

# Case Study - Leads Scoring

## 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%.

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%.

## Goals of the Case Study

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

In [253...]

```
#importing libraries

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

import warnings
warnings.filterwarnings('ignore')

from sklearn.preprocessing import StandardScaler
```

In [254...]

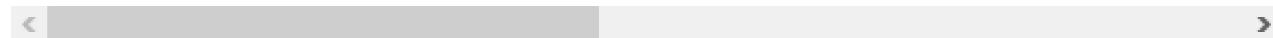
```
#importing dataset

leads=pd.read_csv("Leads.csv")
leads.head()
```

Out[254...]

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

5 rows × 37 columns



In [255...]

leads.shape

Out[255...]

(9240, 37)

In [256...]

leads.info()

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9240 entries, 0 to 9239
Data columns (total 37 columns):
 #   Column           Non-Null Count Dtype
 --- 
 0   Prospect ID     9240 non-null   object
 1   Lead Number     9240 non-null   int64
 2   Lead Origin     9240 non-null   object
 3   Lead Source     9204 non-null   object
 4   Do Not Email    9240 non-null   object
 5   Do Not Call     9240 non-null   object
 6   Converted       9240 non-null   int64
 7   TotalVisits     9103 non-null   float64
 8   Total Time Spent on Website 9240 non-null   int64
 9   Page Views Per Visit  9103 non-null   float64
 10  Last Activity   9137 non-null   object
 11  Country         6779 non-null   object
```

```

12 Specialization                      7802 non-null  object
13 How did you hear about X Education   7033 non-null  object
14 What is your current occupation      6550 non-null  object
15 What matters most to you in choosing a course 6531 non-null  object
16 Search                             9240 non-null  object
17 Magazine                           9240 non-null  object
18 Newspaper Article                  9240 non-null  object
19 X Education Forums                 9240 non-null  object
20 Newspaper                           9240 non-null  object
21 Digital Advertisement              9240 non-null  object
22 Through Recommendations            9240 non-null  object
23 Receive More Updates About Our Courses 9240 non-null  object
24 Tags                               5887 non-null  object
25 Lead Quality                       4473 non-null  object
26 Update me on Supply Chain Content  9240 non-null  object
27 Get updates on DM Content          9240 non-null  object
28 Lead Profile                        6531 non-null  object
29 City                               7820 non-null  object
30 Asymmetrique Activity Index       5022 non-null  object
31 Asymmetrique Profile Index        5022 non-null  object
32 Asymmetrique Activity Score      5022 non-null  float64
33 Asymmetrique Profile Score        5022 non-null  float64
34 I agree to pay the amount through cheque 9240 non-null  object
35 A free copy of Mastering The Interview 9240 non-null  object
36 Last Notable Activity             9240 non-null  object
dtypes: float64(4), int64(3), object(30)
memory usage: 2.6+ MB

```

In [257...]

`leads.describe()`

Out[257...]

	Lead Number	Converted	TotalVisits	Total Time Spent on Website	Page Views Per Visit	Asymmetrique Activity Score	Asymmetrique Profile Score
<b>count</b>	9240.000000	9240.000000	9103.000000	9240.000000	9103.000000	5022.000000	5022.000000
<b>mean</b>	617188.435606	0.385390	3.445238	487.698268	2.362820	14.306252	16.34488
<b>std</b>	23405.995698	0.486714	4.854853	548.021466	2.161418	1.386694	1.81139
<b>min</b>	579533.000000	0.000000	0.000000	0.000000	0.000000	7.000000	11.00000
<b>25%</b>	596484.500000	0.000000	1.000000	12.000000	1.000000	14.000000	15.00000
<b>50%</b>	615479.000000	0.000000	3.000000	248.000000	2.000000	14.000000	16.00000
<b>75%</b>	637387.250000	1.000000	5.000000	936.000000	3.000000	15.000000	18.00000
<b>max</b>	660737.000000	1.000000	251.000000	2272.000000	55.000000	18.000000	20.00000

## EXPLORATORY DATA ANALYSIS

### Data understanding, preparation

In [258...]

`#check for duplicates`

```
sum(leads.duplicated(subset = 'Prospect ID')) == 0
sum(leads.duplicated(subset = 'Lead Number')) == 0
```

Out[258...]

No duplicate values found in Prospect ID & Lead Number

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

In [259...]

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

In [260...]

```
#Converting 'Select' values to NaN.
leads = leads.replace('Select', np.nan)
```

In [261...]

```
#checking null values in each rows
leads.isnull().sum()
```

Out[261...]

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
dtype: int64
```

0

In [262...]

```
#checking percentage of null values in each column

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

Out[262...]

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 [263...]

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

cols=leads.columns

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

In [264...]

```
#checking null values percentage
```

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

Out[264...]

Lead Origin	0.00
-------------	------

Lead Source	0.39
Do Not Email	0.00
Do Not Call	0.00
Converted	0.00
TotalVisits	1.48
Total Time Spent on Website	0.00
Page Views Per Visit	1.48
Last Activity	1.11
Country	26.63
Specialization	36.58
What is your current occupation	29.11
What matters most to you in choosing a course	29.32
Search	0.00
Magazine	0.00
Newspaper Article	0.00
X Education Forums	0.00
Newspaper	0.00
Digital Advertisement	0.00
Through Recommendations	0.00
Receive More Updates About Our Courses	0.00
Tags	36.29
Update me on Supply Chain Content	0.00
Get updates on DM Content	0.00
City	39.71
I agree to pay the amount through cheque	0.00
A free copy of Mastering The Interview	0.00
Last Notable Activity	0.00

dtype: float64

### Categorical Attributes Analysis:

```
In [265...]: #checking value counts of Country column
leads['Country'].value_counts(dropna=False)
```

```
Out[265...]:
```

India	6492
NaN	2461
United States	69
United Arab Emirates	53
Singapore	24
Saudi Arabia	21
United Kingdom	15
Australia	13
Qatar	10
Bahrain	7
Hong Kong	7
Oman	6
France	6
unknown	5
Kuwait	4
South Africa	4
Canada	4
Nigeria	4
Germany	4
Sweden	3
Philippines	2
Uganda	2
Italy	2
Bangladesh	2

```

Netherlands          2
Asia/Pacific Region 2
China                2
Belgium              2
Ghana                2
Kenya                1
Sri Lanka             1
Tanzania              1
Malaysia              1
Liberia              1
Switzerland           1
Denmark              1
Russia                1
Vietnam              1
Indonesia             1
Name: Country, dtype: int64

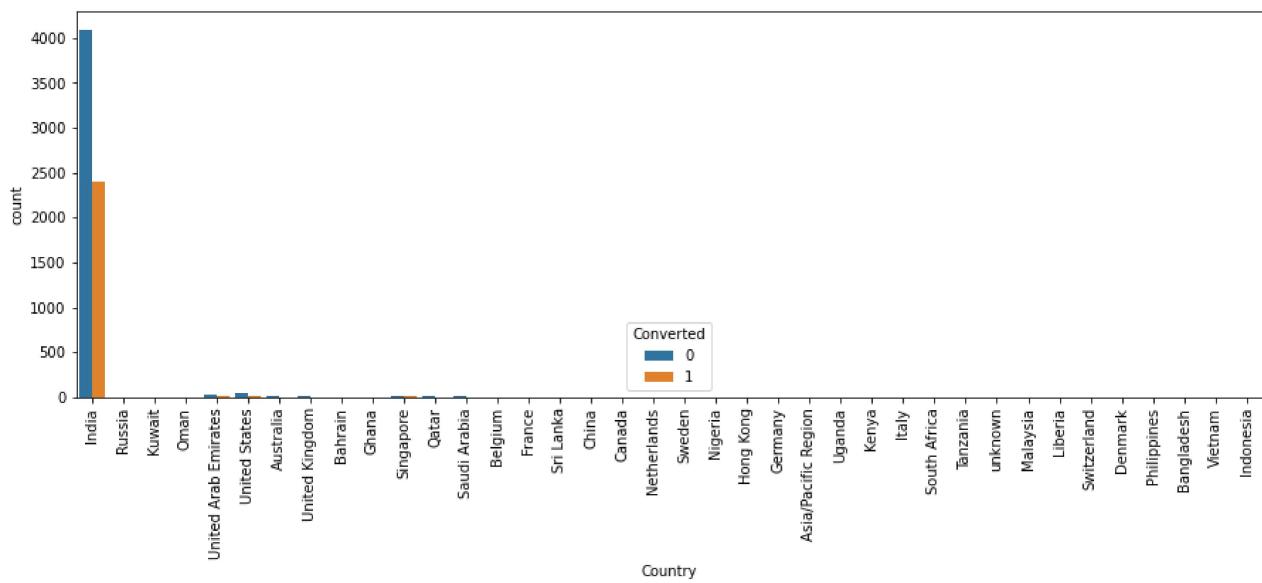
```

In [266...]

```

#plotting spread of Country column
plt.figure(figsize=(15,5))
s1=sns.countplot(leads.Country, hue=leads.Converted)
s1.set_xticklabels(s1.get_xticklabels(),rotation=90)
plt.show()

```



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

In [267...]

```

#creating a List of columns to be dropped
cols_to_drop=['Country']

```

In [268...]

```

#checking value counts of "City" column
leads['City'].value_counts(dropna=False)

```

Out[268...]

NaN	3669
Mumbai	3222
Thane & Outskirts	752
Other Cities	686

```
Other Cities of Maharashtra      457
Other Metro Cities             380
Tier II Cities                 74
Name: City, dtype: int64
```

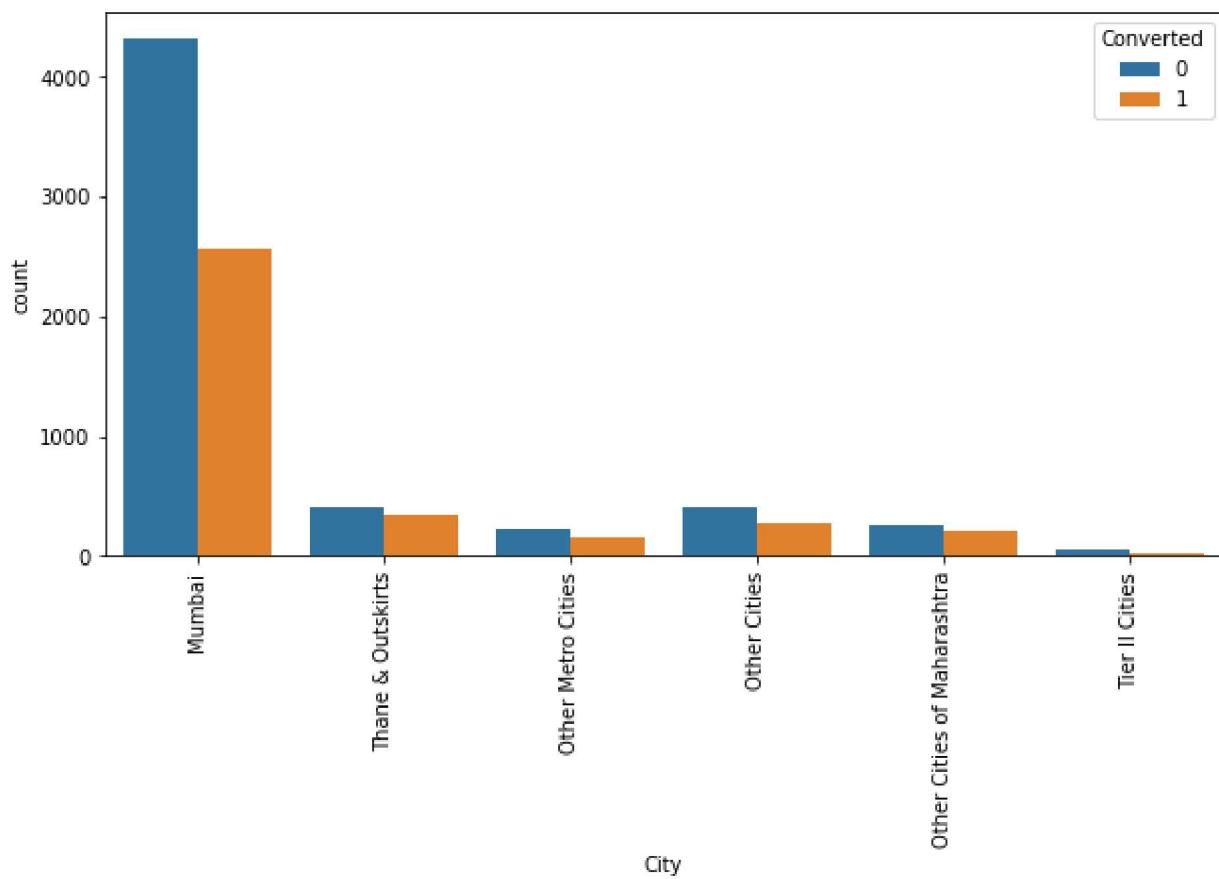
In [269]:

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

In [270]:

```
#plotting spread of City column after replacing NaN values
```

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



In [271]:

```
#checking value counts of Specialization column
```

```
leads['Specialization'].value_counts(dropna=False)
```

Out[271]:

Specialization	Count
NaN	3380
Finance Management	976
Human Resource Management	848
Marketing Management	838
Operations Management	503
Business Administration	403
IT Projects Management	366
Supply Chain Management	349
Banking, Investment And Insurance	338
Travel and Tourism	203
Media and Advertising	203

```
International Business          178
Healthcare Management          159
Hospitality Management          114
E-COMMERCE                      112
Retail Management                 100
Rural and Agribusiness            73
E-Business                        57
Services Excellence                  40
Name: Specialization, dtype: int64
```

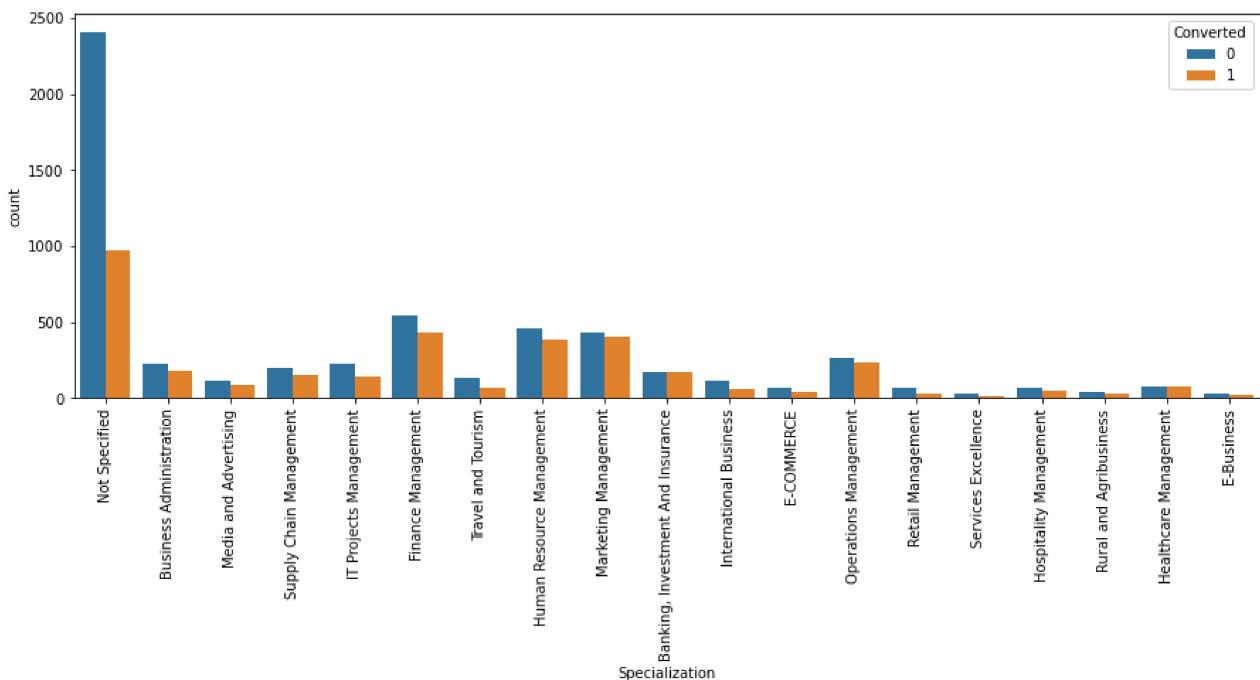
In [272...]

```
# Lead may not have mentioned specialization because it was not in the list or maybe they
# and don't have a specialization yet. So we will replace NaN values here with 'Not Specified'
leads['Specialization'] = leads['Specialization'].replace(np.nan, 'Not Specified')
```

In [273...]

*#plotting spread of Specialization column*

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



We see that specialization with Management in them have higher number of leads as well as leads converted. So this is definitely a significant variable and should not be dropped.

In [274...]

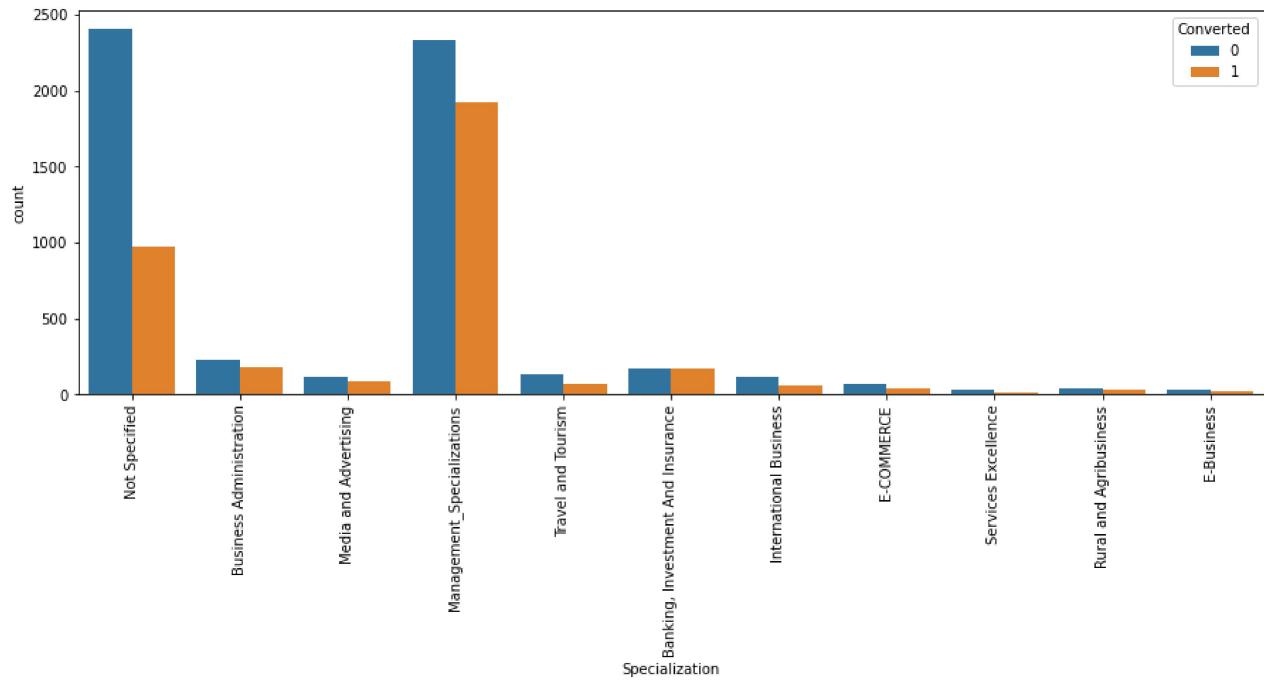
*#combining Management Specializations because they show similar trends*

```
leads['Specialization'] = leads['Specialization'].replace(['Finance Management', 'Human
'Marketing Management', 'Oper
'IT Projects Management', 'Su
'Healthcare Management', 'Hospitalit
'Retail Management'], 'Management')
```

In [275...]

```
#visualizing count of Variable based on Converted value
```

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



In [276...]

```
#What is your current occupation
```

```
leads['What is your current occupation'].value_counts(dropna=False)
```

Out[276...]

Unemployed	5600
NaN	2690
Working Professional	706
Student	210
Other	16
Housewife	10
Businessman	8

Name: What is your current occupation, dtype: int64

In [277...]

```
#imputing Nan values with mode "Unemployed"
```

```
leads['What is your current occupation'] = leads['What is your current occupation'].rep
```

In [278...]

```
#checking count of values
```

```
leads['What is your current occupation'].value_counts(dropna=False)
```

Out[278...]

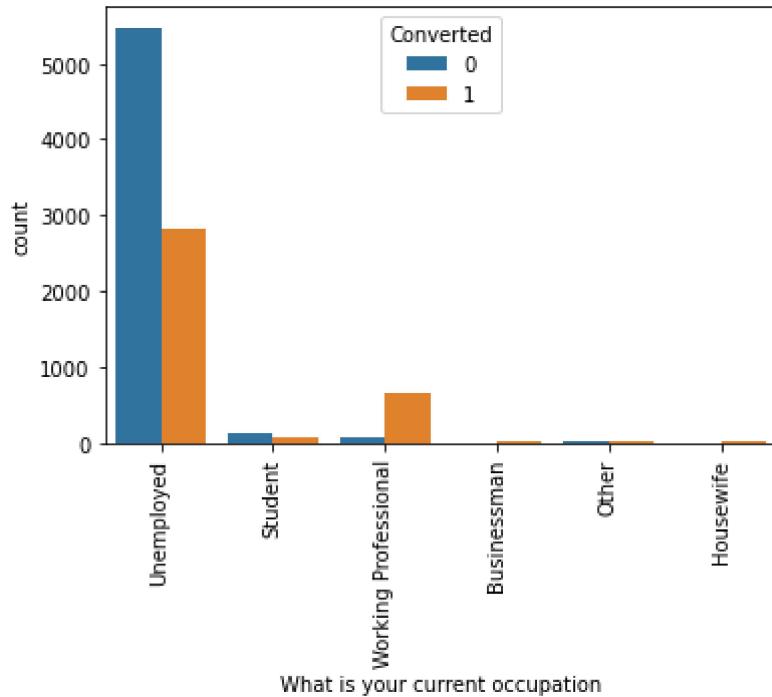
Unemployed	8290
Working Professional	706
Student	210
Other	16
Housewife	10
Businessman	8

Name: What is your current occupation, dtype: int64

In [279...]

```
#visualizing count of Variable based on Converted value
```

```
s1=sns.countplot(leads['What is your current occupation'], hue=leads.Converted)
s1.set_xticklabels(s1.get_xticklabels(),rotation=90)
plt.show()
```



Working Professionals going for the course have high chances of joining it. Unemployed leads are the most in terms of Absolute numbers.

In [280...]

```
#checking value counts
```

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

Out[280...]

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

In [281...]

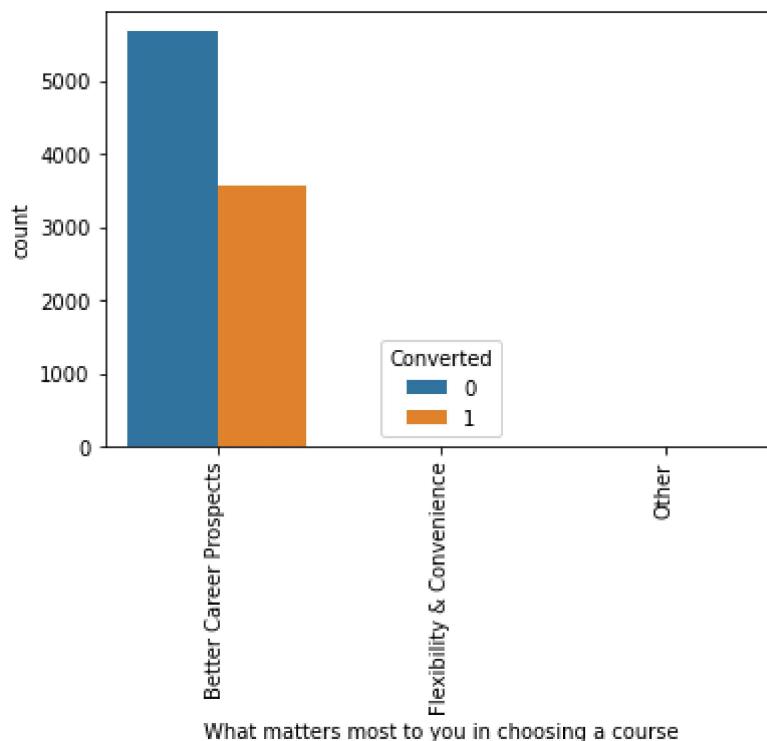
```
#replacing Nan values with Mode "Better Career Prospects"
```

```
leads['What matters most to you in choosing a course'] = leads['What matters most to yo
```

In [282...]

```
#visualizing count of Variable based on Converted value
```

```
s1=sns.countplot(leads['What matters most to you in choosing a course'], hue=leads.Converted)
s1.set_xticklabels(s1.get_xticklabels(),rotation=90)
plt.show()
```



In [283...]

```
#checking value counts of variable
leads['What matters most to you in choosing a course'].value_counts(dropna=False)
```

Out[283...]

Category	Count
Better Career Prospects	9237
Flexibility & Convenience	2
Other	1

Name: What matters most to you in choosing a course, dtype: int64

In [284...]

```
#Here again we have another Column that is worth Dropping. So we Append to the cols_to_drop
cols_to_drop.append('What matters most to you in choosing a course')
cols_to_drop
```

Out[284...]

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

In [285...]

```
#checking value counts of Tag variable
leads['Tags'].value_counts(dropna=False)
```

Out[285...]

Tag	Count
NaN	3353
Will revert after reading the email	2072
Ringing	1203
Interested in other courses	513
Already a student	465
Closed by Horizzon	358
switched off	240
Busy	186
Lost to EINS	175
Not doing further education	145
Interested in full time MBA	117
Graduation in progress	111
invalid number	83
Diploma holder (Not Eligible)	63
wrong number given	47
opp hangup	33

number not provided	27
in touch with EINS	12
Lost to Others	7
Still Thinking	6
Want to take admission but has financial problems	6
In confusion whether part time or DLP	5
Interested in Next batch	5
Lateral student	3
Shall take in the next coming month	2
University not recognized	2
Recognition issue (DEC approval)	1
Name: Tags, dtype: int64	

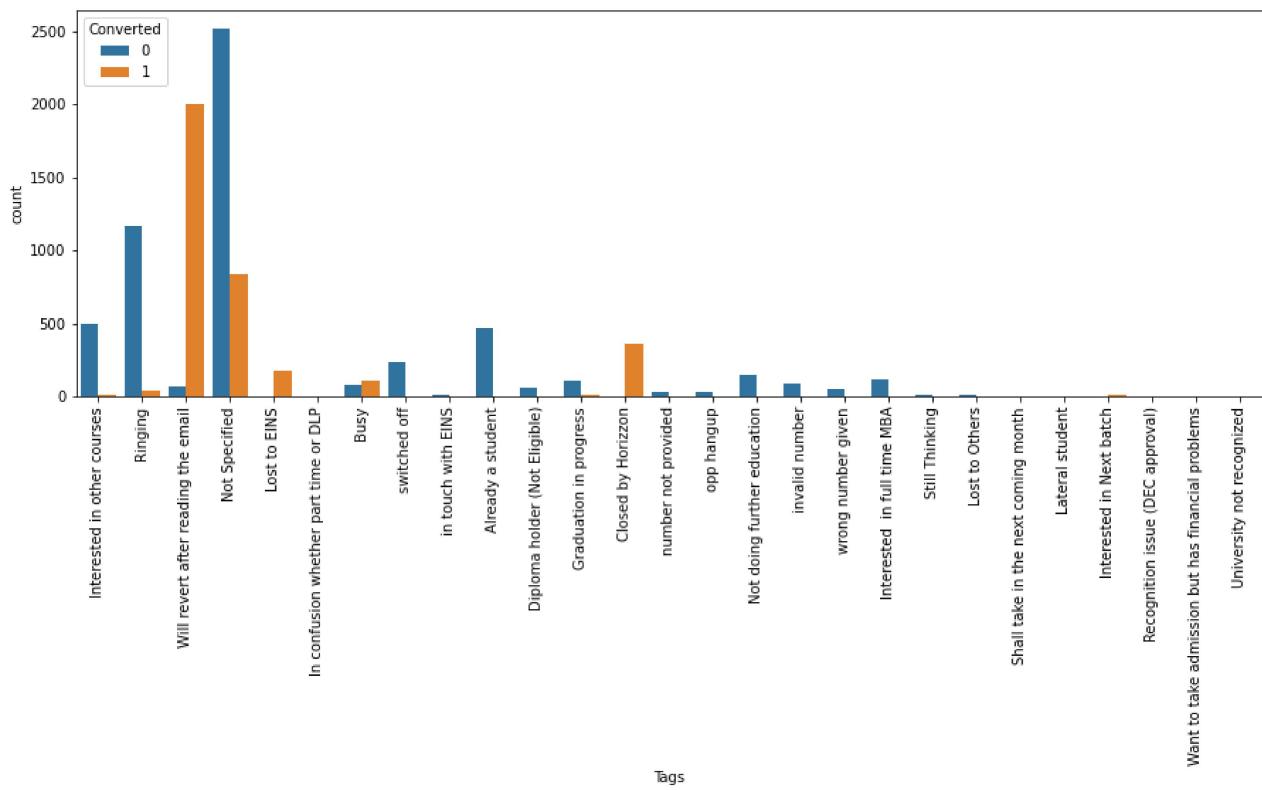
In [286...]

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

In [287...]

```
#visualizing count of Variable based on Converted value
```

```
plt.figure(figsize=(15,5))
s1=sns.countplot(leads['Tags'], hue=leads.Converted)
s1.set_xticklabels(s1.get_xticklabels(), rotation=90)
plt.show()
```



In [288...]

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

leads['Tags'] = leads['Tags'].replace(['switched off',
                                         'Already a student',
```

```
'Not doing further education',
'invalid number',
'wrong number given',
'Interested in full time MBA'] , 'Other_Tags')
```

In [289...]

```
#checking percentage of missing values
round(100*(leads.isnull().sum()/len(leads.index)), 2)
```

Out[289...]

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	0.00
What matters most to you in choosing a course	0.00
Search	0.00
Magazine	0.00
Newspaper Article	0.00
X Education Forums	0.00
Newspaper	0.00
Digital Advertisement	0.00
Through Recommendations	0.00
Receive More Updates About Our Courses	0.00
Tags	0.00
Update me on Supply Chain Content	0.00
Get updates on DM Content	0.00
City	0.00
I agree to pay the amount through cheque	0.00
A free copy of Mastering The Interview	0.00
Last Notable Activity	0.00

dtype: float64

In [290...]

```
#checking value counts of Lead Source column
```

```
leads['Lead Source'].value_counts(dropna=False)
```

Out[290...]

Google	2868
Direct Traffic	2543
Olark Chat	1755
Organic Search	1154
Reference	534
Welingak Website	142
Referral Sites	125
Facebook	55
NaN	36
bing	6
google	5
Click2call	4
Press_Release	2
Social Media	2
Live Chat	2
youtubechannel	1

```
testone          1
Pay per Click Ads 1
welearnblog_Home 1
WeLearn          1
blog             1
NC_EDM           1
Name: Lead Source, dtype: int64
```

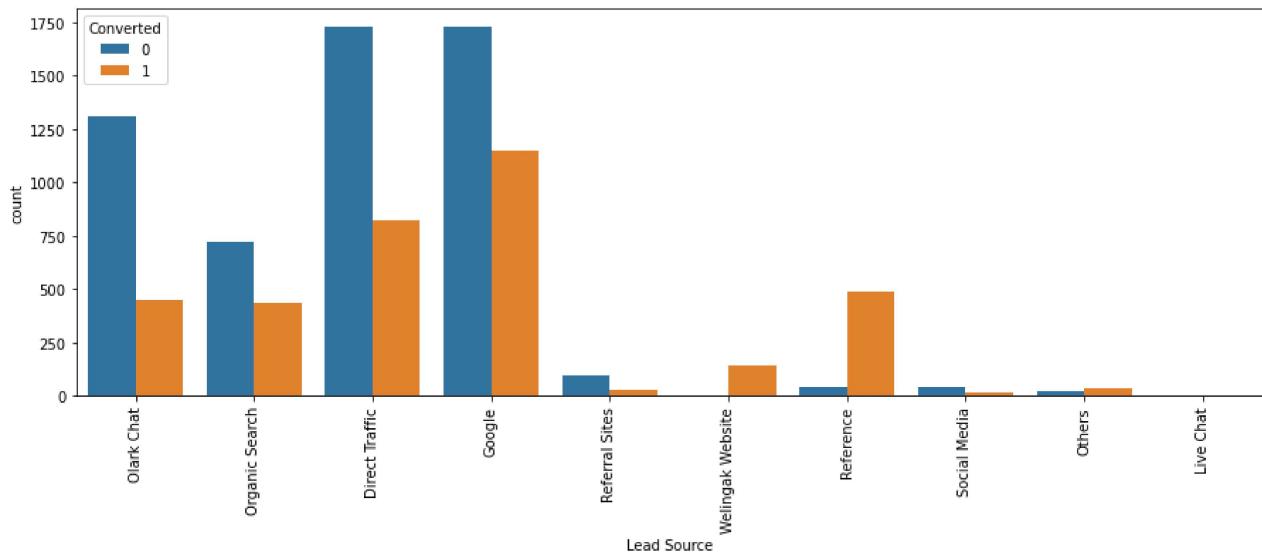
In [291...]

```
#replacing Nan Values and combining Low frequency values
leads['Lead Source'] = leads['Lead Source'].replace(np.nan, 'Others')
leads['Lead Source'] = leads['Lead Source'].replace('google', 'Google')
leads['Lead Source'] = leads['Lead Source'].replace('Facebook', 'Social Media')
leads['Lead Source'] = leads['Lead Source'].replace(['bing', 'Click2call', 'Press_Release',
                                                    'youtubechannel', 'welearnblog_Home',
                                                    'WeLearn', 'blog', 'Pay per Click Ad',
                                                    'testone', 'NC_EDM'], 'Others')
```

We can group some of the lower frequency occurring labels under a common label 'Others'

In [292...]

```
#visualizing count of Variable based on Converted value
plt.figure(figsize=(15,5))
s1=sns.countplot(leads['Lead Source'], hue=leads.Converted)
s1.set_xticklabels(s1.get_xticklabels(), rotation=90)
plt.show()
```



Inference

1. Maximum number of leads are generated by Google and Direct traffic.
2. Conversion Rate of reference leads and leads through welingak website is high.
3. To improve overall lead conversion rate, focus should be on improving lead conversion of olark chat, organic search, direct traffic, and google leads and generate more leads from reference and welingak website.

In [293...]

```
# Last Activity:
```

```
leads['Last Activity'].value_counts(dropna=False)
```

```
Out[293... Email Opened      3437
          SMS Sent        2745
          Olark Chat Conversation 973
          Page Visited on Website 640
          Converted to Lead    428
          Email Bounced       326
          Email Link Clicked   267
          Form Submitted on Website 116
          NaN                 103
          Unreachable         93
          Unsubscribed        61
          Had a Phone Conversation 30
          Approached upfront    9
          View in browser link Clicked 6
          Email Received       2
          Email Marked Spam     2
          Visited Booth in Tradeshow 1
          Resubscribed to emails 1
          Name: Last Activity, dtype: int64
```

In [294... *#replacing Nan Values and combining Low frequency values*

```
leads['Last Activity'] = leads['Last Activity'].replace(np.nan, 'Others')
leads['Last Activity'] = leads['Last Activity'].replace(['Unreachable', 'Unsubscribed',
                                                       'Had a Phone Conversation',
                                                       'Approached upfront',
                                                       'View in browser link Clicked',
                                                       'Email Marked Spam',
                                                       'Email Received', 'Resubscribed',
                                                       'Visited Booth in Tradeshow'],
```

In [295... *# Last Activity:*

```
leads['Last Activity'].value_counts(dropna=False)
```

```
Out[295... Email Opened      3437
          SMS Sent        2745
          Olark Chat Conversation 973
          Page Visited on Website 640
          Converted to Lead    428
          Email Bounced       326
          Others              308
          Email Link Clicked   267
          Form Submitted on Website 116
          Name: Last Activity, dtype: int64
```

In [296... *#Check the Null Values in ALL Columns:*

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

```
Out[296... Lead Origin      0.00
          Lead Source       0.00
          Do Not Email      0.00
          Do Not Call       0.00
          Converted         0.00
          TotalVisits        1.48
          Total Time Spent on Website 0.00
          Page Views Per Visit 1.48
```

```
Last Activity          0.00
Country              26.63
Specialization        0.00
What is your current occupation    0.00
What matters most to you in choosing a course 0.00
Search               0.00
Magazine             0.00
Newspaper Article    0.00
X Education Forums   0.00
Newspaper             0.00
Digital Advertisement 0.00
Through Recommendations 0.00
Receive More Updates About Our Courses 0.00
Tags                0.00
Update me on Supply Chain Content 0.00
Get updates on DM Content 0.00
City                0.00
I agree to pay the amount through cheque 0.00
A free copy of Mastering The Interview 0.00
Last Notable Activity 0.00
dtype: float64
```

In [297...]: #Drop all rows which have Nan Values. Since the number of Dropped rows is Less than 2%,  
leads = leads.dropna()

In [298...]: #Checking percentage of Null Values in All Columns:  
round(100\*(leads.isnull().sum()/len(leads.index)), 2)

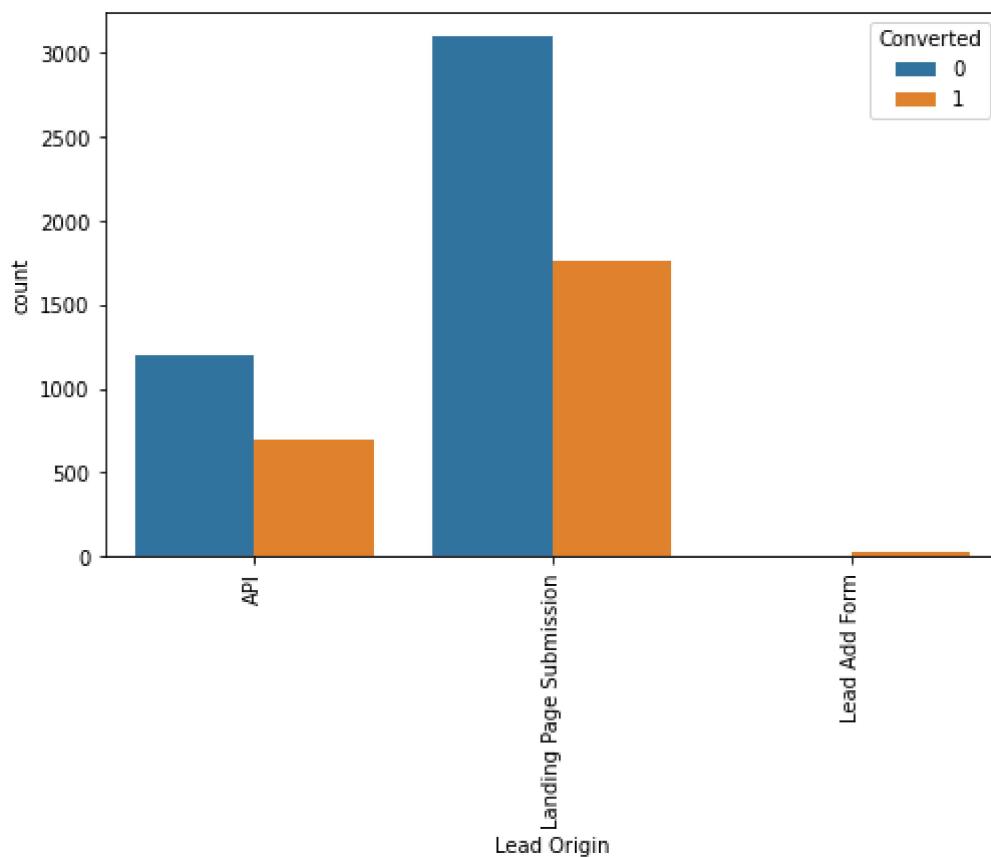
```
Out[298...]: Lead Origin          0.0
Lead Source           0.0
Do Not Email          0.0
Do Not Call           0.0
Converted             0.0
TotalVisits           0.0
Total Time Spent on Website 0.0
Page Views Per Visit 0.0
Last Activity          0.0
Country               0.0
Specialization         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
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 [299... #Lead Origin
leads['Lead Origin'].value_counts(dropna=False)
```

```
Out[299... Landing Page Submission    4850
          API                  1902
          Lead Add Form        27
          Name: Lead Origin, dtype: int64
```

```
In [300... #visualizing count of Variable based on Converted value
```

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



### Inference

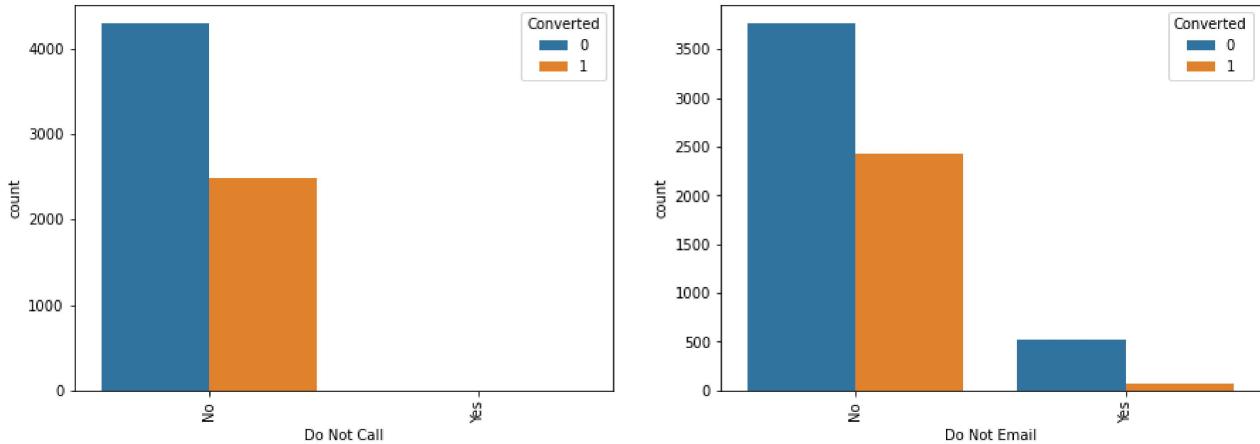
1. API and Landing Page Submission bring higher number of leads as well as conversion.
2. Lead Add Form has a very high conversion rate but count of leads are not very high.
3. Lead Import and Quick Add Form get very few leads.
4. In order to improve overall lead conversion rate, we have to improve lead converion of API and Landing Page Submission origin and generate more leads from Lead Add Form.

```
In [301... #Do Not Email & Do Not Call
#visualizing count of Variable based on Converted value

plt.figure(figsize=(15,5))
ax1=plt.subplot(1, 2, 1)
```

```
ax1=sns.countplot(leads['Do Not Call'], hue=leads.Converted)
ax1.set_xticklabels(ax1.get_xticklabels(), rotation=90)

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



In [302...]

```
#checking value counts for Do Not Call
leads['Do Not Call'].value_counts(dropna=False)
```

Out[302...]

```
No      6778
Yes      1
Name: Do Not Call, dtype: int64
```

In [303...]

```
#checking value counts for Do Not Email
leads['Do Not Email'].value_counts(dropna=False)
```

Out[303...]

```
No      6186
Yes     593
Name: Do Not Email, dtype: int64
```

We Can append the Do Not Call Column to the list of Columns to be Dropped since > 90% is of only one Value

In [304...]

```
cols_to_drop.append('Do Not Call')
cols_to_drop
```

Out[304...]

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

## IMBALANCED VARIABLES THAT CAN BE DROPPED

In [305...]

```
leads.Search.value_counts(dropna=False)
```

Out[305...]

```
No      6765
Yes      14
Name: Search, dtype: int64
```

In [306...]

```
leads.Magazine.value_counts(dropna=False)
```

```
Out[306...]: No    6779  
              Name: Magazine, dtype: int64
```

```
In [307...]: leads['Newspaper Article'].value_counts(dropna=False)
```

```
Out[307...]: No    6777  
              Yes   2  
              Name: Newspaper Article, dtype: int64
```

```
In [308...]: leads['X Education Forums'].value_counts(dropna=False)
```

```
Out[308...]: No    6778  
              Yes   1  
              Name: X Education Forums, dtype: int64
```

```
In [309...]: leads['Newspaper'].value_counts(dropna=False)
```

```
Out[309...]: No    6778  
              Yes   1  
              Name: Newspaper, dtype: int64
```

```
In [310...]: leads['Digital Advertisement'].value_counts(dropna=False)
```

```
Out[310...]: No    6775  
              Yes   4  
              Name: Digital Advertisement, dtype: int64
```

```
In [311...]: leads['Through Recommendations'].value_counts(dropna=False)
```

```
Out[311...]: No    6772  
              Yes   7  
              Name: Through Recommendations, dtype: int64
```

```
In [312...]: leads['Receive More Updates About Our Courses'].value_counts(dropna=False)
```

```
Out[312...]: No    6779  
              Name: Receive More Updates About Our Courses, dtype: int64
```

```
In [313...]: leads['Update me on Supply Chain Content'].value_counts(dropna=False)
```

```
Out[313...]: No    6779  
              Name: Update me on Supply Chain Content, dtype: int64
```

```
In [314...]: leads['Get updates on DM Content'].value_counts(dropna=False)
```

```
Out[314...]: No    6779  
              Name: Get updates on DM Content, dtype: int64
```

```
In [315...]: leads['I agree to pay the amount through cheque'].value_counts(dropna=False)
```

```
Out[315...]: No    6779  
              Name: I agree to pay the amount through cheque, dtype: int64
```

```
In [316...]: leads['A free copy of Mastering The Interview'].value_counts(dropna=False)
```

```
Out[316...]: No      3913
           Yes     2866
           Name: A free copy of Mastering The Interview, dtype: int64
```

```
In [317...]: #adding imbalanced columns to the list of columns to be dropped
```

```
cols_to_drop.extend(['Search', 'Magazine', 'Newspaper Article', 'X Education Forums', 'News Digital Advertisement', 'Through Recommendations', 'Receive More Update', 'Update me on Supply Chain Content', 'Get updates on DM Content', 'I agree to pay the amount through cheque'])
```

```
In [318...]: #checking value counts of Last Notable Activity
leads['Last Notable Activity'].value_counts()
```

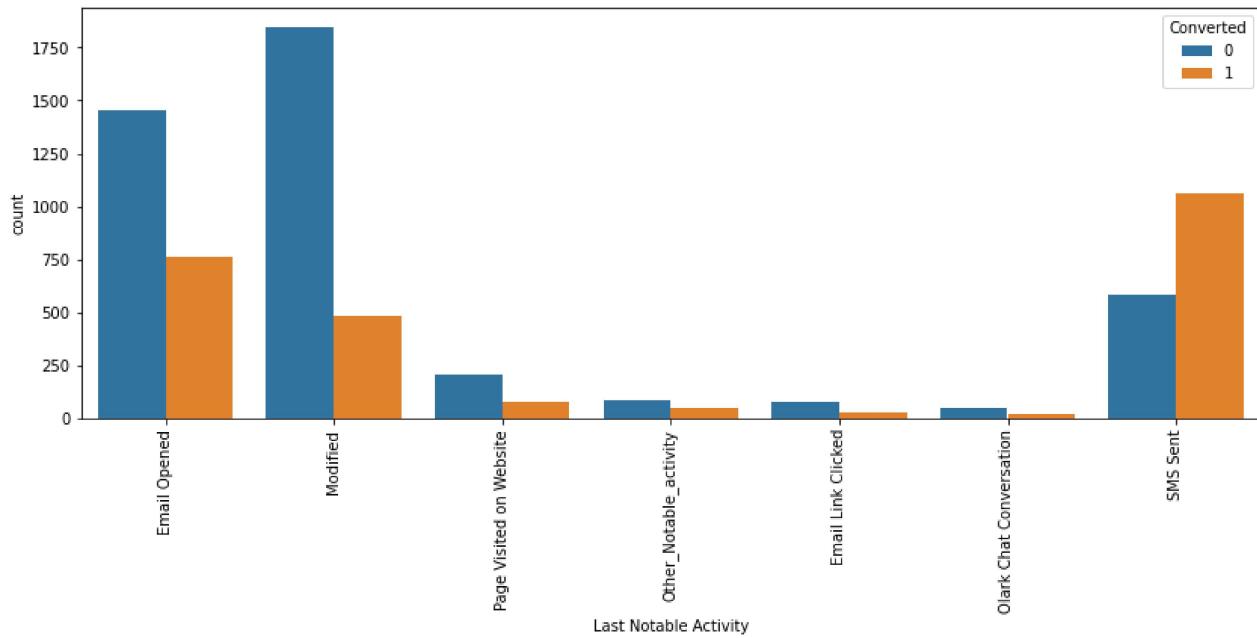
```
Out[318...]: Modified              2331
Email Opened          2212
SMS Sent              1643
Page Visited on Website 286
Email Link Clicked    106
Olark Chat Conversation 68
Email Bounced         48
Unsubscribed          40
Unreachable            26
Had a Phone Conversation 14
Approached upfront     1
View in browser link Clicked 1
Form Submitted on Website 1
Email Received         1
Email Marked Spam      1
Name: Last Notable Activity, dtype: int64
```

```
In [319...]: #clubbing lower frequency values
```

```
leads['Last Notable Activity'] = leads['Last Notable Activity'].replace(['Had a Phone C', 'Email Marked Sp', 'Unreachable', 'Unsubscribed', 'Email Bounced', 'Resubscribed to', 'View in browser', 'Approached upfr', 'Form Submitted', 'Email Received'])
```

```
In [320...]: #visualizing count of Variable based on Converted value
```

```
plt.figure(figsize = (14,5))
ax1=sns.countplot(x = "Last Notable Activity", hue = "Converted", data = leads)
ax1.set_xticklabels(ax1.get_xticklabels(),rotation=90)
plt.show()
```



```
In [321... #checking value counts for variable
```

```
leads['Last Notable Activity'].value_counts()
```

```
Out[321... Modified          2331
Email Opened      2212
SMS Sent          1643
Page Visited on Website  286
Other_Notable_activity 133
Email Link Clicked   106
Olark Chat Conversation 68
Name: Last Notable Activity, dtype: int64
```

```
In [322... #list of columns to be dropped
cols_to_drop
```

```
Out[322... ['Country',
'What matters most to you in choosing a course',
'Do Not Call',
'Search',
'Magazine',
'Newspaper Article',
'X Education Forums',
'Newspaper',
'Digital Advertisement',
'Through Recommendations',
'Receive More Updates About Our Courses',
'Update me on Supply Chain Content',
'Get updates on DM Content',
'I agree to pay the amount through cheque']
```

```
In [323... #dropping columns
leads = leads.drop(cols_to_drop,1)
leads.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 6779 entries, 1 to 9239
```

Data columns (total 14 columns):

#	Column	Non-Null Count	Dtype
0	Lead Origin	6779 non-null	object
1	Lead Source	6779 non-null	object
2	Do Not Email	6779 non-null	object
3	Converted	6779 non-null	int64
4	TotalVisits	6779 non-null	float64
5	Total Time Spent on Website	6779 non-null	int64
6	Page Views Per Visit	6779 non-null	float64
7	Last Activity	6779 non-null	object
8	Specialization	6779 non-null	object
9	What is your current occupation	6779 non-null	object
10	Tags	6779 non-null	object
11	City	6779 non-null	object
12	A free copy of Mastering The Interview	6779 non-null	object
13	Last Notable Activity	6779 non-null	object

dtypes: float64(2), int64(2), object(10)  
memory usage: 1.0+ MB

### Numerical Attributes Analysis:

In [324...]

```
#Check the % of Data that has Converted Values = 1:

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

Out[324...]

36.65732408909869

In [325...]

```
#Checking correlations of numeric values
# figure size
plt.figure(figsize=(10,8))

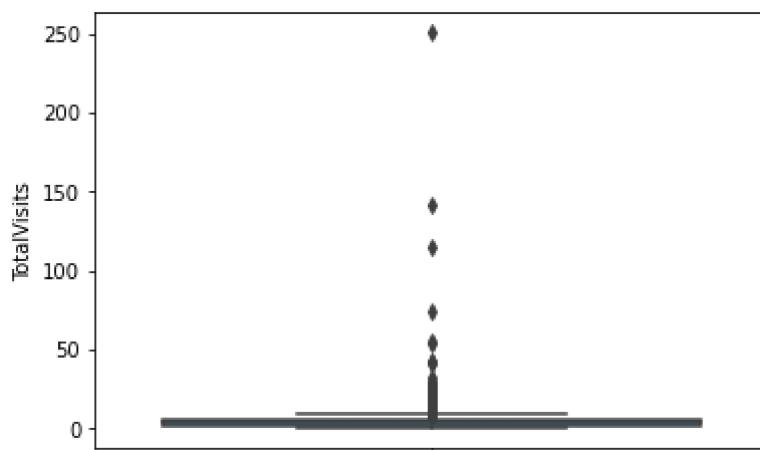
# heatmap
sns.heatmap(leads.corr(), cmap="YlGnBu", annot=True)
plt.show()
```



In [326]:

```
#Total Visits
#visualizing spread of variable

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



We can see presence of outliers here

In [327...]

```
#checking percentile values for "Total Visits"

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

Out[327...]

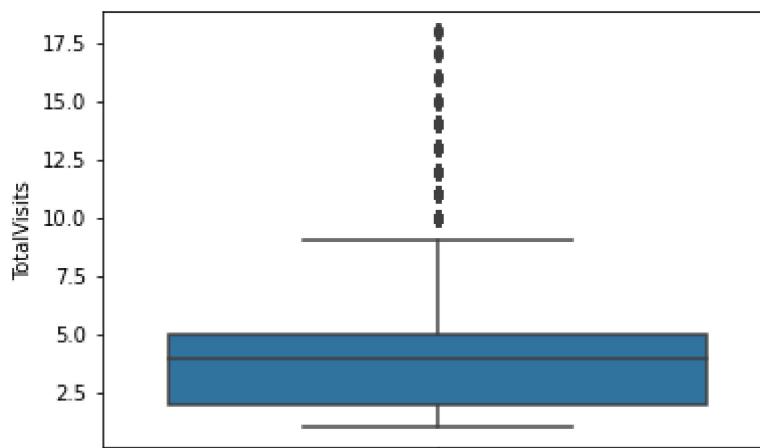
count	6779.000000
mean	4.553917
std	5.138148
min	0.000000
5%	1.000000
25%	2.000000
50%	4.000000
75%	5.000000
90%	8.000000
95%	11.000000
99%	18.220000
max	251.000000

Name: TotalVisits, dtype: float64

In [328...]

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

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



In [329...]

```
leads.shape
```

Out[329...]

```
(6706, 14)
```

Check for the Next Numerical Column:

In [330...]

```
#checking percentiles for "Total Time Spent on Website"
```

```
leads['Total Time Spent on Website'].describe(percentiles=[0.05,.25, .5, .75, .90, .95,
```

Out[330...]

count	6706.000000
mean	631.736206
std	541.763382

```

min      0.000000
5%      33.000000
25%     171.000000
50%     397.000000
75%    1104.000000
90%    1451.500000
95%    1606.000000
99%    1862.950000
max    2272.000000
Name: Total Time Spent on Website, dtype: float64

```

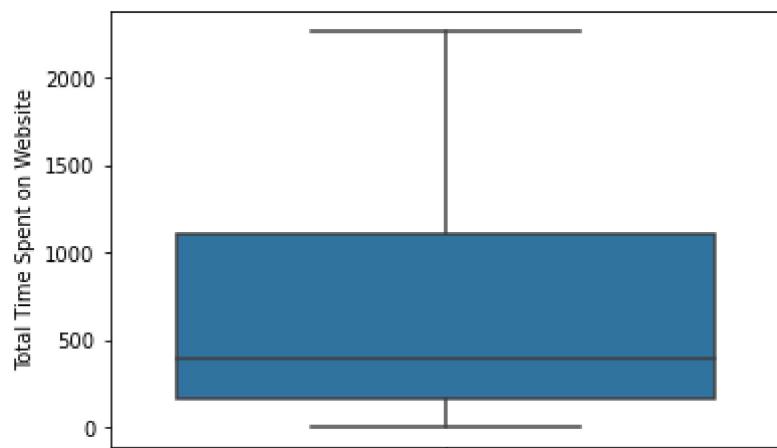
In [331...]

```

#visualizing spread of numeric variable

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

```



Since there are no major Outliers for the above variable we don't do any Outlier Treatment for this above Column

Check for Page Views Per Visit:

In [332...]

```

#checking spread of "Page Views Per Visit"

leads['Page Views Per Visit'].describe()

```

Out[332...]

```

count    6706.000000
mean      3.104545
std       1.817244
min      1.000000
25%      2.000000
50%      3.000000
75%      4.000000
max     16.000000
Name: Page Views Per Visit, dtype: float64

```

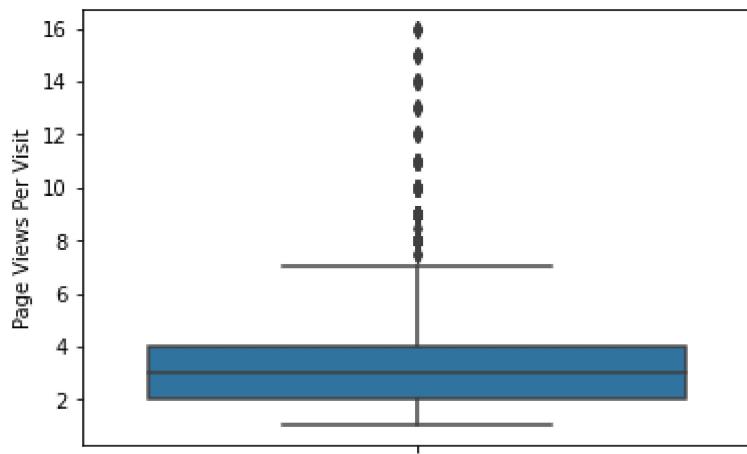
In [333...]

```

#visualizing spread of numeric variable

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

```



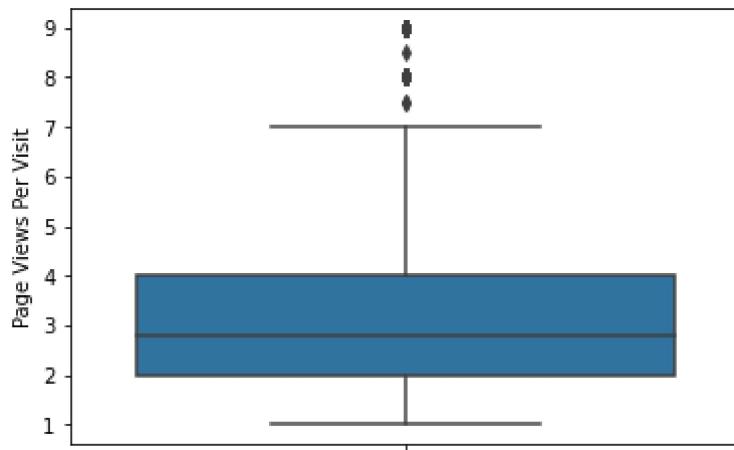
In [334...]

#Outlier Treatment: Remove top &amp; bottom 1%

```

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

```



In [335...]

leads.shape

Out[335...]

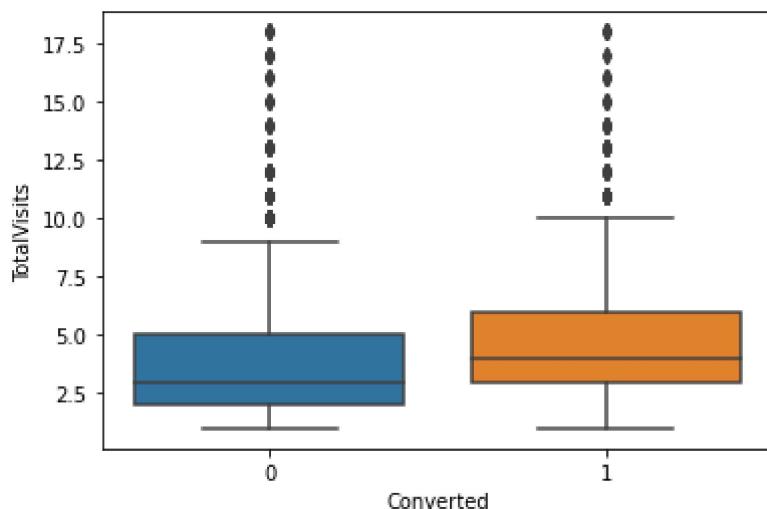
(6640, 14)

In [336...]

```

#checking Spread of "Total Visits" vs Converted variable
sns.boxplot(y = 'TotalVisits', x = 'Converted', data = leads)
plt.show()

```

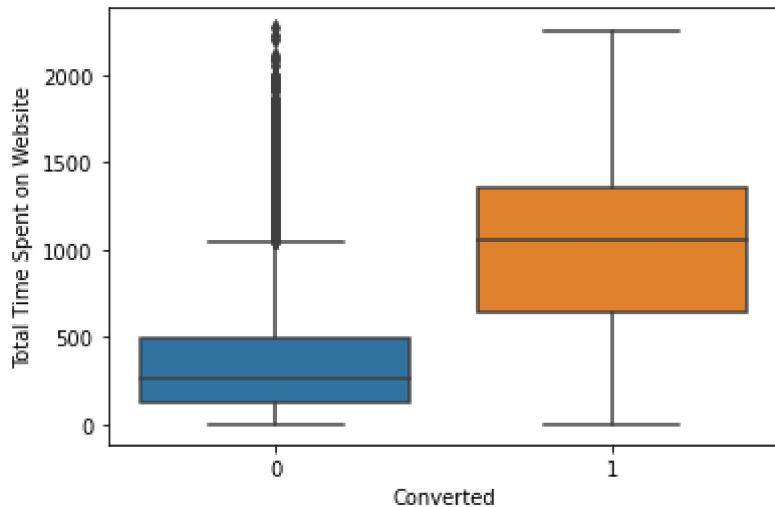


Inference

1. Median for converted and not converted leads are the close.
2. Nothing conclusive can be said on the basis of Total Visits

In [337...]

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

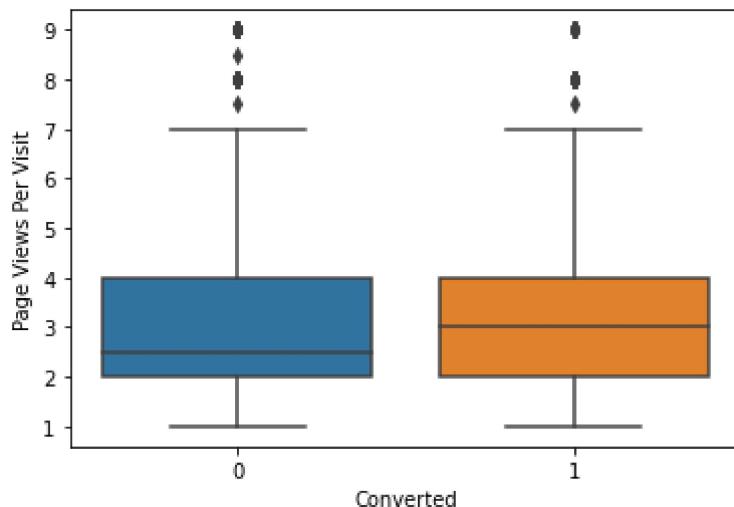


Inference

1. Leads spending more time on the website are more likely to be converted.
2. Website should be made more engaging to make leads spend more time.

In [338...]

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



### Inference

1. Median for converted and unconverted leads is the same.
2. Nothing can be said specifically for lead conversion from Page Views Per Visit

In [339...]

```
#checking missing values in leftover columns/
round(100*(leads.isnull().sum()/len(leads.index)),2)
```

Out[339...]

Lead Origin	0.0
Lead Source	0.0
Do Not Email	0.0
Converted	0.0
TotalVisits	0.0
Total Time Spent on Website	0.0
Page Views Per Visit	0.0
Last Activity	0.0
Specialization	0.0
What is your current occupation	0.0
Tags	0.0
City	0.0
A free copy of Mastering The Interview	0.0
Last Notable Activity	0.0

dtype: float64

There are no missing values in the columns to be analyzed further

### Dummy Variable Creation:

In [340...]

```
#getting a list of categorical columns
cat_cols= leads.select_dtypes(include=['object']).columns
cat_cols
```

Out[340...]

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

In [341...]

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

```

varlist = ['A free copy of Mastering The Interview', 'Do Not Email']

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

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

```

In [342...]

```

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

leads = pd.concat([leads,dummy],1)

```

In [343...]

```

dummy = pd.get_dummies(leads['Specialization'], prefix = 'Specialization')
dummy = dummy.drop(['Specialization_Not Specified'], 1)
leads = pd.concat([leads, dummy], axis = 1)

```

In [344...]

```

dummy = pd.get_dummies(leads['Lead Source'], prefix = 'Lead Source')
dummy = dummy.drop(['Lead Source_Others'], 1)
leads = pd.concat([leads, dummy], axis = 1)

```

In [345...]

```

dummy = pd.get_dummies(leads['Last Activity'], prefix = 'Last Activity')
dummy = dummy.drop(['Last Activity_Others'], 1)
leads = pd.concat([leads, dummy], axis = 1)

```

In [346...]

```

dummy = pd.get_dummies(leads['Last Notable Activity'], prefix = 'Last Notable Activity')
dummy = dummy.drop(['Last Notable Activity_Other_Notable_activity'], 1)
leads = pd.concat([leads, dummy], axis = 1)

```

In [347...]

```

dummy = pd.get_dummies(leads['Tags'], prefix = 'Tags')
dummy = dummy.drop(['Tags_Not Specified'], 1)
leads = pd.concat([leads, dummy], axis = 1)

```

In [348...]

```

#dropping the original columns after dummy variable creation

leads.drop(cat_cols,1,inplace = True)

```

In [349...]

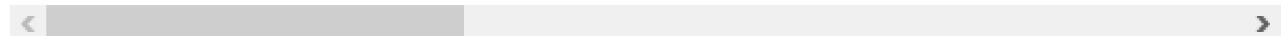
```
leads.head()
```

Out[349...]

Converted	TotalVisits	Total Time Spent on Website	Page Views Per Visit	Origin_Landing Page Submission	Lead Page	Lead Origin_Lead Add Form	What is your current occupation_Housewife	Wh occupati
1	0	5.0	674	2.5	0	0	0	0

	Converted	TotalVisits	Total Time Spent on Website	Page Views Per Visit	Lead Origin_Landing Page Submission	Lead Origin_Lead Add Form	What is your current occupation_Housewife	Wh occupati
2	1	2.0	1532	2.0	1	0	0	0
3	0	1.0	305	1.0	1	0	0	0
4	1	2.0	1428	1.0	1	0	0	0
6	1	2.0	1640	2.0	1	0	0	0

5 rows × 55 columns



#### Train-Test Split & Logistic Regression Model Building:

In [350...]

```
from sklearn.model_selection import train_test_split

# Putting response variable to y
y = leads['Converted']

y.head()

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

In [351...]

```
# Splitting the data into train and test
X_train, X_test, y_train, y_test = train_test_split(X, y, train_size=0.7, test_size=0.3)
```

In [352...]

```
X_train.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 4648 entries, 1979 to 7849
Data columns (total 54 columns):
 #   Column                                     Non-Null Count  Dtype  
--- 
 0   TotalVisits                                4648 non-null   float64
 1   Total Time Spent on Website                 4648 non-null   int64  
 2   Page Views Per Visit                       4648 non-null   float64
 3   Lead Origin_Landing Page Submission        4648 non-null   uint8  
 4   Lead Origin_Lead Add Form                  4648 non-null   uint8  
 5   What is your current occupation_Housewife 4648 non-null   uint8  
 6   What is your current occupation_Other       4648 non-null   uint8  
 7   What is your current occupation_Student     4648 non-null   uint8  
 8   What is your current occupation_Unemployed 4648 non-null   uint8  
 9   What is your current occupation_Working Professional 4648 non-null   uint8  
 10  City_Other Cities                          4648 non-null   uint8  
 11  City_Other Cities of Maharashtra          4648 non-null   uint8  
 12  City_Other Metro Cities                   4648 non-null   uint8  
 13  City_Thane & Outskirts                    4648 non-null   uint8  
 14  City_Tier II Cities                      4648 non-null   uint8  
 15  Specialization_Banking, Investment And Insurance 4648 non-null   uint8  
 16  Specialization_Business Administration    4648 non-null   uint8  
 17  Specialization_E-Business                 4648 non-null   uint8
```

```

18 Specialization_E-COMMERCE           4648 non-null  uint8
19 Specialization_International Business 4648 non-null  uint8
20 Specialization_Management_Specializations 4648 non-null  uint8
21 Specialization_Media and Advertising   4648 non-null  uint8
22 Specialization_Rural and Agribusiness 4648 non-null  uint8
23 Specialization_Services Excellence    4648 non-null  uint8
24 Specialization_Travel and Tourism     4648 non-null  uint8
25 Lead Source_Direct Traffic          4648 non-null  uint8
26 Lead Source_Google                 4648 non-null  uint8
27 Lead Source_Olark Chat             4648 non-null  uint8
28 Lead Source_Organic Search         4648 non-null  uint8
29 Lead Source_Reference              4648 non-null  uint8
30 Lead Source_Referral Sites         4648 non-null  uint8
31 Lead Source_Social Media           4648 non-null  uint8
32 Lead Source_Welingak Website      4648 non-null  uint8
33 Last Activity_Converted to Lead   4648 non-null  uint8
34 Last Activity_Email Bounced       4648 non-null  uint8
35 Last Activity_Email Link Clicked  4648 non-null  uint8
36 Last Activity_Email Opened        4648 non-null  uint8
37 Last Activity_Form Submitted on Website 4648 non-null  uint8
38 Last Activity_Olark Chat Conversation 4648 non-null  uint8
39 Last Activity_Page Visited on Website 4648 non-null  uint8
40 Last Activity_SMS Sent            4648 non-null  uint8
41 Last Notable Activity_Email Link Clicked 4648 non-null  uint8
42 Last Notable Activity_Email Opened 4648 non-null  uint8
43 Last Notable Activity_Modified    4648 non-null  uint8
44 Last Notable Activity_Olark Chat Conversation 4648 non-null  uint8
45 Last Notable Activity_Page Visited on Website 4648 non-null  uint8
46 Last Notable Activity_SMS Sent    4648 non-null  uint8
47 Tags_Busy                        4648 non-null  uint8
48 Tags_Closed by Horizzon          4648 non-null  uint8
49 Tags_Interested in other courses 4648 non-null  uint8
50 Tags_Lost to EIMS                4648 non-null  uint8
51 Tags_Other_Tags                 4648 non-null  uint8
52 Tags_Ringing                     4648 non-null  uint8
53 Tags_Will revert after reading the email 4648 non-null  uint8
dtypes: float64(2), int64(1), uint8(51)
memory usage: 376.7 KB

```

### Scaling of Data:

In [353...]

```

#scaling numeric columns

from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()

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

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

X_train.head()

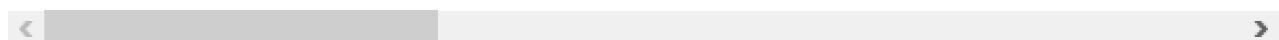
```

Out[353...]

TotalVisits	Total Time Spent on Website	Page Views Per Visit	Lead Origin_Landing Page Submission	Lead Origin_Lead Add Form	What is your current occupation_Housewife	What is cu occupation_(
-------------	-----------------------------	----------------------	-------------------------------------	---------------------------	---	-------------------------

	TotalVisits	Total Time Spent on Website	Page Views Per Visit	Lead Origin_Landing Page Submission	Lead Origin_Lead Add Form	What is your current occupation_Housewife	What is cu occupation_Other
1979	0.668033	1.836717	1.846001	1	0	0	
6249	1.418399	1.230405	-0.225650	1	0	0	
5557	1.793583	1.857051	3.712353	1	0	0	
7865	1.793583	-0.645835	3.712353	1	0	0	
8491	-0.082334	0.086176	0.601766	1	0	0	

5 rows × 54 columns



### Model Building using Stats Model & RFE:

In [354...]

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

In [355...]

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

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

In [356...]

```
rfe.support_
```

Out[356...]

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

In [357...]

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

Out[357...]

```
[('TotalVisits', False, 29),
 ('Total Time Spent on Website', True, 1),
 ('Page Views Per Visit', False, 31),
 ('Lead Origin_Landing Page Submission', False, 6),
 ('Lead Origin_Lead Add Form', True, 1),
 ('What is your current occupation_Housewife', False, 27),
 ('What is your current occupation_Other', False, 32),
 ('What is your current occupation_Student', False, 17),
 ('What is your current occupation_Unemployed', False, 4),
 ('What is your current occupation_Working Professional', True, 1),
 ('City_Other Cities', False, 40),
 ('City_Other Cities of Maharashtra', False, 35),
 ('City_Other Metro Cities', False, 28),
 ('City_Thane & Outskirts', False, 38),
 ('City_Tier II Cities', False, 7),
```

```
('Specialization_Banking, Investment And Insurance', False, 14),
('Specialization_Business Administration', False, 11),
('Specialization_E-Business', False, 19),
('Specialization_E-COMMERCE', False, 9),
('Specialization_International Business', False, 39),
('Specialization_Management_Specializations', False, 15),
('Specialization_Media and Advertising', False, 12),
('Specialization_Rural and Agribusiness', False, 10),
('Specialization_Services Excellence', False, 20),
('Specialization_Travel and Tourism', False, 23),
('Lead Source_Direct Traffic', False, 22),
('Lead Source_Google', False, 34),
('Lead Source_Olark Chat', False, 13),
('Lead Source_Organic Search', False, 33),
('Lead Source_Reference', False, 36),
('Lead Source_Referral Sites', False, 21),
('Lead Source_Social Media', False, 3),
('Lead Source_Welingak Website', False, 8),
('Last Activity_Converted to Lead', True, 1),
('Last Activity_Email Bounced', True, 1),
('Last Activity_Email Link Clicked', False, 18),
('Last Activity_Email Opened', False, 25),
('Last Activity_Form Submitted on Website', False, 26),
('Last Activity_Olark Chat Conversation', False, 37),
('Last Activity_Page Visited on Website', True, 1),
('Last Activity_SMS Sent', False, 2),
('Last Notable Activity_Email Link Clicked', False, 16),
('Last Notable Activity_Email Opened', False, 24),
('Last Notable Activity_Modified', True, 1),
('Last Notable Activity_Olark Chat Conversation', False, 5),
('Last Notable Activity_Page Visited on Website', False, 30),
('Last Notable Activity_SMS Sent', True, 1),
('Tags_Busy', True, 1),
('Tags_Closed by Horizzon', True, 1),
('Tags_Interested in other courses', True, 1),
('Tags_Lost to EINS', True, 1),
('Tags_Other_Tags', True, 1),
('Tags_Ringing', True, 1),
('Tags_Will revert after reading the email', True, 1)]
```

In [358...]

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

Out[358...]

```
Index(['Total Time Spent on Website', 'Lead Origin_Lead Add Form',
       'What is your current occupation_Working Professional',
       'Last Activity_Converted to Lead', 'Last Activity_Email Bounced',
       'Last Activity_Page Visited on Website',
       'Last Notable Activity_Modified', 'Last Notable Activity_SMS Sent',
       'Tags_Busy', 'Tags_Closed by Horizzon',
       'Tags_Interested in other courses', 'Tags_Lost to EINS',
       'Tags_Other_Tags', 'Tags_Ringing',
       'Tags_Will revert after reading the email'],
      dtype='object')
```

In [359...]

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

Out[359...]

```
Index(['TotalVisits', 'Page Views Per Visit',
       'Lead Origin_Landing Page Submission',
```

```
'What is your current occupation_Housewife',
'What is your current occupation_Other',
'What is your current occupation_Student',
'What is your current occupation_Unemployed', 'City_Other Cities',
'City_Other Cities of Maharashtra', 'City_Other Metro Cities',
'City_Thane & Outskirts', 'City_Tier II Cities',
'Specialization_Banking, Investment And Insurance',
'Specialization_Business Administration', 'Specialization_E-Business',
'Specialization_E-COMMERCE', 'Specialization_International Business',
'Specialization_Management_Specializations',
'Specialization_Media and Advertising',
'Specialization_Rural and Agribusiness',
'Specialization_Services Excellence',
'Specialization_Travel and Tourism', 'Lead Source_Direct Traffic',
'Lead Source_Google', 'Lead Source_Olark Chat',
'Lead Source_Organic Search', 'Lead Source_Reference',
'Lead Source_Referral Sites', 'Lead Source_Social Media',
'Lead Source_Welingak Website', 'Last Activity_Email Link Clicked',
'Last Activity_Email Opened', 'Last Activity_Form Submitted on Website',
'Last Activity_Olark Chat Conversation', 'Last Activity_SMS Sent',
'Last Notable Activity_Email Link Clicked',
'Last Notable Activity_Email Opened',
'Last Notable Activity_Olark Chat Conversation',
'Last Notable Activity_Page Visited on Website'],
dtype='object')
```

In [360...]

```
#BUILDING MODEL #1

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

Out[360...]

## Generalized Linear Model Regression Results

<b>Dep. Variable:</b>	Converted	<b>No. Observations:</b>	4648
<b>Model:</b>	GLM	<b>Df Residuals:</b>	4632
<b>Model Family:</b>	Binomial	<b>Df Model:</b>	15
<b>Link Function:</b>	logit	<b>Scale:</b>	1.0000
<b>Method:</b>	IRLS	<b>Log-Likelihood:</b>	-873.41
<b>Date:</b>	Sat, 10 Sep 2022	<b>Deviance:</b>	1746.8
<b>Time:</b>	14:50:20	<b>Pearson chi2:</b>	6.21e+03
<b>No. Iterations:</b>	24		
<b>Covariance Type:</b>	nonrobust		

		coef	std err	z	P> z	[0.025	0.975]
	<b>const</b>	-1.4676	0.107	-13.703	0.000	-1.678	-1.258
	<b>Total Time Spent on Website</b>	1.1229	0.065	17.281	0.000	0.996	1.250
	<b>Lead Origin_Lead Add Form</b>	23.9666	4.01e+04	0.001	1.000	-7.85e+04	7.86e+04
	<b>What is your current occupation_Working Professional</b>	1.3016	0.506	2.572	0.010	0.310	2.293

<b>Last Activity_Converted to Lead</b>	-0.8961	0.359	-2.498	0.012	-1.599	-0.193
<b>Last Activity_Email Bounced</b>	-1.6560	0.541	-3.062	0.002	-2.716	-0.596
<b>Last Activity_Page Visited on Website</b>	-0.8274	0.277	-2.992	0.003	-1.369	-0.285
<b>Last Notable Activity_Modified</b>	-0.4829	0.164	-2.943	0.003	-0.804	-0.161
<b>Last Notable Activity_SMS Sent</b>	1.8894	0.157	12.048	0.000	1.582	2.197
<b>Tags_Busy</b>	0.9760	0.247	3.957	0.000	0.493	1.459
<b>Tags_Closed by Horizzon</b>	26.8976	1.82e+04	0.001	0.999	-3.56e+04	3.57e+04
<b>Tags_Interested in other courses</b>	-2.1792	0.427	-5.104	0.000	-3.016	-1.342
<b>Tags_Lost to EINS</b>	5.1003	0.620	8.226	0.000	3.885	6.315
<b>Tags_Other_Tags</b>	-2.4656	0.241	-10.241	0.000	-2.937	-1.994
<b>Tags_Ringing</b>	-3.3904	0.285	-11.909	0.000	-3.948	-2.832
<b>Tags_Will revert after reading the email</b>	4.5943	0.232	19.839	0.000	4.140	5.048

p-value of variable Lead Origin\_Lead Add Form is high, so we can drop it.

In [361...]

```
#dropping column with high p-value
col = col.drop('Lead Origin_Lead Add Form',1)
```

In [362...]

```
#BUILDING MODEL #2

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

Out[362...]

### Generalized Linear Model Regression Results

<b>Dep. Variable:</b>	Converted	<b>No. Observations:</b>	4648
<b>Model:</b>	GLM	<b>Df Residuals:</b>	4633
<b>Model Family:</b>	Binomial	<b>Df Model:</b>	14
<b>Link Function:</b>	logit	<b>Scale:</b>	1.0000
<b>Method:</b>	IRLS	<b>Log-Likelihood:</b>	-876.03
<b>Date:</b>	Sat, 10 Sep 2022	<b>Deviance:</b>	1752.1
<b>Time:</b>	14:50:20	<b>Pearson chi2:</b>	6.22e+03
<b>No. Iterations:</b>	23		
<b>Covariance Type:</b>	nonrobust		

	<b>coef</b>	<b>std err</b>	<b>z</b>	<b>P&gt; z </b>	<b>[0.025</b>	<b>0.975]</b>
<b>const</b>	-1.4565	0.107	-13.650	0.000	-1.666	-1.247
<b>Total Time Spent on Website</b>	1.1256	0.065	17.333	0.000	0.998	1.253

<b>What is your current occupation_Working Professional</b>	1.2994	0.505	2.571	0.010	0.309	2.290
<b>Last Activity_Converted to Lead</b>	-0.8890	0.358	-2.480	0.013	-1.591	-0.187
<b>Last Activity_Email Bounced</b>	-1.6548	0.540	-3.062	0.002	-2.714	-0.595
<b>Last Activity_Page Visited on Website</b>	-0.7755	0.272	-2.849	0.004	-1.309	-0.242
<b>Last Notable Activity_Modified</b>	-0.4988	0.164	-3.045	0.002	-0.820	-0.178
<b>Last Notable Activity_SMS Sent</b>	1.8808	0.157	12.006	0.000	1.574	2.188
<b>Tags_Busy</b>	0.9679	0.246	3.927	0.000	0.485	1.451
<b>Tags_Closed by Horizzon</b>	26.0114	1.13e+04	0.002	0.998	-2.21e+04	2.22e+04
<b>Tags_Interested in other courses</b>	-2.1845	0.427	-5.116	0.000	-3.021	-1.348
<b>Tags_Lost to EINS</b>	5.0986	0.620	8.221	0.000	3.883	6.314
<b>Tags_Other_Tags</b>	-2.4732	0.241	-10.275	0.000	-2.945	-2.001
<b>Tags_Ringing</b>	-3.3970	0.285	-11.933	0.000	-3.955	-2.839
<b>Tags_Will revert after reading the email</b>	4.5870	0.231	19.820	0.000	4.133	5.041

p-value of variable Tags\_Closed by Horizzon is high, so we can drop it.

In [363...]

```
#dropping column with high p-value
col = col.drop('Tags_Closed by Horizzon',1)
```

In [364...]

```
#BUILDING MODEL #2

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

Out[364...]

### Generalized Linear Model Regression Results

<b>Dep. Variable:</b>	Converted	<b>No. Observations:</b>	4648
<b>Model:</b>	GLM	<b>Df Residuals:</b>	4634
<b>Model Family:</b>	Binomial	<b>Df Model:</b>	13
<b>Link Function:</b>	logit	<b>Scale:</b>	1.0000
<b>Method:</b>	IRLS	<b>Log-Likelihood:</b>	-1002.0
<b>Date:</b>	Sat, 10 Sep 2022	<b>Deviance:</b>	2004.0
<b>Time:</b>	14:50:20	<b>Pearson chi2:</b>	6.73e+03
<b>No. Iterations:</b>	7		
<b>Covariance Type:</b>	nonrobust		

	coef	std err	z	P> z	[0.025	0.975]
<b>const</b>	-1.1852	0.097	-12.165	0.000	-1.376	-0.994

<b>Total Time Spent on Website</b>	1.1275	0.060	18.689	0.000	1.009	1.246
<b>What is your current occupation_Working Professional</b>	2.5887	0.383	6.752	0.000	1.837	3.340
<b>Last Activity_Converted to Lead</b>	-1.2933	0.342	-3.780	0.000	-1.964	-0.623
<b>Last Activity_Email Bounced</b>	-1.8427	0.482	-3.819	0.000	-2.788	-0.897
<b>Last Activity_Page Visited on Website</b>	-0.4957	0.219	-2.259	0.024	-0.926	-0.066
<b>Last Notable Activity_Modified</b>	-0.2242	0.142	-1.581	0.114	-0.502	0.054
<b>Last Notable Activity_SMS Sent</b>	1.7159	0.153	11.210	0.000	1.416	2.016
<b>Tags_Busy</b>	0.6781	0.241	2.818	0.005	0.207	1.150
<b>Tags_Interested in other courses</b>	-2.7184	0.430	-6.322	0.000	-3.561	-1.876
<b>Tags_Lost to EINS</b>	4.7163	0.624	7.559	0.000	3.493	5.939
<b>Tags_Other_Tags</b>	-2.7707	0.238	-11.655	0.000	-3.237	-2.305
<b>Tags_Ringing</b>	-3.6068	0.283	-12.725	0.000	-4.162	-3.051
<b>Tags_Will revert after reading the email</b>	4.1923	0.224	18.680	0.000	3.752	4.632

p-value of variable Last Notable Activity\_Modified is high, so we can drop it.

```
In [365...]: #dropping column with high p-value
          col = col.drop('Last Notable Activity_Modified', 1)
```

```
In [366...]: #BUILDING MODEL #3
X_train_sm = sm.add_constant(X_train[col])
logm2 = sm.GLM(y_train,X_train_sm, family = sm.families.Binomial())
res = logm2.fit()
res.summary()
```

Out[366...]: Generalized Linear Model Regression Results

<b>Dep. Variable:</b>	Converted	<b>No. Observations:</b>	4648
<b>Model:</b>	GLM	<b>Df Residuals:</b>	4635
<b>Model Family:</b>	Binomial	<b>Df Model:</b>	12
<b>Link Function:</b>	logit	<b>Scale:</b>	1.0000
<b>Method:</b>	IRLS	<b>Log-Likelihood:</b>	-1003.3
<b>Date:</b>	Sat, 10 Sep 2022	<b>Deviance:</b>	2006.5
<b>Time:</b>	14:52:20	<b>Pearson chi2:</b>	6.94e+03
<b>No. Iterations:</b>	7		
<b>Covariance Type:</b>	nonrobust		

	coef	std err	z	P> z	[0.025	0.975]
<b>const</b>	-1.2716	0.082	-15.563	0.000	-1.432	-1.111

## Case Study - Leads Scoring

<b>Total Time Spent on Website</b>	1.1247	0.060	18.673	0.000	1.007	1.243
<b>What is your current occupation_Working Professional</b>	2.6097	0.384	6.788	0.000	1.856	3.363
<b>Last Activity_Converted to Lead</b>	-1.4313	0.331	-4.323	0.000	-2.080	-0.782
<b>Last Activity_Email Bounced</b>	-1.9216	0.479	-4.011	0.000	-2.861	-0.983
<b>Last Activity_Page Visited on Website</b>	-0.5171	0.219	-2.364	0.018	-0.946	-0.088
<b>Last Notable Activity_SMS Sent</b>	1.8006	0.144	12.504	0.000	1.518	2.083
<b>Tags_Busy</b>	0.6753	0.241	2.803	0.005	0.203	1.148
<b>Tags_Interested in other courses</b>	-2.7730	0.429	-6.461	0.000	-3.614	-1.932
<b>Tags_Lost to EINS</b>	4.6804	0.624	7.496	0.000	3.457	5.904
<b>Tags_Other_Tags</b>	-2.7670	0.238	-11.633	0.000	-3.233	-2.301
<b>Tags_Ringing</b>	-3.5927	0.284	-12.671	0.000	-4.148	-3.037
<b>Tags_Will revert after reading the email</b>	4.2091	0.224	18.793	0.000	3.770	4.648

p-value of variable Last Activity\_Page Visited on Website is high so we can drop it

In [367...]

```
#dropping column with high p-value
col = col.drop('Last Activity_Page Visited on Website',1)
```

In [368...]

```
#BUILDING MODEL #4

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

Out[368...]

## Generalized Linear Model Regression Results

<b>Dep. Variable:</b>	Converted	<b>No. Observations:</b>	4648
<b>Model:</b>	GLM	<b>Df Residuals:</b>	4636
<b>Model Family:</b>	Binomial	<b>Df Model:</b>	11
<b>Link Function:</b>	logit	<b>Scale:</b>	1.0000
<b>Method:</b>	IRLS	<b>Log-Likelihood:</b>	-1006.2
<b>Date:</b>	Sat, 10 Sep 2022	<b>Deviance:</b>	2012.4
<b>Time:</b>	14:54:16	<b>Pearson chi2:</b>	6.89e+03
<b>No. Iterations:</b>	7		
<b>Covariance Type:</b>	nonrobust		

	coef	std err	z	P> z	[0.025	0.975]
<b>const</b>	-1.3342	0.078	-17.090	0.000	-1.487	-1.181
<b>Total Time Spent on Website</b>	1.1176	0.060	18.646	0.000	1.000	1.235

<b>What is your current occupation_Working Professional</b>	2.5705	0.389	6.603	0.000	1.807	3.334
<b>Last Activity_Converted to Lead</b>	-1.3744	0.331	-4.153	0.000	-2.023	-0.726
<b>Last Activity_Email Bounced</b>	-1.8573	0.479	-3.878	0.000	-2.796	-0.919
<b>Last Notable Activity_SMS Sent</b>	1.8629	0.142	13.130	0.000	1.585	2.141
<b>Tags_Busy</b>	0.6630	0.241	2.756	0.006	0.192	1.135
<b>Tags_Interested in other courses</b>	-2.7372	0.427	-6.408	0.000	-3.574	-1.900
<b>Tags_Lost to EINS</b>	4.7141	0.623	7.561	0.000	3.492	5.936
<b>Tags_Other_Tags</b>	-2.7626	0.237	-11.646	0.000	-3.228	-2.298
<b>Tags_Ringing</b>	-3.5794	0.283	-12.628	0.000	-4.135	-3.024
<b>Tags_Will revert after reading the email</b>	4.2244	0.224	18.873	0.000	3.786	4.663

p-value of variable Tags\_Busy is high so we can drop it

In [369...]

```
#dropping column with high p-value
col = col.drop('Tags_Busy',1)
```

In [370...]

```
#BUILDING MODEL #5

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

Out[370...]

### Generalized Linear Model Regression Results

<b>Dep. Variable:</b>	Converted	<b>No. Observations:</b>	4648
<b>Model:</b>	GLM	<b>Df Residuals:</b>	4637
<b>Model Family:</b>	Binomial	<b>Df Model:</b>	10
<b>Link Function:</b>	logit	<b>Scale:</b>	1.0000
<b>Method:</b>	IRLS	<b>Log-Likelihood:</b>	-1010.0
<b>Date:</b>	Sat, 10 Sep 2022	<b>Deviance:</b>	2020.0
<b>Time:</b>	14:56:46	<b>Pearson chi2:</b>	6.91e+03
<b>No. Iterations:</b>	7		
<b>Covariance Type:</b>	nonrobust		

	coef	std err	z	P> z	[0.025	0.975]
<b>const</b>	-1.2907	0.076	-17.010	0.000	-1.439	-1.142
<b>Total Time Spent on Website</b>	1.1140	0.060	18.673	0.000	0.997	1.231
<b>What is your current occupation_Working Professional</b>	2.5490	0.391	6.520	0.000	1.783	3.315
<b>Last Activity_Converted to Lead</b>	-1.3894	0.330	-4.211	0.000	-2.036	-0.743
<b>Last Activity_Email Bounced</b>	-1.8856	0.477	-3.953	0.000	-2.821	-0.951

<b>Last Notable Activity_SMS Sent</b>	1.9213	0.141	13.639	0.000	1.645	2.197
<b>Tags_Interested in other courses</b>	-2.7810	0.427	-6.512	0.000	-3.618	-1.944
<b>Tags_Lost to EINS</b>	4.6742	0.623	7.497	0.000	3.452	5.896
<b>Tags_Other_Tags</b>	-2.8286	0.237	-11.956	0.000	-3.292	-2.365
<b>Tags_Ringing</b>	-3.6596	0.283	-12.941	0.000	-4.214	-3.105
<b>Tags_Will revert after reading the email</b>	4.1764	0.223	18.712	0.000	3.739	4.614

Since 'All' the p-values are less we can check the Variance Inflation Factor to see if there is any correlation between the variables

In [371...]

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

In [372...]

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

Out[372...]

	Features	VIF
9	Tags_Will revert after reading the email	1.59
4	Last Notable Activity_SMS Sent	1.38
1	What is your current occupation_Working Profes...	1.26
0	Total Time Spent on Website	1.16
7	Tags_Other_Tags	1.11
8	Tags_Ringing	1.11
2	Last Activity_Converted to Lead	1.07
3	Last Activity_Email Bounced	1.04
5	Tags_Interested in other courses	1.03
6	Tags_Lost to EINS	1.01

So the Values all seem to be in order so now, Moving on to derive the Probabilities, Lead Score, Predictions on Train Data:

In [373...]

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

Out[373...]

1979	0.935633
6249	0.519975
5557	0.113992
7865	0.003437

```
8491    0.007735
2573    0.131868
474     0.024910
4477    0.972852
1952    0.072865
1816    0.000739
dtype: float64
```

In [374...]  
`y_train_pred = y_train_pred.values.reshape(-1)  
y_train_pred[:10]`

Out[374...]  
`array([9.35632849e-01, 5.19975073e-01, 1.13991916e-01, 3.43695224e-03,
 7.73465966e-03, 1.31867572e-01, 2.49103360e-02, 9.72851784e-01,
 7.28647705e-02, 7.39024107e-04])`

In [375...]  
`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[375...]  

	Converted	Converted_prob	Prospect ID
<b>0</b>	1	0.935633	1979
<b>1</b>	0	0.519975	6249
<b>2</b>	0	0.113992	5557
<b>3</b>	0	0.003437	7865
<b>4</b>	0	0.007735	8491

In [376...]  
`y_train_pred_final['Predicted'] = y_train_pred_final.Converted_prob.map(lambda x: 1 if
# Let's see the head
y_train_pred_final.head()`

Out[376...]  

	Converted	Converted_prob	Prospect ID	Predicted
<b>0</b>	1	0.935633	1979	1
<b>1</b>	0	0.519975	6249	1
<b>2</b>	0	0.113992	5557	0
<b>3</b>	0	0.003437	7865	0
<b>4</b>	0	0.007735	8491	0

In [377...]  
`from sklearn import metrics

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

[[2807 154]
 [ 263 1424]]`

```
In [378...]: # Let's check the overall accuracy.
print(metrics.accuracy_score(y_train_pred_final.Converted, y_train_pred_final.Predicted))

0.9102839931153184

In [379...]: TP = confusion[1,1] # true positive
TN = confusion[0,0] # true negatives
FP = confusion[0,1] # false positives
FN = confusion[1,0] # false negatives

In [380...]: # Let's see the sensitivity of our Logistic regression model
TP / float(TP+FN)

Out[380...]: 0.8441019561351512

In [381...]: # Let us calculate specificity
TN / float(TN+FP)

Out[381...]: 0.9479905437352246

In [382...]: # Calculate False Positive Rate - predicting conversion when customer does not have conv
print(FP / float(TN+FP))

0.05200945626477541

In [383...]: # positive predictive value
print (TP / float(TP+FP))

0.9024081115335868

In [384...]: # Negative predictive value
print (TN / float(TN+ FN))

0.9143322475570033



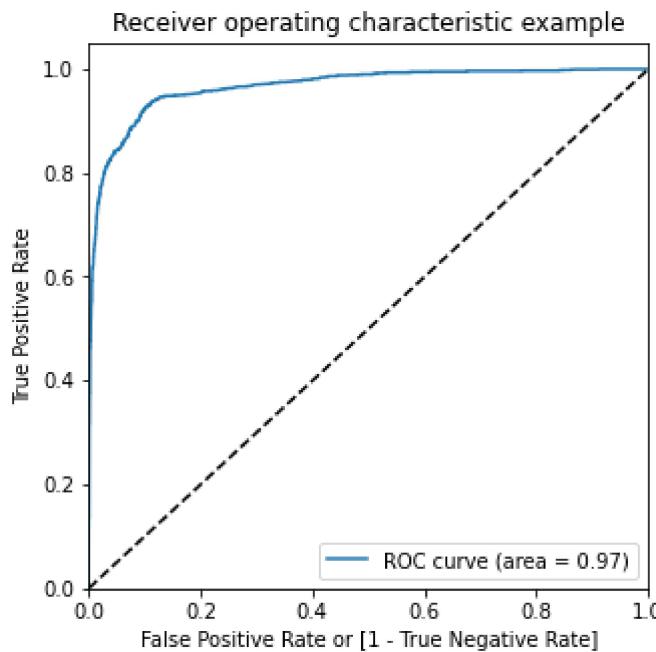
## PLOTTING ROC CURVE


In [385...]: 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 [386... fpr, tpr, thresholds = metrics.roc_curve( y_train_pred_final.Converted, y_train_pred_fi
```

```
In [387... draw_roc(y_train_pred_final.Converted, y_train_pred_final.Converted_prob)
```



The ROC Curve should be a value close to 1. We are getting a good value of 0.97 indicating a good predictive model.

### Finding Optimal Cutoff Point

Above we had chosen an arbitrary cut-off value of 0.5. We need to determine the best cut-off value and the below section deals with that:

```
In [388... # 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()
```

	Converted	Converted_prob	Prospect ID	Predicted	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
<b>0</b>	1	0.935633	1979	1	1	1	1	1	1	1	1	1	1	1
<b>1</b>	0	0.519975	6249	1	1	1	1	1	1	1	0	0	0	0
<b>2</b>	0	0.113992	5557	0	1	1	0	0	0	0	0	0	0	0
<b>3</b>	0	0.003437	7865	0	1	0	0	0	0	0	0	0	0	0
<b>4</b>	0	0.007735	8491	0	1	0	0	0	0	0	0	0	0	0

```
In [389... # Now let's calculate accuracy sensitivity and specificity for various probability cutoffs
cutoff_df = pd.DataFrame( columns = ['prob','accuracy','sensi','speci'])
from sklearn.metrics import confusion_matrix
```

```

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

num = [0.0,0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9]
for i in num:
    cm1 = metrics.confusion_matrix(y_train_pred_final.Converted, y_train_pred_final[i])
    total1=sum(sum(cm1))
    accuracy = (cm1[0,0]+cm1[1,1])/total1

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

```

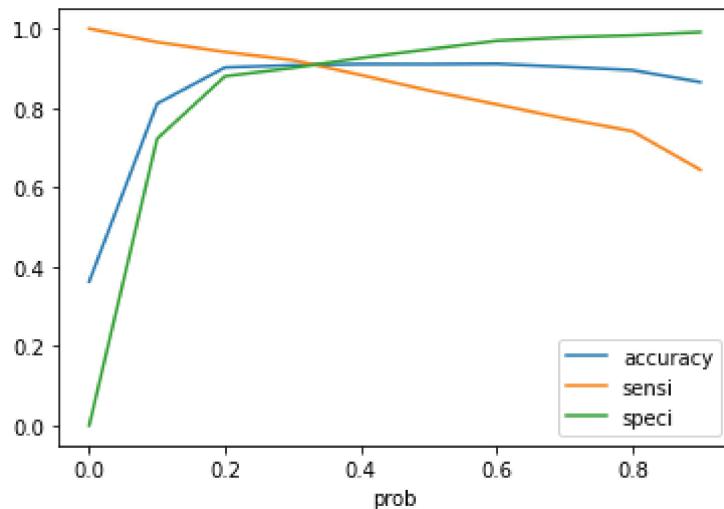
	prob	accuracy	sensi	speci
0.0	0.0	0.362952	1.000000	0.000000
0.1	0.1	0.810886	0.966212	0.722391
0.2	0.2	0.902324	0.941316	0.880108
0.3	0.3	0.908133	0.920569	0.901047
0.4	0.4	0.910499	0.883225	0.926039
0.5	0.5	0.910284	0.844102	0.947991
0.6	0.6	0.911360	0.809129	0.969605
0.7	0.7	0.903830	0.773563	0.978048
0.8	0.8	0.895439	0.741553	0.983114
0.9	0.9	0.865318	0.644339	0.991219

In [390...]

```

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

```



In [391...]

```

#### From the curve above, 0.3 is the optimum point to take it as a cutoff probability.

y_train_pred_final['final_Predicted'] = y_train_pred_final.Converted_prob.map( lambda x
y_train_pred_final.head()

```

Out[391...]

	Converted	Converted_prob	Prospect ID	Predicted	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	fl
0	1	0.935633	1979	1	1	1	1	1	1	1	1	1	1	1	1
1	0	0.519975	6249	1	1	1	1	1	1	1	0	0	0	0	0
2	0	0.113992	5557	0	1	1	0	0	0	0	0	0	0	0	0
3	0	0.003437	7865	0	1	0	0	0	0	0	0	0	0	0	0
4	0	0.007735	8491	0	1	0	0	0	0	0	0	0	0	0	0

In [392...]

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

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

Out[392...]

	Converted	Converted_prob	Prospect ID	final_Predicted	Lead_Score
0	1	0.935633	1979	1	94
1	0	0.519975	6249	1	52
2	0	0.113992	5557	0	11
3	0	0.003437	7865	0	0
4	0	0.007735	8491	0	1

In [393...]

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

Out[393...]

0.9081325301204819

In [394...]

```
confusion2 = metrics.confusion_matrix(y_train_pred_final.Converted, y_train_pred_final.final_Predicted)

confusion2
```

Out[394...]

```
array([[2668, 293],
       [134, 1553]], dtype=int64)
```

In [395...]

```
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 [396...]

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

Out[396...]

0.920569057498518

In [397...]

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

Out[397... 0.901046943600135

Observation:

So as we can see above the model seems to be performing well. The ROC curve has a value of 0.97, which is very good. We have the following values for the Train Data:

Accuracy : 90.81%

Sensitivity : 92.05%

Specificity : 90.10%

Some of the other Stats are derived below, indicating the False Positive Rate, Positive Predictive Value,Negative Predictive Values, Precision & Recall.

In [398... # Calculate False Postive Rate - predicting conversion when customer does not have conv  
print(FP/ float(TN+FP))

0.09895305639986492

In [399... # Positive predictive value  
print (TP / float(TP+FP))

0.8412784398699892

In [400... # Negative predictive value  
print (TN / float(TN+ FN))

0.9521770164168452

In [401... #Looking at the confusion matrix again

```
confusion = metrics.confusion_matrix(y_train_pred_final.Converted, y_train_pred_final.f  
confusion
```

Out[401... array([[2668, 293],  
[ 134, 1553]], dtype=int64)

In [402... ##### Precision  
TP / TP + FP  
  
confusion[1,1]/(confusion[0,1]+confusion[1,1])

Out[402... 0.8412784398699892

In [403... ##### Recall  
TP / TP + FN  
  
confusion[1,1]/(confusion[1,0]+confusion[1,1])

Out[403... 0.920569057498518

```
In [404...]: from sklearn.metrics import precision_score, recall_score
```

```
In [405...]: precision_score(y_train_pred_final.Converted , y_train_pred_final.final_Predicted)
```

```
Out[405...]: 0.8412784398699892
```

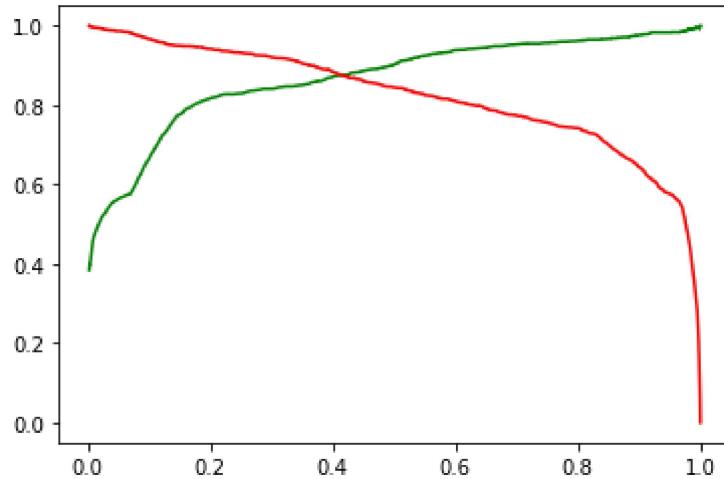
```
In [406...]: recall_score(y_train_pred_final.Converted, y_train_pred_final.final_Predicted)
```

```
Out[406...]: 0.920569057498518
```

```
In [407...]: from sklearn.metrics import precision_recall_curve
```

```
In [408...]: y_train_pred_final.Converted, y_train_pred_final.final_Predicted
p, r, thresholds = precision_recall_curve(y_train_pred_final.Converted, y_train_pred_final.final_Predicted)
```

```
In [409...]: plt.plot(thresholds, p[:-1], "g-")
plt.plot(thresholds, r[:-1], "r-")
plt.show()
```



```
In [410...]: #scaling test set

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

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

X_test.head()
```

```
Out[410...]:
```

TotalVisits	Total Time Spent on Website	Page Views Per Visit	Lead Origin_Landing Page Submission	Lead Origin_Lead Add Form	What is your current occupation_Housewife	What is cu occupation_(
8970	-1.144209	-0.560701	-1.266390	0	0	0

	TotalVisits	Total Time Spent on Website	Page Views Per Visit	Lead Origin_Landing Page Submission	Lead Origin_Lead Add Form	What is your current occupation_Housewife	What is cu occupation_C
8280	-0.783851	-0.236439	-0.628396	0	0	0	0
6941	-0.783851	-1.021301	-1.266390	0	0	0	0
7343	4.621518	0.028867	-0.545456	1	0	0	0
3734	-0.063135	-0.582810	0.647593	1	0	0	0

5 rows × 54 columns

In [411...]

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

Out[411...]

	Total Time Spent on Website	What is your current occupation_Working Professional	Last Activity_Converted to Lead	Last Activity_Email Bounced	Last Notable Activity_SMS Sent	Tags_Interested in other courses
8970	-0.560701	0	1	0	0	1
8280	-0.236439	0	0	0	0	0
6941	-1.021301	0	0	0	0	1
7343	0.028867	0	0	0	0	0
3734	-0.582810	0	0	0	0	0

In [412...]

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

## PREDICTIONS ON TEST SET

In [413...]

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

In [414...]

```
y_test_pred[:10]
```

Out[414...]

8970	0.002270
8280	0.174495
6941	0.005435
7343	0.948724
3734	0.125653
7889	0.598194
837	0.005358
8227	0.019962
7245	0.999419

```
8622    0.913947
dtype: float64
```

In [415...]

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

In [416...]

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

Out[416...]

	0
<b>8970</b>	0.002270
<b>8280</b>	0.174495
<b>6941</b>	0.005435
<b>7343</b>	0.948724
<b>3734</b>	0.125653

In [417...]

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

In [418...]

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

In [419...]

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

In [420...]

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

In [421...]

```
y_pred_final.head()
```

Out[421...]

	Converted	Prospect ID	0
<b>0</b>	0	8970	0.002270
<b>1</b>	0	8280	0.174495
<b>2</b>	0	6941	0.005435
<b>3</b>	0	7343	0.948724
<b>4</b>	0	3734	0.125653

In [422...]

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

In [423...]

```
y_pred_final.head()
```

Out[423...]

	Converted	Prospect ID	Converted_prob
<b>0</b>	0	8970	0.002270
<b>1</b>	0	8280	0.174495
<b>2</b>	0	6941	0.005435
<b>3</b>	0	7343	0.948724
<b>4</b>	0	3734	0.125653

In [424...]

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

In [425...]

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

Out[425...]

	Prospect ID	Converted	Converted_prob	Lead_Score
<b>0</b>	8970	0	0.002270	0
<b>1</b>	8280	0	0.174495	17
<b>2</b>	6941	0	0.005435	1
<b>3</b>	7343	0	0.948724	95
<b>4</b>	3734	0	0.125653	13

In [426...]

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

In [427...]

```
y_pred_final.head()
```

Out[427...]

	Prospect ID	Converted	Converted_prob	Lead_Score	final_Predicted
<b>0</b>	8970	0	0.002270	0	0
<b>1</b>	8280	0	0.174495	17	0
<b>2</b>	6941	0	0.005435	1	0
<b>3</b>	7343	0	0.948724	95	1
<b>4</b>	3734	0	0.125653	13	0

In [428...]

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

Out[428...]

```
0.9091365461847389
```

```
In [429... confusion2 = metrics.confusion_matrix(y_pred_final.Converted, y_pred_final.final_Predic
confusion2
```

```
Out[429... array([[1140, 118],
       [ 63, 671]], dtype=int64)
```

```
In [430... 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 [431... # Let's see the sensitivity of our Logistic regression model
TP / float(TP+FN)
```

```
Out[431... 0.9141689373297003
```

```
In [432... # Let us calculate specificity
TN / float(TN+FP)
```

```
Out[432... 0.9062003179650239
```

```
In [433... precision_score(y_pred_final.Converted , y_pred_final.final_Predicted)
```

```
Out[433... 0.8504435994930292
```

```
In [434... recall_score(y_pred_final.Converted, y_pred_final.final_Predicted)
```

```
Out[434... 0.9141689373297003
```

Observation:

After running the model on the Test Data these are the figures we obtain:

Accuracy : 90.92%

Sensitivity : 91.41%

Specificity : 90.62%

Final Observation:

Let us compare the values obtained for Train & Test:

Train Data:

Accuracy : 90.81%

Sensitivity : 92.05%

Specificity : 90.10%

Test Data:

Accuracy : 90.92%

Sensitivity : 91.41%

Specificity : 90.62%

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