','Newspaper',
           'Digital Advertisement','Through Recommendations','Receive More Updates About Our Course
s','Update me on Supply Chain Content',
           'Get updates on DM Content','I agree to pay the amount through cheque','A free copy of
Mastering The Interview','Country'],1)
```

In [92]:

```
lead_data.shape
```

Out[92]:

```
(9074, 16)
```

In [93]:

```
lead_data.head()
```

Out[93]:

	Prospect ID	Lead Origin	Lead Source	Do Not Email	Do Not Call	Converted	TotalVisits	Total Time Spent on Website	Page Views Per Visit	Last Activity	Specialization	What i you currer occupatio
0	7927b2df-8bba-4d29-b9a2-b6e0beafe620	API	Olark Chat	No	No	0	0.0	0	0.0	Page Visited on Website	Other_Specialization	Unemploye
1	2a272436-5132-4136-86fa-dcc88c88f482	API	Organic Search	No	No	0	5.0	674	2.5	Email Opened	Other_Specialization	Unemploye
2	8cc8c611-a219-4f35-ad23-fdfd2656bd8a	Landing Page Submission	Direct Traffic	No	No	1	2.0	1532	2.0	Email Opened	Business Administration	Studer
3	0cc2df48-7cf4-4e39-9de9-19797f9b38cc	Landing Page Submission	Direct Traffic	No	No	0	1.0	305	1.0	Unreachable	Media and Advertising	Unemploye
4	3256f628-e534-4826-9d63-4a8b88782852	Landing Page Submission	Google	No	No	1	2.0	1428	1.0	Converted to Lead	Other_Specialization	Unemploye

Data Preparation

converting soe binary variables (Yes/No) to 1/0

In [94]:

```
# List of variables to map
```

```
varlist =  ['Do Not Email', 'Do Not Call']

# Defining the map function
def binary_map(x):
    return x.map({'Yes': 1, "No": 0})

# Applying the function to the housing list
lead_data[varlist] = lead_data[varlist].apply(binary_map)
```

For categorical variables with multiple levels, create dummy features (one-hot encoded)

In [95]:

```
# Creating a dummy variable for some of the categorical variables and dropping the first one.
dummy1 = pd.get_dummies(lead_data[['Lead Origin', 'Lead Source', 'Last Activity', 'Specialization',
'What is your current occupation',
                              'Tags','Lead Quality','City','Last Notable Activity']], drop_first=Tr
e)
dummy1.head()
```

Out[95]:

	Lead Origin_Landing Page Submission	Lead Origin_Lead Add Form	Lead Origin_Lead Import	Lead Source_Facebook	Lead Source_Google	Lead Source_Olark Chat	Lead Source_Organic Search	Lead Source_Others	Sourc
0	0	0	0	0	0	1	0	0	
1	0	0	0	0	0	0	1	0	
2	1	0	0	0	0	0	0	0	
3	1	0	0	0	0	0	0	0	
4	1	0	0	0	1	0	0	0	

5 rows × 80 columns

In [96]:

```
# Adding the results to the master dataframe
lead_data = pd.concat([lead_data, dummy1], axis=1)
lead_data.head()
```

Out[96]:

	Prospect ID	Lead Origin	Lead Source	Do Not Email	Do Not Call	Converted	TotalVisits	Total Time Spent on Website	Page Views Per Visit	Last Activity	...	Last Notable Activity_Form Submitted on Website	Last Notabl Activity_Ha a Phon Conversatio
0	7927b2df-8bba-4d29-b9a2-b6e0beafe620	API	Olark Chat	0	0	0	0.0	0	0.0	Page Visited on Website	...	0	
1	2a272436-5132-4136-86fa-dcc88c88f482	API	Organic Search	0	0	0	5.0	674	2.5	Email Opened	...	0	
2	8cc8c611-a219-4f35-ad23-fdfd2656bd8a	Landing Page Submission	Direct Traffic	0	0	1	2.0	1532	2.0	Email Opened	...	0	
3	0cc2df48-7cf4-4e39-9de9-19797f9b38cc	Landing Page Submission	Direct Traffic	0	0	0	1.0	305	1.0	Unreachable	...	0	
4	3256f628-e534-4826-9d63-4a8b88782852	Landing Page Submission	Google	0	0	1	2.0	1428	1.0	Converted to Lead	...	0	

5 rows × 96 columns

In [97]:

```
lead_data = lead_data.drop(['Lead Origin', 'Lead Source', 'Last Activity', 'Specialization','What is your current occupation','Tags','Lead Quality','City','Last Notable Activity'], axis = 1)
```

In [98]:

```
lead_data.head()
```

Out[98]:

	Prospect ID	Do Not Email	Do Not Call	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	...	Last Notable Activity_Form Submitted on Website	L A C
0	7927b2df-8bba-4d29-b9a2-b6e0beafe620	0	0	0	0.0	0	0.0	0	0	0	...	0	
1	2a272436-5132-4136-86fa-dcc88c88f482	0	0	0	5.0	674	2.5	0	0	0	...	0	
2	8cc8c611-a219-4f35-ad23-fdfd2656bd8a	0	0	1	2.0	1532	2.0	1	0	0	...	0	
3	0cc2df48-7cf4-4e39-9de9-19797f9b38cc	0	0	0	1.0	305	1.0	1	0	0	...	0	
4	3256f628-e534-4826-9d63-4a8b88782852	0	0	1	2.0	1428	1.0	1	0	0	...	0	

5 rows × 87 columns

In [99]:

```
from sklearn.model_selection import train_test_split

# Putting feature variable to X
X = lead_data.drop(['Prospect ID','Converted'], axis=1)
```

In [100]:

```
X.head()
```

Out[100]:

	Do Not Email	Do Not Call	TotalVisits	Total Time Spent on Website	Page Views Per Visit	Lead Origin_Landing Page Submission	Lead Origin_Lead Add Form	Lead Origin_Lead Import	Lead Source_Facebook	Lead Source_Google	...	Last No Activity_ Submitte We
0	0	0	0.0	0	0.0	0	0	0	0	0	...	
1	0	0	5.0	674	2.5	0	0	0	0	0	...	
2	0	0	2.0	1532	2.0	1	0	0	0	0	...	
3	0	0	1.0	305	1.0	1	0	0	0	0	...	
4	0	0	2.0	1428	1.0	1	0	0	0	1	...	

5 rows × 85 columns

In [101]:

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

```
y.head()
```

Out[101]:

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

In [102]:

```
# 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,
random_state=100)
```

Feature Scaling

In [103]:

```
from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()

X_train[['TotalVisits','Total Time Spent on Website','Page Views Per Visit']] =
scaler.fit_transform(X_train[['TotalVisits','Total Time Spent on Website','Page Views Per Visit']]
)

X_train.head()
```

Out[103]:

	Do Not Email	Do Not Call	TotalVisits	Total Time Spent on Website	Page Views Per Visit	Lead Origin_Landing Page Submission	Lead Origin_Lead Add Form	Lead Origin_Lead Import	Lead Source_Facebook	Lead Source_Google	...
3009	0	0	-0.432779	-0.160255	-0.155018	1	0	0	0	0	...
1012	1	0	-0.432779	-0.540048	-0.155018	1	0	0	0	0	...
9226	0	0	-1.150329	-0.888650	-1.265540	0	0	0	0	0	...
4750	0	0	-0.432779	1.643304	-0.155018	1	0	0	0	0	...
7987	0	0	0.643547	2.017593	0.122613	1	0	0	0	0	...

5 rows × 85 columns

In [104]:

```
# Checking the Churn Rate
Converted = (sum(lead_data['Converted'])/len(lead_data['Converted'].index))*100
Converted
```

Out[104]:

```
37.85541106458012
```

We have almost 38% conversion

Model Building

In [105]:

```
import statsmodels.api as sm
```

In [106]:

```
# Logistic regression model
logm1 = sm.GLM(y_train,(sm.add_constant(X_train)), family = sm.families.Binomial())
logm1.fit().summary()
```

Out[106]:

Generalized Linear Model Regression Results

Dep. Variable:	Converted	**No. Observations:**	6351
Model:	GLM	**Df Residuals:**	6265
Model Family:	Binomial	**Df Model:**	85
Link Function:	logit	**Scale:**	1.0000
Method:	IRLS	**Log-Likelihood:**	-1250.0
Date:	Mon, 11 May 2020	**Deviance:**	2500.0
Time:	13:00:17	**Pearson chi2:**	3.87e+04
No. Iterations:	24		
Covariance Type:	nonrobust		

	coef	std err	z	P>|z|	[0.025	0.975]
const	23.1423	2.16e+05	0.000	1.000	-4.23e+05	4.23e+05
Do Not Email	-1.3882	0.327	-4.243	0.000	-2.030	-0.747
Do Not Call	23.7150	1.37e+05	0.000	1.000	-2.68e+05	2.68e+05
TotalVisits	0.1816	0.087	2.093	0.036	0.012	0.352
Total Time Spent on Website	1.1457	0.064	17.913	0.000	1.020	1.271
Page Views Per Visit	-0.3272	0.099	-3.309	0.001	-0.521	-0.133
Lead Origin_Landing Page Submission	-0.9762	0.221	-4.420	0.000	-1.409	-0.543
Lead Origin_Lead Add Form	-0.4165	1.287	-0.324	0.746	-2.940	2.107
Lead Origin_Lead Import	29.7289	2.16e+05	0.000	1.000	-4.23e+05	4.23e+05
Lead Source_Facebook	-28.6305	2.16e+05	-0.000	1.000	-4.23e+05	4.23e+05
Lead Source_Google	0.2017	0.155	1.302	0.193	-0.102	0.505
Lead Source_Olark Chat	0.8633	0.234	3.693	0.000	0.405	1.321
Lead Source_Organic Search	0.2278	0.210	1.083	0.279	-0.185	0.640
Lead Source_Others	0.7602	0.816	0.931	0.352	-0.839	2.360
Lead Source_Reference	1.7732	1.344	1.319	0.187	-0.861	4.407
Lead Source_Referral Sites	-0.0945	0.491	-0.193	0.847	-1.056	0.867
Lead Source_Welingak Website	5.4722	1.486	3.682	0.000	2.559	8.385
Last Activity_Email Bounced	-0.5488	0.870	-0.631	0.528	-2.254	1.157
Last Activity_Email Link Clicked	0.8429	0.644	1.309	0.190	-0.419	2.105
Last Activity_Email Opened	-0.0003	0.384	-0.001	0.999	-0.754	0.753
Last Activity_Form Submitted on Website	0.1337	0.593	0.225	0.822	-1.028	1.296
Last Activity_Olark Chat Conversation	-0.5464	0.392	-1.395	0.163	-1.314	0.221
Last Activity_Other_Activity	1.4578	1.200	1.214	0.225	-0.895	3.811
Last Activity_Page Visited on Website	0.5059	0.456	1.110	0.267	-0.387	1.399
Last Activity_SMS Sent	1.1289	0.360	3.134	0.002	0.423	1.835
Last Activity_Unreachable	0.6479	0.840	0.771	0.441	-0.999	2.294
Last Activity_Unsubscribed	0.8348	1.571	0.531	0.595	-2.245	3.914
Specialization_Business Administration	-0.2329	0.392	-0.594	0.553	-1.002	0.536
Specialization_E-Business	-0.3661	0.715	-0.512	0.609	-1.767	1.035
Specialization_E-COMMERCE	0.5774	0.587	0.983	0.326	-0.574	1.728
Specialization_Finance Management	-0.4463	0.346	-1.291	0.197	-1.124	0.231

Specialization_Healthcare Management	-0.5197	0.510	-1.018	0.308	-1.520	0.480
Specialization_Hospitality Management	-0.1701	0.544	-0.312	0.755	-1.237	0.897
Specialization_Human Resource Management	-0.2918	0.347	-0.840	0.401	-0.973	0.389
Specialization_IT Projects Management	-0.0187	0.411	-0.045	0.964	-0.824	0.787
Specialization_International Business	-0.8406	0.460	-1.828	0.068	-1.742	0.061
Specialization_Marketing Management	0.0389	0.349	0.112	0.911	-0.645	0.722
Specialization_Media and Advertising	-0.5447	0.488	-1.116	0.264	-1.501	0.412
Specialization_Operations Management	-0.1345	0.392	-0.343	0.732	-0.904	0.635
Specialization_Other_Specialization	-0.7987	0.359	-2.228	0.026	-1.501	-0.096
Specialization_Retail Management	-0.2404	0.562	-0.428	0.669	-1.342	0.861
Specialization_Rural and Agribusiness	0.0798	0.688	0.116	0.908	-1.269	1.428
Specialization_Services Excellence	-0.0560	0.971	-0.058	0.954	-1.960	1.848
Specialization_Supply Chain Management	-0.4389	0.426	-1.030	0.303	-1.274	0.397
Specialization_Travel and Tourism	-0.7866	0.512	-1.537	0.124	-1.790	0.217
What is your current occupation_Housewife	20.6162	7.16e+04	0.000	1.000	-1.4e+05	1.4e+05
What is your current occupation_Other_Occupation	-0.7446	2.036	-0.366	0.715	-4.736	3.246
What is your current occupation_Student	-1.3109	1.548	-0.847	0.397	-4.345	1.723
What is your current occupation_Unemployed	-2.1034	1.446	-1.455	0.146	-4.937	0.730
What is your current occupation_Working Professional	-0.7884	1.483	-0.532	0.595	-3.694	2.117
Tags_Busy	3.9167	0.849	4.611	0.000	2.252	5.582
Tags_Closed by Horizzon	8.8694	1.138	7.792	0.000	6.638	11.100
Tags_Interested in full time MBA	0.3509	1.227	0.286	0.775	-2.054	2.756
Tags_Interested in other courses	0.2322	0.888	0.261	0.794	-1.509	1.973
Tags_Lost to EINS	9.7272	1.087	8.946	0.000	7.596	11.858
Tags_Not doing further education	-0.0911	1.502	-0.061	0.952	-3.035	2.853
Tags_Other_Tags	1.0318	0.865	1.193	0.233	-0.663	2.726
Tags_Ringing	-1.1124	0.857	-1.298	0.194	-2.792	0.568
Tags_Will revert after reading the email	4.1719	0.812	5.138	0.000	2.581	5.763
Tags_invalid number	-22.5334	2.22e+04	-0.001	0.999	-4.35e+04	4.34e+04
Tags_switched off	-1.8183	1.014	-1.792	0.073	-3.807	0.170
Tags_wrong number given	-22.8008	3.02e+04	-0.001	0.999	-5.92e+04	5.91e+04
Lead Quality_Low in Relevance	-0.6390	0.433	-1.474	0.140	-1.488	0.211
Lead Quality_Might be	-1.3393	0.394	-3.403	0.001	-2.111	-0.568
Lead Quality_Not Sure	-4.1142	0.377	-10.912	0.000	-4.853	-3.375
Lead Quality_Worst	-4.8082	1.015	-4.736	0.000	-6.798	-2.819
City_Other Cities	-0.2032	0.224	-0.908	0.364	-0.642	0.236
City_Other Cities of Maharashtra	-0.0075	0.261	-0.029	0.977	-0.518	0.504
City_Other Metro Cities	0.1117	0.287	0.389	0.697	-0.451	0.674
City_Thane & Outskirts	-0.1038	0.218	-0.477	0.634	-0.530	0.323
City_Tier II Cities	0.9188	0.654	1.405	0.160	-0.363	2.200
Last Notable Activity_Email Bounced	-20.1482	2.16e+05	-9.33e-05	1.000	-4.23e+05	4.23e+05
Last Notable Activity_Email Link Clicked	-23.1902	2.16e+05	-0.000	1.000	-4.23e+05	4.23e+05
Last Notable Activity_Email Marked Spam	0.5190	2.56e+05	2.02e-06	1.000	-5.02e+05	5.02e+05
Last Notable Activity_Email Opened	-21.4918	2.16e+05	-9.95e-05	1.000	-4.23e+05	4.23e+05
Last Notable Activity_Email Received	-1.9838	3.05e+05	-6.49e-06	1.000	-5.99e+05	5.99e+05
Last Notable Activity_Form Submitted on Website	-45.2737	3.05e+05	-0.000	1.000	-5.99e+05	5.99e+05
Last Notable Activity_Had a Phone Conversation	-21.6355	2.16e+05	-0.000	1.000	-4.23e+05	4.23e+05
Last Notable Activity_Modified	-22.7328	2.16e+05	-0.000	1.000	-4.23e+05	4.23e+05
Last Notable Activity_Olark Chat Conversation	-22.7846	2.16e+05	-0.000	1.000	-4.23e+05	4.23e+05
Last Notable Activity_Page Visited on Website	-22.5728	2.16e+05	-0.000	1.000	-4.23e+05	4.23e+05

Last Notable Activity_Resubscribed to emails	-1.7953	3.05e+05	-5.88e-06	1.000	-5.99e+05	5.99e+05
Last Notable Activity_SMS Sent	-20.2275	2.16e+05	-9.37e-05	1.000	-4.23e+05	4.23e+05
Last Notable Activity_Unreachable	-21.1488	2.16e+05	-9.79e-05	1.000	-4.23e+05	4.23e+05
Last Notable Activity_Unsubscribed	-21.3606	2.16e+05	-9.89e-05	1.000	-4.23e+05	4.23e+05
Last Notable Activity_View in browser link Clicked	-43.1067	3.05e+05	-0.000	1.000	-5.99e+05	5.99e+05

Feature Selection Using RFE

In [107]:

```
from sklearn.linear_model import LogisticRegression
logreg = LogisticRegression()

from sklearn.feature_selection import RFE
rfe = RFE(logreg, 15)             # running RFE with 15 variables as output
rfe = rfe.fit(X_train, y_train)
```

In [108]:

```
rfe.support_
```

Out[108]:

```
array([ True, False, False, False, False, False,  True, False, False,
       False, False, False, False, 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, False,  True, False,  True,  True, False, False,  True,
       False, False,  True,  True,  True,  True,  True, False, False,
        True,  True, False, False, False, False, False, False, False,
       False, False, False, False, False, False, False, False, False,
        True, False, False, False])
```

In [109]:

```
list(zip(X_train.columns, rfe.support_, rfe.ranking_))
```

Out[109]:

```
[('Do Not Email', True, 1),
 ('Do Not Call', False, 33),
 ('TotalVisits', False, 39),
 ('Total Time Spent on Website', False, 3),
 ('Page Views Per Visit', False, 38),
 ('Lead Origin_Landing Page Submission', False, 16),
 ('Lead Origin_Lead Add Form', True, 1),
 ('Lead Origin_Lead Import', False, 2),
 ('Lead Source_Facebook', False, 44),
 ('Lead Source_Google', False, 41),
 ('Lead Source_Olark Chat', False, 5),
 ('Lead Source_Organic Search', False, 42),
 ('Lead Source_Others', False, 47),
 ('Lead Source_Reference', False, 68),
 ('Lead Source_Referral Sites', False, 51),
 ('Lead Source_Welingak Website', True, 1),
 ('Last Activity_Email Bounced', False, 29),
 ('Last Activity_Email Link Clicked', False, 35),
 ('Last Activity_Email Opened', False, 66),
 ('Last Activity_Form Submitted on Website', False, 67),
 ('Last Activity_Olark Chat Conversation', False, 13),
 ('Last Activity_Other_Activity', False, 9),
 ('Last Activity_Page Visited on Website', False, 36),
 ('Last Activity_SMS Sent', False, 7),
 ('Last Activity_Unreachable', False, 14),
 ('Last Activity_Unsubscribed', False, 18),
 ('Specialization_Business Administration', False, 60),
 ('Specialization_E-Business', False, 63),
 ('Specialization_E-COMMERCE', False, 15),
```

```
('Specialization_E-COMMERCE', False, 15),
('Specialization_Finance Management', False, 40),
('Specialization_Healthcare Management', False, 37),
('Specialization_Hospitality Management', False, 61),
('Specialization_Human Resource Management', False, 55),
('Specialization_IT Projects Management', False, 50),
('Specialization_International Business', False, 21),
('Specialization_Marketing Management', False, 32),
('Specialization_Media and Advertising', False, 34),
('Specialization_Operations Management', False, 70),
('Specialization_Other_Specialization', False, 20),
('Specialization_Retail Management', False, 58),
('Specialization_Rural and Agribusiness', False, 48),
('Specialization_Services Excellence', False, 56),
('Specialization_Supply Chain Management', False, 46),
('Specialization_Travel and Tourism', False, 25),
('What is your current occupation_Housewife', False, 43),
('What is your current occupation_Other_Occupation', False, 45),
('What is your current occupation_Student', False, 8),
('What is your current occupation_Unemployed', True, 1),
('What is your current occupation_Working Professional', False, 26),
('Tags_Busy', True, 1),
('Tags_Closed by Horizzon', True, 1),
('Tags_Interested  in full time MBA', False, 19),
('Tags_Interested in other courses', False, 11),
('Tags_Lost to EINS', True, 1),
('Tags_Not doing further education', False, 17),
('Tags_Other_Tags', False, 27),
('Tags_Ringing', True, 1),
('Tags_Will revert after reading the email', True, 1),
('Tags_invalid number', True, 1),
('Tags_switched off', True, 1),
('Tags_wrong number given', True, 1),
('Lead Quality_Low in Relevance', False, 69),
('Lead Quality_Might be', False, 10),
('Lead Quality_Not Sure', True, 1),
('Lead Quality_Worst', True, 1),
('City_Other Cities', False, 49),
('City_Other Cities of Maharashtra', False, 64),
('City_Other Metro Cities', False, 59),
('City_Thane & Outskirts', False, 52),
('City_Tier II Cities', False, 24),
('Last Notable Activity_Email Bounced', False, 23),
('Last Notable Activity_Email Link Clicked', False, 12),
('Last Notable Activity_Email Marked Spam', False, 54),
('Last Notable Activity_Email Opened', False, 57),
('Last Notable Activity_Email Received', False, 71),
('Last Notable Activity_Form Submitted on Website', False, 53),
('Last Notable Activity_Had a Phone Conversation', False, 30),
('Last Notable Activity_Modified', False, 6),
('Last Notable Activity_Olark Chat Conversation', False, 4),
('Last Notable Activity_Page Visited on Website', False, 22),
('Last Notable Activity_Resubscribed to emails', False, 65),
('Last Notable Activity_SMS Sent', True, 1),
('Last Notable Activity_Unreachable', False, 28),
('Last Notable Activity_Unsubscribed', False, 31),
('Last Notable Activity_View in browser link Clicked', False, 62)]
```

In [110]:

```
col = X_train.columns[rfe.support_]
col
```

Out[110]:

```
Index(['Do Not Email', 'Lead Origin_Lead Add Form',
       'Lead Source_Welingak Website',
       'What is your current occupation_Unemployed', 'Tags_Busy',
       'Tags_Closed by Horizzon', 'Tags_Lost to EINS', 'Tags_Ringing',
       'Tags_Will revert after reading the email', 'Tags_invalid number',
       'Tags_switched off', 'Tags_wrong number given', 'Lead Quality_Not Sure',
       'Lead Quality_Worst', 'Last Notable Activity_SMS Sent'],
      dtype='object')
```

In [111]:

```
X_train.columns[~rfe.support_]
```

Out[111]:

```
Index(['Do Not Call', 'TotalVisits', 'Total Time Spent on Website',
       'Page Views Per Visit', 'Lead Origin_Landing Page Submission',
       'Lead Origin_Lead Import', 'Lead Source_Facebook', 'Lead Source_Google',
       'Lead Source_Olark Chat', 'Lead Source_Organic Search',
       'Lead Source_Others', 'Lead Source_Reference',
       'Lead Source_Referral Sites', '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_Other_Activity',
       'Last Activity_Page Visited on Website', 'Last Activity_SMS Sent',
       'Last Activity_Unreachable', 'Last Activity_Unsubscribed',
       '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_Other_Specialization',
       'Specialization_Retail Management',
       'Specialization_Rural and Agribusiness',
       'Specialization_Services Excellence',
       'Specialization_Supply Chain Management',
       'Specialization_Travel and Tourism',
       'What is your current occupation_Housewife',
       'What is your current occupation_Other_Occupation',
       'What is your current occupation_Student',
       'What is your current occupation_Working Professional',
       'Tags_Interested  in full time MBA', 'Tags_Interested in other courses',
       'Tags_Not doing further education', 'Tags_Other_Tags',
       'Lead Quality_Low in Relevance', 'Lead Quality_Might be',
       'City_Other Cities', 'City_Other Cities of Maharashtra',
       'City_Other Metro Cities', 'City_Thane & Outskirts',
       'City_Tier II Cities', 'Last Notable Activity_Email Bounced',
       'Last Notable Activity_Email Link Clicked',
       'Last Notable Activity_Email Marked Spam',
       'Last Notable Activity_Email Opened',
       'Last Notable Activity_Email Received',
       'Last Notable Activity_Form Submitted on Website',
       'Last Notable Activity_Had a Phone Conversation',
       'Last Notable Activity_Modified',
       'Last Notable Activity_Olark Chat Conversation',
       'Last Notable Activity_Page Visited on Website',
       'Last Notable Activity_Resubscribed to emails',
       'Last Notable Activity_Unreachable',
       'Last Notable Activity_Unsubscribed',
       'Last Notable Activity_View in browser link Clicked'],
      dtype='object')
```

Assessing the model with StatsModels

In [112]:

```
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[112]:

Generalized Linear Model Regression Results

Dep. Variable:	Converted	**No. Observations:**	6351
Model:	GLM	**Df Residuals:**	6335
Model Family:	Binomial	**Df Model:**	15

Link Function:	logit	**Scale:**	1.0000
Method:	IRLS	**Log-Likelihood:**	-1580.1
Date:	Mon, 11 May 2020	**Deviance:**	3160.2
Time:	13:00:31	**Pearson chi2:**	3.11e+04
No. Iterations:	24		
Covariance Type:	nonrobust		

	coef	std err	z	P>|z|	[0.025	0.975]
const	-0.7920	0.278	-2.845	0.004	-1.338	-0.246
Do Not Email	-1.3202	0.212	-6.236	0.000	-1.735	-0.905
Lead Origin_Lead Add Form	1.0521	0.363	2.897	0.004	0.340	1.764
Lead Source_Welingak Website	3.4638	0.819	4.231	0.000	1.859	5.068
What is your current occupation_Unemployed	-1.1148	0.237	-4.713	0.000	-1.578	-0.651
Tags_Busy	3.5772	0.333	10.752	0.000	2.925	4.229
Tags_Closed by Horizzon	7.7760	0.762	10.203	0.000	6.282	9.270
Tags_Lost to EINS	8.9986	0.754	11.931	0.000	7.520	10.477
Tags_Ringing	-1.9203	0.340	-5.640	0.000	-2.588	-1.253
Tags_Will revert after reading the email	3.7576	0.229	16.412	0.000	3.309	4.206
Tags_invalid number	-23.4125	2.21e+04	-0.001	0.999	-4.34e+04	4.34e+04
Tags_switched off	-2.5224	0.589	-4.279	0.000	-3.678	-1.367
Tags_wrong number given	-23.0270	3.17e+04	-0.001	0.999	-6.21e+04	6.2e+04
Lead Quality_Not Sure	-3.3269	0.129	-25.702	0.000	-3.581	-3.073
Lead Quality_Worst	-3.9922	0.832	-4.798	0.000	-5.623	-2.361
Last Notable Activity_SMS Sent	2.7952	0.122	22.846	0.000	2.555	3.035

In [113]:

```
col1 = col.drop('Tags_invalid number',1)
```

In [114]:

```
col1
```

Out[114]:

```
Index(['Do Not Email', 'Lead Origin_Lead Add Form',
       'Lead Source_Welingak Website',
       'What is your current occupation_Unemployed', 'Tags_Busy',
       'Tags_Closed by Horizzon', 'Tags_Lost to EINS', 'Tags_Ringing',
       'Tags_Will revert after reading the email', 'Tags_switched off',
       'Tags_wrong number given', 'Lead Quality_Not Sure',
       'Lead Quality_Worst', 'Last Notable Activity_SMS Sent'],
      dtype='object')
```

In [115]:

```
X_train_sm = sm.add_constant(X_train[col1])
logm2 = sm.GLM(y_train,X_train_sm, family = sm.families.Binomial())
res = logm2.fit()
res.summary()
```

Out[115]:

Generalized Linear Model Regression Results

Dep. Variable:	Converted	**No. Observations:**	6351
Model:	GLM	**Df Residuals:**	6336
Model Family:	Binomial	**Df Model:**	14

Link Function:	logit	**Scale:**	1.0000
Method:	IRLS	**Log-Likelihood:**	-1585.9
Date:	Mon, 11 May 2020	**Deviance:**	3171.8
Time:	13:00:31	**Pearson chi2:**	3.07e+04
No. Iterations:	22		
Covariance Type:	nonrobust		

	coef	std err	z	P>|z|	[0.025	0.975]
const	-0.9144	0.282	-3.245	0.001	-1.467	-0.362
Do Not Email	-1.3129	0.211	-6.218	0.000	-1.727	-0.899
Lead Origin_Lead Add Form	1.0839	0.365	2.969	0.003	0.368	1.800
Lead Source_Welingak Website	3.4275	0.819	4.184	0.000	1.822	5.033
What is your current occupation_Unemployed	-1.1577	0.239	-4.848	0.000	-1.626	-0.690
Tags_Busy	3.7579	0.331	11.338	0.000	3.108	4.407
Tags_Closed by Horizzon	7.9271	0.763	10.394	0.000	6.432	9.422
Tags_Lost to EINS	9.1535	0.755	12.128	0.000	7.674	10.633
Tags_Ringing	-1.7229	0.339	-5.089	0.000	-2.386	-1.059
Tags_Will revert after reading the email	3.9200	0.230	17.026	0.000	3.469	4.371
Tags_switched off	-2.3187	0.588	-3.942	0.000	-3.471	-1.166
Tags_wrong number given	-20.8331	1.17e+04	-0.002	0.999	-2.29e+04	2.28e+04
Lead Quality_Not Sure	-3.3174	0.129	-25.685	0.000	-3.571	-3.064
Lead Quality_Worst	-3.9830	0.834	-4.777	0.000	-5.617	-2.349
Last Notable Activity_SMS Sent	2.7537	0.121	22.849	0.000	2.518	2.990

In [116]:

```
col2 = col1.drop('Tags_wrong number given',1)
```

In [117]:

```
col2
```

Out[117]:

```
Index(['Do Not Email', 'Lead Origin_Lead Add Form',
       'Lead Source_Welingak Website',
       'What is your current occupation_Unemployed', 'Tags_Busy',
       'Tags_Closed by Horizzon', 'Tags_Lost to EINS', 'Tags_Ringing',
       'Tags_Will revert after reading the email', 'Tags_switched off',
       'Lead Quality_Not Sure', 'Lead Quality_Worst',
       'Last Notable Activity_SMS Sent'],
      dtype='object')
```

In [118]:

```
X_train_sm = sm.add_constant(X_train[col2])
logm2 = sm.GLM(y_train,X_train_sm, family = sm.families.Binomial())
res = logm2.fit()
res.summary()
```

Out[118]:

Generalized Linear Model Regression Results

Dep. Variable:	Converted	**No. Observations:**	6351
Model:	GLM	**Df Residuals:**	6337
Model Family:	Binomial	**Df Model:**	13
Link Function:	logit	**Scale:**	1.0000
Method:	IRLS	**Log-Likelihood:**	-1587.9

Date:	Mon, 11 May 2020	**Deviance:**	3175.8
Time:	13:00:32	**Pearson chi2:**	3.08e+04
No. Iterations:	8		
Covariance Type:	nonrobust		

	coef	std err	z	P>|z|	[0.025	0.975]
const	-0.9661	0.283	-3.417	0.001	-1.520	-0.412
Do Not Email	-1.3127	0.211	-6.223	0.000	-1.726	-0.899
Lead Origin_Lead Add Form	1.0963	0.366	2.995	0.003	0.379	1.814
Lead Source_Welingak Website	3.4147	0.820	4.166	0.000	1.808	5.021
What is your current occupation_Unemployed	-1.1746	0.240	-4.899	0.000	-1.644	-0.705
Tags_Busy	3.8305	0.330	11.598	0.000	3.183	4.478
Tags_Closed by Horizzon	7.9914	0.763	10.480	0.000	6.497	9.486
Tags_Lost to EINS	9.2178	0.755	12.217	0.000	7.739	10.697
Tags_Ringing	-1.6472	0.337	-4.885	0.000	-2.308	-0.986
Tags_Will revert after reading the email	3.9881	0.229	17.380	0.000	3.538	4.438
Tags_switched off	-2.2412	0.587	-3.816	0.000	-3.392	-1.090
Lead Quality_Not Sure	-3.3158	0.129	-25.690	0.000	-3.569	-3.063
Lead Quality_Worst	-3.9600	0.836	-4.734	0.000	-5.599	-2.321
Last Notable Activity_SMS Sent	2.7443	0.120	22.856	0.000	2.509	2.980

In [119]:

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

Out[119]:

```
3009    0.187192
1012    0.167079
9226    0.000821
4750    0.781753
7987    0.977276
1281    0.989966
2880    0.187192
4971    0.753675
7536    0.863827
1248    0.000821
dtype: float64
```

In [120]:

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

Out[120]:

```
array([1.87191534e-01, 1.67078806e-01, 8.21369066e-04, 7.81753466e-01,
       9.77276034e-01, 9.89966304e-01, 1.87191534e-01, 7.53674840e-01,
       8.63826796e-01, 8.21369066e-04])
```

Creating a dataframe with the actual churn flag and the predicted probabilities

In [121]:

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

Out[121]:

	Converted	Converted_prob	Prospect ID
0	0	0.187192	3009
1	0	0.167079	1012
2	0	0.000821	9226
3	1	0.781753	4750
4	1	0.977276	7987

Creating new column 'predicted' with 1 if Churn_Prob > 0.5 else 0¶

In [122]:

```
y_train_pred_final['predicted'] = y_train_pred_final.Converted_prob.map(lambda x: 1 if x > 0.5 else
0)

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

Out[122]:

	Converted	Converted_prob	Prospect ID	predicted
0	0	0.187192	3009	0
1	0	0.167079	1012	0
2	0	0.000821	9226	0
3	1	0.781753	4750	1
4	1	0.977276	7987	1

In [123]:

```
from sklearn import metrics

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

```
[[3756  149]
 [ 363 2083]]
```

In [124]:

```
# Predicted     not_churn    churn
# Actual
# not_churn        3270       365
# churn            579        708
```

In [125]:

```
# Let's check the overall accuracy.
print(metrics.accuracy_score(y_train_pred_final.Converted, y_train_pred_final.predicted))
```

```
0.9193827743662415
```

Checking VIFs

In [126]:

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

In [127]:

```python
# Create a dataframe that will contain the names of all the feature variables and their respective VIFs
vif = pd.DataFrame()
vif['Features'] = X_train[col2].columns
vif['VIF'] = [variance_inflation_factor(X_train[col].values, i) for i in range(X_train[col2].shape[1])]
vif['VIF'] = round(vif['VIF'], 2)
vif = vif.sort_values(by = "VIF", ascending = False)
vif
```

Out[127]:

	Features	VIF
3	What is your current occupation_Unemployed	7.37
12	Last Notable Activity_SMS Sent	4.05
8	Tags_Will revert after reading the email	4.02
7	Tags_Ringing	1.86
1	Lead Origin_Lead Add Form	1.58
2	Lead Source_Welingak Website	1.34
5	Tags_Closed by Horizzon	1.25
10	Lead Quality_Not Sure	1.17
4	Tags_Busy	1.15
0	Do Not Email	1.11
6	Tags_Lost to EINS	1.08
9	Tags_switched off	1.06
11	Lead Quality_Worst	1.03

Metrics beyond simply accuracy

In [128]:

```python
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 [129]:

```python
# Let's see the sensitivity of our logistic regression model
TP / float(TP+FN)
```

Out[129]:

0.8515944399018807

In [130]:

```python
# Let us calculate specificity
TN / float(TN+FP)
```

Out[130]:

0.9618437900128041

In [131]:

```python
# Calculate false postive rate - predicting churn when customer does not have churned
print(FP/ float(TN+FP))
```

0.038156209987195905

In [132]:

```
# positive predictive value
print (TP / float(TP+FP))
```

```
0.9332437275985663
```

In [133]:

```
# Negative predictive value
print (TN / float(TN+ FN))
```

```
0.9118718135469774
```

Plotting the ROC Curve

An ROC curve demonstrates several things:

It shows the tradeoff between sensitivity and specificity (any increase in sensitivity will be accompanied by a decrease in specificity). The closer the curve follows the left-hand border and then the top border of the ROC space, the more accurate the test. The closer the curve comes to the 45-degree diagonal of the ROC space, the less accurate the test.

In [134]:

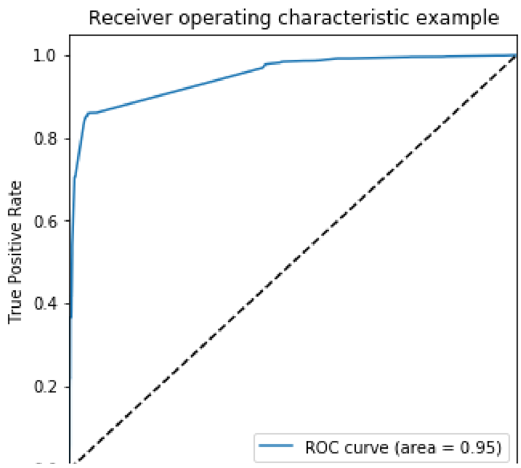
```
def draw_roc( actual, probs ):
    fpr, tpr, thresholds = metrics.roc_curve( actual, probs,
                                              drop_intermediate = False )
    auc_score = metrics.roc_auc_score( actual, probs )
    plt.figure(figsize=(5, 5))
    plt.plot( fpr, tpr, label='ROC curve (area = %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
```

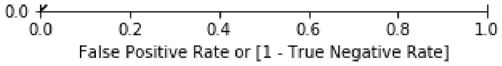
In [135]:

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

In [136]:

```
draw_roc(y_train_pred_final.Converted, y_train_pred_final.Converted_prob)
```





Finding Optimal Cutoff Point

In [137]:

```
# Let's 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 else 0)
y_train_pred_final.head()
```

Out[137]:

	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	0	0.187192	3009	0	1	1	0	0	0	0	0	0	0	0
1	0	0.167079	1012	0	1	1	0	0	0	0	0	0	0	0
2	0	0.000821	9226	0	1	0	0	0	0	0	0	0	0	0
3	1	0.781753	4750	1	1	1	1	1	1	1	1	1	0	0
4	1	0.977276	7987	1	1	1	1	1	1	1	1	1	1	1

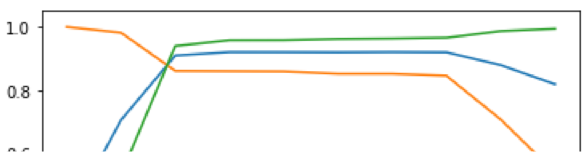
In [138]:

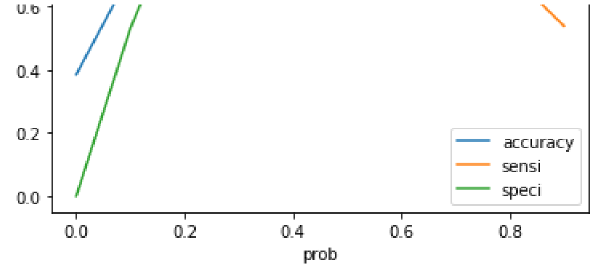
```
# Now let's calculate accuracy sensitivity and specificity for various probability cutoffs.
cutoff_df = pd.DataFrame( columns = ['prob','accuracy','sensi','speci'])
from sklearn.metrics import confusion_matrix

# 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  0.385136  1.000000  0.000000
0.1   0.1  0.705086  0.981603  0.531882
0.2   0.2  0.909148  0.860589  0.939565
0.3   0.3  0.920013  0.859771  0.957746
0.4   0.4  0.919855  0.858953  0.958003
0.5   0.5  0.919383  0.851594  0.961844
0.6   0.6  0.920170  0.851594  0.963124
0.7   0.7  0.919383  0.845462  0.965685
0.8   0.8  0.878917  0.706868  0.986684
0.9   0.9  0.818769  0.538839  0.994110
```

In [139]:

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





In [140]:

```
#### From the curve above, 0.2 is the optimum point to take it as a cutoff probability.

y_train_pred_final['final_predicted'] = y_train_pred_final.Converted_prob.map( lambda x: 1 if x > 0
.2 else 0)

y_train_pred_final.head()
```

Out[140]:

	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_predicted
0	0	0.187192	3009	0	1	1	0	0	0	0	0	0	0	0	0
1	0	0.167079	1012	0	1	1	0	0	0	0	0	0	0	0	0
2	0	0.000821	9226	0	1	0	0	0	0	0	0	0	0	0	0
3	1	0.781753	4750	1	1	1	1	1	1	1	1	1	0	0	1
4	1	0.977276	7987	1	1	1	1	1	1	1	1	1	1	1	1

Assigning Lead Score

In [141]:

```
y_train_pred_final['Lead_Score'] = y_train_pred_final.Converted_prob.map( lambda x: round(x*100))

y_train_pred_final.head()
```

Out[141]:

	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_predicted	Lead_Score
0	0	0.187192	3009	0	1	1	0	0	0	0	0	0	0	0	0	19
1	0	0.167079	1012	0	1	1	0	0	0	0	0	0	0	0	0	17
2	0	0.000821	9226	0	1	0	0	0	0	0	0	0	0	0	0	0
3	1	0.781753	4750	1	1	1	1	1	1	1	1	1	0	0	1	78
4	1	0.977276	7987	1	1	1	1	1	1	1	1	1	1	1	1	98

In [142]:

```
# Let's check the overall accuracy.
metrics.accuracy_score(y_train_pred_final.Converted, y_train_pred_final.final_predicted)

confusion2 = metrics.confusion_matrix(y_train_pred_final.Converted,
y_train_pred_final.final_predicted )
confusion2

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 [143]:

```
# Let's see the sensitivity of our logistic regression model
TP / float(TP+FN)
```

Out[143]:

```
0.8605887162714636
```

In [144]:

```
# Let us calculate specificity
TN / float(TN+FP)
```

Out[144]:

```
0.9395646606914213
```

In [145]:

```
# Calculate false postive rate - predicting churn when customer does not have churned
print(FP/ float(TN+FP))
```

```
0.060435339308578744
```

In [146]:

```
# Positive predictive value
print (TP / float(TP+FP))
```

```
0.8991883810337462
```

In [147]:

```
# Negative predictive value
print (TN / float(TN+ FN))
```

```
0.9149625935162095
```

Precision and Recall

In [148]:

```
#Looking at the confusion matrix again

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

Out[148]:

```
array([[3756,  149],
       [ 363, 2083]], dtype=int64)
```

In [149]:

```
##### Precision
TP / TP + FP

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

Out[149]:

```
0.9332437275985663
```

In [150]:

```
##### Recall
TP / TP + FN

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

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

Out[150]:

```
0.8515944399018807
```

Using sklearn utilities for the same

In [151]:

```
from sklearn.metrics import precision_score, recall_score
```

In [152]:

```
precision_score(y_train_pred_final.Converted , y_train_pred_final.predicted)
```

Out[152]:

```
0.9332437275985663
```

In [153]:

```
recall_score(y_train_pred_final.Converted, y_train_pred_final.predicted)
```

Out[153]:

```
0.8515944399018807
```

Precision and recall tradeoff

In [154]:

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

In [155]:

```
y_train_pred_final.Converted, y_train_pred_final.predicted
```

Out[155]:

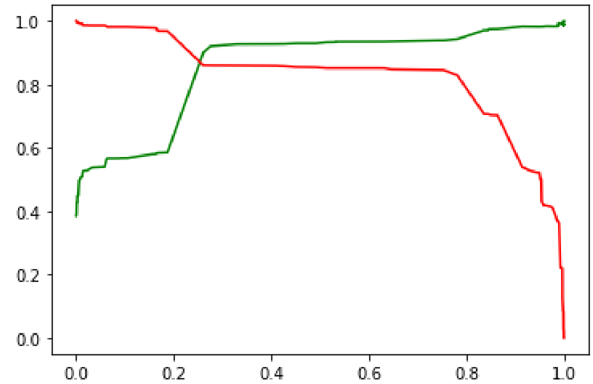
```
(0       0
 1       0
 2       0
 3       1
 4       1
        ..
 6346    0
 6347    1
 6348    0
 6349    0
 6350    0
 Name: Converted, Length: 6351, dtype: int64,
 0       0
 1       0
 2       0
 3       1
 4       1
        ..
 6346    0
 6347    1
 6348    0
 6349    0
 6350    0
 Name: predicted,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

In [156]:

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

In [157]:

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



Making predictions on the test set

In [158]:

```
X_test[['TotalVisits','Total Time Spent on Website','Page Views Per Visit']] =
scaler.fit_transform(X_test[['TotalVisits','Total Time Spent on Website','Page Views Per Visit']])

X_train.head()
```

Out[158]:

	Do Not Email	Do Not Call	TotalVisits	Total Time Spent on Website	Page Views Per Visit	Lead Origin_Landing Page Submission	Lead Origin_Lead Add Form	Lead Origin_Lead Import	Lead Source_Facebook	Lead Source_Google	...
3009	0	0	-0.432779	-0.160255	-0.155018	1	0	0	0	0	...
1012	1	0	-0.432779	-0.540048	-0.155018	1	0	0	0	0	...
9226	0	0	-1.150329	-0.888650	-1.265540	0	0	0	0	0	...
4750	0	0	-0.432779	1.643304	-0.155018	1	0	0	0	0	...
7987	0	0	0.643547	2.017593	0.122613	1	0	0	0	0	...

5 rows × 85 columns

In [159]:

```
X_test = X_test[col2]
X_test.head()
```

Out[159]:

	Do Not Email	Lead Origin_Lead Add Form	Lead Source_Welingak Website	What is your current occupation_Unemployed	Tags_Busy	Tags_Closed by Horizzon	Tags_Lost to EINS	Tags_Ringing	Tags_Will revert after reading the email
3271	0	0	0	1	0	0	0	0	1
1490	0	0	0	0	0	0	0	0	1
7936	0	0	0	1	0	0	0	0	1
4216	0	1	0	0	0	1	0	0	0
3830	0	0	0	1	0	0	0	0	1

3830	0	0	0	1	0	0	0	0	1

Tags_Will

In [160]:

```
X_test_sm = sm.add_constant(X_test)
```

Making predictions on the test set

In [161]:

```
y_test_pred = res.predict(X_test_sm)
```

In [162]:

```
y_test_pred[:10]
```

Out[162]:

```
3271    0.187192
1490    0.953558
7936    0.187192
4216    0.999703
3830    0.187192
1800    0.953558
6507    0.012624
4821    0.000454
4223    0.996625
4714    0.187192
dtype: float64
```

In [163]:

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

In [164]:

```
# Let's see the head
y_pred_1.head()
```

Out[164]:

	0
3271	0.187192
1490	0.953558
7936	0.187192
4216	0.999703
3830	0.187192

In [165]:

```
# Converting y_test to dataframe
y_test_df = pd.DataFrame(y_test)
```

In [166]:

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

In [167]:

```
# 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 [168]:

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

In [169]:

```
y_pred_final.head()
```

Out[169]:

	Converted	Prospect ID	0
0	0	3271	0.187192
1	1	1490	0.953558
2	0	7936	0.187192
3	1	4216	0.999703
4	0	3830	0.187192

In [172]:

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

In [175]:

```
# Let's see the head of y_pred_final
y_pred_final.head()
```

Out[175]:

	Converted	Prospect ID	Converted_prob
0	0	3271	0.187192
1	1	1490	0.953558
2	0	7936	0.187192
3	1	4216	0.999703
4	0	3830	0.187192

In [176]:

```
y_pred_final['final_predicted'] = y_pred_final.Converted_prob.map(lambda x: 1 if x > 0.2 else 0)
```

In [177]:

```
y_pred_final.head()
```

Out[177]:

	Converted	Prospect ID	Converted_prob	final_predicted
0	0	3271	0.187192	0
1	1	1490	0.953558	1
2	0	7936	0.187192	0
3	1	4216	0.999703	1
4	0	3830	0.187192	0

In [178]:

```
# Let's check the overall accuracy.
```

```
metrics.accuracy_score(y_pred_final.Converted, y_pred_final.final_predicted)
```

Out[178]:

```
0.9045170767535806
```

In [179]:

```
confusion2 = metrics.confusion_matrix(y_pred_final.Converted, y_pred_final.final_predicted )
confusion2
```

Out[179]:

```
array([[1628,  106],
       [ 154,  835]], dtype=int64)
```

In [180]:

```
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 [181]:

```
# Let's see the sensitivity of our logistic regression model
TP / float(TP+FN)
```

Out[181]:

```
0.8442871587462083
```

In [182]:

```
# Let us calculate specificity
TN / float(TN+FP)
```

Out[182]:

```
0.9388696655132641
```