Lead Scoring Case Study

Importing necessary libraries

In [1]:

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

#supressing warnings
import warnings
warnings.filterwarnings("ignore")

from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
import statsmodels.api as sm
from statsmodels.stats.outliers_influence import variance_inflation_factor
from sklearn.linear_model import LogisticRegression
from sklearn.feature_selection import RFE

# model evaluation
from sklearn import metrics
from sklearn.metrics import confusion_matrix
from sklearn.metrics import precision_score
from sklearn.metrics import precision_recall_curve

pd.set_option("display.max_columns",None)

plt.style.use("ggplot")
```

Reading the file

In [2]:

```
df = pd.read_csv("/home/arvin/Downloads/Lead+Scoring+Case+Study/Lead Scoring Assignment/Leads.csv")
df
```

Out[2]:

	Prospect ID	Lead Number	Lead Origin	Lead Source	Do Not Email	Do Not Call	Converted	TotalVisits	Total Time Spent on Website	Page Views Per Visit	Last Activity	Country	Specializatio
0	7927b2df-8bba-4d29-b9a2-b6e0beafe620	660737	API	Olark Chat	No	No	0	0.0	0	0.00	Page Visited on Website	NaN	Sele
1	2a272436-5132-4136-86fa-dcc88c88f482	660728	API	Organic Search	No	No	0	5.0	674	2.50	Email Opened	India	Sele
2	8cc8c611-a219-4f35-ad23-fdfd2656bd8a	660727	Landing Page Submission	Direct Traffic	No	No	1	2.0	1532	2.00	Email Opened	India	Busines Administratio
3	0cc2df48-7cf4-4e39-9de9-19797f9b38cc	660719	Landing Page Submission	Direct Traffic	No	No	0	1.0	305	1.00	Unreachable	India	Media an Advertisin
4	3256f628-e534-4826-9d63-4a8b88782852	660681	Landing Page Submission	Google	No	No	1	2.0	1428	1.00	Converted to Lead	India	Sele
...	...	...	...	...	...	...	...	...	...	...	...	...	.
9235	19d6451e-fcd6-407c-b83b-48e1af805ea9	579564	Landing Page Submission	Direct Traffic	Yes	No	1	8.0	1845	2.67	Email Marked Spam	Saudi Arabia	IT Project Manageme
9236	82a7005b-7196-4d56-95ce-a79f937a158d	579546	Landing Page Submission	Direct Traffic	No	No	0	2.0	238	2.00	SMS Sent	India	Media an Advertisin
9237	aac550fe-a586-452d-8d3c-f1b62c94e02c	579545	Landing Page Submission	Direct Traffic	Yes	No	0	2.0	199	2.00	SMS Sent	India	Busines Administratio
9238	5330a7d1-2f2b-4df4-85d6-64ca2f6b95b9	579538	Landing Page Submission	Google	No	No	1	3.0	499	3.00	SMS Sent	India	Huma Resourc Manageme
9239	571b5c8e-a5b2-4d57-8574-f2ffb06fdeff	579533	Landing Page Submission	Direct Traffic	No	No	1	6.0	1279	3.00	SMS Sent	Bangladesh	Supply Chai Manageme

9240 rows × 37 columns

In [3]:

```
#Check statistical summary of data frame
df.describe(percentiles=[0.25,.50,.75,.99])
```

Out[3]:

	Lead Number	Converted	TotalVisits	Total Time Spent on Website	Page Views Per Visit	Asymmetrique Activity Score	Asymmetrique Profile Score
count	9240.000000	9240.000000	9103.000000	9240.000000	9103.000000	5022.000000	5022.000000
mean	617188.435606	0.385390	3.445238	487.698268	2.362820	14.306252	16.344883
std	23405.995698	0.486714	4.854853	548.021466	2.161418	1.386694	1.811395
min	579533.000000	0.000000	0.000000	0.000000	0.000000	7.000000	11.000000
25%	596484.500000	0.000000	1.000000	12.000000	1.000000	14.000000	15.000000
50%	615479.000000	0.000000	3.000000	248.000000	2.000000	14.000000	16.000000
75%	637387.250000	1.000000	5.000000	936.000000	3.000000	15.000000	18.000000
99%	659592.980000	1.000000	17.000000	1840.610000	9.000000	17.000000	20.000000
max	660737.000000	1.000000	251.000000	2272.000000	55.000000	18.000000	20.000000

In [4]:

```
#Check data type, number of unique values, missing values percentage in each column
temp = {col:[df[col].dtype, df[col].nunique(), round(100*df[col].isnull().sum()/len(df[col]),2)] for col in df.
temp_df = pd.DataFrame(temp)
temp_df
```

Out[4]:

	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	Last Activity	Country	Specialization	How did you hear about X Education	occ
0	object	int64	object	object	object	object	int64	float64	int64	float64	object	object	object	object	
1	9240	9240	5	21	2	2	2	41	1731	114	17	38	19	10	
2	0.0	0.0	0.0	0.39	0.0	0.0	0.0	1.48	0.0	1.48	1.11	26.63	15.56	23.89	

In [5]:

```
#Checking the labels of the remaining categorical columns
for col in df.iloc[:,1:].select_dtypes(include='object').columns:
    print(col)
    print("___________________________________________________________________________________________________")
    print(df[col].value_counts(normalize= True)*100)
    print("___________________________________________________________________________________________________")
```

```
Lead Origin
___________________________________________________________________________________________________
Landing Page Submission    52.878788
API                        38.744589
Lead Add Form               7.770563
Lead Import                 0.595238
Quick Add Form              0.010823
Name: Lead Origin, dtype: float64
___________________________________________________________________________________________________
Lead Source
___________________________________________________________________________________________________
Google               31.160365
Direct Traffic       27.629292
Olark Chat           19.067797
Organic Search       12.538027
Reference             5.801825
Welingak Website      1.542807
Referral Sites        1.358105
Facebook              0.597566
bing                  0.065189
google                0.054324
Click2call            0.043459
Press_Release         0.021730
Social Media          0.021730
Live Chat             0.021730
youtubechannel        0.010865
testone               0.010865
Pay per Click Ads     0.010865
welearnblog_Home      0.010865
WeLearn               0.010865
blog                  0.010865
NC_EDM                0.010865
Name: Lead Source, dtype: float64
___________________________________________________________________________________________________
Do Not Email
___________________________________________________________________________________________________
No     92.056277
Yes     7.943723
Name: Do Not Email, dtype: float64
___________________________________________________________________________________________________
Do Not Call
___________________________________________________________________________________________________
No     99.978355
Yes     0.021645
Name: Do Not Call, dtype: float64
___________________________________________________________________________________________________
Last Activity
___________________________________________________________________________________________________
Email Opened                    37.616285
SMS Sent                        30.042684
Olark Chat Conversation         10.649010
Page Visited on Website          7.004487
Converted to Lead                4.684251
Email Bounced                    3.567911
Email Link Clicked               2.922185
Form Submitted on Website        1.269563
Unreachable                      1.017840
Unsubscribed                     0.667615
Had a Phone Conversation         0.328335
Approached upfront               0.098501
View in browser link Clicked     0.065667
Email Received                   0.021889
Email Marked Spam                0.021889
Visited Booth in Tradeshow       0.010945
Resubscribed to emails           0.010945
Name: Last Activity, dtype: float64
___________________________________________________________________________________________________
```

```
Country
________________________________________________________________________________
India                   95.766337
United States            1.017849
United Arab Emirates     0.781826
Singapore                0.354035
Saudi Arabia             0.309780
United Kingdom           0.221272
Australia                0.191769
Qatar                    0.147514
Hong Kong                0.103260
Bahrain                  0.103260
Oman                     0.088509
France                   0.088509
unknown                  0.073757
South Africa             0.059006
Nigeria                  0.059006
Germany                  0.059006
Kuwait                   0.059006
Canada                   0.059006
Sweden                   0.044254
China                    0.029503
Asia/Pacific Region      0.029503
Uganda                   0.029503
Bangladesh               0.029503
Italy                    0.029503
Belgium                  0.029503
Netherlands              0.029503
Ghana                    0.029503
Philippines              0.029503
Russia                   0.014751
Switzerland              0.014751
Vietnam                  0.014751
Denmark                  0.014751
Tanzania                 0.014751
Liberia                  0.014751
Malaysia                 0.014751
Kenya                    0.014751
Sri Lanka                0.014751
Indonesia                0.014751
Name: Country, dtype: float64
________________________________________________________________________________
Specialization
________________________________________________________________________________
Select                                 24.891054
Finance Management                     12.509613
Human Resource Management              10.869008
Marketing Management                   10.740836
Operations Management                   6.447065
Business Administration                 5.165342
IT Projects Management                  4.691105
Supply Chain Management                 4.473212
Banking, Investment And Insurance       4.332223
Travel and Tourism                      2.601897
Media and Advertising                   2.601897
International Business                  2.281466
Healthcare Management                   2.037939
Hospitality Management                  1.461164
E-COMMERCE                              1.435529
Retail Management                       1.281723
Rural and Agribusiness                  0.935658
E-Business                              0.730582
Services Excellence                     0.512689
Name: Specialization, dtype: float64
________________________________________________________________________________
How did you hear about X Education
________________________________________________________________________________
Select                   71.704820
Online Search            11.488696
Word Of Mouth             4.948102
Student of SomeSchool     4.407792
Other                     2.644675
Multiple Sources          2.161240
Advertisements            0.995308
Social Media              0.952652
Email                     0.369686
SMS                       0.327030
Name: How did you hear about X Education, dtype: float64
________________________________________________________________________________
What is your current occupation
________________________________________________________________________________
Unemployed              85.496183
Working Professional    10.778626
Student                  3.206107
Other                    0.244275
Housewife                0.152672
Businessman              0.122137
Name: What is your current occupation, dtype: float64
________________________________________________________________________________
```

```
What matters most to you in choosing a course
________________________________________________________________________________________________
Better Career Prospects      99.954065
Flexibility & Convenience     0.030623
Other                         0.015312
Name: What matters most to you in choosing a course, dtype: float64
________________________________________________________________________________________________
Search
________________________________________________________________________________________________
No     99.848485
Yes     0.151515
Name: Search, dtype: float64
________________________________________________________________________________________________
Magazine
________________________________________________________________________________________________
No    100.0
Name: Magazine, dtype: float64
________________________________________________________________________________________________
Newspaper Article
________________________________________________________________________________________________
No     99.978355
Yes     0.021645
Name: Newspaper Article, dtype: float64
________________________________________________________________________________________________
X Education Forums
________________________________________________________________________________________________
No     99.989177
Yes     0.010823
Name: X Education Forums, dtype: float64
________________________________________________________________________________________________
Newspaper
________________________________________________________________________________________________
No     99.989177
Yes     0.010823
Name: Newspaper, dtype: float64
________________________________________________________________________________________________
Digital Advertisement
________________________________________________________________________________________________
No     99.95671
Yes     0.04329
Name: Digital Advertisement, dtype: float64
________________________________________________________________________________________________
Through Recommendations
________________________________________________________________________________________________
No     99.924242
Yes     0.075758
Name: Through Recommendations, dtype: float64
________________________________________________________________________________________________
Receive More Updates About Our Courses
________________________________________________________________________________________________
No    100.0
Name: Receive More Updates About Our Courses, dtype: float64
________________________________________________________________________________________________
Tags
________________________________________________________________________________________________
Will revert after reading the email                  35.196195
Ringing                                              20.434856
Interested in other courses                           8.714116
Already a student                                     7.898760
Closed by Horizzon                                    6.081196
switched off                                          4.076779
Busy                                                  3.159504
Lost to EINS                                          2.972652
Not doing further education                           2.463054
Interested  in full time MBA                          1.987430
Graduation in progress                                1.885510
invalid number                                        1.409886
Diploma holder (Not Eligible)                         1.070155
wrong number given                                    0.798369
opp hangup                                            0.560557
number not provided                                   0.458638
in touch with EINS                                    0.203839
Lost to Others                                        0.118906
Still Thinking                                        0.101919
Want to take admission but has financial problems     0.101919
In confusion whether part time or DLP                 0.084933
Interested in Next batch                              0.084933
Lateral student                                       0.050960
Shall take in the next coming month                   0.033973
University not recognized                             0.033973
Recognition issue (DEC approval)                      0.016987
Name: Tags, dtype: float64
________________________________________________________________________________________________
Lead Quality
________________________________________________________________________________________________
Might be             34.875922
Not Sure             24.413146
High in Relevance    14.241002
Worst                13.436173
```

```
Low in Relevance     13.033758
Name: Lead Quality, dtype: float64
```

```
Update me on Supply Chain Content
```

```
No    100.0
Name: Update me on Supply Chain Content, dtype: float64
```

```
Get updates on DM Content
```

```
No    100.0
Name: Get updates on DM Content, dtype: float64
```

```
Lead Profile
```

```
Select                         63.481856
Potential Lead                 24.697596
Other Leads                     7.456745
Student of SomeSchool           3.690093
Lateral Student                 0.367478
Dual Specialization Student     0.306232
Name: Lead Profile, dtype: float64
```

```
City
```

```
Mumbai                         41.202046
Select                         28.759591
Thane & Outskirts               9.616368
Other Cities                    8.772379
Other Cities of Maharashtra     5.843990
Other Metro Cities              4.859335
Tier II Cities                  0.946292
Name: City, dtype: float64
```

```
Asymmetrique Activity Index
```

```
02.Medium    76.443648
01.High      16.348068
03.Low        7.208284
Name: Asymmetrique Activity Index, dtype: float64
```

```
Asymmetrique Profile Index
```

```
02.Medium    55.515731
01.High      43.866985
03.Low        0.617284
Name: Asymmetrique Profile Index, dtype: float64
```

```
I agree to pay the amount through cheque
```

```
No    100.0
Name: I agree to pay the amount through cheque, dtype: float64
```

```
A free copy of Mastering The Interview
```

```
No     68.744589
Yes    31.255411
Name: A free copy of Mastering The Interview, dtype: float64
```

```
Last Notable Activity
```

```
Modified                        36.872294
Email Opened                    30.595238
SMS Sent                        23.506494
Page Visited on Website          3.441558
Olark Chat Conversation          1.980519
Email Link Clicked               1.872294
Email Bounced                    0.649351
Unsubscribed                     0.508658
Unreachable                      0.346320
Had a Phone Conversation         0.151515
Email Marked Spam                0.021645
Approached upfront               0.010823
Resubscribed to emails           0.010823
View in browser link Clicked     0.010823
Form Submitted on Website        0.010823
Email Received                   0.010823
Name: Last Notable Activity, dtype: float64
```

OBSERVATIONS:

- Outliers exists in the numeric variables
- Columns with single values needs to be dropped
- Columns with more than 70% missing values needs to be removed
- Bivariate categorical variables needs to be encoded
- Missing values needs to be handled

- NaN values needs to be filled in place of 'Select'
- Too much variations in the columns ('Asymmetrique Activity Index','Asymmetrique Activity Score','Asymmetrique Profile Index','Asymmetrique Profile Score') and it is not safer to impute any values in the columns and hence we will drop these columns with very high percentage of missing data

Data Preparation, Preprocessing & Missing value treatment

In [6]:

```
#writing a funcion to preprocess, clean, replace missing values in the data


def pre_process(df):
    #Droping columns with single values throughout
    df.drop(['Magazine', 'Receive More Updates About Our Courses', 'I agree to pay the amount through cheque',

    #Dropping columns with too much variations in values and high NaN values
    df.drop(['Asymmetrique Activity Index','Asymmetrique Activity Score','Asymmetrique Profile Index','Asymmetri

    #Encoding the variables with yes/no labels
    encode_list = ['Do Not Email', 'Do Not Call', 'Search', 'Newspaper Article', 'X Education Forums', 'Newspap
    for col in encode_list:
        df[col].replace({'Yes':1, 'No':0}, inplace = True)

    #Converting all selects to NaN as the user didn't select any option from the list and "Select" is as good a
    df.replace('Select', np.nan, inplace = True)

    #Replacing Other with Other_Occupation in the column
    df['What is your current occupation'].replace("Other", 'Other_Occupation', inplace = True)

    #As Lead Quality depends on employees intuition, it's safer to update the NaN to "Not Sure"
    df['Lead Quality'].replace(np.nan, 'Not Sure', inplace = True)

    #We can impute the MUMBAI into all the NULLs as most of the values belong to MUMBAI
    df['City'].replace(np.nan, 'Mumbai', inplace = True)

    #Since there is no significant difference among top 3 specialisation , hence it will be safer to impute NaN
    df['Specialization'].replace(np.nan, 'Other_Specialization', inplace = True)

    #For Tags column, more than 30% data is for "Will revert after reading the email" and hence we can impute N
    df['Tags'].replace(np.nan, 'Will revert after reading the email', inplace = True)

    #More than 99% data is of "Better Career Prospects" and hence it is safer to impute NULLS with this value
    df['What matters most to you in choosing a course'].replace(np.nan, 'Better Career Prospects', inplace = Tr

    #More than 85% data is of "Unemployed" and hence it is safer to impute NULLS with this value
    df['What is your current occupation'].replace(np.nan, 'Unemployed', inplace = True)

    #More than 95% data is of "India" and hence it is safer to impute NULLS with this value
    df['Country'].replace(np.nan, 'India', inplace = True)

    #Dropping columns having more than 60% null values
    df = df.drop(df.columns[round(100*df.isnull().sum()/len(df),2)>60], axis = 1, inplace = True)

    return df
```

In [7]:

```
# calling the function to preprocess the data

pre_process(df)

# Checking the null values count after preprocessing the data
100*df.isnull().sum()/len(df)
```

Out[7]:

```
Prospect ID                                      0.000000
Lead Number                                      0.000000
Lead Origin                                      0.000000
Lead Source                                      0.389610
Do Not Email                                     0.000000
Do Not Call                                      0.000000
Converted                                        0.000000
TotalVisits                                      1.482684
Total Time Spent on Website                      0.000000
Page Views Per Visit                             1.482684
Last Activity                                    1.114719
Country                                          0.000000
Specialization                                   0.000000
What is your current occupation                  0.000000
What matters most to you in choosing a course    0.000000
Search                                           0.000000
Newspaper Article                                0.000000
X Education Forums                               0.000000
Newspaper                                        0.000000
Digital Advertisement                            0.000000
Through Recommendations                          0.000000
Tags                                             0.000000
Lead Quality                                     0.000000
City                                             0.000000
A free copy of Mastering The Interview           0.000000
Last Notable Activity                            0.000000
dtype: float64
```

In [8]:

```python
#Remaining NULL values are less than 2% and hence these rows can be directly dropped
df.dropna(inplace = True)
```

Exploratory Data Analysis

In [9]:

```python
#Checking the target variable and analysing it
df.Converted.value_counts(normalize = True)*100
```

Out[9]:

```
0    62.144589
1    37.855411
Name: Converted, dtype: float64
```

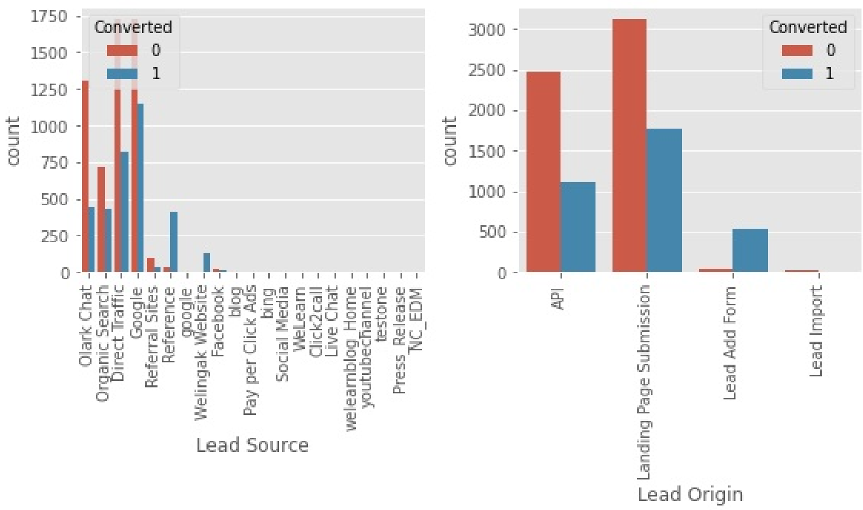
In [ ]:

OBSERVATIONS:

- There seems to be a good representation of both the classes of data and hence we are good to go with the further analysis

In [10]:

```python
fig = plt.subplots(figsize = (12, 12))

for i, feature in enumerate(['Lead Source', 'Lead Origin']):
    plt.subplot(3, 3, i+1)
    plt.subplots_adjust(hspace = 2.0)
    sns.countplot(df[feature], hue = df["Converted"])
    plt.xticks(rotation = 90)
    plt.tight_layout()
```



OBSERVATION:

- API and Landing Page Submission has less conversion rate(~30%) but counts of the leads from them are considerable
- The count of leads from the Lead Add Form is pretty low but the conversion rate is very high
- Lead Import has very less count as well as conversion rate and hence can be ignored

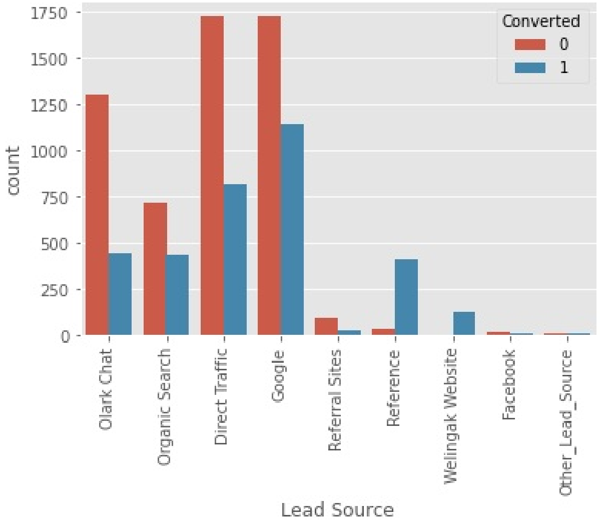
To improve the overall lead conversion rate, we need to focus on increasing the conversion rate of 'API' and 'Landing Page Submission' and also increasing the number of leads from 'Lead Add Form'

In [11]:

```
# We can clearly observe that the count of leads from various sources are close to negligible and hence we can

df['Lead Source'].replace(['Click2call', 'Live Chat', 'NC_EDM', 'Pay per Click Ads', 'Press_Release',
  'Social Media', 'WeLearn', 'bing', 'blog', 'testone', 'welearnblog_Home', 'youtubechannel'], 'Other_Lead_Sou

df['Lead Source'].replace("google", 'Google', inplace = True)
```

In [12]:

```
# Plotting Lead Source again

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



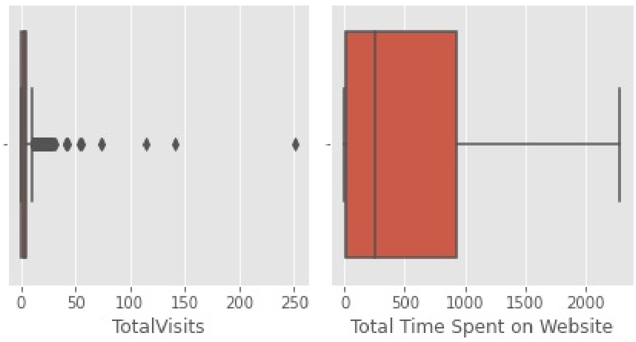
OBSERVATION:

- **The count of leads from the Google and Direct Traffic is maximum**
- **The conversion rate of the leads from Reference and Welingak Website is maximum**

To improve the overall lead conversion rate, we need to focus on increasing the conversion rate of 'Google', 'Olark Chat', 'Organic Search', 'Direct Traffic' and also increasing the number of leads from 'Reference' and 'Welingak Website'

In [13]:

```
fig=plt.subplots(figsize=(6, 6))

for i, feature in enumerate(["TotalVisits", "Total Time Spent on Website"]):
    plt.subplot(2, 2, i+1)
    plt.subplots_adjust(hspace = 2.0)
    sns.boxplot(df[feature])
    plt.tight_layout()
```



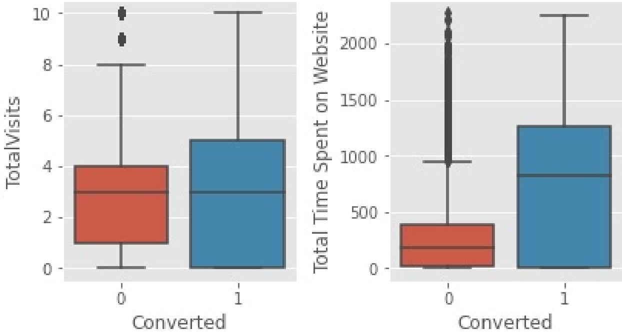
In [14]:

```
# There are lot of outliers in the Total Visits columns and we can cap this variable to 95 percetile

q1 = df["TotalVisits"].quantile(0.95)
df["TotalVisits"][df["TotalVisits"] >= q1] = q1
```

In [15]:

```
fig=plt.subplots(figsize=(6, 6))
```

```
for i, feature in enumerate(["TotalVisits", "Total Time Spent on Website"]):
    plt.subplot(2, 2, i+1)
    plt.subplots_adjust(hspace = 2.0)
    sns.boxplot(y = feature, x = 'Converted', data = df)
    plt.tight_layout()
```

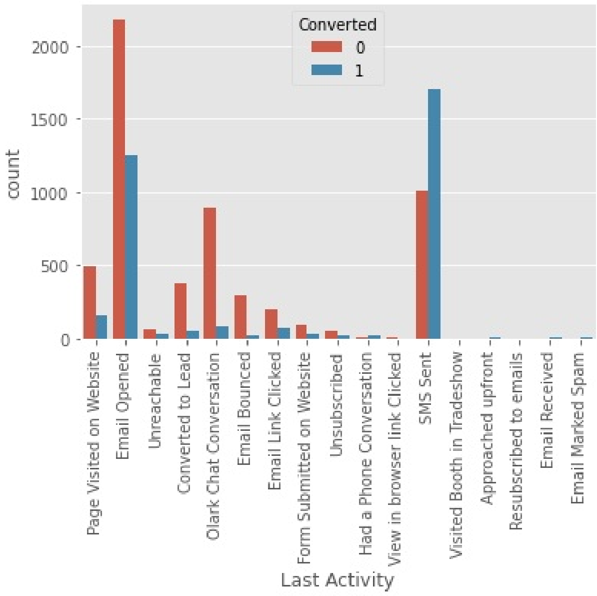


OBSERVATIONS:

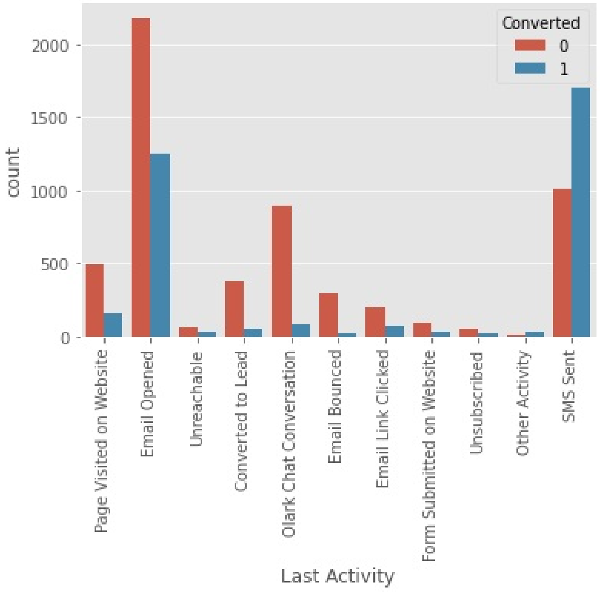
- **The median of both the conversion and non-conversion are same and hence nothing conclusive can be said using this information**
- **Users spending more time on the website are more likely to get converted**

  Website can be made more appealing so as to increase the time of the Users on websites

In [16]:
```
# Plotting Last Activity column

sns.countplot(x = "Last Activity", hue = "Converted", data= df)
plt.xticks(rotation = 'vertical')
plt.show()
```



In [17]:
```
# Converting all the low count categories to the 'Others' category
df['Last Activity'].replace(['Had a Phone Conversation', 'View in browser link Clicked',
                                                        'Visited Booth in Tradeshow', 'Approached upfront',
                                                        'Resubscribed to emails','Email Received', 'Email Markec

# lets plot the Last Activity again
sns.countplot(x = "Last Activity", hue = "Converted", data = df)
plt.xticks( rotation = 'vertical')
plt.show()
```



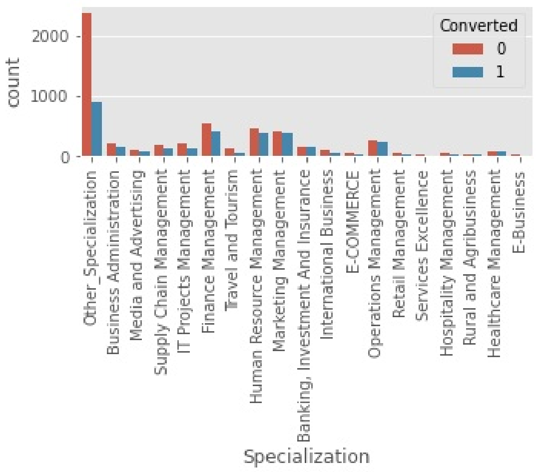
OBSERVATIONS:

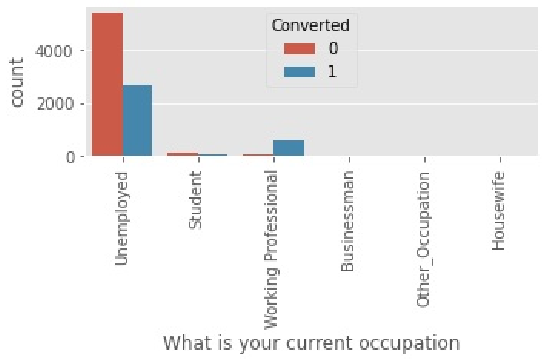
- **The count of lst activity as "Email Opened" is max**
- **The conversion rate of SMS sent as last activity is maximum**

We should focus on increasing the conversion rate of those having last activity as Email Opened by making a call to those leads and also try to increase the count of the ones having last activity as SMS sent

In [18]:

```
fig=plt.subplots(figsize=(10, 6))

for i, feature in enumerate(["Specialization", "What is your current occupation"]):
    plt.subplot(2, 2, i+1)
    plt.subplots_adjust(hspace = 2.0)
    sns.countplot(x = feature, hue = "Converted", data = df)
    plt.xticks( rotation = 'vertical')
    plt.tight_layout()
```





OBSERVATIONS:

- **Looking at above plot, no particular inference can be made for Specialization**
- **Looking at above plot, we can say that working professionals have high conversion rate**
- **Number of Unemployed leads are more than any other category**

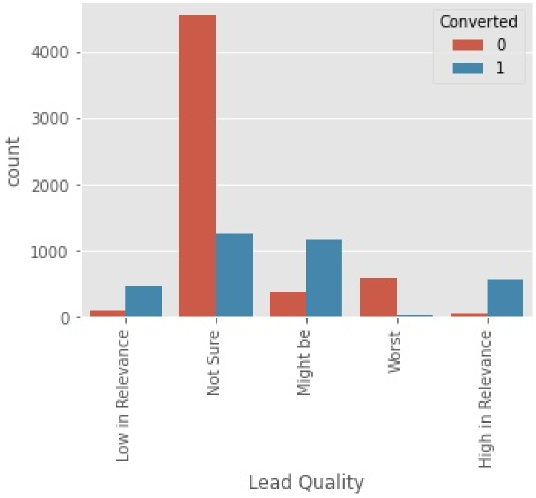
To increase overall conversion rate, we need to increase the number of Working Professional leads by reaching out to them through different social sites such as LinkedIn etc. and also on increasing the conversion rate of Unemployed leads

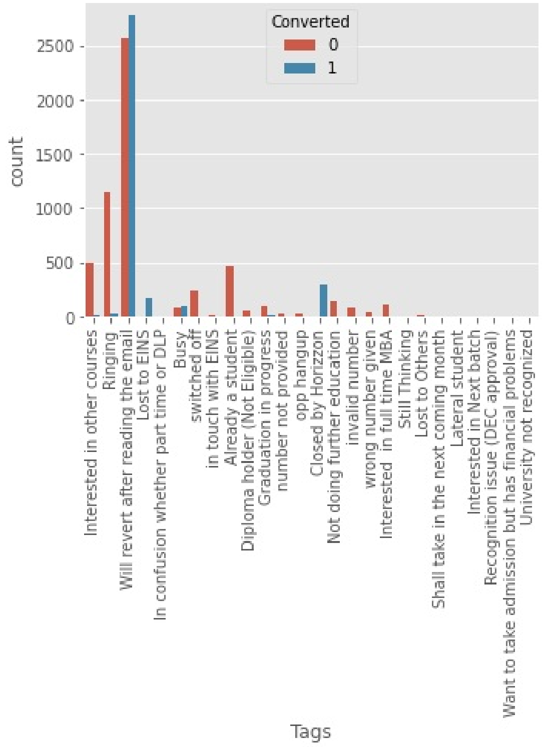
- **Country, What matters most to you in choosing a course, City columns have most values corresponding to one value such as India for Country, Mumbai for city and hence there is no particular insights for these columns**

In [19]:

```
fig=plt.subplots(figsize=(10, 10))

for i, feature in enumerate(["Lead Quality", "Tags"]):
    plt.subplot(2, 2, i+1)
    plt.subplots_adjust(hspace = 2.0)
    sns.countplot(x = feature, hue = "Converted", data = df)
```

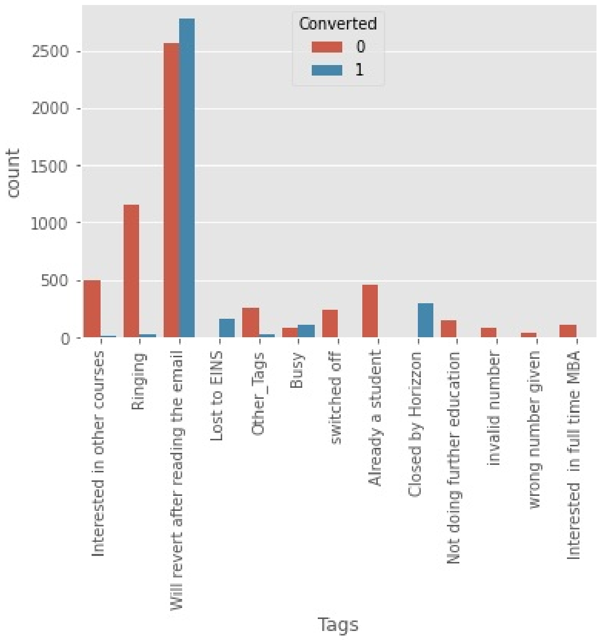
```
    plt.xticks( rotation = 'vertical')
    plt.tight_layout()
```





In [20]:

```
# Converting all low count categories to Others category
df['Tags'].replace(['In confusion whether part time or DLP', 'in touch with EINS','Diploma holder (Not Eligible
                    'Approached upfront','Graduation in progress','number not provided', 'opp hangup','Stil
                    'Lost to Others','Shall take in the next coming month','Lateral student','Interested ir
                    'Recognition issue (DEC approval)','Want to take admission but has financial problems',
                    'University not recognized'], 'Other_Tags', inplace = True)


# lets plot the Tags again
sns.countplot(x = "Tags", hue = "Converted", data= df)
plt.xticks( rotation = 'vertical')
plt.show()
```



OBSERVATION:

- **'Will revert after reading the email' and 'Closed by Horizzon' have high conversion rate #### SUMMARY:**
- **To improve the overall lead conversion rate, we need to focus on increasing the conversion rate of 'API' and 'Landing Page Submission' Lead Origins and also increasing the number of leads from 'Lead Add Form'**
- **To improve the overall lead conversion rate, we need to focus on increasing the conversion rate of 'Google', 'Olark Chat', 'Organic Search', 'Direct Traffic' and also increasing the number of leads from 'Reference' and 'Welingak Website'**
- **Websites can be made more appealing so as to increase the time of the Users on websites**
- **We should focus on increasing the conversion rate of those having last activity as Email Opened by making a call to those**

leads and also try to increase the count of the ones having last activity as SMS sent

- To increase overall conversion rate, we need to increase the number of Working Professional leads by reaching out to them through different social sites such as LinkedIn etc. and also on increasing the conversion rate of Unemployed leads
- We also observed that there are multiple columns which contains data of a single value only. As these columns do not contribute towards any inference, we can remove them from further analysis

In [21]:

```
# Dropping unnecessary columns
df.drop(['Lead Number','What matters most to you in choosing a course','Search','Newspaper Article','X Educatic
            'Digital Advertisement','Through Recommendations','A free copy of Mastering The Interview','Country'

df
```

Out[21]:

	Prospect ID	Lead Origin	Lead Source	Do Not Email	Converted	TotalVisits	Total Time Spent on Website	Page Views Per Visit	Last Activity	Specialization	What is your current occupation	
0	7927b2df-8bba-4d29-b9a2-b6e0beafe620	API	Olark Chat	0	0	0.0	0	0.00	Page Visited on Website	Other_Specialization	Unemployed	In
1	2a272436-5132-4136-86fa-dcc88c88f482	API	Organic Search	0	0	5.0	674	2.50	Email Opened	Other_Specialization	Unemployed	
2	8cc8c611-a219-4f35-ad23-fdfd2656bd8a	Landing Page Submission	Direct Traffic	0	1	2.0	1532	2.00	Email Opened	Business Administration	Student	t
3	0cc2df48-7cf4-4e39-9de9-19797f9b38cc	Landing Page Submission	Direct Traffic	0	0	1.0	305	1.00	Unreachable	Media and Advertising	Unemployed	
4	3256f628-e534-4826-9d63-4a8b88782852	Landing Page Submission	Google	0	1	2.0	1428	1.00	Converted to Lead	Other_Specialization	Unemployed	t
...	...	...	...	...	...	...	...	...	...	...	...	
9235	19d6451e-fcd6-407c-b83b-48e1af805ea9	Landing Page Submission	Direct Traffic	1	1	8.0	1845	2.67	Other Activity	IT Projects Management	Unemployed	t
9236	82a7005b-7196-4d56-95ce-a79f937a158d	Landing Page Submission	Direct Traffic	0	0	2.0	238	2.00	SMS Sent	Media and Advertising	Unemployed	
9237	aac550fe-a586-452d-8d3c-f1b62c94e02c	Landing Page Submission	Direct Traffic	1	0	2.0	199	2.00	SMS Sent	Business Administration	Unemployed	
9238	5330a7d1-2f2b-4df4-85d6-64ca2f6b95b9	Landing Page Submission	Google	0	1	3.0	499	3.00	SMS Sent	Human Resource Management	Unemployed	t
9239	571b5c8e-a5b2-4d57-8574-f2ffb06fdeff	Landing Page Submission	Direct Traffic	0	1	6.0	1279	3.00	SMS Sent	Supply Chain Management	Unemployed	t

9074 rows × 15 columns

Dummy Variable Creation

In [22]:

```
dummy = pd.get_dummies(df[['Lead Origin', 'Lead Source', 'Last Activity', 'Specialization','What is your currer
                        'Tags','Lead Quality','City','Last Notable Activity']], drop_first=True)
dummy.head()
```

Out[22]:

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

In [23]:

```
#Droping the original columns after dummy variable creation
df.drop(['Lead Origin', 'Lead Source', 'Last Activity', 'Specialization','What is your current occupation',
         'Tags','Lead Quality','City','Last Notable Activity'], axis=1, inplace = True)

#merging dataframe with dummy
df = pd.concat([df, dummy], axis=1)
df
```

Out[23]:

	Prospect ID	Do Not Email	Converted	TotalVisits	Total Time Spent on Website	Page Views Per Visit	Lead Origin_Landing Page Submission	Lead Origin_Lead Add Form	Lead Origin_Lead Import	Lead Source_Facebook
0	7927b2df-8bba-4d29-b9a2-b6e0beafe620	0	0	0.0	0	0.00	0	0	0	0
1	2a272436-5132-4136-86fa-dcc88c88f482	0	0	5.0	674	2.50	0	0	0	0
2	8cc8c611-a219-4f35-ad23-fdfd2656bd8a	0	1	2.0	1532	2.00	1	0	0	0
3	0cc2df48-7cf4-4e39-9de9-19797f9b38cc	0	0	1.0	305	1.00	1	0	0	0
4	3256f628-e534-4826-9d63-4a8b88782852	0	1	2.0	1428	1.00	1	0	0	0
...	...	...	...	...	...	...	...	...	...	...
9235	19d6451e-fcd6-407c-b83b-48e1af805ea9	1	1	8.0	1845	2.67	1	0	0	0
9236	82a7005b-7196-4d56-95ce-a79f937a158d	0	0	2.0	238	2.00	1	0	0	0
9237	aac550fe-a586-452d-8d3c-f1b62c94e02c	1	0	2.0	199	2.00	1	0	0	0
9238	5330a7d1-2f2b-4df4-85d6-64ca2f6b95b9	0	1	3.0	499	3.00	1	0	0	0
9239	571b5c8e-a5b2-4d57-8574-f2ffb06fdeff	0	1	6.0	1279	3.00	1	0	0	0

9074 rows × 86 columns

Test-Train Split

In [24]:

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

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

In [25]:
```python
# Splitting the data into train and test

X_train, X_test, y_train, y_test = train_test_split(X, y, train_size=0.7, test_size=0.3, random_state=100)
```

Feature Scaling

In [26]:
```python
scaler = StandardScaler()

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

In [27]:
```python
# Checking the Conversion Rate

print("Conversion rate is ", (sum(df['Converted'])/len(df['Converted'].index))*100)
```

```
Conversion rate is  37.85541106458012
```

Looking at Correlations

In [28]:
```python
# Correlation between different numerical variables for both the Converted and not-converted cases
conv_corr = df.corr()

# Unstacking the correlation matrix to find out top correlations
conv_corr_unstacked = conv_corr.unstack().sort_values(kind="quicksort")
conv_corr.where(np.triu(np.ones(conv_corr.shape), k=1).astype(np.bool)).stack().sort_values(ascending=False).he
```

Out[28]:
```
Lead Origin_Lead Import                Lead Source_Facebook                          0.983684
Last Activity_Unsubscribed             Last Notable Activity_Unsubscribed            0.872656
Lead Origin_Lead Add Form              Lead Source_Reference                         0.866191
Last Activity_Email Opened             Last Notable Activity_Email Opened            0.861636
Last Activity_SMS Sent                 Last Notable Activity_SMS Sent                0.853102
Last Activity_Email Link Clicked       Last Notable Activity_Email Link Clicked      0.800686
TotalVisits                            Page Views Per Visit                          0.737996
Last Activity_Page Visited on Website  Last Notable Activity_Page Visited on Website 0.691811
Do Not Email                           Last Activity_Email Bounced                   0.620041
Last Activity_Unreachable              Last Notable Activity_Unreachable             0.594369
dtype: float64
```

In [29]:
```python
# Dropping highly correlated features

X_test.drop(['Lead Source_Facebook','Last Notable Activity_Unsubscribed','Last Notable Activity_SMS Sent',
                'Last Notable Activity_Email Opened','Last Notable Activity_Unreachable','Last Notable Activity
X_train.drop(['Lead Source_Facebook','Last Notable Activity_Unsubscribed','Last Notable Activity_SMS Sent',
                'Last Notable Activity_Email Opened','Last Notable Activity_Unreachable','Last Notable Activity
```

In [30]:
```python
conv_corr = X_train.corr()
```

In [31]:
```python
conv_corr.where(np.triu(np.ones(conv_corr.shape), k=1).astype(np.bool)).stack().sort_values(ascending=False).he
```

Out[31]:
```
Lead Origin_Lead Add Form     Lead Source_Reference                          0.859537
TotalVisits                   Page Views Per Visit                           0.756104
Do Not Email                  Last Activity_Email Bounced                    0.624939
Last Activity_Other Activity  Last Notable Activity_Had a Phone Conversation 0.593057
Lead Source_Olark Chat        Specialization_Other_Specialization            0.505771
Page Views Per Visit          Lead Origin_Landing Page Submission            0.493007
Lead Origin_Lead Add Form     Lead Source_Welingak Website                   0.468225
Last Activity_Email Bounced   Last Notable Activity_Email Bounced            0.450911
TotalVisits                   Lead Origin_Landing Page Submission            0.447765
Lead Source_Olark Chat        Last Activity_Olark Chat Conversation          0.419173
dtype: float64
```

Model Building

In [32]:
```python
# Logistic regression model

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

Out[32]:

Generalized Linear Model Regression Results

Dep. Variable:	Converted	No. Observations:	6351
Model:	GLM	Df Residuals:	6273
Model Family:	Binomial	Df Model:	77
Link Function:	Logit	Scale:	1.0000
Method:	IRLS	Log-Likelihood:	-1275.8
Date:	Sun, 04 Sep 2022	Deviance:	2551.6

Time:	18:08:59	Pearson chi2:	3.54e+04
No. Iterations:	24	Pseudo R-squ. (CS):	0.6059
Covariance Type:	nonrobust		

	coef	std err	z	P>|z|	[0.025	0.975]
const	2.0041	1.763	1.137	0.256	-1.451	5.460
Do Not Email	-1.3817	0.317	-4.362	0.000	-2.002	-0.761
TotalVisits	0.0695	0.085	0.815	0.415	-0.098	0.237
Total Time Spent on Website	1.1466	0.063	18.110	0.000	1.022	1.271
Page Views Per Visit	-0.1212	0.085	-1.432	0.152	-0.287	0.045
Lead Origin_Landing Page Submission	-1.0310	0.218	-4.722	0.000	-1.459	-0.603
Lead Origin_Lead Add Form	-0.3581	1.310	-0.273	0.785	-2.926	2.210
Lead Origin_Lead Import	1.1699	0.820	1.427	0.154	-0.437	2.777
Lead Source_Google	0.1897	0.152	1.247	0.212	-0.108	0.488
Lead Source_Olark Chat	0.9961	0.227	4.385	0.000	0.551	1.441
Lead Source_Organic Search	0.1893	0.206	0.917	0.359	-0.215	0.594
Lead Source_Other_Lead_Source	0.9457	0.829	1.140	0.254	-0.680	2.571
Lead Source_Reference	1.8298	1.367	1.339	0.181	-0.849	4.508
Lead Source_Referral Sites	-0.1186	0.490	-0.242	0.809	-1.079	0.842
Lead Source_Welingak Website	5.5399	1.508	3.674	0.000	2.584	8.496
Last Activity_Email Bounced	-0.5811	0.880	-0.661	0.509	-2.305	1.143
Last Activity_Email Link Clicked	-0.8403	0.458	-1.833	0.067	-1.739	0.058
Last Activity_Email Opened	-0.4341	0.348	-1.249	0.212	-1.115	0.247
Last Activity_Form Submitted on Website	0.1271	0.591	0.215	0.830	-1.031	1.285
Last Activity_Olark Chat Conversation	-0.5769	0.392	-1.470	0.142	-1.346	0.192
Last Activity_Other Activity	1.4306	1.186	1.207	0.228	-0.893	3.754
Last Activity_Page Visited on Website	-0.4234	0.402	-1.054	0.292	-1.211	0.364
Last Activity_SMS Sent	1.6437	0.342	4.813	0.000	0.974	2.313
Last Activity_Unreachable	0.5769	0.570	1.012	0.312	-0.541	1.694
Last Activity_Unsubscribed	0.5471	0.799	0.684	0.494	-1.020	2.114
Specialization_Business Administration	-0.2070	0.387	-0.535	0.593	-0.966	0.552
Specialization_E-Business	-0.3311	0.688	-0.481	0.630	-1.680	1.018
Specialization_E-COMMERCE	0.6914	0.575	1.202	0.229	-0.436	1.819
Specialization_Finance Management	-0.4219	0.341	-1.236	0.216	-1.091	0.247
Specialization_Healthcare Management	-0.4287	0.510	-0.841	0.400	-1.428	0.570
Specialization_Hospitality Management	-0.2095	0.539	-0.389	0.698	-1.266	0.847
Specialization_Human Resource Management	-0.2823	0.345	-0.819	0.413	-0.958	0.394
Specialization_IT Projects Management	0.0279	0.404	0.069	0.945	-0.764	0.820
Specialization_International Business	-0.8352	0.455	-1.836	0.066	-1.727	0.057
Specialization_Marketing Management	0.1043	0.345	0.303	0.762	-0.571	0.780
Specialization_Media and Advertising	-0.5267	0.478	-1.101	0.271	-1.464	0.411
Specialization_Operations Management	-0.0511	0.387	-0.132	0.895	-0.809	0.707
Specialization_Other_Specialization	-0.7647	0.355	-2.156	0.031	-1.460	-0.070
Specialization_Retail Management	-0.2702	0.557	-0.485	0.628	-1.362	0.821
Specialization_Rural and Agribusiness	-0.0355	0.685	-0.052	0.959	-1.378	1.307
Specialization_Services Excellence	-0.3058	0.963	-0.318	0.751	-2.192	1.581
Specialization_Supply Chain Management	-0.4489	0.422	-1.063	0.288	-1.276	0.378
Specialization_Travel and Tourism	-0.6891	0.509	-1.355	0.175	-1.686	0.308
What is your current occupation_Housewife	20.6365	7.11e+04	0.000	1.000	-1.39e+05	1.39e+05
What is your current occupation_Other_Occupation	-0.2721	2.079	-0.131	0.896	-4.347	3.803
What is your current occupation_Student	-1.3454	1.557	-0.864	0.387	-4.396	1.706
What is your current occupation_Unemployed	-1.9887	1.457	-1.365	0.172	-4.845	0.867
What is your current occupation_Working Professional	-0.6322	1.493	-0.423	0.672	-3.559	2.295
Tags_Busy	3.9145	0.852	4.596	0.000	2.245	5.584
Tags_Closed by Horizzon	9.2360	1.149	8.039	0.000	6.984	11.488

Tags_Interested in full time MBA	0.4010	1.239	0.324	0.746	-2.027	2.829
Tags_Interested in other courses	0.2442	0.891	0.274	0.784	-1.502	1.990
Tags_Lost to EINS	9.9001	1.091	9.074	0.000	7.762	12.039
Tags_Not doing further education	-0.2751	1.510	-0.182	0.855	-3.235	2.685
Tags_Other_Tags	1.0350	0.866	1.195	0.232	-0.663	2.733
Tags_Ringing	-0.9231	0.858	-1.076	0.282	-2.605	0.759
Tags_Will revert after reading the email	4.1487	0.816	5.085	0.000	2.550	5.748
Tags_invalid number	-22.2227	2.23e+04	-0.001	0.999	-4.38e+04	4.38e+04
Tags_switched off	-1.5685	1.013	-1.549	0.121	-3.553	0.416
Tags_wrong number given	-22.8729	3.04e+04	-0.001	0.999	-5.96e+04	5.95e+04
Lead Quality_Low in Relevance	-0.5156	0.434	-1.188	0.235	-1.366	0.335
Lead Quality_Might be	-1.3432	0.393	-3.419	0.001	-2.113	-0.573
Lead Quality_Not Sure	-4.0935	0.376	-10.883	0.000	-4.831	-3.356
Lead Quality_Worst	-4.8633	1.010	-4.814	0.000	-6.843	-2.883
City_Other Cities	-0.2724	0.219	-1.244	0.214	-0.702	0.157
City_Other Cities of Maharashtra	-0.0812	0.259	-0.314	0.754	-0.588	0.426
City_Other Metro Cities	0.0951	0.280	0.340	0.734	-0.454	0.644
City_Thane & Outskirts	-0.1630	0.217	-0.753	0.452	-0.587	0.261
City_Tier II Cities	0.9292	0.667	1.394	0.163	-0.378	2.236
Last Notable Activity_Email Bounced	0.8522	1.011	0.843	0.399	-1.130	2.834
Last Notable Activity_Email Marked Spam	21.6297	1.39e+05	0.000	1.000	-2.72e+05	2.72e+05
Last Notable Activity_Email Received	19.1228	2.16e+05	8.85e-05	1.000	-4.23e+05	4.23e+05
Last Notable Activity_Form Submitted on Website	-24.1463	2.16e+05	-0.000	1.000	-4.23e+05	4.23e+05
Last Notable Activity_Had a Phone Conversation	-0.5854	1.762	-0.332	0.740	-4.040	2.869
Last Notable Activity_Modified	-1.6992	0.148	-11.489	0.000	-1.989	-1.409
Last Notable Activity_Olark Chat Conversation	-1.7134	0.482	-3.553	0.000	-2.659	-0.768
Last Notable Activity_Resubscribed to emails	19.0436	2.16e+05	8.82e-05	1.000	-4.23e+05	4.23e+05
Last Notable Activity_View in browser link Clicked	-21.8761	2.16e+05	-0.000	1.000	-4.23e+05	4.23e+05

Feature Selection Using RFE

In [33]:

```
# Starting with 15 features selected by RFE
# We will then optimize the model further by inspecting VIF and p-value of the features

logreg = LogisticRegression()
rfe = RFE(logreg, n_features_to_select = 15)
rfe = rfe.fit(X_train, y_train)

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

Out[33]:
```
[('Do Not Email', True, 1),
 ('TotalVisits', False, 54),
 ('Total Time Spent on Website', False, 4),
 ('Page Views Per Visit', False, 53),
 ('Lead Origin_Landing Page Submission', False, 13),
 ('Lead Origin_Lead Add Form', True, 1),
 ('Lead Origin_Lead Import', False, 14),
 ('Lead Source_Google', False, 46),
 ('Lead Source_Olark Chat', False, 3),
 ('Lead Source_Organic Search', False, 47),
 ('Lead Source_Other_Lead_Source', False, 33),
 ('Lead Source_Reference', False, 60),
 ('Lead Source_Referral Sites', False, 31),
 ('Lead Source_Welingak Website', True, 1),
 ('Last Activity_Email Bounced', False, 29),
 ('Last Activity_Email Link Clicked', False, 22),
 ('Last Activity_Email Opened', False, 26),
 ('Last Activity_Form Submitted on Website', False, 52),
 ('Last Activity_Olark Chat Conversation', False, 21),
 ('Last Activity_Other Activity', False, 6),
 ('Last Activity_Page Visited on Website', False, 25),
 ('Last Activity_SMS Sent', True, 1),
 ('Last Activity_Unreachable', False, 11),
 ('Last Activity_Unsubscribed', False, 15),
 ('Specialization_Business Administration', False, 58),
 ('Specialization_E-Business', False, 61),
 ('Specialization_E-COMMERCE', False, 12),
 ('Specialization_Finance Management', False, 37),
 ('Specialization_Healthcare Management', False, 38),
 ('Specialization_Hospitality Management', False, 55),
 ('Specialization_Human Resource Management', False, 51),
 ('Specialization_IT Projects Management', False, 42),
 ('Specialization_International Business', False, 19),
 ('Specialization_Marketing Management', False, 27),
 ('Specialization_Media and Advertising', False, 32),
 ('Specialization_Operations Management', False, 44),
 ('Specialization_Other_Specialization', False, 17),
 ('Specialization_Retail Management', False, 49),
 ('Specialization_Rural and Agribusiness', False, 45),
 ('Specialization_Services Excellence', False, 36),
 ('Specialization_Supply Chain Management', False, 35),
 ('Specialization_Travel and Tourism', False, 20),
 ('What is your current occupation_Housewife', False, 39),
 ('What is your current occupation_Other_Occupation', False, 30),
 ('What is your current occupation_Student', False, 7),
 ('What is your current occupation_Unemployed', False, 5),
 ('What is your current occupation_Working Professional', False, 23),
 ('Tags_Busy', True, 1),
 ('Tags_Closed by Horizzon', True, 1),
 ('Tags_Interested  in full time MBA', False, 16),
 ('Tags_Interested in other courses', False, 9),
 ('Tags_Lost to EINS', True, 1),
 ('Tags_Not doing further education', False, 10),
 ('Tags_Other_Tags', False, 24),
 ('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', False, 2),
 ('Lead Quality_Low in Relevance', False, 48),
 ('Lead Quality_Might be', False, 8),
 ('Lead Quality_Not Sure', True, 1),
 ('Lead Quality_Worst', True, 1),
 ('City_Other Cities', False, 34),
 ('City_Other Cities of Maharashtra', False, 59),
 ('City_Other Metro Cities', False, 56),
 ('City_Thane & Outskirts', False, 43),
 ('City_Tier II Cities', False, 18),
 ('Last Notable Activity_Email Bounced', False, 28),
 ('Last Notable Activity_Email Marked Spam', False, 50),
 ('Last Notable Activity_Email Received', False, 63),
 ('Last Notable Activity_Form Submitted on Website', False, 40),
 ('Last Notable Activity_Had a Phone Conversation', False, 41),
 ('Last Notable Activity_Modified', True, 1),
 ('Last Notable Activity_Olark Chat Conversation', True, 1),
 ('Last Notable Activity_Resubscribed to emails', False, 62),
 ('Last Notable Activity_View in browser link Clicked', False, 57)]
```

In [34]:
```
col = X_train.columns[rfe.support_]
```

Assessing the model with StatsModels

In [35]:
```
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[35]:

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:	-1553.1
Date:	Sun, 04 Sep 2022	Deviance:	3106.2
Time:	18:09:06	Pearson chi2:	4.04e+04
No. Iterations:	23	Pseudo R-squ. (CS):	0.5700
Covariance Type:	nonrobust		

	coef	std err	z	P>|z|	[0.025	0.975]
const	-1.0794	0.217	-4.963	0.000	-1.506	-0.653
Do Not Email	-1.1895	0.221	-5.376	0.000	-1.623	-0.756
Lead Origin_Lead Add Form	0.8693	0.366	2.372	0.018	0.151	1.587
Lead Source_Welingak Website	3.2594	0.820	3.976	0.000	1.653	4.866
Last Activity_SMS Sent	1.9538	0.103	19.039	0.000	1.753	2.155
Tags_Busy	3.4717	0.323	10.757	0.000	2.839	4.104
Tags_Closed by Horizzon	8.4090	0.775	10.849	0.000	6.890	9.928
Tags_Lost to EINS	9.4298	0.766	12.317	0.000	7.929	10.930
Tags_Ringing	-1.9594	0.331	-5.911	0.000	-2.609	-1.310
Tags_Will revert after reading the email	3.6656	0.231	15.900	0.000	3.214	4.117
Tags_invalid number	-22.4206	1.34e+04	-0.002	0.999	-2.62e+04	2.62e+04
Tags_switched off	-2.5297	0.584	-4.331	0.000	-3.674	-1.385
Lead Quality_Not Sure	-3.4872	0.130	-26.738	0.000	-3.743	-3.232
Lead Quality_Worst	-3.9571	0.834	-4.745	0.000	-5.592	-2.323
Last Notable Activity_Modified	-1.6959	0.107	-15.830	0.000	-1.906	-1.486
Last Notable Activity_Olark Chat Conversation	-1.3029	0.352	-3.699	0.000	-1.993	-0.612

In [36]:

```
# Getting the predicted values on the train set
y_train_pred = res.predict(X_train_sm)
y_train_pred = y_train_pred.values.reshape(-1)
y_train_pred[:10]
```

Out[36]:

```
array([0.28883901, 0.11002273, 0.00189224, 0.7413066 , 0.99406588,
       0.98943879, 0.28883901, 0.70243735, 0.92996485, 0.00189224])
```

Creating a dataframe with the true convertion status and the predicted probabilities

In [37]:

```
y_train_pred_final = pd.DataFrame({'Convert':y_train.values, 'Convert_Prob':y_train_pred})
y_train_pred_final['Pros_ID'] = y_train.index
y_train_pred_final.head()
```

Out[37]:

	Convert	Convert_Prob	Pros_ID
0	0	0.288839	3009
1	0	0.110023	1012
2	0	0.001892	9226
3	1	0.741307	4750
4	1	0.994066	7987

In [38]:

```
# Creating new column 'predicted' with 1 if Convert_Prob > 0.5 else 0

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

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

Out[38]:

	Convert	Convert_Prob	Pros_ID	predicted
0	0	0.288839	3009	0
1	0	0.110023	1012	0
2	0	0.001892	9226	0
3	1	0.741307	4750	1
4	1	0.994066	7987	1

In [39]:

```
print("Accuracy score", metrics.accuracy_score(y_train_pred_final.Convert, y_train_pred_final.predicted))
```

```
Accuracy score 0.9209573295544009
```

Checking VIFs

In [40]:

```
def calculate_vif(X_train):
    vif_df = pd.DataFrame()
    vif_df['Features'] = X_train.columns
    vif_df['Variance Inflation Factor'] = [variance_inflation_factor(X_train.values, i) for i in range(X_train.
    vif_df['Variance Inflation Factor'] = round(vif_df['Variance Inflation Factor'], 2)
    vif_df = vif_df.sort_values(by = 'Variance Inflation Factor', ascending = False)
    print(vif_df)

calculate_vif(X_train[col])
```

```
                                    Features  Variance Inflation Factor
11                      Lead Quality_Not Sure                       3.02
8        Tags_Will revert after reading the email                   2.70
13               Last Notable Activity_Modified                     1.69
3                        Last Activity_SMS Sent                     1.63
1                    Lead Origin_Lead Add Form                      1.58
7                                 Tags_Ringing                      1.53
2                  Lead Source_Welingak Website                     1.35
5                        Tags_Closed by Horizzon                    1.17
0                                 Do Not Email                      1.13
12                          Lead Quality_Worst                      1.13
4                                    Tags_Busy                      1.11
10                           Tags_switched off                      1.10
6                            Tags_Lost to EINS                      1.06
14  Last Notable Activity_Olark Chat Conversation                   1.05
9                          Tags_invalid number                      1.04
```

All variables have a good value of VIF. But we observed earlier that the column "Tags_invalid number" has high p-value and hence we will drop this column and remake the model.

In [41]:

```
col = col.drop('Tags_invalid number')
col
```

Out[41]:

```
Index(['Do Not Email', 'Lead Origin_Lead Add Form',
       'Lead Source_Welingak Website', 'Last Activity_SMS Sent', '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_Modified',
       'Last Notable Activity_Olark Chat Conversation'],
      dtype='object')
```

In [42]:

```
# Let's re-run the model using the selected variables
X_train_sm = sm.add_constant(X_train[col])
logm = sm.GLM(y_train,X_train_sm, family = sm.families.Binomial())
res = logm.fit()
res.summary()
```

Out[42]:

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:	-1559.1
Date:	Sun, 04 Sep 2022	Deviance:	3118.3
Time:	18:09:07	Pearson chi2:	3.94e+04
No. Iterations:	8	Pseudo R-squ. (CS):	0.5692
Covariance Type:	nonrobust		

	coef	std err	z	P>|z|	[0.025	0.975]
const	-1.2486	0.218	-5.725	0.000	-1.676	-0.821
Do Not Email	-1.1805	0.221	-5.350	0.000	-1.613	-0.748
Lead Origin_Lead Add Form	0.9081	0.369	2.464	0.014	0.186	1.630
Lead Source_Welingak Website	3.2182	0.820	3.923	0.000	1.611	4.826
Last Activity_SMS Sent	1.9270	0.102	18.901	0.000	1.727	2.127
Tags_Busy	3.6495	0.322	11.338	0.000	3.019	4.280
Tags_Closed by Horizzon	8.5559	0.776	11.031	0.000	7.036	10.076
Tags_Lost to EINS	9.5786	0.766	12.504	0.000	8.077	11.080
Tags_Ringing	-1.7714	0.330	-5.368	0.000	-2.418	-1.125
Tags_Will revert after reading the email	3.8317	0.231	16.579	0.000	3.379	4.285
Tags_switched off	-2.3367	0.583	-4.008	0.000	-3.479	-1.194
Lead Quality_Not Sure	-3.4792	0.130	-26.743	0.000	-3.734	-3.224
Lead Quality_Worst	-3.9437	0.836	-4.720	0.000	-5.581	-2.306
Last Notable Activity_Modified	-1.6821	0.107	-15.737	0.000	-1.892	-1.473
Last Notable Activity_Olark Chat Conversation	-1.3049	0.352	-3.706	0.000	-1.995	-0.615

In [43]:

```
y_train_pred = res.predict(X_train_sm).values.reshape(-1)
y_train_pred_final['Convert_Prob'] = y_train_pred

# Creating new column 'predicted' with 1 if Convert_Prob > 0.5 else 0
y_train_pred_final['predicted'] = y_train_pred_final.Convert_Prob.map(lambda x: 1 if x > 0.5 else 0)
y_train_pred_final.head()
```

Out[43]:

	Convert	Convert_Prob	Pros_ID	predicted
0	0	0.289842	3009	0
1	0	0.111387	1012	0
2	0	0.001918	9226	0
3	1	0.737087	4750	1
4	1	0.993914	7987	1

In [44]:

```
# Let's check the overall accuracy.
print("Accuracy score", metrics.accuracy_score(y_train_pred_final.Convert, y_train_pred_final.predicted))
```

```
Accuracy score 0.920642418516769
```

The accuracy is still practically the same.

Let's now check the VIFs again

In [45]:

```
calculate_vif(X_train[col])
```

```
                                        Features  Variance Inflation Factor
10                         Lead Quality_Not Sure                       2.97
8        Tags_Will revert after reading the email                      2.66
12                  Last Notable Activity_Modified                     1.68
3                           Last Activity_SMS Sent                     1.62
1                         Lead Origin_Lead Add Form                    1.58
7                                     Tags_Ringing                     1.51
2                     Lead Source_Welingak Website                     1.35
5                          Tags_Closed by Horizzon                     1.17
0                                    Do Not Email                      1.12
11                              Lead Quality_Worst                     1.12
4                                       Tags_Busy                      1.11
9                                 Tags_switched off                    1.09
6                                 Tags_Lost to EINS                    1.06
13  Last Notable Activity_Olark Chat Conversation                      1.05
```

OBSERVATIONS:

-All variables have a good value of VIF and p-values. So we need not drop any more variables and we can proceed with making predictions using this model only

In [46]:

```
# function name : evaluate_model
# argumet : y_true, y_predicted
# prints Confusion matrix, accuracy, Sensitivity, Specificity, False Positive Rate, Positive Predictive Value
# returns accuracy, Sensitivity, Specificity

def evaluate_model(y_true, y_predicted, print_score=False):
    confusion = metrics.confusion_matrix(y_true, y_predicted)
    # Predicted     not_converted    converted
    # Actual
    # not_converted        TN          FP
    # converted            FN          TP

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

    accuracy_sc = metrics.accuracy_score(y_true, y_predicted)
    sensitivity_score = TP / float(TP+FN)
    specificity_score = TN / float(TN+FP)
    precision_sc = precision_score(y_true, y_predicted)

    if print_score:
        print("Confusion Matrix :\n", confusion)
        print("Accuracy :", accuracy_sc)
        print("Sensitivity :", sensitivity_score)
        print("Specificity :", specificity_score)
        print("Precision :", precision_sc)

    return accuracy_sc, sensitivity_score, specificity_score, precision_sc
```

In [47]:

```
# Evaluating model
evaluate_model(y_train_pred_final.Convert, y_train_pred_final.predicted, print_score=True)
```

```
Confusion Matrix :
 [[3761  144]
 [ 360 2086]]
Accuracy : 0.920642418516769
Sensitivity : 0.8528209321340965
Specificity : 0.963124199743918
Precision : 0.9354260089686098
```

Out[47]:

```
(0.920642418516769, 0.8528209321340965, 0.963124199743918, 0.9354260089686098)
```

Plotting the ROC Curve

An ROC curve

- **shows tradeoff between sensitivity and specificity (increase in one will cause decrease in other).**
- **The closer the curve follows the y-axis and then the top border of the ROC space, means more area under the curve and the more accurate the test.**
- **The closer the curve comes to the 45-degree diagonal of the ROC space i.e. the reference line, means less area and the less accurate is the test.**

  Here, our goal is to have achieve good sensitivity score

In [48]:

```
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))
```

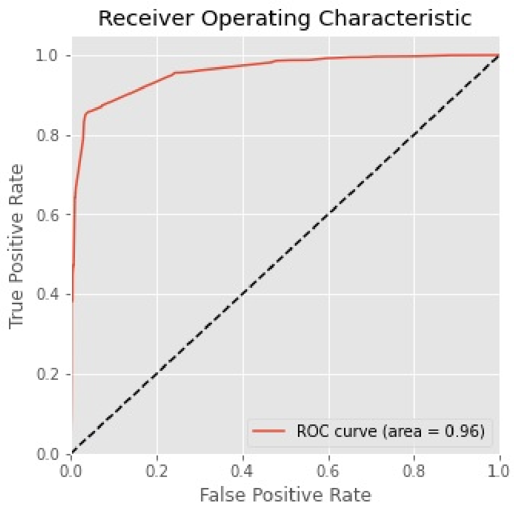
```
    plt.plot( fpr, tpr, label='ROC curve (area = %0.2f)' % auc_score )
    plt.plot([0, 1], [0, 1], 'k--')
    plt.xlim([0.0, 1.0])
    plt.ylim([0.0, 1.05])
    plt.xlabel('False Positive Rate')
    plt.ylabel('True Positive Rate')
    plt.title('Receiver Operating Characteristic')
    plt.legend(loc="lower right")
    plt.show()

    return None
```

In [49]:

```
fpr, tpr, thresholds = metrics.roc_curve( y_train_pred_final.Convert, y_train_pred_final.Convert_Prob, drop_int
```

In [50]:

```
draw_roc(y_train_pred_final.Convert, y_train_pred_final.Convert_Prob)
```



Finding optimal value of the cut off

In [51]:

```
# Predicting Convert status with different probability cutoffs

for i in [float(x)/10 for x in range(10)]:
    y_train_pred_final[i]= y_train_pred_final.Convert_Prob.map(lambda x: 1 if x > i else 0)
y_train_pred_final.head()
```

Out[51]:

	Convert	Convert_Prob	Pros_ID	predicted	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
0	0	0.289842	3009	0	1	1	1	0	0	0	0	0	0	0
1	0	0.111387	1012	0	1	1	0	0	0	0	0	0	0	0
2	0	0.001918	9226	0	1	0	0	0	0	0	0	0	0	0
3	1	0.737087	4750	1	1	1	1	1	1	1	1	1	0	0
4	1	0.993914	7987	1	1	1	1	1	1	1	1	1	1	1

In [52]:

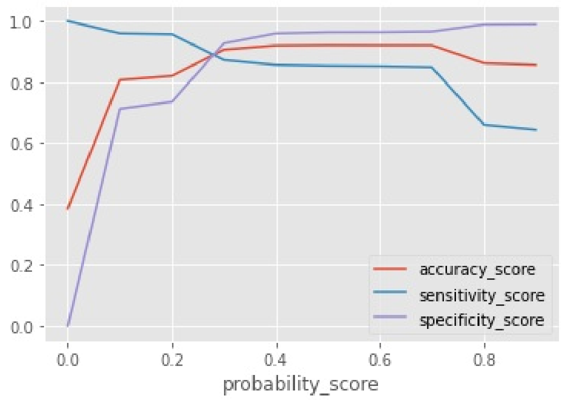
```
# Calculating accuracy, sensitivity and specificity for various probability cutoffs from 0.1 to 0.9.

df = pd.DataFrame(columns = ['probability_score','accuracy_score','sensitivity_score','specificity_score','pred

for i in [float(x)/10 for x in range(10)]:
    (accuracy_score,sensitivity_score,specificity_score,precision_sc) = evaluate_model(y_train_pred_final.Conve
    df.loc[i] =[i,accuracy_score,sensitivity_score,specificity_score,precision_sc]

df
```

Out[52]:

	probability_score	accuracy_score	sensitivity_score	specificity_score	precision_score
0.0	0.0	0.385136	1.000000	0.000000	0.385136
0.1	0.1	0.807117	0.959526	0.711652	0.675785
0.2	0.2	0.820343	0.956664	0.734955	0.693333
0.3	0.3	0.905999	0.872445	0.927017	0.882183
0.4	0.4	0.919540	0.856092	0.959283	0.929427
0.5	0.5	0.920642	0.852821	0.963124	0.935426
0.6	0.6	0.920328	0.851594	0.963380	0.935759
0.7	0.7	0.920328	0.848324	0.965429	0.938914
0.8	0.8	0.861912	0.659853	0.988476	0.972875
0.9	0.9	0.856086	0.643500	0.989245	0.974010

In [53]:

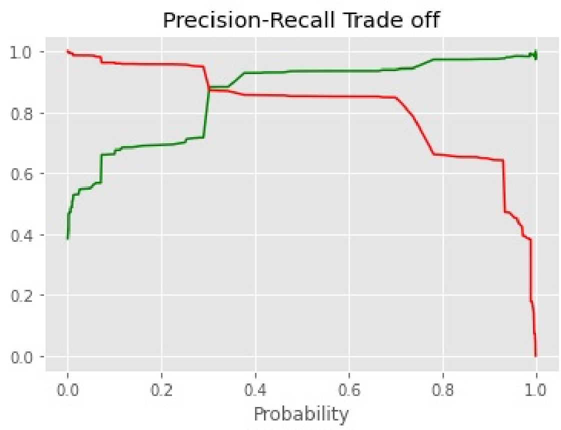
```
df.plot.line(x='probability_score', y=['accuracy_score','sensitivity_score','specificity_score'])
plt.show()
```



Precision-Recall Trade off

In [54]:

```
p, r, thresholds = precision_recall_curve(y_train_pred_final.Convert, y_train_pred_final.Convert_Prob)

plt.plot(thresholds, p[:-1], "g-")
plt.plot(thresholds, r[:-1], "r-")
plt.xlabel("Probability")
plt.title("Precision-Recall Trade off")
plt.show()
```



OBSERVATIONS:

In Sensitivity-Specificity-Accuracy plot 0.27 probability looks optimal. In Precision-Recall Curve 0.3 looks optimal.

We are taking 0.27 is the optimum point as a cutoff probability and assigning Lead Score in training data.

In [55]:

```
y_train_pred_final = y_train_pred_final.iloc[:, :3]
y_train_pred_final['Convert_predicted'] = y_train_pred_final.Convert_Prob.map(lambda x: 1 if x > 0.27 else 0)

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

Out[55]:

	Convert	Convert_Prob	Pros_ID	Convert_predicted	Lead_Score
0	0	0.289842	3009	1	29
1	0	0.111387	1012	0	11
2	0	0.001918	9226	0	0
3	1	0.737087	4750	1	74
4	1	0.993914	7987	1	99

In [56]:

```
# Evaluating model performance on training data

evaluate_model(y_train_pred_final.Convert, y_train_pred_final.Convert_predicted, print_score=True)
```

```
Confusion Matrix :
 [[2987  918]
 [ 124 2322]]
Accuracy : 0.8359313493937962
Sensitivity : 0.9493049877350777
Specificity : 0.7649167733674775
Precision : 0.7166666666666667
```

Out[56]:

```
(0.8359313493937962,
 0.9493049877350777,
 0.7649167733674775,
 0.7166666666666667)
```

In [57]:

```
# Getting the predicted values on the train set
X_test_sm = sm.add_constant(X_test[col])
y_test_pred = res.predict(X_test_sm)

y_test_df = pd.DataFrame(y_test)
y_test_pred_df = pd.DataFrame(y_test_pred, columns=["Converting_Probability"])
y_test_df['Prospect ID'] = y_test_df.index

y_predicted_final = pd.concat([y_test_df.reset_index(drop=True), y_test_pred_df.reset_index(drop=True)],axis=1)
y_predicted_final['final_predicted'] = y_predicted_final.Converting_Probability.map(lambda x: 1 if x > 0.27 els
y_predicted_final['Lead_Score'] = y_predicted_final.Converting_Probability.map(lambda x: round(x*100))

y_predicted_final.head()
```

Out[57]:

	Converted	Prospect ID	Converting_Probability	final_predicted	Lead_Score
0	0	3271	0.289842	1	29
1	1	1490	0.929765	1	93
2	0	7936	0.289842	1	29
3	1	4216	0.998548	1	100
4	0	3830	0.289842	1	29

In [58]:

```
# Evaluating model performance on test data

evaluate_model(y_predicted_final.Converted, y_predicted_final.final_predicted, print_score=True)
```

```
Confusion Matrix :
 [[1303  431]
 [  71  918]]
Accuracy : 0.8156445097319134
Sensitivity : 0.9282103134479271
Specificity : 0.751441753171857
Precision : 0.6805040770941438
```

Out[58]:

```
(0.8156445097319134, 0.9282103134479271, 0.751441753171857, 0.6805040770941438)
```

Final Model

In [59]:

```
# Builds a logistic regression model and returns predicted values on training dataset
# when training data, test data and probability cutoff is given

def build_model_cutoff(X_train, y_train, X_test, y_test, cutoff=0.5):

    # Train model
    X_train_sm = sm.add_constant(X_train)
    logm = sm.GLM(y_train,X_train_sm, family = sm.families.Binomial())
    res = logm.fit()

    y_train_pred = res.predict(X_train_sm).values.reshape(-1)

    y_train_pred_final = pd.DataFrame({'Prospect ID':y_train.index, 'Converted':y_train.values, 'Convert_Probab
    y_train_pred_final['Convert_predicted'] = y_train_pred_final.Convert_Probability.map(lambda x: 1 if x > cut
    y_train_pred_final['Lead_Score'] = y_train_pred_final.Convert_Probability.map(lambda x: round(x*100))
    print("------------------Result of training data-------------------")
```

```python
    print(y_train_pred_final.head())

    # Predicting Lead Score on Test data
    X_test_sm = sm.add_constant(X_test)
    y_test_pred = res.predict(X_test_sm)

    y_test_pred_final = pd.DataFrame({'Prospect ID':y_test.index, 'Converted':y_test.values, 'Convert_Probabili
    y_test_pred_final['Convert_predicted'] = y_test_pred_final.Convert_Probability.map(lambda x: 1 if x > cutof
    y_test_pred_final['Lead_Score'] = y_test_pred_final.Convert_Probability.map(lambda x: round(x*100))
    y_test_pred_final.reset_index(inplace=True, drop=True)
    print("------------------Result of test data-------------------")
    print(y_test_pred_final.head())

    print("------------------Model Evaluation Metrics-------------------")
    evaluate_model(y_test_pred_final.Converted, y_test_pred_final.Convert_predicted, print_score=True)

    return y_test_pred_final
```

In [60]:

```python
build_model_cutoff(X_train[col], y_train, X_test[col], y_test, cutoff=0.27)
```

```
------------------Result of training data------------------
   Prospect ID  Converted  Convert_Probability  Convert_predicted  Lead_Score
0         3009          0             0.289842                  1          29
1         1012          0             0.111387                  0          11
2         9226          0             0.001918                  0           0
3         4750          1             0.737087                  1          74
4         7987          1             0.993914                  1          99
------------------Result of test data------------------
   Prospect ID  Converted  Convert_Probability  Convert_predicted  Lead_Score
0         3271          0             0.289842                  1          29
1         1490          1             0.929765                  1          93
2         7936          0             0.289842                  1          29
3         4216          1             0.998548                  1         100
4         3830          0             0.289842                  1          29
------------------Model Evaluation Metrics------------------
Confusion Matrix :
 [[1303  431]
 [  71  918]]
Accuracy : 0.8156445097319134
Sensitivity : 0.9282103134479271
Specificity : 0.751441753171857
Precision : 0.6805040770941438
```

Out[60]:

	Prospect ID	Converted	Convert_Probability	Convert_predicted	Lead_Score
0	3271	0	0.289842	1	29
1	1490	1	0.929765	1	93
2	7936	0	0.289842	1	29
3	4216	1	0.998548	1	100
4	3830	0	0.289842	1	29
...	...	...	...	...	...
2718	850	0	0.070553	0	7
2719	2879	0	0.001642	0	0
2720	6501	1	0.989122	1	99
2721	7155	0	0.070553	0	7
2722	376	0	0.070553	0	7

2723 rows × 5 columns

In [61]:

```python
print("Features used in Final Model :", col)

print("----------------------Feature Importance-------------------")
print(res.params)
```

```
Features used in Final Model : Index(['Do Not Email', 'Lead Origin_Lead Add Form',
       'Lead Source_Welingak Website', 'Last Activity_SMS Sent', '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_Modified',
       'Last Notable Activity_Olark Chat Conversation'],
      dtype='object')
-----------------------Feature Importance-------------------
const                                             -1.248649
Do Not Email                                      -1.180501
Lead Origin_Lead Add Form                          0.908052
Lead Source_Welingak Website                       3.218160
Last Activity_SMS Sent                             1.927033
Tags_Busy                                          3.649486
Tags_Closed by Horizzon                            8.555901
Tags_Lost to EINS                                  9.578632
Tags_Ringing                                      -1.771378
Tags_Will revert after reading the email           3.831727
Tags_switched off                                 -2.336683
Lead Quality_Not Sure                             -3.479228
Lead Quality_Worst                                -3.943680
Last Notable Activity_Modified                    -1.682075
Last Notable Activity_Olark Chat Conversation     -1.304940
dtype: float64
```

Conclusion:

- **The logistic regression model predicts the probability of the target variable having a certain value, rather than predicting the value of the target variable directly. Then a cutoff of the probability is used to obtain the predicted value of the target variable.**
- **Here, the logistic regression model is used to predict the probabilty of conversion of a customer.**
- **Optimum cut off is chosen to be 0.27 i.e. any lead with greater than 0.27 probability of converting is predicted as Hot Lead (customer will convert) and any lead with 0.27 or less probability of converting is predicted as Cold Lead (customer will not convert)**
- **Our final Logistic Regression Model is built with 14 features.**
- **Features used in final model are:**
  ['Do Not Email', 'Lead Origin_Lead Add Form', 'Lead Source_Welingak Website',
  'Last Activity_SMS Sent', '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_Modified', 'Last Notable Activity_Olark Chat Conversation']
- **The top three categorical/dummy variables in the final model are ‘Tags_Lost to EINS’, ‘Tags_Closed by Horizzon’, ‘Lead Quality_Worst’ with respect to the absolute value of their coefficient factors.**

  ‘Tags_Lost to EINS’, ‘Tags_Closed by Horizzon’ are obtained by encoding original categorical variable ‘Tags’. ‘Lead Quality_Worst’ is obtained by encoding the categorical variable ‘Lead Quality’.

- **Tags_Lost to EINS (Coefficient factor = 9.578632)**
- **Tags_Closed by Horizzon (Coefficient factor = 8.555901)**
- **Lead Quality_Worst (Coefficient factor =-3.943680)**
- **The final model has Sensitivity of 0.928, this means the model is able to predict 92% customers out of all the converted customers, (Positive conversion) correctly.**
- **The final model has Precision of 0.68, this means 68% of predicted hot leads are True Hot Leads.**
- **We have also built an reusable code block which will predict Convert value and Lead Score given training, test data and a cut-off. Different cutoffs can be used depending on the use-cases (for eg. when high sensitivity is required, when model have optimum precision score etc.)**

Subjective Question

1. **X Education has a period of 2 months every year during which they hire some interns. The sales team, in particular, has around 10 interns allotted to them. So during this phase, they wish to make the lead conversion more aggressive. So they want almost all of the potential leads (i.e. the customers who have been predicted as 1 by the model) to be converted and hence, want to make phone calls to as much of such people as possible. Suggest a good strategy they should employ at this stage.**

In [62]:
```
build_model_cutoff(X_train[col], y_train, X_test[col], y_test, cutoff=0.1)
```

```
------------------Result of training data------------------
   Prospect ID  Converted  Convert_Probability  Convert_predicted  Lead_Score
0         3009          0             0.289842                  1          29
1         1012          0             0.111387                  1          11
2         9226          0             0.001918                  0           0
3         4750          1             0.737087                  1          74
4         7987          1             0.993914                  1          99
------------------Result of test data------------------
   Prospect ID  Converted  Convert_Probability  Convert_predicted  Lead_Score
0         3271          0             0.289842                  1          29
1         1490          1             0.929765                  1          93
2         7936          0             0.289842                  1          29
3         4216          1             0.998548                  1         100
4         3830          0             0.289842                  1          29
------------------Model Evaluation Metrics------------------
Confusion Matrix :
 [[1221  513]
 [  44  945]]
Accuracy : 0.7954461990451708
Sensitivity : 0.9555106167846309
Specificity : 0.7041522491349481
Precision : 0.6481481481481481
```

Out[62]:

	Prospect ID	Converted	Convert_Probability	Convert_predicted	Lead_Score
0	3271	0	0.289842	1	29
1	1490	1	0.929765	1	93
2	7936	0	0.289842	1	29
3	4216	1	0.998548	1	100
4	3830	0	0.289842	1	29
...	...	...	...	...	...
2718	850	0	0.070553	0	7
2719	2879	0	0.001642	0	0
2720	6501	1	0.989122	1	99
2721	7155	0	0.070553	0	7
2722	376	0	0.070553	0	7

2723 rows × 5 columns

Subjective Question

1. **Similarly, at times, the company reaches its target for a quarter before the deadline. During this time, the company wants the sales team to focus on some new work as well. So during this time, the company's aim is to not make phone calls unless it's extremely necessary, i.e. they want to minimize the rate of useless phone calls. Suggest a strategy they should employ at this stage.**

In [63]:

```
build_model_cutoff(X_train[col], y_train, X_test[col], y_test, cutoff=0.9)
```

```
------------------Result of training data------------------
   Prospect ID  Converted  Convert_Probability  Convert_predicted  Lead_Score
0         3009          0             0.289842                  0          29
1         1012          0             0.111387                  0          11
2         9226          0             0.001918                  0           0
3         4750          1             0.737087                  0          74
4         7987          1             0.993914                  1          99
------------------Result of test data------------------
   Prospect ID  Converted  Convert_Probability  Convert_predicted  Lead_Score
0         3271          0             0.289842                  0          29
1         1490          1             0.929765                  1          93
2         7936          0             0.289842                  0          29
3         4216          1             0.998548                  1         100
4         3830          0             0.289842                  0          29
------------------Model Evaluation Metrics------------------
Confusion Matrix :
 [[1721   13]
 [ 370  619]]
Accuracy : 0.8593463092177746
Sensitivity : 0.6258847320525783
Specificity : 0.9925028835063437
Precision : 0.9794303797468354
```

Out[63]:

	Prospect ID	Converted	Convert_Probability	Convert_predicted	Lead_Score
0	3271	0	0.289842	0	29
1	1490	1	0.929765	1	93
2	7936	0	0.289842	0	29
3	4216	1	0.998548	1	100
4	3830	0	0.289842	0	29
...	...	...	...	...	...
2718	850	0	0.070553	0	7
2719	2879	0	0.001642	0	0
2720	6501	1	0.989122	1	99
2721	7155	0	0.070553	0	7
2722	376	0	0.070553	0	7

2723 rows × 5 columns

In [ ]: