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.

Lead Conversion Process - Demonstrated as a 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.

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.

Goal

1. 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.

In [1]:

```
# Supress Warnings import
warnings
warnings.filterwarnings('ignore')
 # Importing libraries import
numpy as np import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
 # visulaisation from
matplotlib.pyplot import xticks
%matplotlib inline
 # Data display coustomization
pd.set_option('display.max_rows', 100)
pd.set_option('display.max_columns',
100)
```

Data Preparation

Data Loading

In [2]:

```
data = pd.DataFrame(pd.read_csv('../input/Leads.csv'))
data.head(5)
```

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
0	7927b2df8bba-4d29b9a2b6e0beafe620	660737	API	Olark Chat	No	No	0	0.0	0	0.0
1	2a2724365132-413686fadcc88c88f482	660728	API	Organic Search	No	No	0	5.0	674	2.5
2	8cc8c611a219-4f35ad23fdfd2656bd8a	660727	Landing Page Submission	Direct Traffic	No	No	1	2.0	1532	2.0
3	0cc2df48-7cf4-4e39-9de919797f9b38cc	660719	Landing Page Submission	Direct Traffic	No	No	0	1.0	305	1.0
4	3256f628e534-4826-9d63-4a8b88782852	660681	Landing Page Submission	Google	No	No	1	2.0	1428	1.0

In [3]:

```
#checking duplicates sum(data.duplicated(subset
= 'Prospect ID')) == 0
# No duplicate values
```

Out[3]:

```
True
```

Data Inspection

In [4]:

```
data.shape
```

Out[4]:

```
(9240, 37)
```

In [5]:

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9240 entries, 0 to 9239
Data columns (total 37 columns):
Prospect ID                                      9240 non-null object
Lead Number                                      9240 non-null int64
Lead Origin                                      9240 non-null object
Lead Source                                      9204 non-null object
Do Not Email                                     9240 non-null object
Do Not Call                                      9240 non-null object
Converted                                        9240 non-null int64
TotalVisits                                      9103 non-null float64
Total Time Spent on Website                      9240 non-null int64
Page Views Per Visit                             9103 non-null float64
Last Activity                                    9137 non-null object
Country                                          6779 non-null object
Specialization                                   7802 non-null object
How did you hear about X Education               7033 non-null object
What is your current occupation                  6550 non-null object
What matters most to you in choosing a course    6531 non-null object
Search                                           9240 non-null object
Magazine                                         9240 non-null object
Newspaper Article                                9240 non-null object
X Education Forums                               9240 non-null object
Newspaper                                        9240 non-null object
Digital Advertisement                            9240 non-null object
Through Recommendations                          9240 non-null object
Receive More Updates About Our Courses           9240 non-null object
Tags                                             5887 non-null object
Lead Quality                                     4473 non-null object
Update me on Supply Chain Content                9240 non-null object
Get updates on DM Content                        9240 non-null object
Lead Profile                                     6531 non-null object
City                                             7820 non-null object
Asymmetrique Activity Index                      5022 non-null object
Asymmetrique Profile Index                       5022 non-null object
Asymmetrique Activity Score                      5022 non-null float64
Asymmetrique Profile Score                       5022 non-null float64
I agree to pay the amount through cheque         9240 non-null object
A free copy of Mastering The Interview           9240 non-null
object Last Notable Activity                            9240 non-
```

```
null object dtypes: float64(4), int64(3), object(30) memory usage:
2.6+ MB
```

In [6]:

```
data.describe()
```

Out[6]:

	Lead Number	Converted	TotalVisits	Total Time Spent on Website	Page Views Per Visit	Asymmetrique Activity Score	Asymmet Profile S
count	9240.000000	9240.000000	9103.000000	9240.000000	9103.000000	5022.000000	5022.000
mean	617188.435606	0.385390	3.445238	487.698268	2.362820	14.306252	16.34488
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	11.00000
25%	596484.500000	0.000000	1.000000	12.000000	1.000000	14.000000	15.00000
50%	615479.000000	0.000000	3.000000	248.000000	2.000000	14.000000	16.00000
75%	637387.250000	1.000000	5.000000	936.000000	3.000000	15.000000	18.00000
max	660737.000000	1.000000	251.000000	2272.000000	55.000000	18.000000	20.00000

Data Cleaning

In [7]:

```
# As we can observe that there are select values for many column.
#This is because customer did not select any option from the list, hence it shows
select.
# Select values are as good as NULL.
 # Converting 'Select' values to
NaN. data = data.replace('Select',
np.nan)
```

In [8]:

```python
data.isnull().sum()
```

Out[8]:

```
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
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 [9]:

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

Out[9]:

```
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 [10]:

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

In [11]:

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

In [12]:

```
# Lead Quality: Indicates the quality of lead based on the data and intuition the
the employee who has been assigned to the lead
```

In [13]:

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

Out[13]:

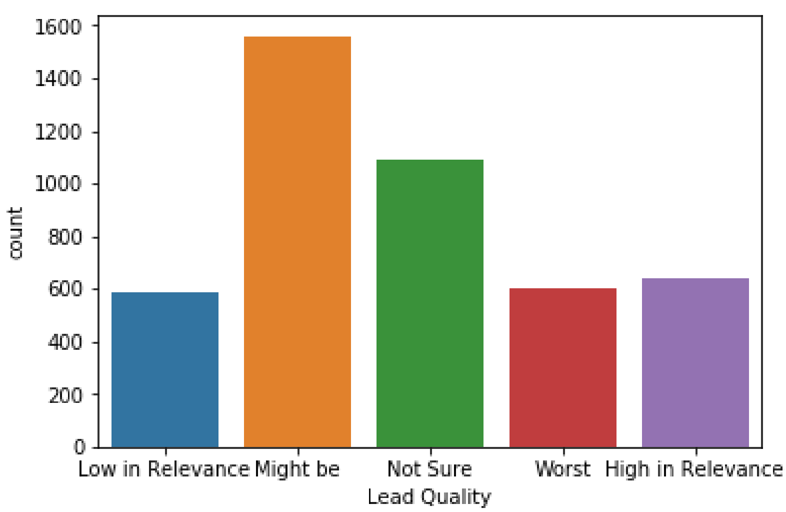
```
count          4473
unique
5 top        Might
be freq
1560
Name: Lead Quality, dtype: object
```

In [14]:

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

Out[14]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f17590b3668>
```



In [15]:

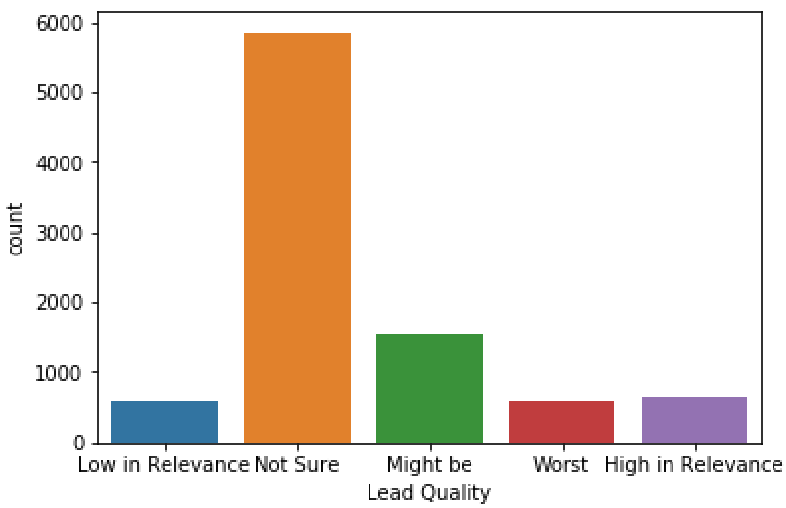
```
# As Lead quality is based on the intution of employee, so if left blank we can i
mpute 'Not Sure' in NaN safely.
data['Lead Quality'] = data['Lead Quality'].replace(np.nan, 'Not Sure')
```

In [16]:

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

Out[16]:

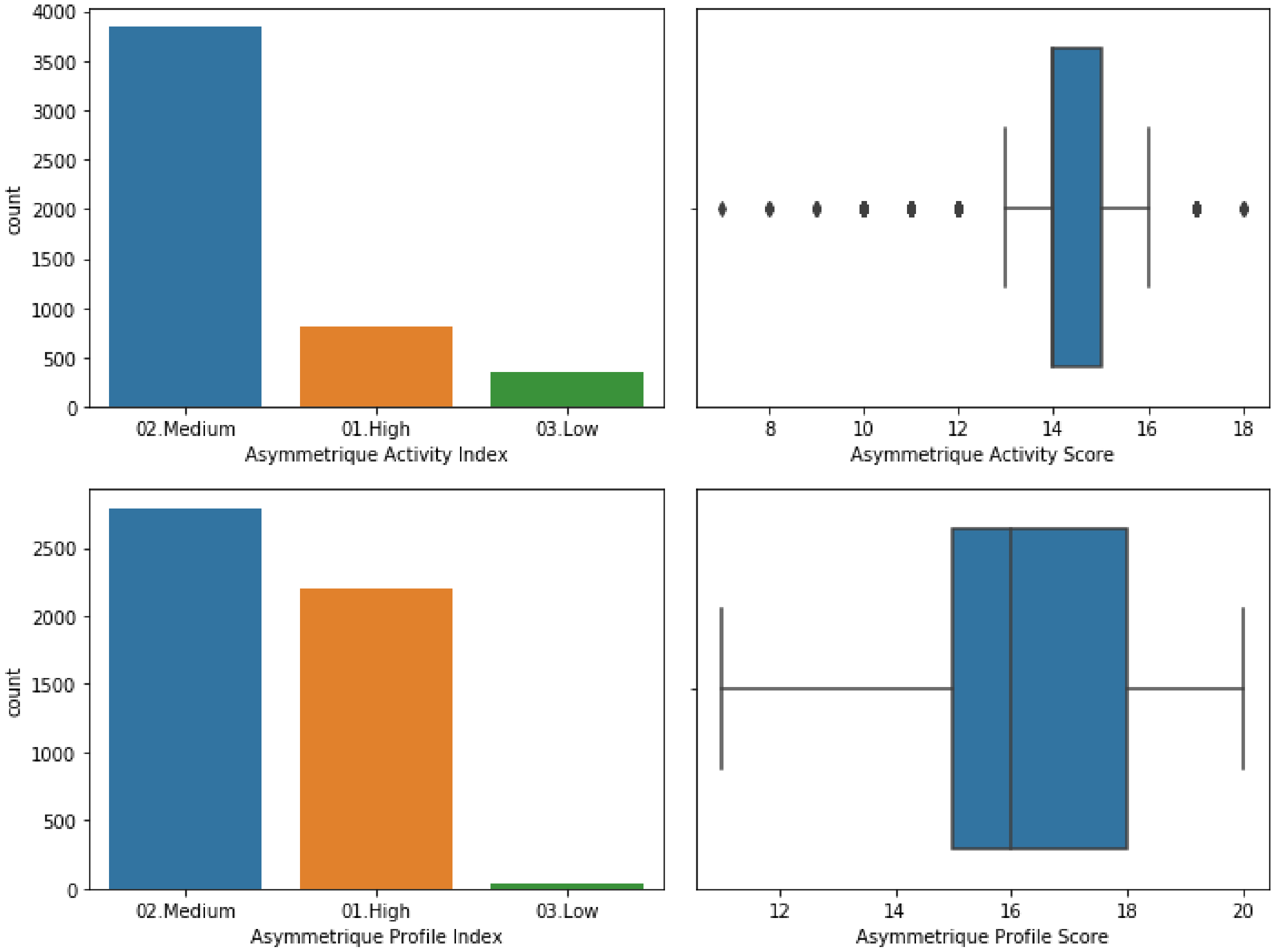
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f1758d8b5c0>
```



In [17]:
In [18]:

```
# Asymmetrique Activity Index  |
# Asymmetrique Profile Index   \    An index and score assigned to each customer
# Asymmetrique Activity Score  |     based on their activity and their profile
# Asymmetrique Profile Score   \
```

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



In [19]:

```
# There is too much variation in thes parameters so its not reliable to impute
an y value in it. # 45% null values means we need to drop these columns.
```

In [20]:

```
data = data.drop(['Asymmetrique Activity Index','Asymmetrique Activity Score','As
ymmetrique Profile Index','Asymmetrique Profile Score'],1)
```

In [21]:

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

Out[21]:

```
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 [22]:

```
# City
```

In [23]:

```
data.City.describe(
)
```

Out[23]:

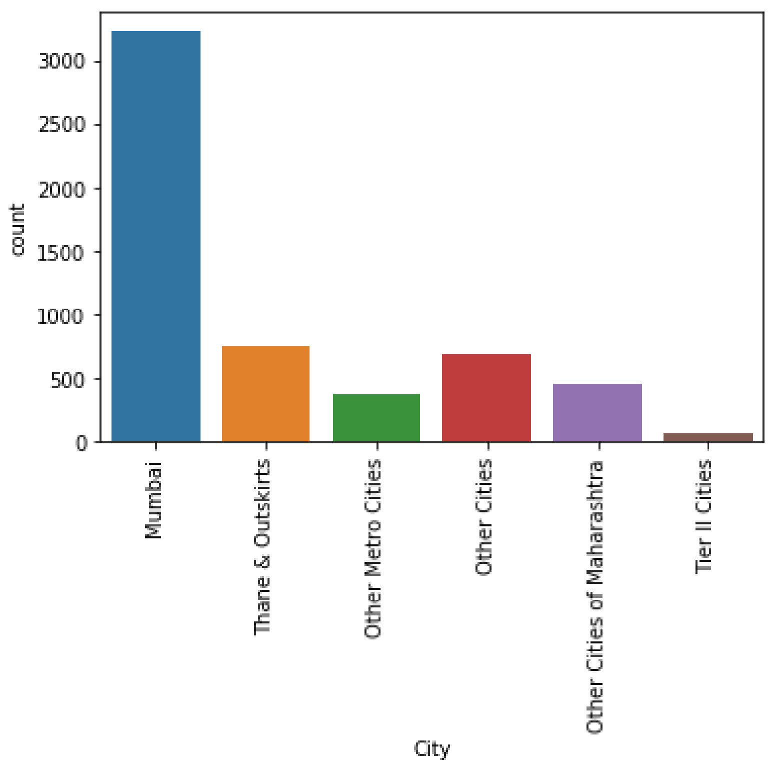
```
count        5571 unique
6 top        Mumbai freq
3222 Name: City, dtype:
object
```

In [24]:

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

Out[24]:

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



In [25]:

```
# Around 60% of the data is Mumbai so we can impute Mumbai in the missing values.
```

In [26]:

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

In [27]:

```
# Specailization
```

In [28]:

```
data.Specialization.describe()
```

Out[28]:

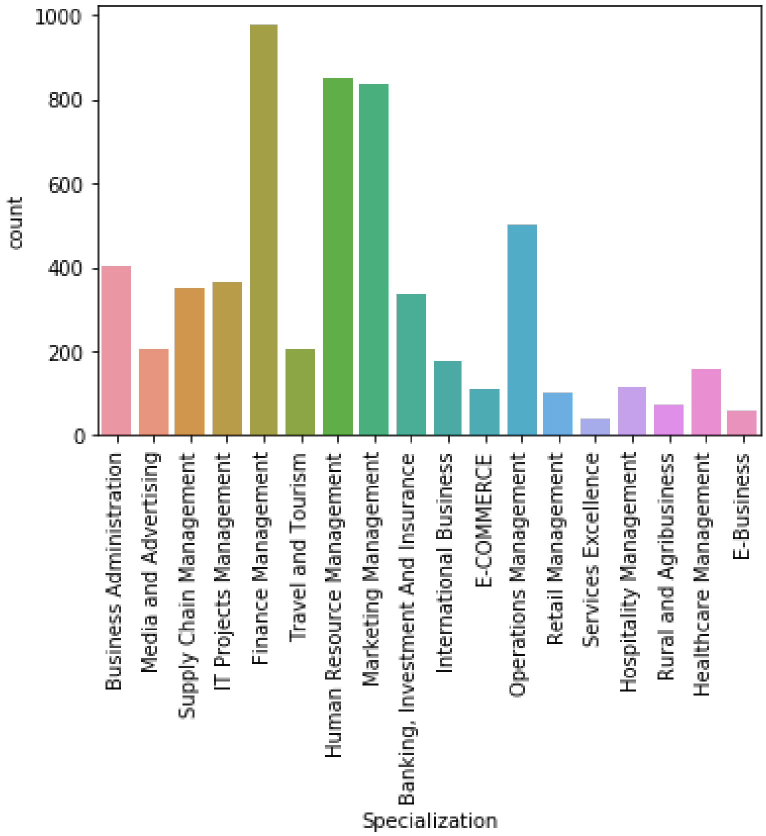
```
count                        5860
unique
18 top          Finance
Management freq
976
Name: Specialization, dtype: object
```

In [29]:

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

Out[29]:

```
(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>)
```



count
1000
800
600
400
200
0
Business Administration
Media and Advertising
Supply Chain Management
IT Projects Management
Finance Management
Travel and Tourism
Human Resource Management
Marketing Management
Banking, Investment And Insurance
International Business
E-COMMERCE
Operations Management
Retail Management
Services Excellence
Hospitality Management
Rural and Agribusiness
Healthcare Management
E-Business
Specialization

In

```
# It maybe the case that lead has not entered any specialization if his/her optio n is not availabe on the list, #  may not have any specialization or is a student.
# Hence we can make a category "Others" for missing values.
```

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

[30]: In [31]:

In [32]:
```
round(100*(data.isnull().sum()/len(data.index)), 2)
```

Out[32]:
```
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
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 [33]:
```
# Tags
```

In [34]:

```
data.Tags.describe()
```

Out[34]:

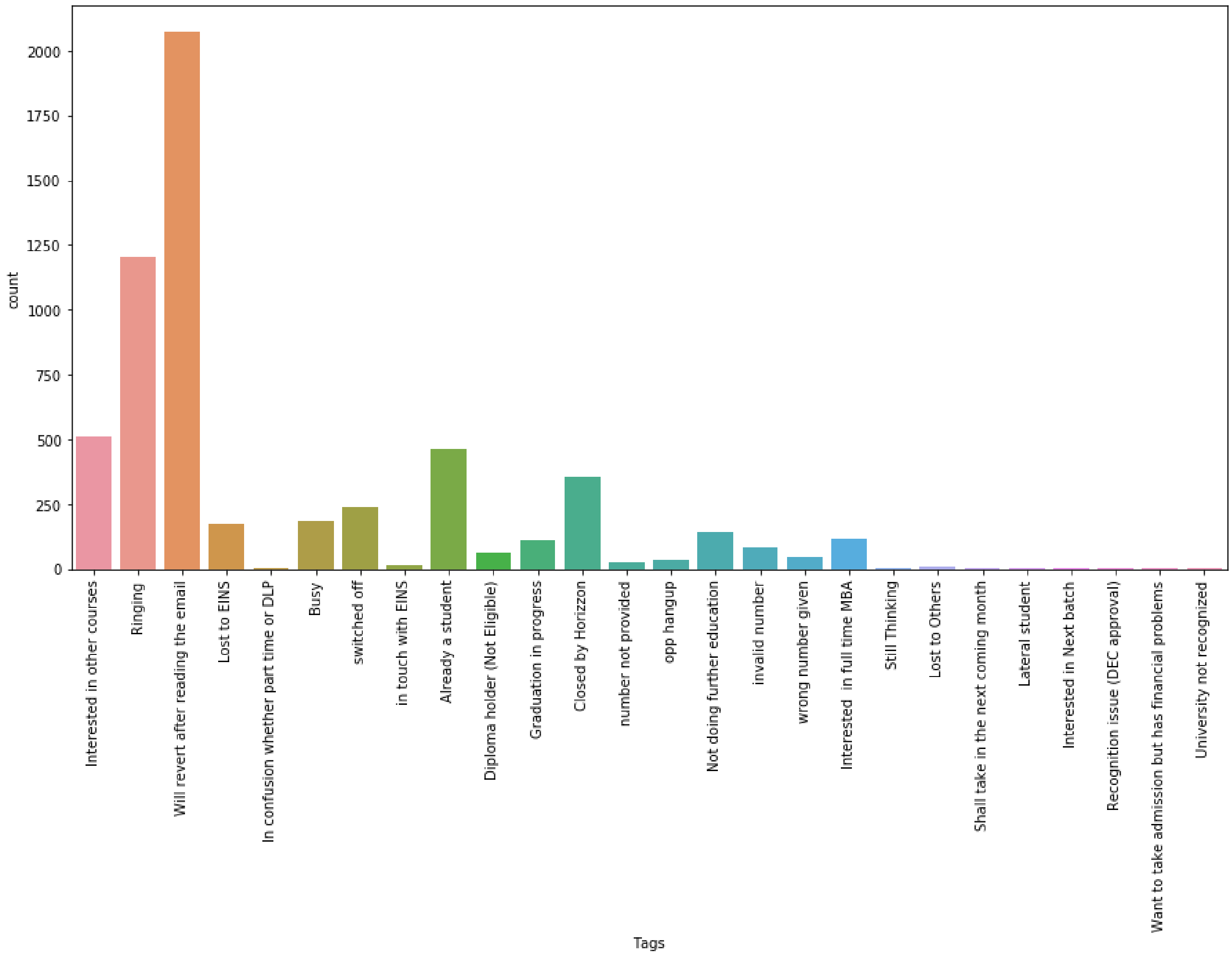
```
count                                   5887
unique                                    26
top         Will revert after reading the email
freq                                    2072
Name: Tags, dtype: object
```

In [35]:

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

Out[35]:

```
(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 [36]:

```
# Blanks in the tag column may be imputed by 'Will revert after reading the emai
l'.
```

In [37]:
In [38]:

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

In [39]:

```
# What matters most to you in choosing a course
```

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

Out[39]:

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

In [40]:

```
# Blanks in the this column may be imputed by 'Better Career Prospects'.
```

In [41]:

```
data['What matters most to you in choosing a course'] = data['What matters most t
o you in choosing a course'].replace(np.nan, 'Better Career Prospects')
```

In [42]:

```
# Occupation
```

In [43]:

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

Out[43]:

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

In [44]:

```
# 86% entries are of Unemployed so we can impute "Unemployed" in it.
```

In [45]:
In [46]:

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

In

```
# Country
```

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

[47]:

In [48]:

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

Out[48]:

```
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 [49]:

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

In [50]:
```
round(100*(data.isnull().sum()/len(data.index)), 2)
```

Out[50]:
```
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 [51]:
```
data.to_csv('Leads_cleaned')
```

Now Data is clean and we can start with the analysis part

Exploratory Data Analytics

Univariate Analysis

Converted

In [52]:
```
# Converted is the target variable, Indicates whether a lead has been successfull
y converted (1) or not (0).
```

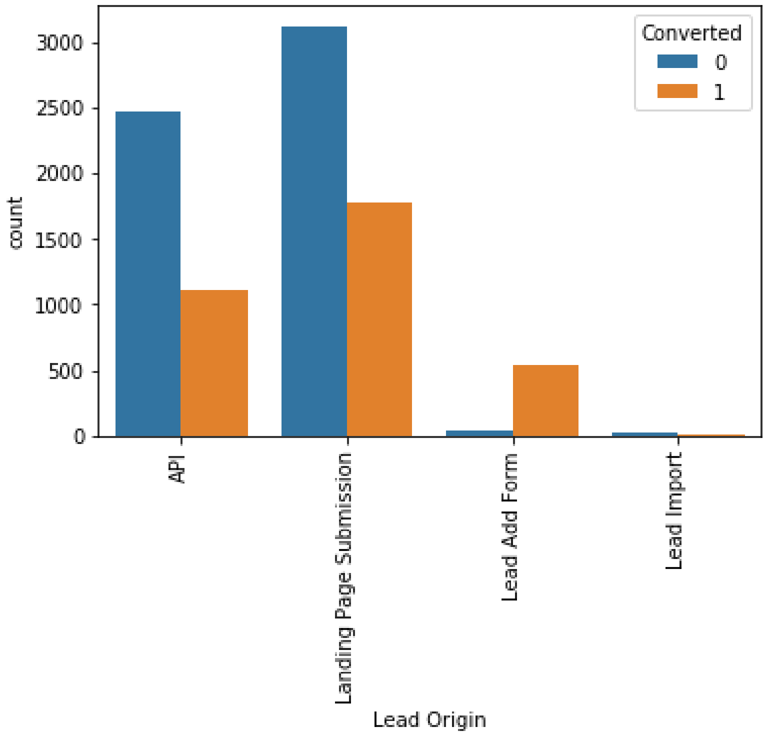
In [53]:
```
Converted = (sum(data['Converted'])/len(data['Converted'].index))*100
Converted
```

Out[53]:
```
37.85541106458012
```

Lead Origin

In [54]:
```
sns.countplot(x = "Lead Origin", hue = "Converted", data = data)
xticks(rotation = 90)
```

Out[54]:
```
(array([0, 1, 2, 3]), <a list of 4 Text xticklabel objects>)
```



Inference

1. API and Landing Page Submission have 30-35% conversion rate but count of lead originated from them are considerable.
2. Lead Add Form has more than 90% conversion rate but count of lead are not very high.
3. Lead Import are very less in count.

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.

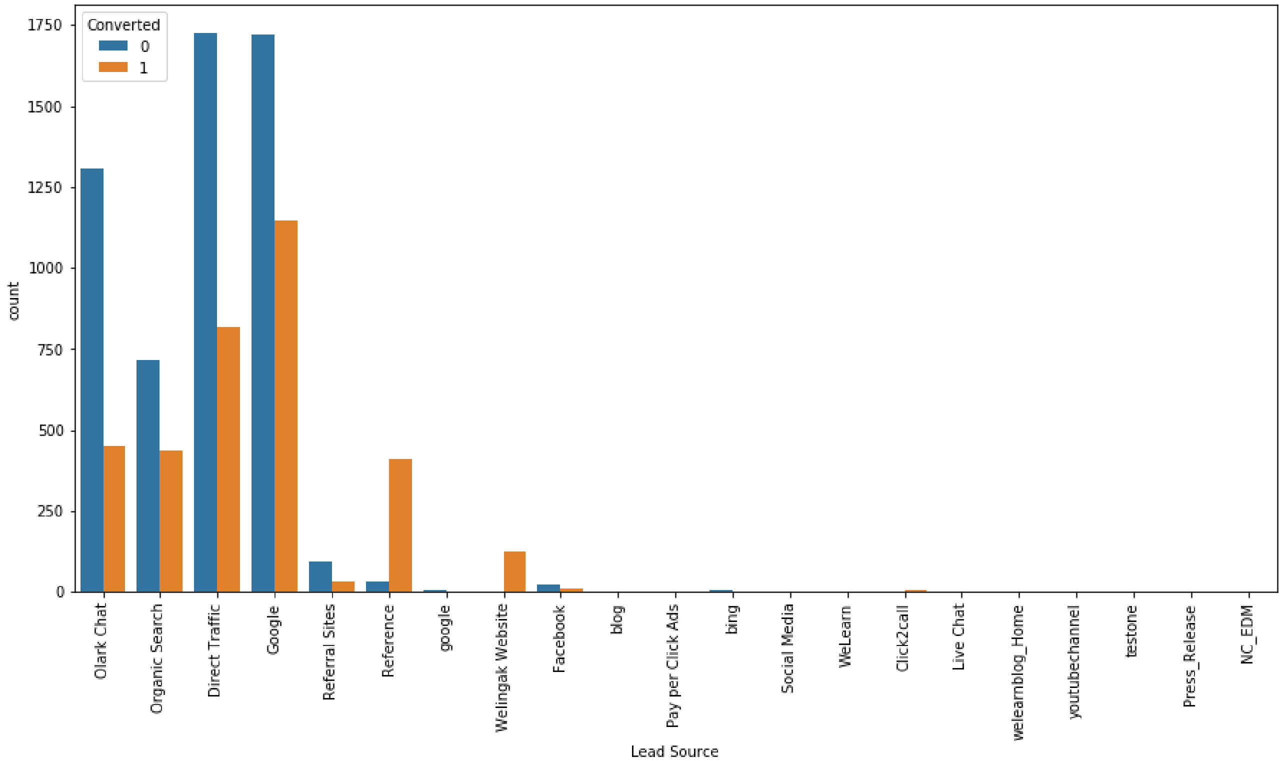
Lead Source

In [55]:

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

Out[55]:

```
(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 [56]:

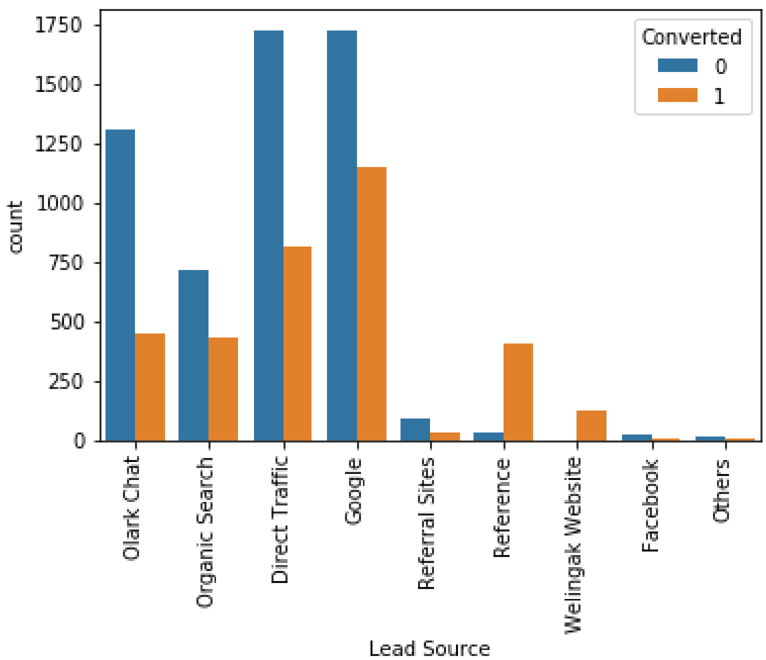
```
data['Lead Source'] = data['Lead Source'].replace(['google'], 'Google')
data['Lead Source'] = data['Lead Source'].replace(['Click2call', 'Live Chat',
'NC _EDM', 'Pay per Click Ads', 'Press_Release',
  'Social Media', 'WeLearn', 'bing', 'blog', 'testone', 'welearnblog_Home', 'yout
ubechannel'], 'Others')
```

In [57]:

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

Out[57]:

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



Inference

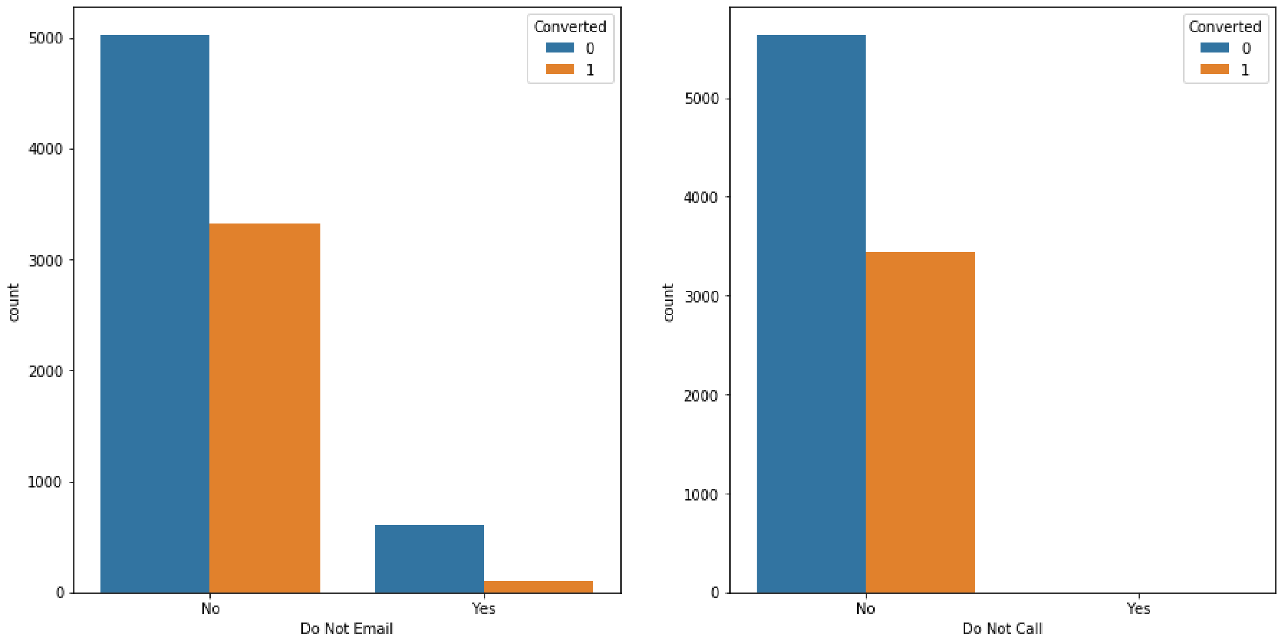
1. Google and Direct traffic generates maximum number of leads.
2. Conversion Rate of reference leads and leads through welingak website is high.

To improve overall lead conversion rate, focus should be on improving lead converion of olark chat, organic search, direct traffic, and google leads and generate more leads from reference and welingak website.

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 = data, ax =
axs[0]) sns.countplot(x = "Do Not Call", hue = "Converted", data = data, ax =
axs[1])
```

```
In [58]:
Out[58]:
<matplotlib.axes._subplots.AxesSubplot at 0x7f174c6ad0f0>
```



Total Visits

```
data['TotalVisits'].describe(percentiles=[0.05,.25, .5, .75, .90, .95, .99])
```

```
Out[59]:
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
```

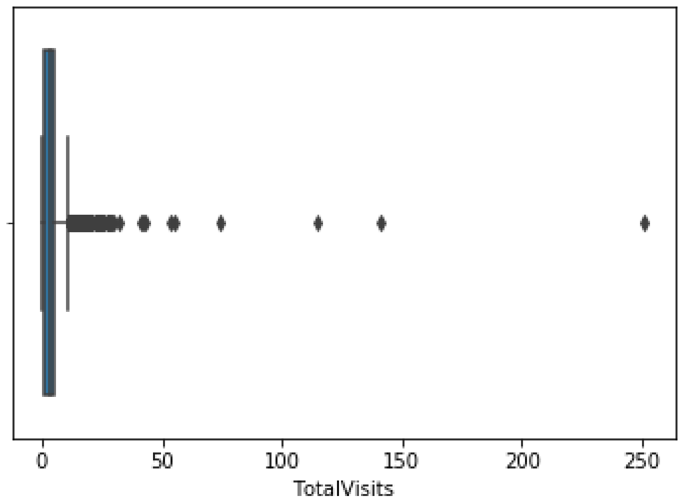
In [59]:

In [60]:

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

Out[60]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f174c62b0f0>
```



In [61]:

```
# As we can see there are a number of outliers in the data.
# We will cap the outliers to 95% value for analysis.
```

In [62]:

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

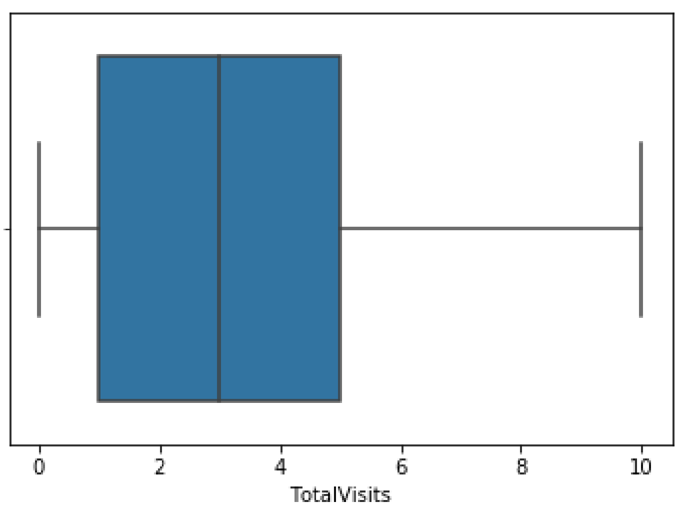
In [63]:

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

Out[63]:

```
&lt;matplotlib.axes._subplots.AxesSubplot at 0x7f174c5edf60&gt;
```

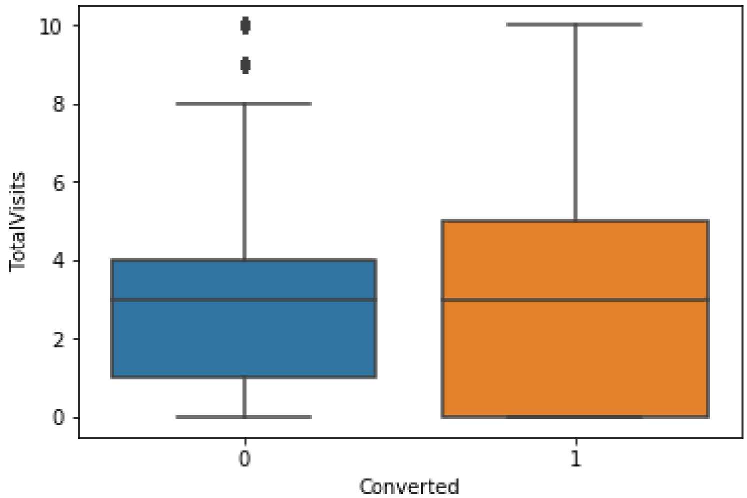
```
sns.boxplot(y = 'TotalVisits', x = 'Converted', data = data)
```



In [64]:

Out[64]:

```
&lt;matplotlib.axes._subplots.AxesSubplot at 0x7f174c5596d8&gt;
```



Inference

1. Median for converted and not converted leads are the same.

Nothng conclusive can be said on the basis of Total Visits.

Total time spent on website

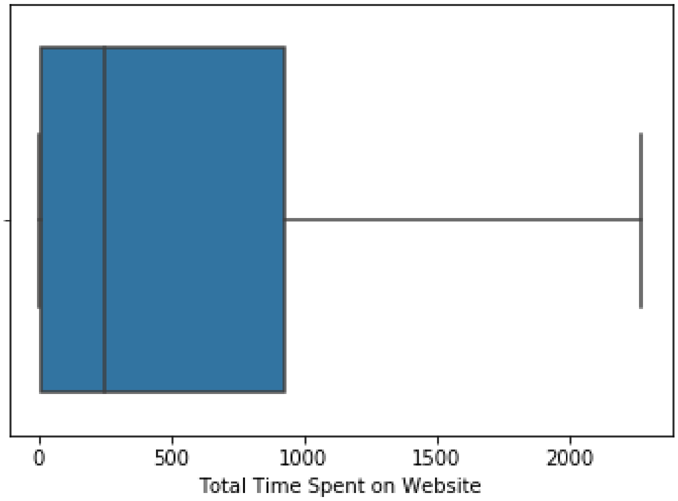
In [65]:
```
data['Total Time Spent on Website'].describe()
```

Out[65]:
```
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 [66]:
```
sns.boxplot(data['Total Time Spent on Website'])
```

Out[66]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f174c52d438>
```

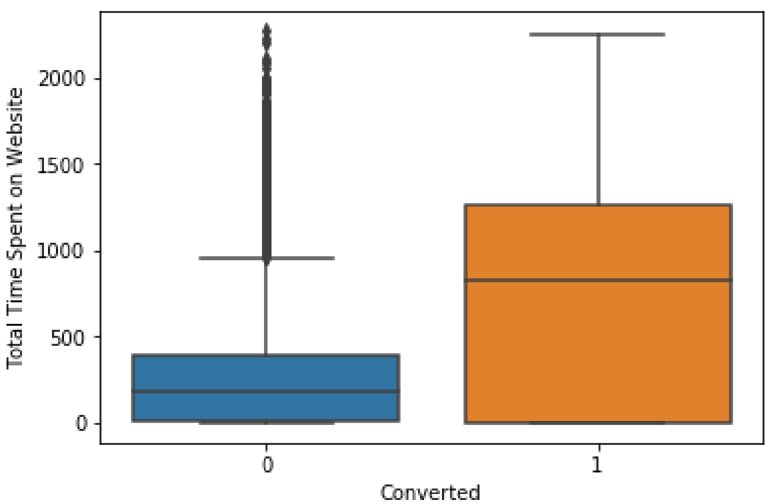


In [67]:

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

Out[67]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f174c582da0>
```



Inference

1. Leads spending more time on the weblise are more likely to be converted.

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

Page views per visit

In [68]:

```
data['Page Views Per Visit'].describe()
```

Out[68]:

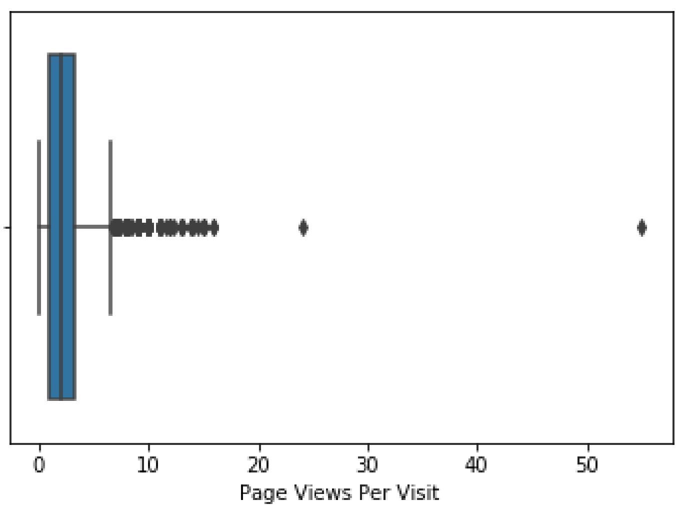
```
count    9074.000000
mean        2.370151
std         2.160871
min         0.000000
25%         1.000000
50%         2.000000
75%         3.200000
max        55.000000
Name: Page Views Per Visit, dtype: float64
```

In [69]:

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

Out[69]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f174c3fee10>
```



In [70]:

```
# As we can see there are a number of outliers in the data.
# We will cap the outliers to 95% value for analysis.
```

In [71]:

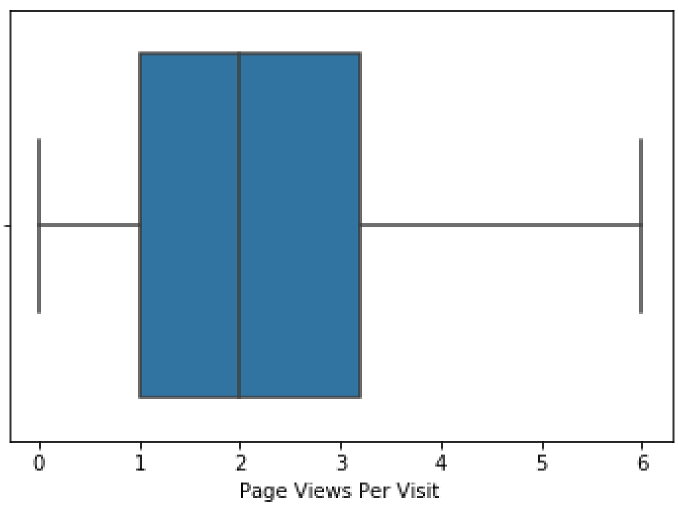
```
percentiles = data['Page Views Per Visit'].quantile([0.05,0.95]).values
data['Page Views Per Visit'][data['Page Views Per Visit'] <= percentiles[0]] =
pe rcentiles[0]
data['Page Views Per Visit'][data['Page Views Per Visit'] >= percentiles[1]] = pe
rcentiles[1]
```

In [72]:

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

Out[72]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f174c525828>
```

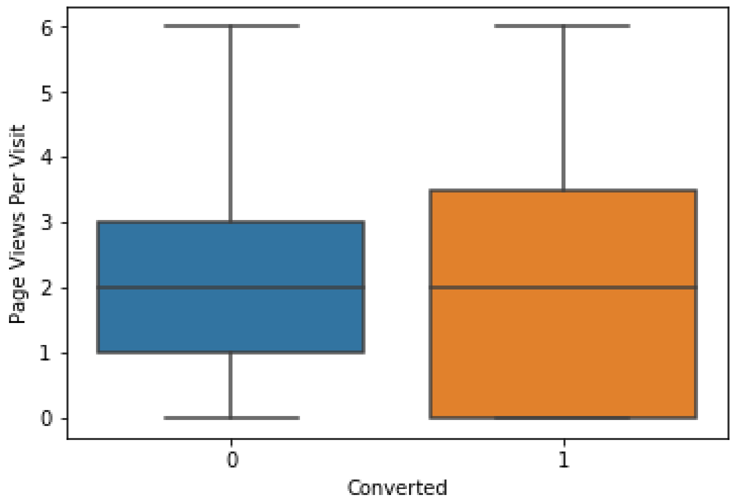


In [73]:

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

Out[73]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f174c338828>
```



Inference

1. Median for converted and unconverted leads is the same.

Nothing can be said specifically for lead conversion from Page Views Per Visit

Last Activity

In [74]:

```
data['Last Activity'].describe()
```

Out[74]:

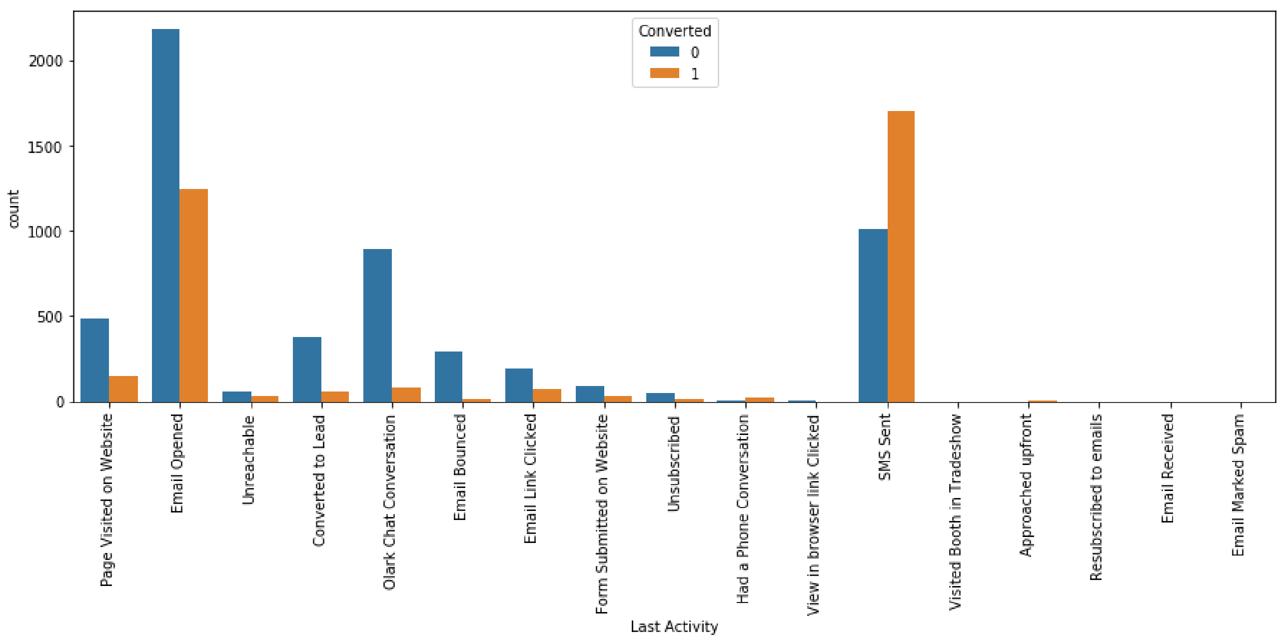
```
count                9074 unique
17 top         Email Opened freq
3432 Name: Last Activity, dtype:
object
```

In [75]:

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

Out[75]:

```
(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>)
```



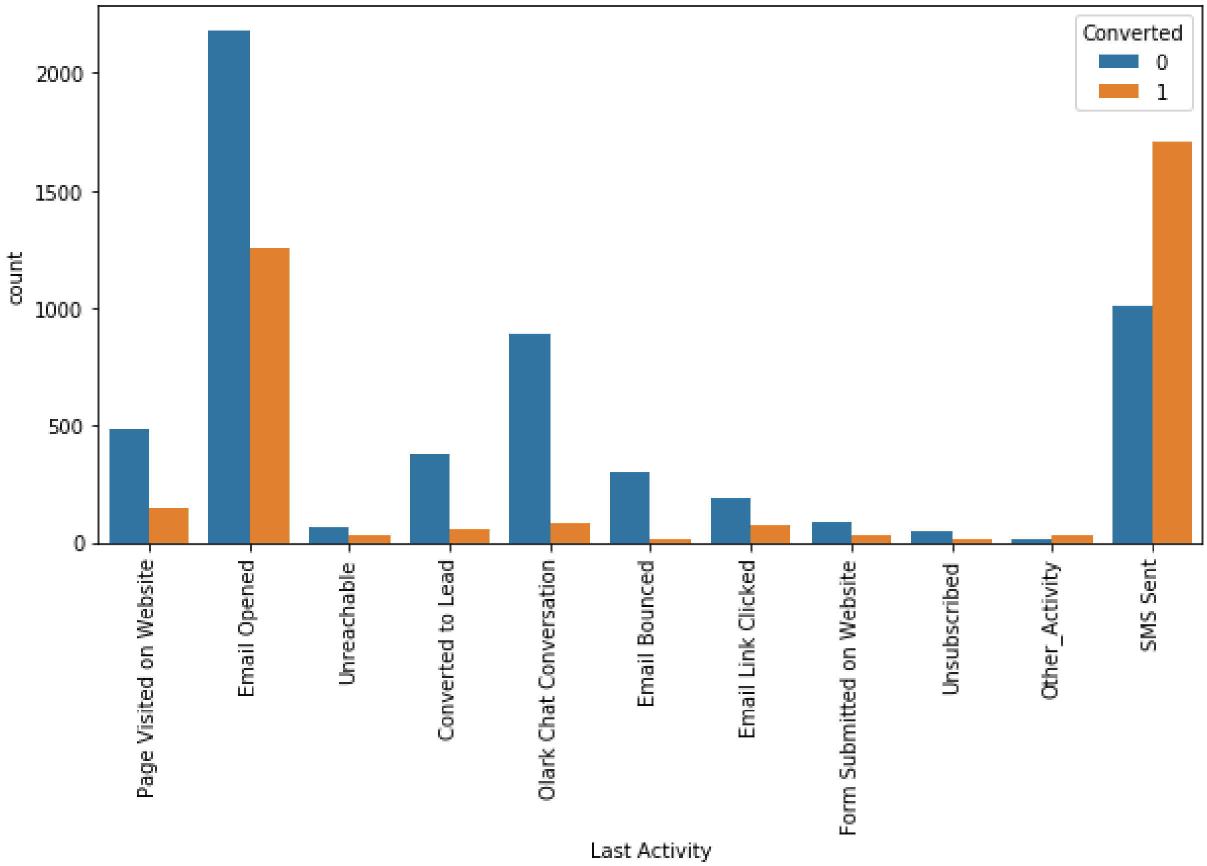
In [76]:
In [77]:

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

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

Out[77]:

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



Inference

1. Most of the lead have their Email opened as their last activity.
2. Conversion rate for leads with last activity as SMS Sent is almost 60%.b

Country

In [78]:

```
data.Country.describe()
```

Out[78]:

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

Inference

Most values are 'India' no such inference can be drawn

Specialization

In [79]:

```
data.Specialization.describe()
```

Out[79]:

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

In [80]:

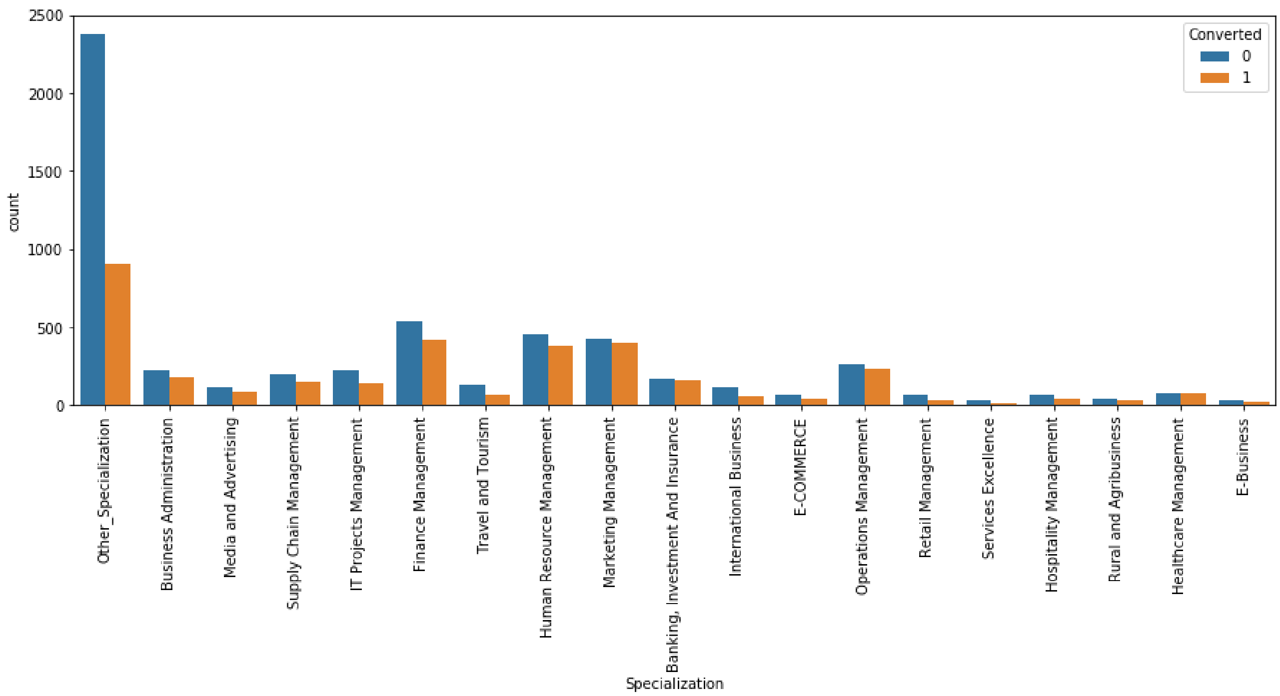
In [81]:

```
data['Specialization'] = data['Specialization'].replace(['Others'], 'Other_Specia
lization')
```

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

Out[81]:

```
(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>)
```



Inference

1. Focus should be more on the Specialization with high conversion rate.

Occupation

In [82]:
```
data['What is your current occupation'].describe()
```

Out[82]:
```
count          9074
unique            6
top      Unemployed
freq           8159
Name: What is your current occupation, dtype: object
```

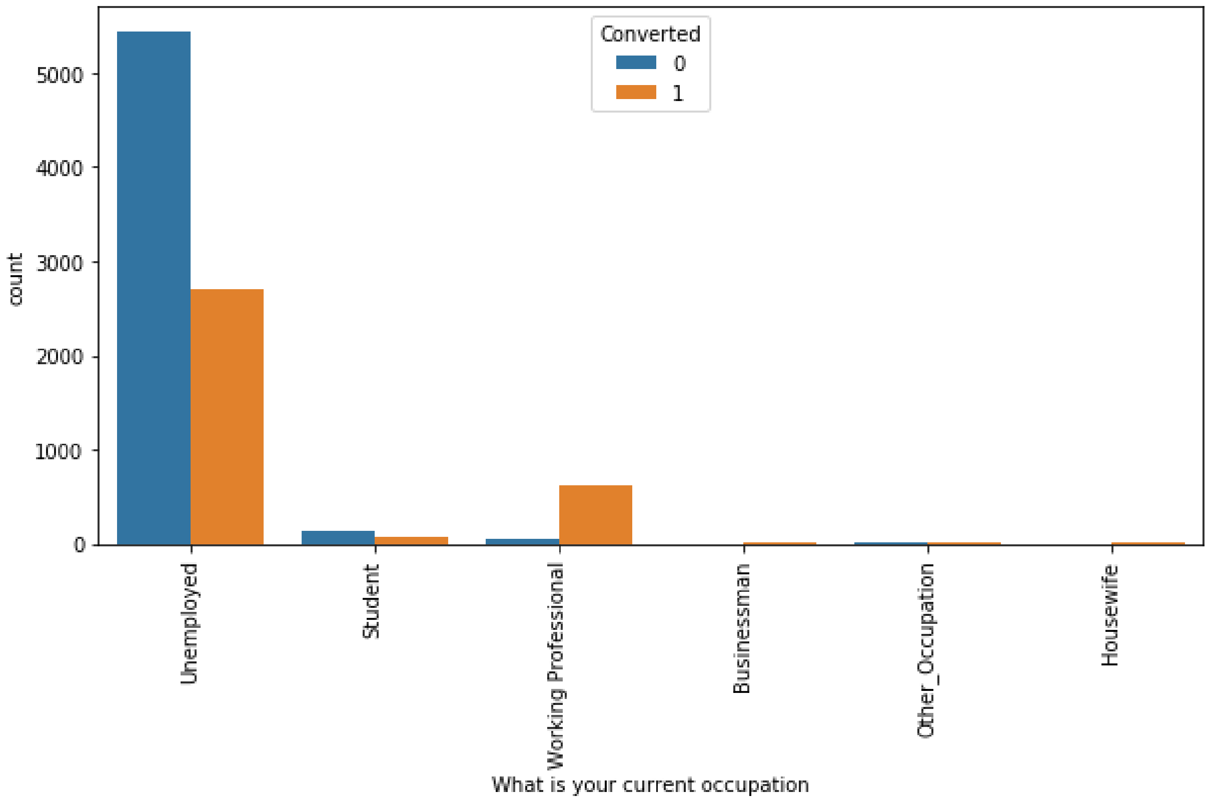
In [83]:
```
data['What is your current occupation'] = data['What is your current occupation']
.replace(['Other'], 'Other_Occupation')
```

In [84]:

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

Out[84]:

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



Inference

1. Working Professionals going for the course have high chances of joining it.
2. Unemployed leads are the most in numbers but has around 30-35% conversion rate.

What matters most to you in choosing a course

In [85]:

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

Out[85]:

```
count                    9074
unique
3 top       Better Career
```

```
Prospects freq
9072
Name: What matters most to you in choosing a course, dtype: object
```

Inference

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

Search

In [86]:

```
data.Search.describe()
```

Out[86]:

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

Inference

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

Magazine

In [87]:

```
data.Magazine.describe()
```

Out[87]:

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

Inference

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

Newspaper Article

In [88]:

```
data['Newspaper Article'].describe()
```

Out[88]:

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

Inference

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

X Education Forums

In [89]:
```
data[
      'X Education Forums'].describe()
```

Out[89]:

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

Inference

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

Newspaper

In [90]:
```
data['Newspaper'].describe()
```

Out[90]:

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

Inference

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

Digital Advertisement

In [91]:

```
data['Digital Advertisement'].describe()
```

Out[91]:

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

Inference

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

Through Recommendations

In [92]:

```
data['Through Recommendations'].describe()
```

Out[92]:

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

Inference

Most entries are

```
data[
```

'No'. No Inference can be drawn with this parameter.

Receive More Updates About Our Courses

In [93]:

```
'Receive More Updates About Our Courses'].describe()
```

Out[93]:

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

Inference

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

Tags

In [94]:

```
data.Tags.describe()
```

Out[94]:

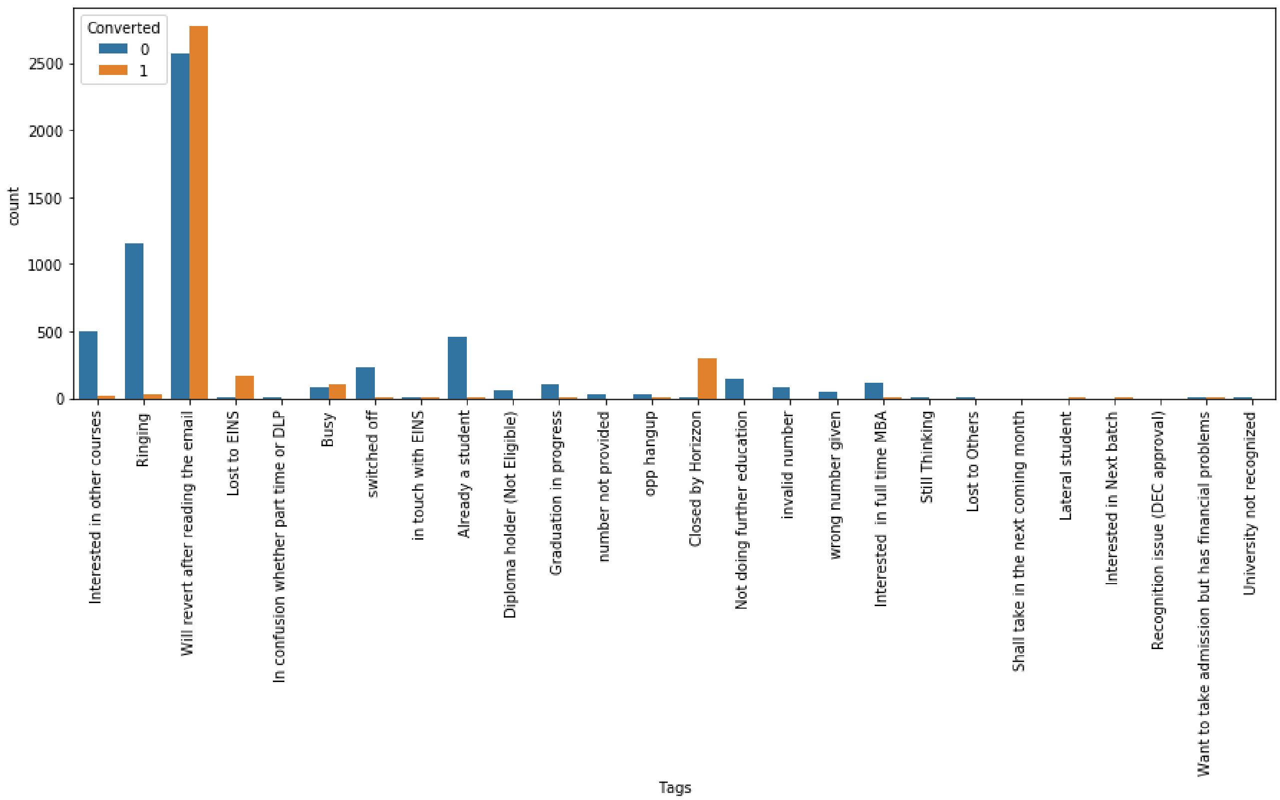
```
count                                    9074
unique                                     26
top         Will revert after reading the email
freq                                     5343
Name: Tags, dtype: object
```

In [95]:

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

Out[95]:

```
(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>)
```



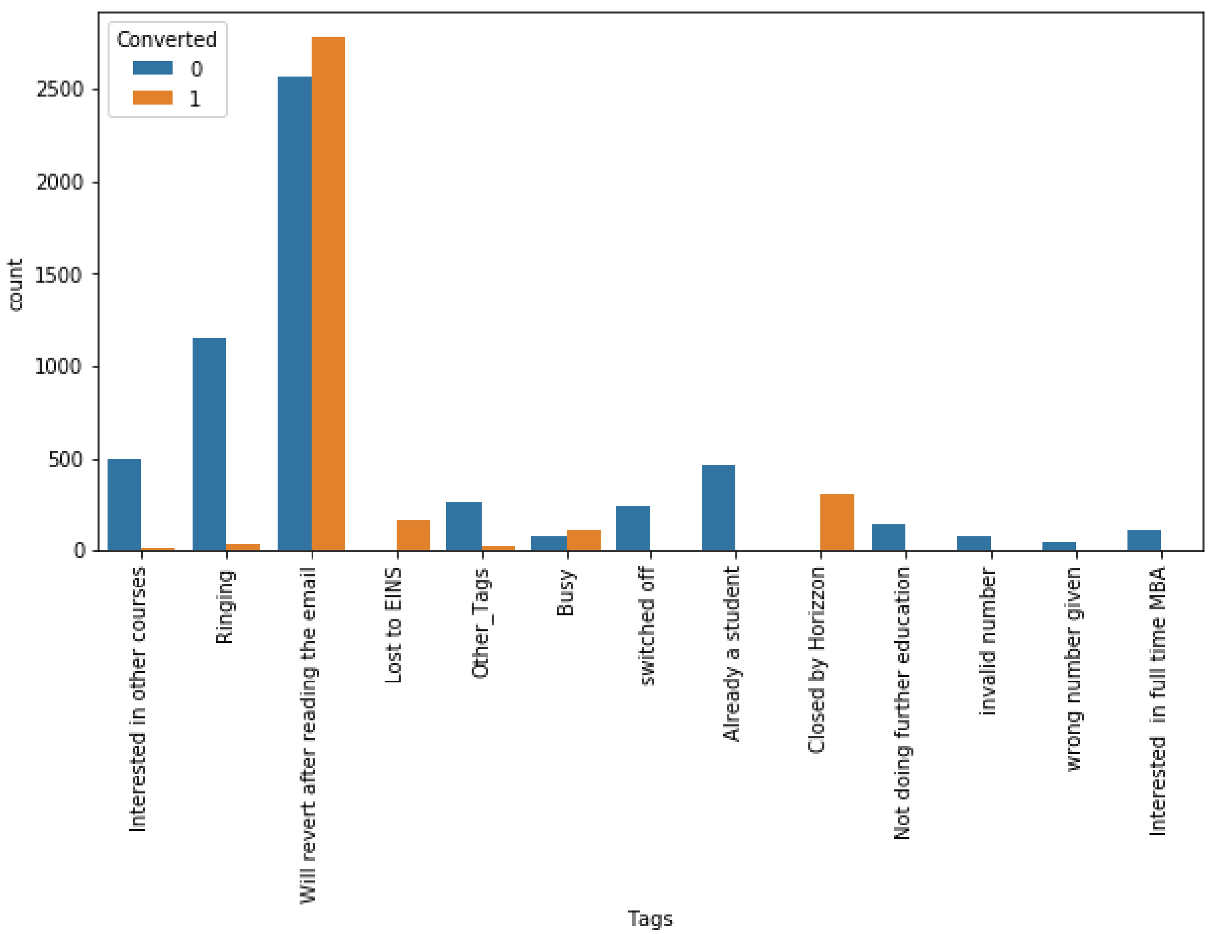
In [96]:
In [97]:

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

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

Out[97]:

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



Inference

```
data[
```

Lead Quality

In [98]:

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

Out[98]:

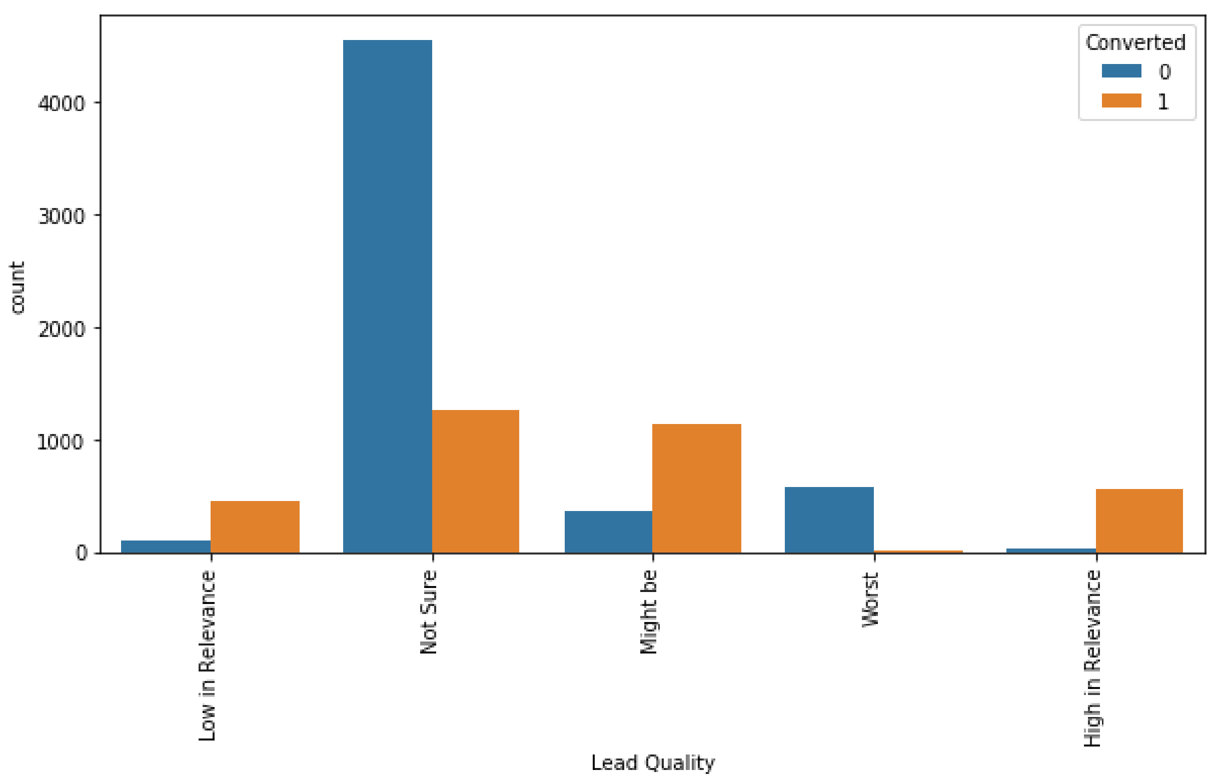
```
count           9074
unique             5
top         Not Sure
freq            5806
Name: Lead Quality, dtype: object
```

In [99]:

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

Out[99]:

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



Update me on Supply Chain Content

In [100]:
```python
data['Update me on Supply Chain Content'].describe()
```

Out[100]:
```
count      9074
unique        1
top          No
freq       9074
Name: Update me on Supply Chain Content, dtype: object
```

Inference

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

Get updates on DM Content

In [101]:
```python
data['Get updates on DM Content'].describe()
```

Out[101]:
```
count      9074
unique        1
top          No
freq       9074
Name: Get updates on DM Content, dtype: object
```

Inference

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

I agree to pay the amount through cheque

In [102]:

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

Out[102]:

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

Inference

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

A free copy of Mastering The Interview

In [103]:

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

Out[103]:

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

Inference

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

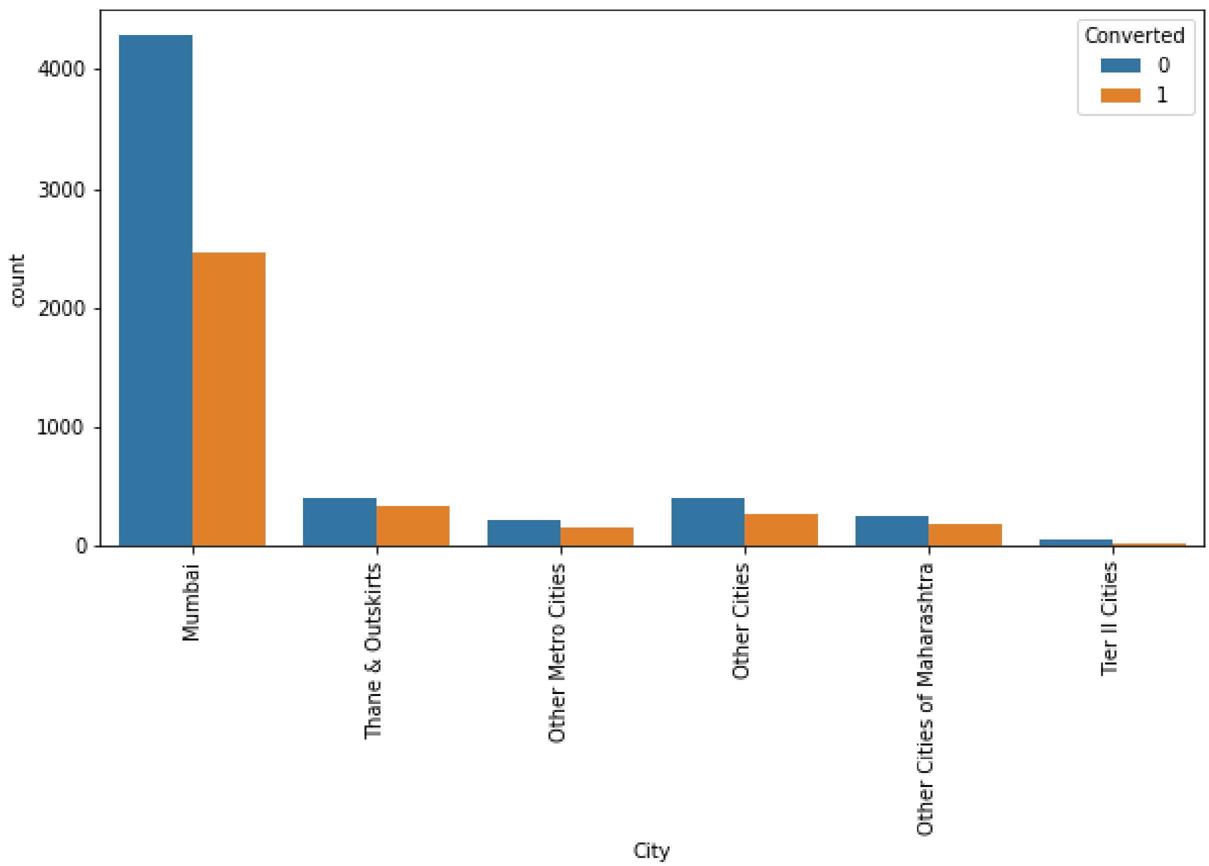
City

In [104]:
```
data[
data.City.describe(
)
```

Out[104]:
```
count        9074 unique
6 top        Mumbai freq
6752 Name: City, dtype:
object
```

In [105]:
```
fig, axs = plt.subplots(figsize = (10,5))
sns.countplot(x = "City", hue = "Converted", data = data)
xticks(rotation = 90)
```

Out[105]:
```
(array([0, 1, 2, 3, 4, 5]), <a list of 6 Text xticklabel objects>)
```



Inference

Most leads are from mumbai with around 30% conversion rate.

Last Notable Activity

In [106]:

```
data['Last Notable Activity'].describe()
```

Out[106]:

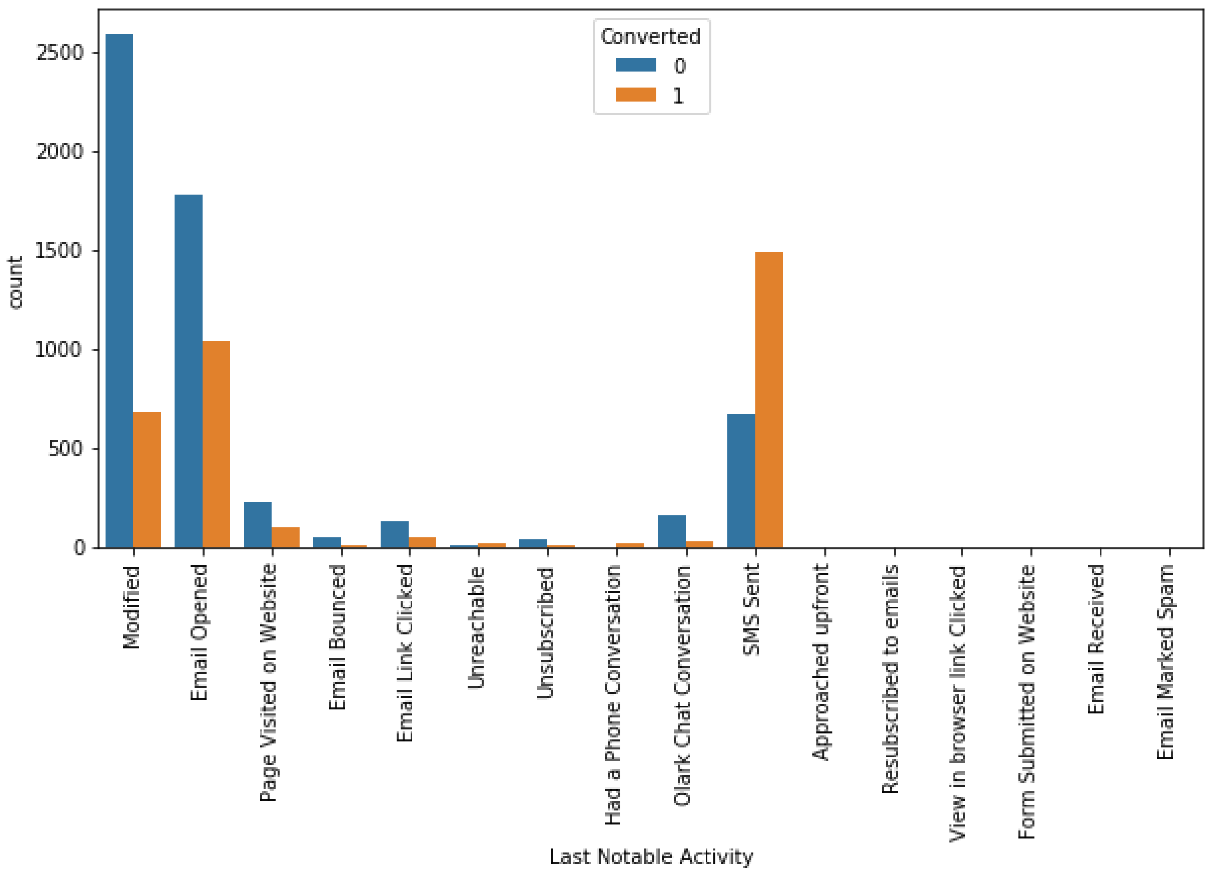
```
count         9074
unique          16
top       Modified
freq          3267
Name: Last Notable Activity, dtype: object
```

In [107]:

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

Out[107]:

```
(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 [108]:

```
data = 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 Update s About
Our Courses','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 [109]:

```
data.shape
```

Out[109]:

```
(9074, 16)
```

In [110]:

```
data.head()
```

Out[110]:

	Prospect ID	Lead Origin	Lead Source	Do Not Email	Do Not Call	Converted	TotalVisits	Total Time Spent on Website	Page Views Per Visit	Last Act
0	7927b2df8bba-4d29b9a2b6e0beafe620	API	Olark Chat	No	No	0	0.0	0	0.0	Page Vi on Web
1	2a2724365132-413686fadcc88c88f482	API	Organic Search	No	No	0	5.0	674	2.5	Email Opened
2	8cc8c611a219-4f35ad23fdfd2656bd8a	Landing Page Submission	Direct Traffic	No	No	1	2.0	1532	2.0	Email Opened
3	0cc2df48-7cf4-4e39-9de919797f9b38cc	Landing Page Submission	Direct Traffic	No	No	0	1.0	305	1.0	Unreach
4	3256f628e534-4826-9d63-4a8b88782852	Landing Page Submission	Google	No	No	1	2.0	1428	1.0	Converte Lead

Data Preparation

Converting some binary variables (Yes/No) to 1/0

In [111]:

```
# 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
data[varlist] =
data[varlist].apply(binary_map)
```

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

In [112]:
Out[112]:

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

	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_O Search
0	0	0	0	0	0	1	0
1	0	0	0	0	0	0	1
2	1	0	0	0	0	0	0
3	1	0	0	0	0	0	0
4	1	0	0	0	1	0	0

In [113]:

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

Out[113]:

	Prospect ID	Lead Origin	Lead Source	Do Not Email	Do Not Call	Converted	TotalVisits	Total Time Spent on Website	Page Views Per Visit	Last Act
0	7927b2df8bba-4d29b9a2b6e0beafe620	API	Olark Chat	0	0	0	0.0	0	0.0	Page Vi on Web
1	2a2724365132-413686fadcc88c88f482	API	Organic Search	0	0	0	5.0	674	2.5	Email Opened
2	8cc8c611a219-4f35ad23fdfd2656bd8a	Landing Page Submission	Direct Traffic	0	0	1	2.0	1532	2.0	Email Opened
3	0cc2df48-7cf4-4e39-9de919797f9b38cc	Landing Page Submission	Direct Traffic	0	0	0	1.0	305	1.0	Unreach
4	3256f628e534-4826-9d63-4a8b88782852	Landing Page Submission	Google	0	0	1	2.0	1428	1.0	Converte Lead

In [114]:
In [115]:

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

```
data.head()
```

Out[115]:

	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
0	7927b2df8bba-4d29b9a2b6e0beafe620	0	0	0	0.0	0	0.0	0	0
1	2a2724365132-413686fadcc88c88f482	0	0	0	5.0	674	2.5	0	0
2	8cc8c611a219-4f35ad23fdfd2656bd8a	0	0	1	2.0	1532	2.0	1	0
3	0cc2df48-7cf4-4e39-9de919797f9b38cc	0	0	0	1.0	305	1.0	1	0
4	3256f628e534-4826-9d63-4a8b88782852	0	0	1	2.0	1428	1.0	1	0

In [116]:
In [117]:

```
from sklearn.model_selection import train_test_split

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

```
X.head()
```

Out[117]:

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

y.head()
```

	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_Faceboo
0	0	0	0.0	0	0.0	0	0	0	0
1	0	0	5.0	674	2.5	0	0	0	0
2	0	0	2.0	1532	2.0	1	0	0	0
3	0	0	1.0	305	1.0	1	0	0	0
4	0	0	2.0	1428	1.0	1	0	0	0

In [118]:

Out[118]:

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

In [119]:

```
# 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_si
ze=0.3, random_state=100)
```

Step 5: Feature Scaling

In [120]:

```
from sklearn.preprocessing import StandardScaler
 scaler =
StandardScaler()

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

X_train.head()
```

```
/opt/conda/lib/python3.6/site-packages/sklearn/preprocessing/data.py:645:
DataCo nversionWarning: Data with input dtype int64, float64 were all converted
to floa t64 by StandardScaler.
  return self.partial_fit(X, y)
/opt/conda/lib/python3.6/site-packages/sklearn/base.py:464:
DataConversionWarnin g: Data with input dtype int64, float64 were all converted
to float64 by Standar dScaler.  return self.fit(X, **fit_params).transform(X)
```

Out[120]:

	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

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

3009	0	0	-0.432779	-0.160255	-0.155018	1	0	0	0
1012	1	0	-0.432779	-0.540048	-0.155018	1	0	0	0
9226	0	0	-1.150329	-0.888650	-1.265540	0	0	0	0
4750	0	0	-0.432779	1.643304	-0.155018	1	0	0	0
7987	0	0	0.643547	2.017593	0.122613	1	0	0	0

In [121]:

Out[121]:

```
37.85541106458012
```

We have almost 38% conversion

Model Building

Running Your First Training Model

In [122]:

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

In [123]:

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

Out[123]:

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:	Fri, 15 Mar 2019	Deviance:	2500.0
Time:	06:22:51	Pearson chi2:	3.87e+04
No. Iterations:	24	Covariance Type:	nonrobust

	coef	std err	z	P>|z|	[0.025	0.975]
const	23.1427	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	-20.1486	2.16e+05	-9.33e-05	1.000	-4.23e+05	4.23e+05
Last Notable Activity_Email Link Clicked	-23.1906	2.16e+05	-0.000	1.000	-4.23e+05	4.23e+05
Last Notable Activity_Email Marked Spam	0.5185	2.56e+05	2.02e-06	1.000	-5.02e+05	5.02e+05

Last Notable Activity_Email Opened	-21.4922	2.16e+05	-9.95e-05	1.000	-4.23e+05	4.23e+05
Last Notable Activity_Email Received	-1.9842	3.05e+05	-6.5e-06	1.000	-5.99e+05	5.99e+05
Last Notable Activity_Form Submitted on Website	-45.2742	3.05e+05	-0.000	1.000	-5.99e+05	5.99e+05
Last Notable Activity_Had a Phone Conversation	-21.6359	2.16e+05	-0.000	1.000	-4.23e+05	4.23e+05
Last Notable Activity_Modified	-22.7333	2.16e+05	-0.000	1.000	-4.23e+05	4.23e+05
Last Notable Activity_Olark Chat Conversation	-22.7851	2.16e+05	-0.000	1.000	-4.23e+05	4.23e+05
Last Notable Activity_Page Visited on Website	-22.5732	2.16e+05	-0.000	1.000	-4.23e+05	4.23e+05
Last Notable Activity_Resubscribed to emails	-1.7957	3.05e+05	-5.88e-06	1.000	-5.99e+05	5.99e+05
Last Notable Activity_SMS Sent	-20.2280	2.16e+05	-9.37e-05	1.000	-4.23e+05	4.23e+05
Last Notable Activity_Unreachable	-21.1492	2.16e+05	-9.79e-05	1.000	-4.23e+05	4.23e+05
Last Notable Activity_Unsubscribed	-21.3610	2.16e+05	-9.89e-05	1.000	-4.23e+05	4.23e+05
Last Notable Activity_View in browser link Clicked	-43.1071	3.05e+05	-0.000	1.000	-5.99e+05	5.99e+05

Feature Selection Using RFE

In [124]:

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

In [125]:

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

```
rfe.support_
```

Out[125]:

```
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, False,  True,  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 [126]:

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

Out[126]:

```
[('Do Not Email', True, 1),
 ('Do Not Call', False, 33),
 ('TotalVisits', False, 43),
 ('Total Time Spent on Website', False, 3),
 ('Page Views Per Visit', False, 40),
 ('Lead Origin_Landing Page Submission', False, 16),
 ('Lead Origin_Lead Add Form', True, 1),
 ('Lead Origin_Lead Import', False, 2),
 ('Lead Source_Facebook', False, 49),
 ('Lead Source_Google', False, 38),
 ('Lead Source_Olark Chat', False, 5),
 ('Lead Source_Organic Search', False, 39),
 ('Lead Source_Others', False, 46),
 ('Lead Source_Reference', False, 70),
 ('Lead Source_Referral Sites', False, 53),
 ('Lead Source_Welingak Website', True, 1),
 ('Last Activity_Email Bounced', False, 28),
 ('Last Activity_Email Link Clicked', False, 36),
 ('Last Activity_Email Opened', False, 60),
 ('Last Activity_Form Submitted on Website', False, 66),
 ('Last Activity_Olark Chat Conversation', False, 13),
('Last Activity_Other_Activity', False, 8),
 ('Last Activity_Page Visited on Website', False, 37),
 ('Last Activity_SMS Sent', False, 7),
 ('Last Activity_Unreachable', False, 14),
 ('Last Activity_Unsubscribed', False, 17),
 ('Specialization_Business Administration', False, 61),
 ('Specialization_E-Business', False, 67),
 ('Specialization_E-COMMERCE', False, 15),
 ('Specialization_Finance Management', False, 44),
 ('Specialization_Healthcare Management', False, 42),
 ('Specialization_Hospitality Management', False, 62),
 ('Specialization_Human Resource Management', False, 57),
 ('Specialization_IT Projects Management', False, 47),
 ('Specialization_International Business', False, 21),
 ('Specialization_Marketing Management', False, 32),
 ('Specialization_Media and Advertising', False, 34),
 ('Specialization_Operations Management', False, 69),
 ('Specialization_Other_Specialization', False, 20),
 ('Specialization_Retail Management', False, 59),
```

```
 ('Specialization_Rural and Agribusiness', False, 45),
 ('Specialization_Services Excellence', False, 56),  ('Specialization_Supply
Chain Management', False, 50),
 ('Specialization_Travel and Tourism', False, 24),
 ('What is your current occupation_Housewife', False, 41),
 ('What is your current occupation_Other_Occupation', False, 26),
 ('What is your current occupation_Student', False, 35),
 ('What is your current occupation_Unemployed', False, 19),
 ('What is your current occupation_Working Professional', True, 1),
 ('Tags_Busy', True, 1),
 ('Tags_Closed by Horizzon', True, 1),
 ('Tags_Interested  in full time MBA', False, 18),
 ('Tags_Interested in other courses', False, 10),
('Tags_Lost to EINS', True, 1),
 ('Tags_Not doing further education', False, 11),
 ('Tags_Other_Tags', False, 30),
 ('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, 63),
 ('Lead Quality_Might be', False, 9),
 ('Lead Quality_Not Sure', True, 1),
 ('Lead Quality_Worst', True, 1),
 ('City_Other Cities', False, 51),
 ('City_Other Cities of Maharashtra', False, 68),
 ('City_Other Metro Cities', False, 58),
 ('City_Thane & Outskirts', False, 52),
 ('City_Tier II Cities', False, 23),
 ('Last Notable Activity_Email Bounced', False, 25),
 ('Last Notable Activity_Email Link Clicked', False, 12),
 ('Last Notable Activity_Email Marked Spam', False, 54),
 ('Last Notable Activity_Email Opened', False, 48),
 ('Last Notable Activity_Email Received', False, 71),
 ('Last Notable Activity_Form Submitted on Website', False, 55),
 ('Last Notable Activity_Had a Phone Conversation', False, 29),
 ('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, 27),
 ('Last Notable Activity_Unsubscribed', False, 31),
 ('Last Notable Activity_View in browser link Clicked', False, 64)]
```

In [127]:

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

Out[127]:

```
Index(['Do Not Email', 'Lead Origin_Lead Add Form',
       'Lead Source_Welingak Website',
       'What is your current occupation_Working Professional', '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 [128]:

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

Out[128]:

```
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_Unemployed',
       '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 [129]:

```
X_train_sm =
logm2 =       sm.families.Binomial()) res = res.summary()

                  sm.add_constant(X_train[col]
                  )
        sm.GLM(y_train,X_train_sm, family =
    logm2.fit()
```

Out[129]:

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:	-1580.6
Date:	Fri, 15 Mar 2019	Deviance:	3161.3
Time:	06:22:54	Pearson chi2:	3.11e+04
No. Iterations:	24	Covariance Type:	nonrobust

	coef	std err	z	P>|z|	[0.025	0.975]
const	-1.8547	0.215	-8.636	0.000	-2.276	-1.434
Do Not Email	-1.3106	0.213	-6.154	0.000	-1.728	-0.893
Lead Origin_Lead Add Form	1.0452	0.360	2.900	0.004	0.339	1.752
Lead Source_Welingak Website	3.4638	0.817	4.238	0.000	1.862	5.066
What is your current occupation_Working Professional	1.2843	0.287	4.476	0.000	0.722	1.847
Tags_Busy	3.5477	0.332	10.680	0.000	2.897	4.199
Tags_Closed by Horizzon	7.7377	0.762	10.152	0.000	6.244	9.231
Tags_Lost to EINS	8.9540	0.753	11.887	0.000	7.478	10.430
Tags_Ringing	-1.9696	0.340	-5.800	0.000	-2.635	-1.304
Tags_Will revert after reading the email	3.7332	0.228	16.340	0.000	3.285	4.181

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

Tags_invalid number	-23.4649	2.21e+04	-0.001	0.999	-4.34e+04	4.33e+04
Tags_switched off	-2.5711	0.589	-4.367	0.000	-3.725	-1.417
Tags_wrong number given	-23.0779	3.17e+04	-0.001	0.999	-6.21e+04	6.2e+04
Lead Quality_Not Sure	-3.3496	0.129	-26.033	0.000	-3.602	-3.097
Lead Quality_Worst	-3.7672	0.848	-4.445	0.000	-5.428	-2.106
Last Notable Activity_SMS Sent	2.7931	0.122	22.838	0.000	2.553	3.033

In [130]:

In [131]:

```
col1
```

Out[131]:

```
Index(['Do Not Email', 'Lead Origin_Lead Add Form',
       'Lead Source_Welingak Website',
       'What is your current occupation_Working Professional',
'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 [132]:

```
X_train_sm =
logm2 =       sm.families.Binomial()) res = res.summary()

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

Out[132]:

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:	-1586.7
Date:	Fri, 15 Mar 2019	Deviance:	3173.3
Time:	06:22:54	Pearson chi2:	3.07e+04
No. Iterations:	22	Covariance Type:	nonrobust

	coef	std err	z	P>|z|	[0.025	0.975]
const	-2.0195	0.217	-9.308	0.000	-2.445	-1.594
Do Not Email	-1.3018	0.212	-6.130	0.000	-1.718	-0.886
Lead Origin_Lead Add Form	1.0769	0.362	2.974	0.003	0.367	1.787
Lead Source_Welingak Website	3.4268	0.818	4.190	0.000	1.824	5.030
What is your current occupation_Working Professional	1.3240	0.290	4.567	0.000	0.756	1.892
Tags_Busy	3.7300	0.331	11.270	0.000	3.081	4.379
Tags_Closed by Horizzon	7.8904	0.763	10.345	0.000	6.396	9.385
Tags_Lost to EINS	9.1124	0.754	12.086	0.000	7.635	10.590
Tags_Ringing	-1.7713	0.338	-5.244	0.000	-2.433	-1.109
Tags_Will revert after reading the email	3.8970	0.230	16.954	0.000	3.446	4.348
Tags_switched off	-2.3666	0.588	-4.028	0.000	-3.518	-1.215

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

Tags_wrong number given	-20.8825	1.17e+04	-0.002	0.999	-2.29e+04	2.28e+04
Lead Quality_Not Sure	-3.3417	0.128	-26.020	0.000	-3.593	-3.090
Lead Quality_Worst	-3.7822	0.848	-4.462	0.000	-5.444	-2.121
Last Notable Activity_SMS Sent	2.7503	0.120	22.841	0.000	2.514	2.986

In [133]:

In [134]:

```
col2
```

Out[134]:

```
Index(['Do Not Email', 'Lead Origin_Lead Add Form',
       'Lead Source_Welingak Website',
       'What is your current occupation_Working Professional',
'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 [135]:

```
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[135]:

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:	-1588.8
Date:	Fri, 15 Mar 2019	Deviance:	3177.6
Time:	06:22:54	Pearson chi2:	3.08e+04
No. Iterations:	8	Covariance Type:	nonrobust

	coef	std err	z	P>|z|	[0.025	0.975]
const	-2.0888	0.216	-9.654	0.000	-2.513	-1.665
Do Not Email	-1.3012	0.212	-6.134	0.000	-1.717	-0.885
Lead Origin_Lead Add Form	1.0894	0.363	3.001	0.003	0.378	1.801
Lead Source_Welingak Website	3.4138	0.818	4.173	0.000	1.810	5.017

In [136]:

What is your current occupation_Working Professional	1.3403	0.291	4.602	0.000	0.769	1.911
Tags_Busy	3.8040	0.330	11.532	0.000	3.157	4.450
Tags_Closed by Horizzon	7.9562	0.763	10.433	0.000	6.461	9.451
Tags_Lost to EINS	9.1785	0.754	12.177	0.000	7.701	10.656
Tags_Ringing	-1.6947	0.337	-5.036	0.000	-2.354	-1.035
Tags_Will revert after reading the email	3.9665	0.229	17.311	0.000	3.517	4.416
Tags_switched off	-2.2882	0.587	-3.900	0.000	-3.438	-1.138
Lead Quality_Not Sure	-3.3406	0.128	-26.026	0.000	-3.592	-3.089
Lead Quality_Worst	-3.7624	0.850	-4.426	0.000	-5.428	-2.096
Last Notable Activity_SMS Sent	2.7406	0.120	22.847	0.000	2.506	2.976

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

Out[136]:

```
3009    0.188037
1012    0.194070
9226    0.000805
4750    0.782077
7987    0.977003
1281    0.990228
2880    0.188037
4971    0.753104
7536    0.867357
1248    0.000805
dtype: float64
```

In [137]:
In [137]:

```
X_train_sm =
logm2 =       sm.families.Binomial()) res = res.summary()
```

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

Out[137]:
```
array([1.88037158e-01, 1.94070077e-01, 8.04879357e-04, 7.82076694e-01,
       9.77003470e-01, 9.90227993e-01, 1.88037158e-01, 7.53103755e-01,
              8.67356930e-01, 8.04879357e-04])
```

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

In

```
[138]:      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[138]:

	Converted	Converted_prob	Prospect ID
0	0	0.188037	3009
1	0	0.194070	1012
2	0	0.000805	9226
3	1	0.782077	4750
4	1	0.977003	7987

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

In [139]:

```
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[139]:

	Converted	Converted_prob	Prospect ID	predicted
0	0	0.188037	3009	0
1	0	0.194070	1012	0
2	0	0.000805	9226	0
3	1	0.782077	4750	1
4	1	0.977003	7987	1

In [140]:

```
from sklearn import metrics
 # Confusion
matrix
confusion = metrics.confusion_matrix(y_train_pred_final.Converted,
y_train_pred_f inal.predicted ) print(confusion)
```

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

In [141]:

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

In [142]:

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

```
0.9193827743662415
```

Checking VIFs

In [143]:

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

In [144]:

```
# 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[144]:

	Features	VIF
8	Tags_Will revert after reading the email	2.89
12	Last Notable Activity_SMS Sent	2.85
1	Lead Origin_Lead Add Form	1.62
7	Tags_Ringing	1.56
2	Lead Source_Welingak Website	1.36
3	What is your current occupation_Working Profes...	1.26
5	Tags_Closed by Horizzon	1.15
0	Do Not Email	1.11
4	Tags_Busy	1.11
10	Lead Quality_Not Sure	1.11
6	Tags_Lost to EINS	1.05
9	Tags_switched off	1.04
11	Lead Quality_Worst	1.02

Metrics beyond simply accuracy

In [145]:
In [146]:

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

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

Out[146]:

```
0.8515944399018807
```

In [147]:

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

Out[147]:

```
0.9618437900128041
```

In [148]:

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

```
0.0381562099871959O5
```

In [149]:

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

```
0.9332437275985663
```

In [150]:

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

```
0.9118718135469774
```

Step 9: 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 [151]:

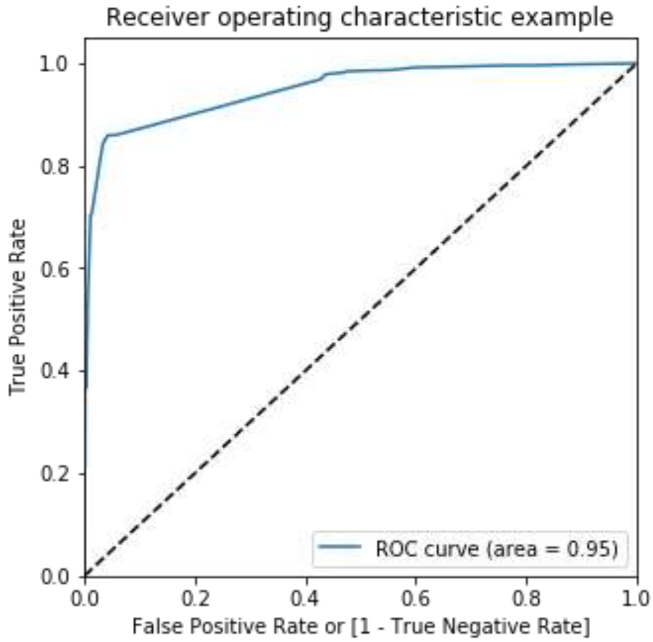
```
def draw_roc( actual, probs ):
    fpr, tpr, thresholds = metrics.roc_curve( actual, probs,
drop_intermediate = False )     auc_score = metrics.roc_auc_score( actual,
probs )     plt.figure(figsize=(5, 5))
    plt.plot( fpr, tpr, label='ROC curve (area = %0.2f)' % auc_score
)     plt.plot([0, 1], [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 [152]:

```
fpr, tpr, thresholds = metrics.roc_curve( y_train_pred_final.Converted, y_train_p
red_final.Converted_prob, drop_intermediate = False )
```

In [153]:

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



Step 10: Finding Optimal Cutoff Point

Optimal cutoff probability is that prob where we get balanced sensitivity and specificity

```
# 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[154]:

	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
0	0	0.188037	3009	0	1	1	0	0	0	0	0	0	0	0
1	0	0.194070	1012	0	1	1	0	0	0	0	0	0	0	0
2	0	0.000805	9226	0	1	0	0	0	0	0	0	0	0	0
3	1	0.782077	4750	1	1	1	1	1	1	1	1	1	0	0
4	1	0.977003	7987	1	1	1	1	1	1	1	1	1	1	1

In [154]:

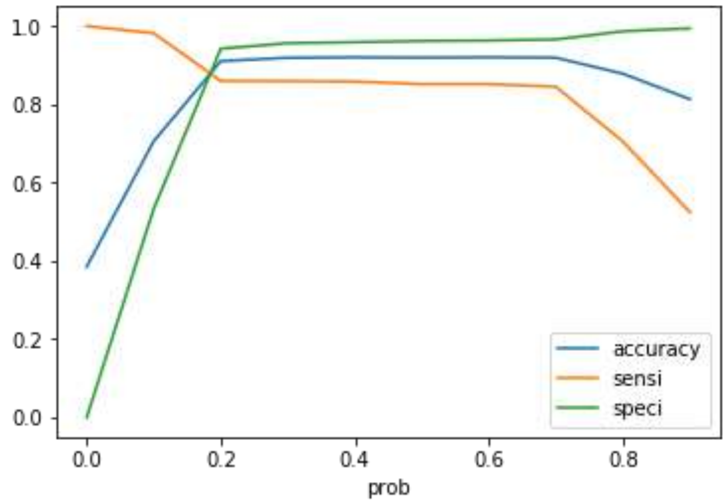
In [155]:

```
# Now let's calculate accuracy sensitivity and specificity for various probabilit y
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_fin
al[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	0.705873	0.981603	0.533163
0.2	0.2	0.910408	0.859771	0.942125
0.3	0.3	0.918910	0.859362	0.956210
0.4	0.4	0.920013	0.858136	0.958771
0.5	0.5	0.919383	0.851594	0.961844
0.6	0.6	0.920170	0.851594	0.963124
0.7	0.7	0.919225	0.845053	0.965685
0.8	0.8	0.878287	0.705233	0.986684
0.9	0.9	0.813258	0.524530	0.994110

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

In [156]:



In [157]:

```
#### From the curve above, 0.2 is the optimum point to take it as a cutoff probab ility.
y_train_pred_final['final_predicted'] = y_train_pred_final.Converted_prob.map( la mbda x: 1 if x > 0.2 else 0)

y_train_pred_final.head()
```

Out[157]:

	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
0	0	0.188037	3009	0	1	1	0	0	0	0	0	0	0	0
1	0	0.194070	1012	0	1	1	0	0	0	0	0	0	0	0
2	0	0.000805	9226	0	1	0	0	0	0	0	0	0	0	0
3	1	0.782077	4750	1	1	1	1	1	1	1	1	1	0	0
4	1	0.977003	7987	1	1	1	1	1	1	1	1	1	1	1

Assigning Lead Score

In [158]:

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

y_train_pred_final.head()
```

Out[158]:

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

	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
0	0	0.188037	3009	0	1	1	0	0	0	0	0	0	0	0
1	0	0.194070	1012	0	1	1	0	0	0	0	0	0	0	0
2	0	0.000805	9226	0	1	0	0	0	0	0	0	0	0	0
3	1	0.782077	4750	1	1	1	1	1	1	1	1	1	0	0
4	1	0.977003	7987	1	1	1	1	1	1	1	1	1	1	1

In [159]:

In [160]:

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

Out[160]:

```
0.8597710547833197
```

In [161]:

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

Out[161]:

```
0.9421254801536492
```

In [162]:

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

```
0.05787451984635083
```

In [163]:

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

```
0.9029626449119794
```

In [164]:

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

```
0.9147190452511188
```

Precision and Recall

In [165]:

```
#Looking at the confusion matrix again
 confusion = metrics.confusion_matrix(y_train_pred_final.Converted,
y_train_pred_f inal.predicted ) confusion
```

Out[165]:

```
array([[3756,  149],
[ 363, 2083]])
```

In [166]:

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

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

Out[166]:

```
0.9332437275985663
```

In [167]:

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

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

Out[167]:

```
0.8515944399018807
```

Using sklearn utilities for the same

In [168]:

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

In [169]:

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

Out[169]:

```
0.9332437275985663
```

In [170]:

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

Out[170]:

```
0.8515944399018807
```

Precision and recall tradeoff

In [171]:

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

In [172]:

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

Out[172]:

```
(0      0
1       0
2       0
3       1
4       1
5       1
6       0
7       1
8       1
9       0
10      0
11      0
12      0
13      1
14      1
15      1
16      0
17      0
18      0
19      0
20      1
21      0
22      0
23      0
24      1
25      0
26      1
27      1
28      0
29      1
30      0
31      1
32      1
33      0
34      1
35      0
36      0
37      0
38      0
39      0
```

40	0
41	0
42	1
43	1
44	1
45	0
46	1
47	0
48	1
49	1

..

6301	1
6302	0
6303	1
6304	1
6305	1
6306	1
6307	0
6308	0
6309	0
6310	1
6311	1
6312	0
6313	0
6314	0
6315	1
6316	1
6317	1
6318	0
6319	0
6320	0
6321	0
6322	1
6323	0
6324	1
6325	0
6326	0
6327	0
6328	1
6329	1
6330	1

```
6331    0
 6332    0
6333    0
6334    0 6335    0
6336    0
6337    0
6338    0
6339    0
6340    0
6341    0
6342    1
 6343    0
 6344    1
 6345    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
5       1
6       0
7       1
8       1
9       0
10      0
11      0
12      0
13      1
14      1
15      1
16      0
17      0
18      0
19      0
20      1
```

```
21      0
22      0
23      0
24      1
25      0
26      0
27      1
28      0
29      1
30      0
31      1
32      0
33      0
34      1
35      0  36      0
37      0
38      0
39      0
40      0
41      0
42      1
43      1
44      1
45      0
46      1
47      0
48      1
49      1
        ..
6301    0
6302    0
6303    1
6304    1
6305    1
6306    1
6307    0
6308    0
6309    0
6310    1
6311    1
```

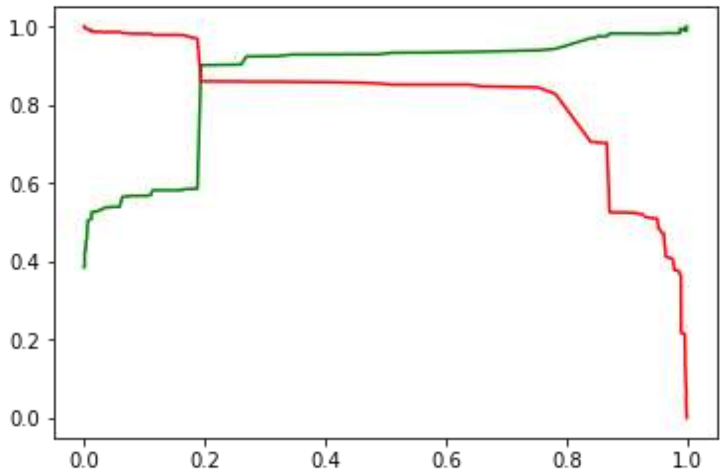
```
6312    0
6313    0
6314    0
6315    1
6316    1
6317    1
6318    0
6319    0 6320    0
6321    0
6322    1
6323    0
6324    1
6325    0
6326    0
6327    0
6328    1
6329    0
6330    1
6331    0
 6332    0  6342                                        164
 6343    0  6344                                        164
 6345                                                   164
 6346    0  6347                                        164

6333    0
6334    0
6335    0
6336    0
6337    0
6338    0
6339    0
6340    0
6341    0
6348    0
6349    0
6350    0
 Name: predicted, Length: 6351, dtype: int64)
```

In [173]:

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

In [174]:

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



Making predictions on the test set

In [175]:

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

X_train.head()
```

```
/opt/conda/lib/python3.6/site-packages/sklearn/preprocessing/data.py:645: DataCo
nversionWarning: Data with input dtype int64, float64 were all converted to
floa t64 by StandardScaler.
  return self.partial_fit(X, y)
/opt/conda/lib/python3.6/site-packages/sklearn/base.py:464:
DataConversionWarnin g: Data with input dtype int64, float64 were all converted
to float64 by Standar dScaler.  return self.fit(X, **fit_params).transform(X)
```

Out[175]:

	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
3009	0	0	-0.432779	-0.160255	-0.155018	1	0	0	0
1012	1	0	-0.432779	-0.540048	-0.155018	1	0	0	0
9226	0	0	-1.150329	-0.888650	-1.265540	0	0	0	0
4750	0	0	-0.432779	1.643304	-0.155018	1	0	0	0
7987	0	0	0.643547	2.017593	0.122613	1	0	0	0

In [176]:

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

Out[176]:

	Do Not Email	Lead Origin_Lead Add Form	Lead Source_Welingak Website	What is your current occupation_Working Professional	Tags_Busy	Tags_Closed by Horizzon	Tags_Lost to EINS
3271	0	0	0	0	0	0	0
1490	0	0	0	1	0	0	0
7936	0	0	0	0	0	0	0
4216	0	1	0	0	0	1	0
3830	0	0	0	0	0	0	0

In [177]:
In [177]:

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

Making predictions on the test set

In [178]:

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

In [179]:

```
y_test_pred[:10]
```

Out[179]:

```
3271    0.188037
1490    0.961508
7936    0.188037
4216    0.999049
3830    0.188037
1800    0.961508
6507    0.012329
4821    0.000445
4223    0.996691
4714    0.188037
dtype: float64
```

In [180]:

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

In [181]:

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

Out[181]:

	0
3271	0.188037
1490	0.961508
7936	0.188037
4216	0.999049
3830	0.188037

In [182]:

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

In [183]:

In [184]:

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

```
# 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)
```

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

In [185]:

In [186]:

```
y_pred_final.head()
```

Out[186]:

	Converted	Prospect ID	0
0	0	3271	0.188037
1	1	1490	0.961508
2	0	7936	0.188037
3	1	4216	0.999049
4	0	3830	0.188037

In [187]:

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

In [188]:

In [189]:

```
# Rearranging the columns
y_pred_final = y_pred_final.reindex_axis(['Prospect ID','Converted','Converted_prob'], axis=1)
```

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

Out[189]:

	Prospect ID	Converted	Converted_prob
0	3271	0	0.188037
1	1490	1	0.961508
2	7936	0	0.188037
3	4216	1	0.999049
4	3830	0	0.188037

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

In [190]:

In [191]:

```
y_pred_final.head()
```

Out[191]:

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

	Prospect ID	Converted	Converted_prob	final_predicted
0	3271	0	0.188037	0
1	1490	1	0.961508	1
2	7936	0	0.188037	0
3	4216	1	0.999049	1
4	3830	0	0.188037	0

In [192]:

Out[192]:

```
0.906720528828498
```

In [193]:

```
confusion2 = metrics.confusion_matrix(y_pred_final.Converted,
y_pred_final.final_ predicted ) confusion2
```

Out[193]:

```
array([[1635,   99],
[ 155,   834]])
```

In [194]:

```
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 [195]:

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

Out[195]:

```
0.8432760364004045
```

In [196]:

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

Out[196]:

```
0.9429065743944637
```